Naponska kontrola elektrode CNC plazma rezača

Sudar, Jakov

Undergraduate thesis / Završni rad

2019

Degree Grantor / Ustanova koja je dodijelila akademski / stručni stupanj: **Josip Juraj Strossmayer University of Osijek, Faculty of Electrical Engineering, Computer Science and Information Technology Osijek / Sveučilište Josipa Jurja Strossmayera u Osijeku, Fakultet elektrotehnike, računarstva i informacijskih tehnologija Osijek**

Permanent link / Trajna poveznica: https://urn.nsk.hr/urn:nbn:hr:200:311574

Rights / Prava: In copyright / Zaštićeno autorskim pravom.

Download date / Datum preuzimanja: **2024-04-18**

Repository / Repozitorij:

Faculty of Electrical Engineering, Computer Science and Information Technology Osijek

SVEUČILIŠTE JOSIPA JURJA STROSSMAYERA U OSIJEKU
FAKULTET ELEKTROTEHNIKE, RAČUNARSTVA I INFORMACIJSKIH TEHNOLOGIJA OSIJEK
Sveučilišni studij

NAPONSKA KONTROLA ELEKTRODE CNC PLAZMA REZAČA

Završni rad

Jakov Sudar

Osijek, 2019.

FERIT
FAKULTET ELEKTROTEHNIKE, RAČUNARSTVA I INFORMACIJSKIH TEHNOLOGIJA OSIJEK

Obrazac Z1P - Obrazac za ocjenu završnog rada na preddiplomskom sveučilišnom studiju

Osijek, 10.09.2019.

Odboru za završne i diplomske ispite

Prijedlog ocjene završnog rada

Ime i prezime studenta:	Jakov Sudar
Studij, smjer:	Preddiplomski sveučilišni studij Računarstvo
Mat. br. studenta, godina upisa:	R3986, 26.09.2018.
OIB studenta:	59336824559
Mentor:	Izv. prof. dr. sc. Davor Vinko
Sumentor:	
Sumentor iz tvrtke:	
Naslov završnog rada:	Naponska kontrola elektrode CNC plazma rezača
Znanstvena grana rada:	**Elektronika (zn. polje elektrotehnika)**
Predložena ocjena završnog rada:	Izvrstan (5)
Kratko obrazloženje ocjene prema Kriterijima za ocjenjivanje završnih i diplomskih radova:	Primjena znanja stečenih na fakultetu: 3 bod/boda Postignuti rezultati u odnosu na složenost zadatka: 2 bod/boda Jasnoća pismenog izražavanja: 3 bod/boda Razina samostalnosti: 3 razina
Datum prijedloga ocjene mentora:	10.09.2019.
Datum potvrde ocjene Odbora:	11.09.2019.

Potpis mentora za predaju konačne verzije rada u Studentsku službu pri završetku studija:	Potpis:
	Datum: 16.09.2019.

FERIT
FAKULTET ELEKTROTEHNIKE, RAČUNARSTVA I INFORMACIJSKIH TEHNOLOGIJA **OSIJEK**

IZJAVA O ORIGINALNOSTI RADA

Osijek, 15.09.2019.

Ime i prezime studenta:	Jakov Sudar
Studij:	Preddiplomski sveučilišni studij Računarstvo
Mat. br. studenta, godina upisa:	R3986, 26.09.2018.
Ephorus podudaranje [%]:	13%

Ovom izjavom izjavljujem da je rad pod nazivom**: Naponska kontrola elektrode CNC plazma rezača**

izrađen pod vodstvom mentora Izv. prof. dr. sc. Davor Vinko

i sumentora

moj vlastiti rad i prema mom najboljem znanju ne sadrži prethodno objavljene ili neobjavljene pisane materijale drugih osoba, osim onih koji su izričito priznati navođenjem literature i drugih izvora informacija.
Izjavljujem da je intelektualni sadržaj navedenog rada proizvod mog vlastitog rada, osim u onom dijelu za koji mi je bila potrebna pomoć mentora, sumentora i drugih osoba, a što je izričito navedeno u radu.

Potpis studenta:

SADRŽAJ

1. UVOD 1
1.1. Zadatak završnog rada 2
2. OSNOVE PLAZMA REZAČA 3
2.1. Plazma 3
2.2. Elektroda 3
2.3. Vrijednosti veličina prilikom rezanja 4
2.4. CNC plazma rezanje 5
2.4.1. Mach3 6
2.4.2. G-kod 7
3. IZVEDBA RADA 8
3.1. Prilagodba napona 8
3.1.1. Naponsko djelilo 8
3.1.2. Low pass filtar 9
3.1.3. Arduino očitavanje napona 10
3.2. Komunikacija Arduino-Mach3 10
3.2.1. Optoizolator 11
3.2.2. TTL 12
3.3. Paljenje elektrode 12
3.3.1. Relej 13
3.3.2. Tranzistor 14
3.4. Upravljanje visinom elektrode 15
3.4.1. Step motor 16
3.4.2. EasyDriver 16
3.4.3. Detektiranje metala 17
3.4.4. Početna visina elektrode 18
3.5. Korisničko sučelje 21
3.5.1. Enkoder 21
3.5.2. Meniji 22
4. Realizacija sustava 24
4.1. Korisničko sučelje 24
4.2. Mehanizam podizanja i spuštanja elektrode 25
4.3. Simulacija 25

5. Zaključak 27
LITERATURA 28
SAŽETAK 29
ABSTRACT 29
ŽIVOTOPIS 30
Prilog 1 31
Prilog 2 32
Prilog 3 42

1. UVOD

Kako bi dobili kvalitetan i čist rez korištenjem plazma rezača, potrebno je ispuniti slijedeće uvjete :

- Konstantan pritisak zraka
- Konstantu udaljenost elektrode i materijala
- Odgovarajuću jakost električne struje [1]

Pritisak zraka i jakost električne struje prilikom rezanja predodređuje rukovatelj, te nakon što započne rezanje nema više utjecaja na te dvije varijable. Preostala varijabla "Konstanta udaljenost elektrode i materijala" ovisi o rukovatelju odnosno u našem slučaju o CNC sustavu. Upravljanje visinom elektrode (engl. *Torch hight control,* u daljnjem tekstu: THC) biti će izvedeno kao zaseban sklop povezan s CNC sustavom. Visina elektrode izravno je povezana s naponom između elektrode i materijala koji se obrađuje. Napon će se povećavati ukoliko se povećava udaljenost između elektrode i materijala, te će se smanjivati ukoliko se udaljenost smanji. Cilj THC-a je mjeriti trenutni napon između elektrode i materijala za obradu te ga uspoređivati sa zadanim naponom. Ovisno o rezultatima usporedbe THC podizati ili spuštati elektrodu. Različite kvalitete reza za različite visine elektrode su prikazane na slici 1.1., pri čemu je elektroda najniža skroz desno, a najviša skroz lijevo.

Slika 1.1. *Rezovi dobiveni različitim visinama elektrode* [2]

1.1. Zadatak završnog rada

Zadatak završnog rada je razviti sustav za kontrolu visine elektrode CNC plazma rezača kako bi se održavao jednak napon električnog luka. Sustav je potrebno razviti korištenjem Arduino mikroupravljačke platforme.

2. OSNOVE PLAZMA REZAČA

Kako bi se što bolje upoznali s problemom kontrole visine elektrode plazma rezača potrebno je razumjeti osnove plazma rezača te način rada istog. U narednim potpoglavljima biti će opisani osnovni pojmovi i načela rada plazma rezača.

2.1. Plazma

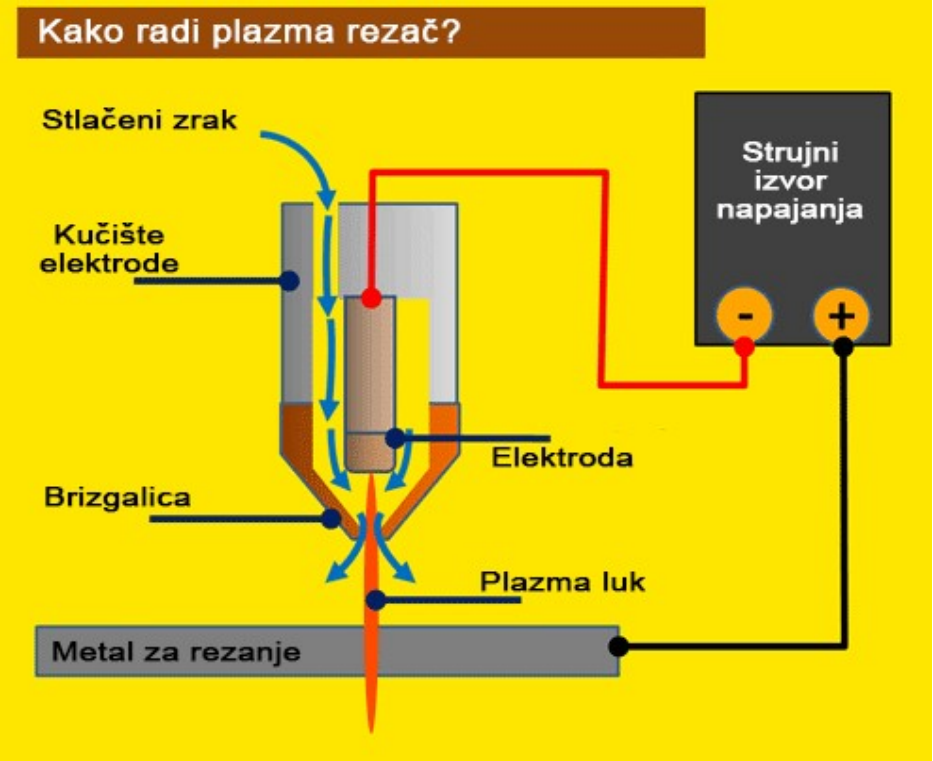
Postoje četiri stanja tvari: čvrsta tvar, tekućina, plin i plazma. Količina topline u nekoj tvari određuje njeno stanje. Na primjer, voda je čvrsta tvar u obliku leda koji možemo zagrijavati da postane tekućina. Daljnjim zagrijavanjem molekule napuštaju slabe veze te tekućina prelazi u stanje plina. Ukoliko bi plin nastavili zagrijavati atomi molekula bi se počeli razdvajati te bi dobili plazmu. Plazma je dakle, ioniziran plin koji ima barem po jedan elektron odvojen od dijela svojih atoma. Zbog velike količine elektrona plazma je jako dobar vodič električne struje.[3] Plazma rezač koristi ovo svojstvo električne provodljivosti za prijenos energije s elektrode na materijal koji se obrađuje. [4]

2.2. Elektroda

Elektroda (engl. *torch*) je naziv za dio plazma rezača iz kojeg izlazi mlaz plazme koji je prikazan na slici 2.1. Može se pronaći i naziv „pištolj“ koji je čest u žargonu. Naziv elektroda dolazi iz činjenice da se unutar kućišta sa slike 2.1 nalazi elektroda koja je izvor električne energije.

Slika 2.1. *Elektroda (pištolj)* [5]

Unutar kućišta se nalazi prolaz za stlačeni plin koji prolazi pored elektrode te kroz nosnicu izlazi van. Prilikom pokretanja plazma rezača na elektrodu se na kratko dovodi visoki napon visoke frekvencije koji služi za okidanje plazme. Nakon što je plazma luk uspostavljen električna energija se prenosi sa elektrode na materijal koji se obrađuje. Nosnica usmjerava plazmu što osigurava čist i oštar rez. (Slika 2.2.)



Slika 2.2. *Unutrašnjost elektrode* [6]

2.3. Vrijednosti veličina prilikom rezanja

Prilikom rezanja jačina električne struje iznosi do 100A, pri čemu je napon između elektrode i materijala za obradu -120V. Plazma rezači koriste negativan napon za rezanje, odnosno uzemljenje je na 0V a elektroda na -120V. Jačina struje ovisi o debljini materijala te se određuje iz tablica koje proizvođač plazma rezača pruža. Primjer takve tablice nalazi se na slici 2.3.

Versa-Cut 60 Plasma Cutter – Air Pressure & Amperage Settings*														
Metal Thickness	1/64"	1/32"	3/64"	1/16"	5/64"	3/32"	7/64"	1/8"	9/64"	5/32"	11/64"	3/16"	13/64"	7/32"
AMPS	20	20	20	20	20	20	20	20	20	20	20	20	30	30
PSI	30	30	30	30	30	30	30	30	30	30	30	30	40	40
Metal Thickness	15/64"	1/4"	17/64"	9/32"	19/64"	5/16"	21/64"	11/32"	23/64"	3/8"	25/64"	13/32"	27/64"	7/16"
AMPS	30	30	30	30	30	40	40	40	40	40	50	50	50	50
PSI	40	40	40	40	40	40	40	40	40	40	40	40	40	40
Metal Thickness	29/64"	15/32"	31/64"	1/2"	33/64"	17/32"	35/64"	9/16"	37/64"	19/32"	39/64"	5/8"	41/64"	21/32"
AMPS	50	50	50	50	50	50	50	50	50	50	50	50	50	50
PSI	40	40	40	50	50	50	50	50	50	50	50	50	60	60
Metal Thickness	43/64"	11/16"	45/64"	23/32"	47/64"	3/4"	49/64"	25/32"	51/64"	13/16"	53/64"	27/32"	55/64"	7/8"
AMPS	50	50	50	50	50	50	60	60	60	60	60	60	60	60
PSI	60	60	60	60	60	60	60	60	60	60	60	60	60	60

*These settings are guidelines and may need to be adjusted, based on your techniques.

Slika 2.3. *Vrijednosti jakosti struje i pritiska zraka za različite debljine materijala* [7]

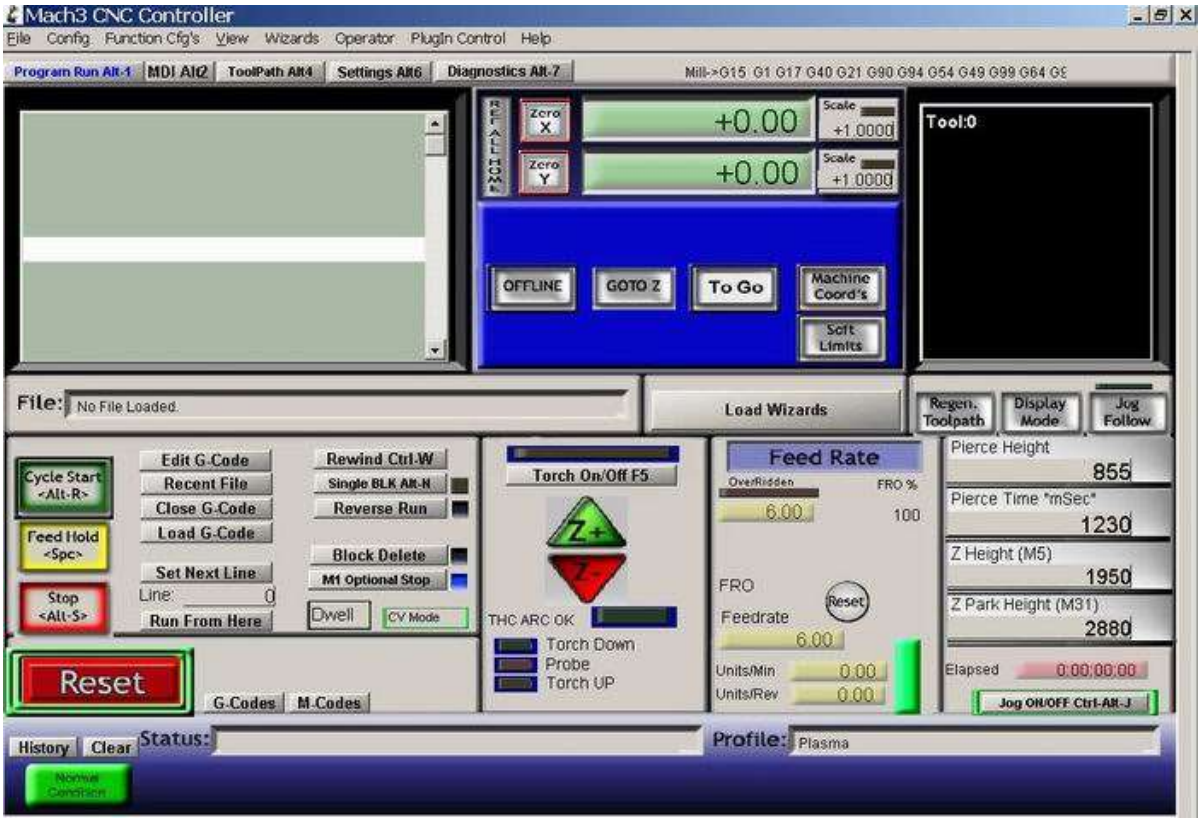
2.4. CNC plazma rezanje

CNC (engl. *Computer Numerical Control*) je računalno numeričko upravljanje alatnim strojevima pomoću kodiranih naredbi. Naredbe se unose na kontrolnoj jedinici pomoću posebno dizajnirane tipkovnice. Primjer takve kontrolne jedinice dan je na slici 3.1. U posljednje vrijeme došlo je do razvoja programskih alata koji pretvaraju osobno računalo u CNC kontrolnu jedinicu. Razvoj takvih alata omogućio je korištenje CNC-a bez kupovine skupih kontrolnih jedinica.

Slika 2.4. *Kontrolna jedinica* [8]

2.4.1. Mach3

Mach3 je program koji pretvara obično računalo u CNC upravljač. Jako je fleksibilan te je dizajniran tako da može upravljati glodalicama, laserskim rezačima, plazma rezačima, tokarskim strojevima i sl. Najrasprostranjeniji je od svih alata slične primjene te će se izrada THC-a u ovom radu temeljiti na Mach3 CNC sustavu [9]. Interface Mach3 sustava je prikazano na slici 3.2. Mach3 prevodi G-kod te ovisno o naredbama šalje upravljačke impulse upravljačkom sklopu.



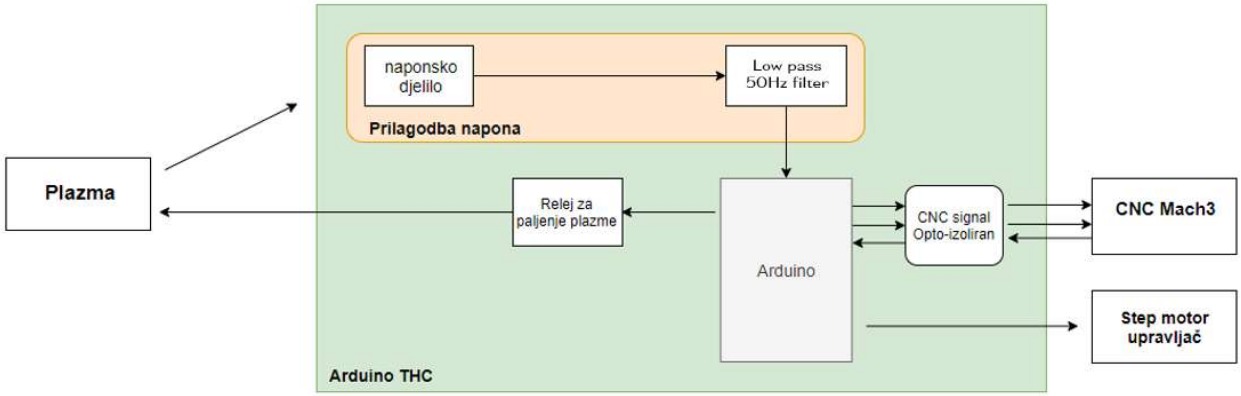
Slika 2.5. *Interface Mach3 sustava* [10]

2.4.2. G-kod

G kod je skup naredbi računala za CNC upravljanje kojima se detaljno opisuju putanje kretanja alata kojeg CNC koristi. Naredbe G koda imaju oblik GXXX, gdje je XXX numerička vrijednost naredbe koja ju identificira. G kod se može pisati izravno unoseći naredbe ili korištenjem CAD alata. Postoje i M naredbe koje određuju pomoćne strojne funkcije. M naredbe imaju oblik MXX gdje XX također označava broj naredbe koji ju identificira. Primjeri pomoćnih funkcija koje M naredbe izvršavaju su: kraj programa, zaustavljanje programa, izmjena alata, uvjetno zaustavljanje, uključenje rashladnog sredstva itd.

3. IZVEDBA RADA

Blok shema sustava prikazana je na slici 3.1.



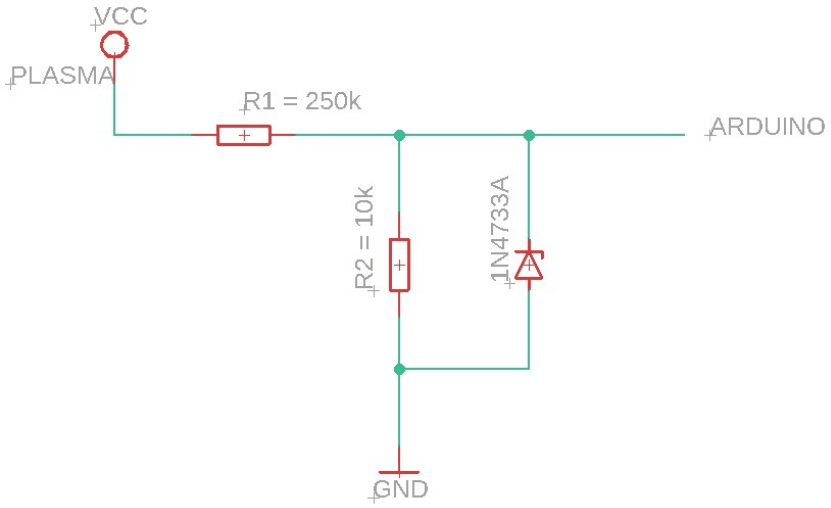
Slika 3.1. *Blok shema sustava*

3.1. Prilagodba napona

Plazma rezač daje visok napon pun šuma koji ne odgovara Arduinu te je potrebno prije dolaska do Arduina provesti nekoliko koraka kako bi se taj napon prilagodio mikroupravljaču. Koraci uključuju snižavanje izvornog napona te uklanjanje šuma.

3.1.1. Naponsko djelilo

Korištenjem jednostavnog naponskog djelila, snižen je napon sa razine rezanja do razine očitavanja Arduina – 5V. U nekim tehničkim specifikacijama plazma rezača naznačeno je da naponi rezanja idu do 200V no prilikom vršenja mjerenja nije uočen napon veći od 140V. Stoga je naponsko djelilo dizajnirano za niže ulazne veličine što je pružilo veću rezoluciju mjerenja. Veća rezolucija mjerenja je dovela do rizika da napon na izlazu djelila poprimi vrijednosti iznad 5V te time uzrokuje štetu na mikroupravljaču. To je spriječeno dodavanje Zener diode kao naponskog osiguranja. Shema naponskog djelila prikazana je na slici 3.2.



Slika 3.2. *Shema naponskog djelila*

Izraz za izračunavanje izlaznog napona naponskog djelila je:

$$V_{out} = \frac{R2}{R1 + R2} * V_{in}$$

$$V_{out} = \frac{10.000}{250.000 + 10.000} * 140 = 5.3$$

3.1.2. Low pass filtar

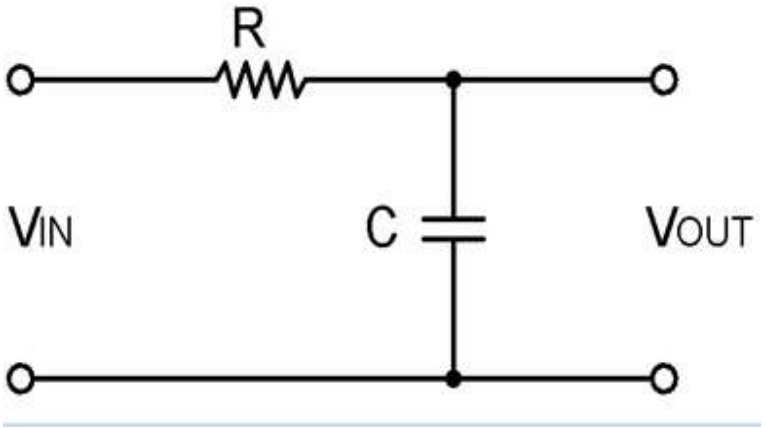
Low pass filtar je elektronički sklop koji uklanja sve signale čije su frekvencije veće od granične frekvencije te propušta sve signale čije su frekvencije manje od granične. Prema literaturi najčešće frekvencije koje se pojavljuju u šumu su od 50Hz [11], stoga su odabrane komponente proračunate za potpuno otklanjane signala frekvencija od 50Hz na više. Shema Low pass filtra dana je slikom 3.3. To je jednostavan filter koji se sastoji od dvije komponente, otpornika i kondenzatora.

Frekvencija filtriranja je frekvencija pri kojoj filtar počinje djelovati na signale no ne otklanja ih u potpunosti. Kako bi u potpunosti otklonili signale frekvencija preko 50Hz potrebno je projektirati filtar sa dosta nižom frekvencijom filtriranja.

Frekvencija filtriranja računa se prema izrazu :

$$f_c = \frac{1}{2\pi RC}$$

Nakon računanja odabiremo vrijednosti 470Ω za otpornik i 68µF za kondenzator. Odabir vrijednosti komponenti je kompromis između frekvencije filtriranja i brzine odziva. Brzina odziva s odabranim komponentama je 7ms [12].



Slika 3.3. *Shema low pass filtra* [13]

3.1.3. Arduino očitavanje napona

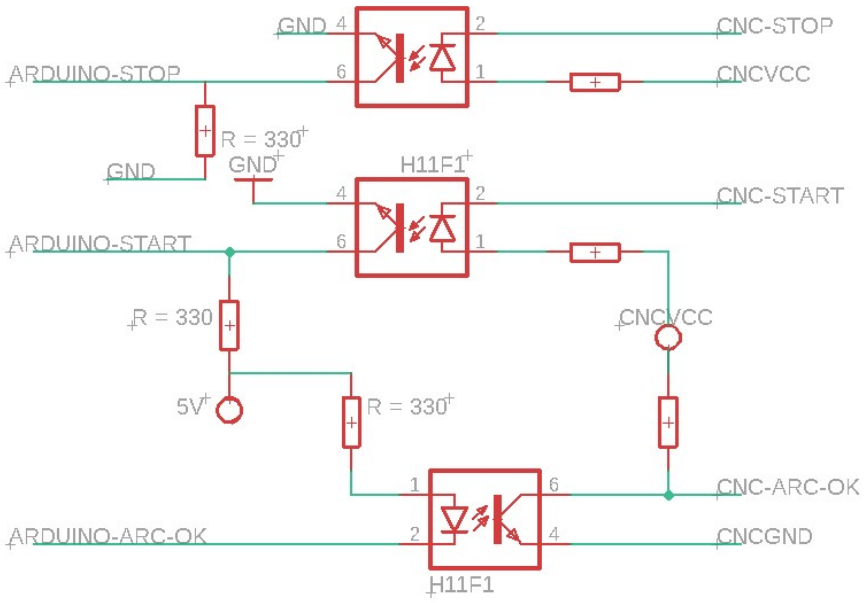
Napon koji se mjeri Arduinom spaja se na analogni pin. Arduino zatim vrši analogno-digitalnu pretvorbu integriranim strujnim krugom. Rezolucija arduina je 10 bitova odnosno 1024 razine, što znači da se analogni ulazni napon pretvara u digitalnu vrijednost između 0 i 1023. 0 predstavlja 0 volta, a 1023 predstavlja referentnu vrijednost. Kako bi smo dobili točnu vrijednost napona potrebno je očitanje sa ulaznog pina pomnožiti sa naponskom vrijednošću jednog pomaka rezolucije. Naponska vrijednost jednog pomaka računa se tako da se podijeli referentna vrijednost napona sa brojem pomaka rezolucije.

3.2. Komunikacija Arduino-Mach3

U ovom projektu Arduino je dodatak za Mach3, stoga je neophodna mogućnost komunikacije između njih. Oblik signala koji je potreban na ulazu Arduina i Mach3 je TTL signal, odnosno digitalni signal. To nam omogućuje da signal bude opto-izoliran te da se uklone bilo kakve opasnosti oštećenja CNC sustava ili Arduina ukoliko dođe do neželjenih ponašanja napona.

Jedna od zadaća Arduina je postaviti elektrodu u početni položaj prilikom početka rezanja. To se radi dok još nema napona na elektrodi gdje Arduino detektira metal pomoću krajnjih prekidača.

Kako bi Arduino znao kada treba postaviti elektrodu u početni položaj, Mach3 mu šalje signal o početku programa te čeka odgovor. Nakon što je Arduino primio takav signal, postavlja elektrodu u početnu poziciju te ju pali pomoću releja. Nakon paljenja, Arduino očitava napon između elektrode i metala te utvrđuje je li uspostavljen strujni luk. Ukoliko je luk uspostavljen, Arduino šalje signal „Arc-OK“ prema Mach3 koji po primitku takvog signala nastavlja sa programom te započinje rezanje. Ukoliko prilikom izvođenja programa dođe do prekida strujnog luka Arduino šalje informaciju Mach3 koji potom zaustavlja program te čeka dok Arduino ponovo ne uspostavi luk. Shema spajanja komunikacije dana je slikom 3.4. Element H11F1 je optoizolator. Iz sheme se vidi da se signali okidaju na logičku nulu. Kada se neki od outputa, CNC-start ili Arduino-arc-ok, dovedu u logičku nulu, fotodioda će postati propusno polarizirana te će se poslati signal.



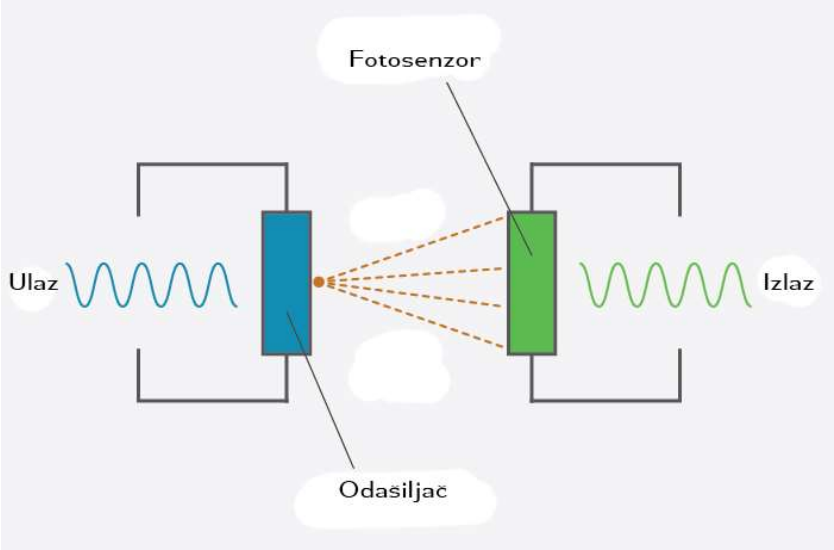
Slika 3.4. *Shema komunikacije Arduino-Mach3*

3.2.1. Optoizolator

Optoizolator je element koji pomoću svjetlosti galvanski odvaja strujne krugove trošila i upravljačkog dijela. Svrha je ista kao i kod releja, samo što optoizolator ne može upravljati velikim trošilima već se uglavnom koristi za prijenos signala. Ovakav strujni krug radio bi i bez optoizolatora, no u tom slučaju nema sigurnosti od neželjenih ponašanja napona na strani CNC-a.

Optoizolator se sastoji od svjetleće diode i foto senzora. Kada se svjetleća dioda propusno polarizira i struja poteče kroz nju, ona emitira svjetlost koja pada na foto senzor te se i on propusno

polarizira. Na taj način smo prenijeli signal sa jednog strujnog kruga na drugi iako oni međusobno nisu u fizičkom kontaktu. Prednost optoizolatora nad relejom je njegova brzina prijenosa signala. Skica optoizolatora prikazana je na slici 3.5.



Slika 3.5. *Optoizolator* [14]

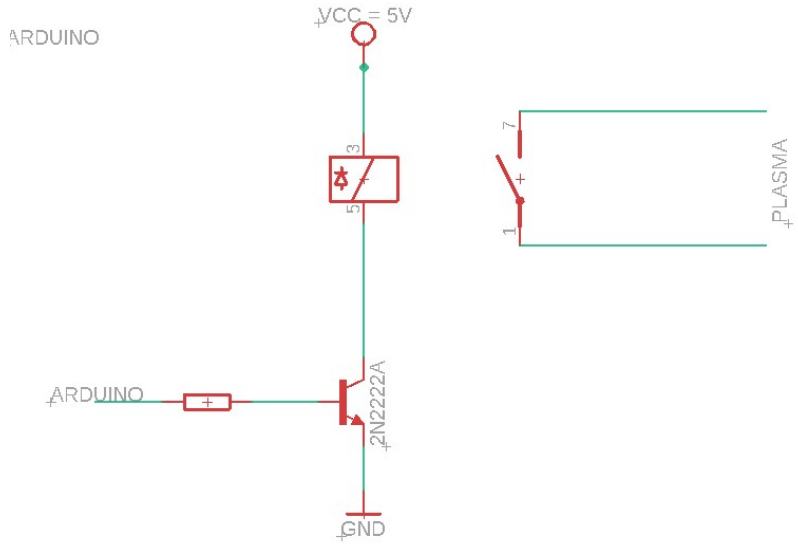
3.2.2. TTL

TTL je skraćenica od „Transistor-Transistor Logic“ te predstavlja sklopovlja izrađena od bipolarnih tranzistora. Naponske razine za TTL signal su idealno 0V za logičku nulu te 5V za logičku jedinicu. Kod Arduina naponske razine su >3V logička jedinica te <2V logička nula, dakle postoji tolerancija od 2V. [15]

3.3. Paljenje elektrode

Kako bi THC bio funkcionalan, mora imati mogućnost paljenja elektrode i samim time uspostavljanja strujnog luka između materijala i elektrode. Sučelje paljenja elektrode nije digitalno, najčešće je izvedeno kao suhi kontakt, odnosno fizički prekidač koji zahtjeva pritisak. Prekidač se u CNC sustavima zamjenjuje relejom kojeg Arduino može kontrolirati.

Izlazna struja sa Arduino digitalnog pina je najviše 50mA. Za aktiviranje zavojnice releja često je potrebno i više od toga. Radi toga ubačen je tranzistor koji će propustiti struju kroz relej kada je to potrebno. Strujni krug prikazan je slikom 3.6.

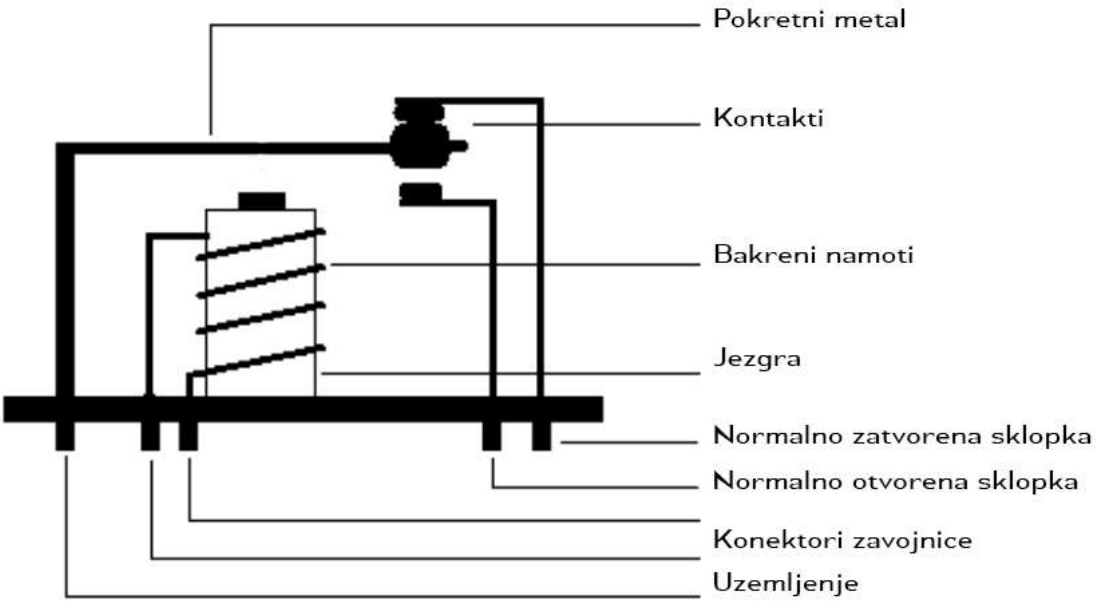


Slika 3.6. *Shema spajanja kruga za paljenje elektrode*

3.3.1. Relej

Releji su prekidači koji se aktiviraju električnim putem. Najrašireniji su releji s elektromagnetom no postoje i solid-state releji. Releji se koriste u situacijama u kojima je potrebno kontrolirati strujne krugove visokih napona ili velikih struja uz pomoć niskonaponskog signala ili u situacijama kada je potrebno kontrolirati više krugova sa samo jednim upravljačkim signalom.

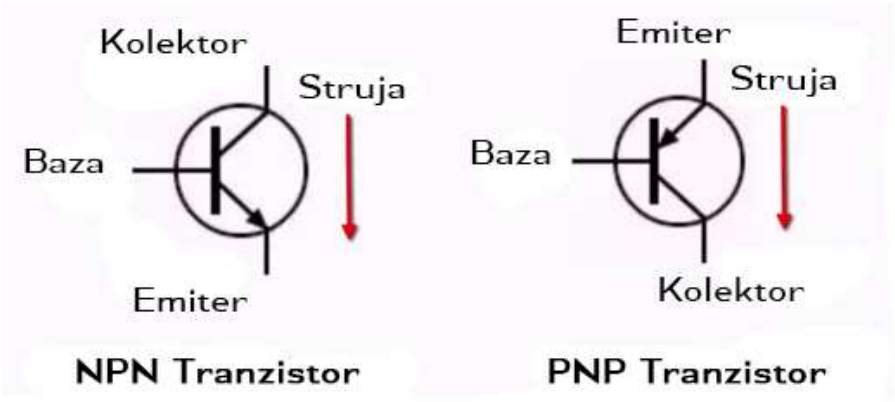
Elektromagnetni releji imaju najčešće jednu zavojnicu namotanu oko jezgre koja predstavlja elektromagnet. Kada struja poteče kroz zavojnicu stvara se elektromagnetsko polje koje privlači kontakte te ih na taj način spaja. Skica releja prikazana je slikom 3.7. Releji su pogodni za korištenje jer galvanski odvajaju strujne krugove trošila i upravljačkog dijela. Mana elektromagnetnih releja je što su relativno spori te ih je nemoguće koristit u situacijama gdje je potrebno brzo mijenjati stanja strujnog kruga. Mana brzine elektromagnetnog releja riješena je u takozvanim solid-state relejima gdje se za zatvaranje strujnog kruga ne koriste pomični dijelovi već poluvodiči. U relejima nalazimo normalno otvorene i normalno zatvorene kontakte. Normalno otvoreni kontakt je kontakt koji je otvoren u slučaju kada nema upravljačkog signala na releju, a zatvara se prilikom dolaska upravljačkog signala. Normalno zatvoreni kontakt je zatvoren dok nema upravljačkog signala na releju.



Slika 3.7. *Skica releja* [16]

3.3.2. Tranzistor

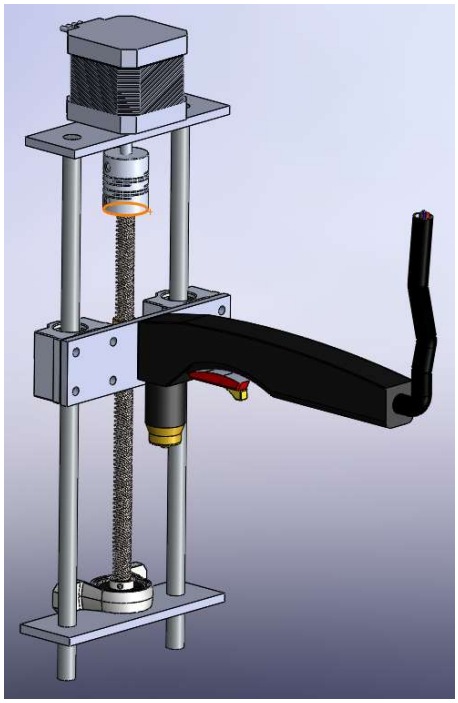
Tranzistor je aktivni poluvodički element sa tri izvoda. Postoji u dvije verzije a to su: bipolarni i unipolarni. Bipolarni je upravljan strujom dok je unipolarni upravljan naponom. Koriste se za razne primjene kao što su: pojačala električnih signala, stabilizaciju napona, modulacije signala, te u ovom slučaju, upravljane sklopke. Tranzistori se mogu izrađivati u 2 tehnologije NPN i PNP. Razlika je u tipu baze te emitera i kolektora. Kod NPN tranzistora baza je P tip poluvodiča, a emiter i kolektor su N-tipovi. Kod PNP tranzistora je obrnuto. Tranzistor se, ovisno o ulaznoj struji baze, prebacuje u područje zapiranja odnosno zasićenja. U području zapiranja tranzistor je kao otvorena sklopka uz zanemariv tok struje, dok je u području zasićenja tranzistor kao zatvorena sklopka uz nizak pad napona između kolektora i emitera. Simboli PNP i NPN tranzistora dani su slikom 3.8.



Slika 3.8. *Simboli NPN i PNP tranzistora* [17]

3.4. Upravljanje visinom elektrode

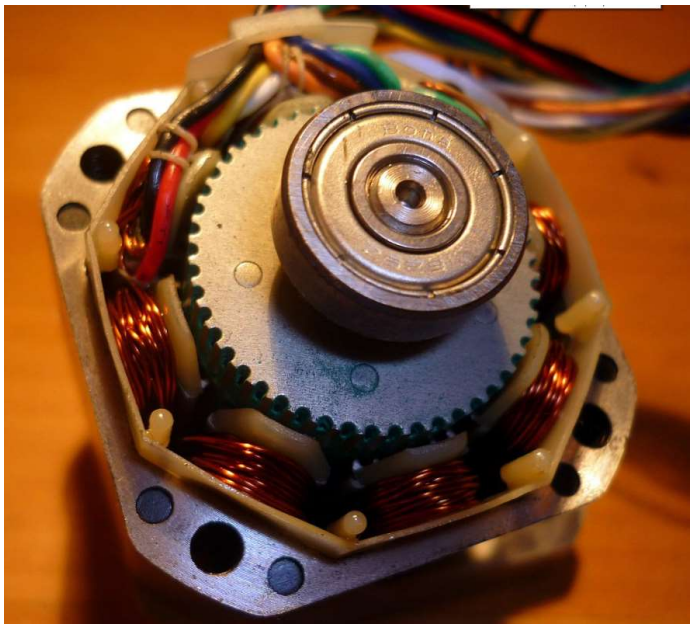
Nakon što je Arduino upalio elektrodu te poslao „Arc-OK“ signal, jedina zadaća koja mu preostaje je reguliranje visine elektrode. Na temelju zadanog parametra napona između elektrode i metala te izmjerenog napona između elektrode i metala, Arduino donosi odluku treba li elektrodu podići ili spustiti. Elektroda je preko linearnog vijka povezana sa step motorom kojim upravlja Arduino. Povezanost elektrode i step motora prikazan je na slici 3.9.



Slika 3.9. *3D model THCa*

3.4.1. Step motor

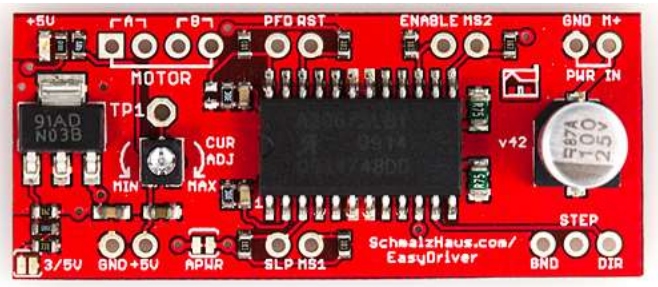
Step motor je istosmjerni električni motor koji jedan okret dijeli na određen broj jednakih koraka. Koliko će se motor okrenuti zadaje se brojem koraka koji će napraviti. Ovo svojstvo daje mu iznimnu preciznost okretanja. Važno je napomenuti da step motor ne koristi četkice kao standardni istosmjerni motori koji se vrte dok god je prisutan napon. Step motori koriste elektromagnetno polje te kako bi se motor vrtio potrebno je pomjerati elektromagnetno polje između namotaja statora. To je složen proces te se za upravljanje step motora koriste razni step upravljači. Unutarnji izgled step motora prikazan je slikom 3.10.



Slika 3.10. *Unutarnji izgled step motora* [18]

3.4.2. EasyDriver

EasyDriver je jedan od najraširenijih upravljača za step motor. Prikaz EasyDrivera je na slici 3.11.



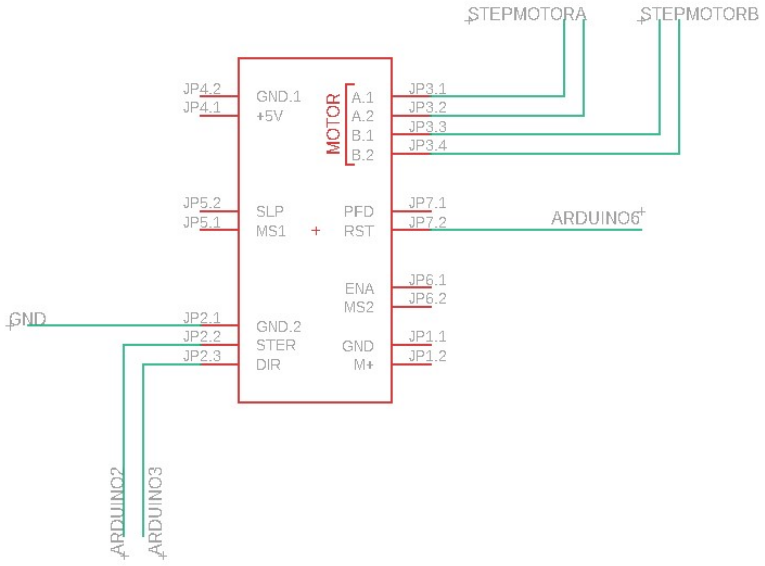
Slika 3.11. *Prikaz EasyDrivera* [19]

U tablici 3.1. su opisani pinovi koji će biti korišteni. Ostatak opisa može se pronaći u literaturi [19].

A	Izlaz za jedan namotaj statora
B	Izlaz za drugi namotaj statora
RST	Logički ulaz. Postavljanjem u logičku jedinicu ignoriraju se sve step naredbe te se motor ne pokreće
GND	Uzemljenje
STEP	Logički ulaz. Svaki padajući brid na ovom ulazu uzrokovat će pomak motora za jedan korak.
DIR	Logički ulaz. Promjenom stanja na ovom ulazu, mijenja se smjer okretanja motora

Tablica 3.1. *Opis korištenih pinova* [19]

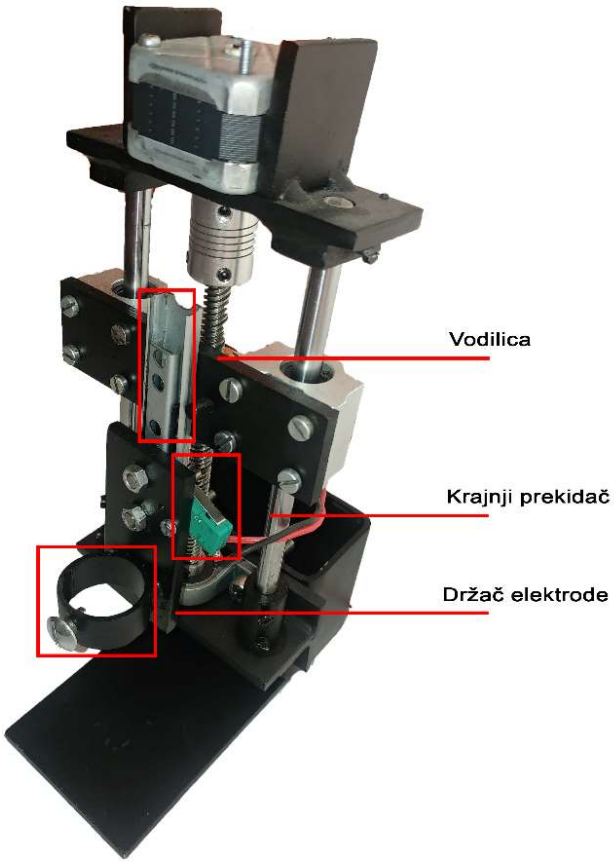
Shema spajanja prikazana je slikom 3.12.



Slika 3.12. *Shema spajanja EasyDrivera*

3.4.3. Detektiranje metala

THC pomoću krajnjih prekidača detektira površinu metala. U stranoj literaturi to je nazvano pojmom „floating head“. Mehanizam je prikazan slikom 3.13.



Slika 3.13. *Mehanizam za kontrolu visine*

Elektroda je montirana na vodilicu po kojoj može klizati gore/dolje. Djelovanjem gravitacijske sile elektroda je uvijek na dnu vodilice. Na nosač elektrode također je montiran spomenuti krajnji prekidač. Prilikom detekcije metala, motor spušta cijelu konstrukciju prema dolje. U trenutku kad elektroda dotakne metal, ona se počinje dizati po vodilici prema gore što dovodi do aktiviranja krajnjeg prekidača čime se šalje signal mikroupravljaču. Važno je napomenuti da je put kojeg elektroda pređe po vodilici od trenutka kada dotakne metal do trenutka kada se aktivira prekidač uvijek isti za sve debljine metala. Ta činjenica će se iskoristiti u slijedećim poglavljima za postavljanje početne visine elektrode.

3.4.4. Početna visina elektrode

Početna visina elektrode je među najbitnijim zadaćama THC-a. Ukoliko bi početna udaljenost elektrode i obrađivanog metala bila prevelika, plazma luk se ne bi uspostavio te rezanje ne bi bilo moguće. Ukoliko je početna udaljenost premala, od frcanja iskri prilikom uspostave luka došlo bi do oštećenja elektrode.

Korisnik će sam preko korisničkog sučelja birati željenu korisničku visinu u milimetrima. Da bi se takva funkcionalnost omogućila bilo je potrebno provesti određena mjerenja kako bi se utvrdilo koliko koraka motora iznosi 1mm. Za to je korišten mikrometar koji je na konstrukciju pričvršćen kao na slici 3.14.

Nosač je ručno spušten tako da se mikrometar otklone za 2 mm. Nakon toga mikrometarska skala dovedena je u nulu. Idući korak bio je okrenuti motor za 50 koraka tako da se nosač kreće u smjeru prema gore. Nakon okrenutih 50 koraka očitanje mikrometra pokazalo je da se nosač podignuo za 2 mm. Mjerenje je ponovljeno deset puta te su odstupanja iznosila +/- 0.04 mm. Iz rezultata mjerenja zaključeno je da za pomak od 1 milimetra potrebno je napraviti 25 koraka motora.

Slika 3.14. *Montaža mikrometra*

Ukupni hod koji nosač mora preći kako bi otklonio elektrodu je zbroj zadane početne visine te „prazni hod“ po vodilici. Put je prikazan na slici 3.15.

Slika 3.15. *Ukupni hod nosača*

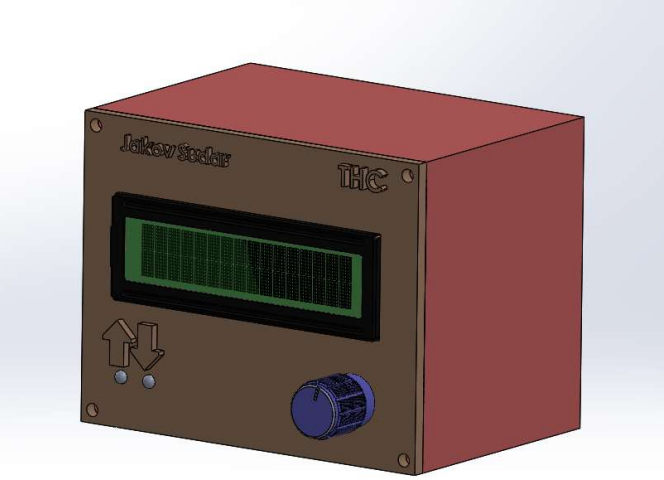
Za utvrđivanje duljine praznog hoda provedeno je mjerenje na način da se elektroda ručno dovede u poziciju gdje dodirne metal te se u tom trenutku pokrene motor da nastavi spuštati nosač sve dok se ne aktivira krajnji prekidač pri čemu program broji koliko je koraka motor napravio. Program korišten u mjerenju dan je u nastavku:

```
while(digitalRead(down)==HIGH){
          Down(700);
          steps++;
          }
          Serial.println(steps);
}
```

Kako je ručno dovođenje elektrode u poziciju gdje dodirne metal subjektivna radnja, postupak je ponovljen 5 puta te je dobivena prosječna vrijednost od 502 koraka uz maksimalna odstupanja od 2 koraka.

3.5. Korisničko sučelje

Korisniku se mora omogućiti postavljanje željenih parametara kao što su napon rezanja i početna visina. Zbog toga izrađeno je korisničko sučelje čiji je model dan slikom 3.16.

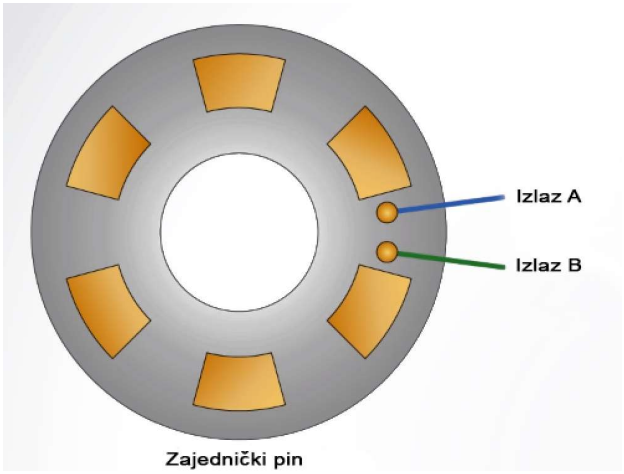


Slika 3.16. *Model korisničkog sučelja*

Korisničko sučelje sastoji se od LCD zaslona za prikaz postavki te enkodera za kretanje kroz menije i postavljanje postavki. U lijevom donjem kutu vidimo 2 LED koje će svijetliti kad se elektroda kreće gore ili dolje.

3.5.1. Enkoder

Enkoder je komponenta koja izgledom podsjeća na potenciometar ali za razliku od potenciometra njegovi izlazi daju digitalne vrijednosti. Skica enkodera dana je slikom 3.17.

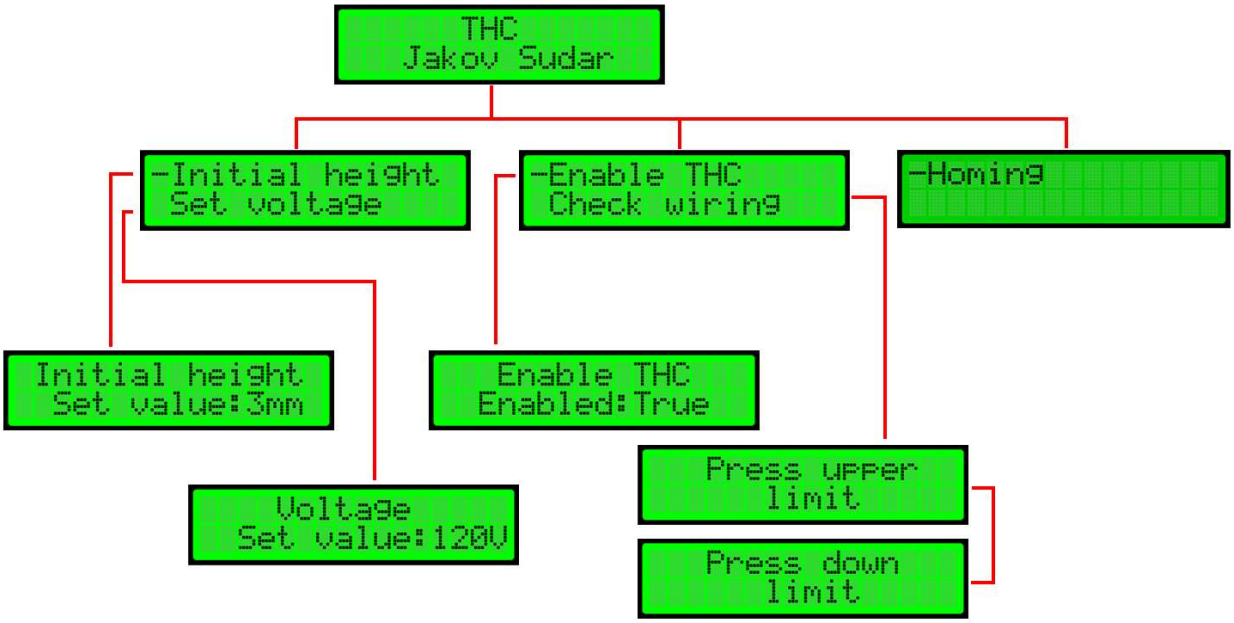


Slika 3.17. *Skica enkodera*

Enkoder ima određena područja kontakta (žuto na slici 3.17.) te 2 konektora. Vrteći osovinu enkodera okreću se kontaktna područja te konektori naizmjenice provode signal. Pošto su konektori odvojeni jedan od drugoga, promatranjem vremena provođenja signala može se utvrditi u kojem smjeru se okreće enkoder. Enkoderi najčešće dolaze s ugrađenim tipkalom te su pogodni za korištenje u korisničkim sučeljima.

3.5.2. Meniji

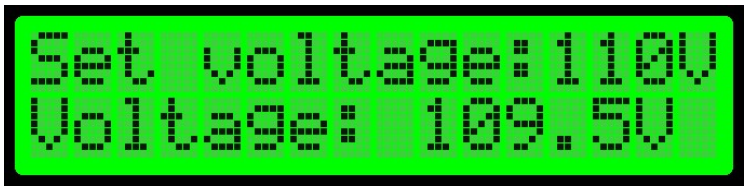
Korisničko sučelje ima nekoliko menija: Initial height, Set voltage, Enable THC i Check wiring te za svaki od tih odgovarajući podmeni. Raspored menija prikazan je slikom 3.18.



Slika 3.18. *Prikaz menija*

Na samom vrhu slike 3.18. nalazi se početni zaslon koji će biti aktivan dok god korisnik ne pomakne enkoder. Pomicanjem enkodera pomjera pokazivač između četiri menija. Klikom enkodera korisnik ulazi u meni te odabire parametar. Parametri se podešavaju okretanjem enkodera. Izuzetak je meni „Check wiring“ koji služi za provjeru spajanja uređaja te meni „Homing“ koji elektrodu podiže na sam vrh nosača. Ulaskom u „Check wiring“ program će zahtijevati pritisak pojedinih prekidača te će ispisati poruku ukoliko je sve dobro spojeno.

Postavljanjem postavke Enable THC u „false“ možemo onemogućiti kretanje motora, no prikaz napona te svijetljenje dioda će i dalje biti prisutno. Tokom rezanja na zaslonu će biti prikazan ekran na slici 3.19. Na navedenom ekranu prikazan je zadani napon te stvarni napon elektrode.



Slika 3.19. *Zaslon prilikom rezanja*

4. Realizacija sustava

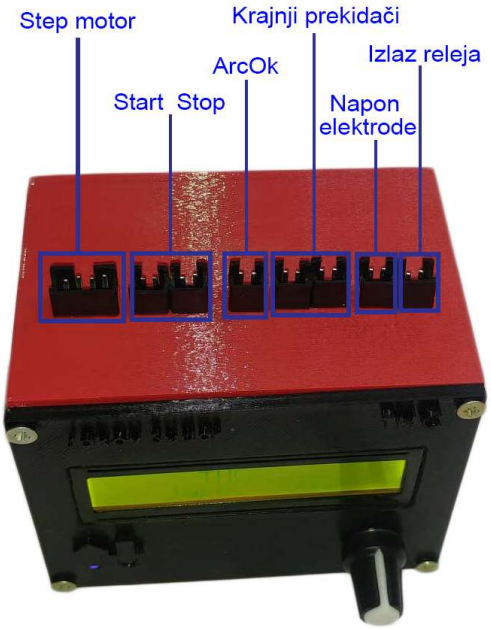
Sustav je uspješno realiziran slijedeći ideje iz prethodno napisanih poglavlja. Sustav se sastoji od dva dijela. To su korisničko sučelje te mehanizam za podizanje i spuštanje elektrode. Električna shema spajanja cijelog sustava nalazi se u prilogu 1.

4.1. Korisničko sučelje

Kućište korisničkog sučelja je modelirano u alatu „Solidworks“ te je isprintano na 3d printeru. Prikaz gotovog korisničkog sučelja je na slici 4.1.

Slika 4.1. *Korisničko sučelje*

Na gornjoj strani kućišta nalaze se konektori sa povezivanje motora, prekidača, CNC sustava i sl. Opis konektora prikazan je slikom 4.2. Zbog oblika konektora nije moguće zamijeniti polaritet spoja. Nakon spajanja potrebno je pokrenuti proceduru „Check wiring“ koja će utvrditi jesu li krajnji prekidači ispravno spojeni. Napon napajanja korisničkog sučelja je 12V. Sa lijeve strane također postoji i USB b priključak koji omogućuje izmjenu programa. Poklopac kućišta pričvršćen je vijcima za kućište te je moguće otvoriti kućište. Kod korišten za programiranje korisničkog sučelja nalazi se u prilogu 2.



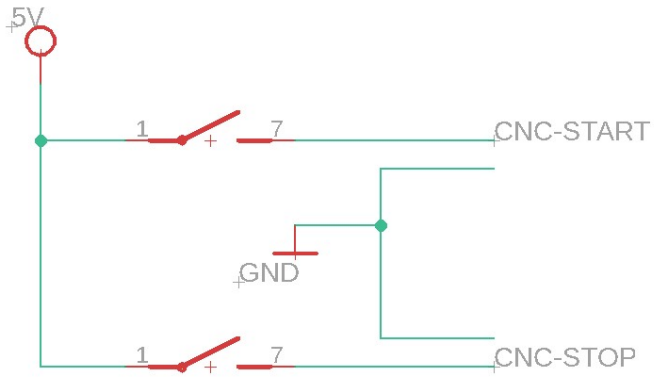
Slika 4.2. *Povezivanje konektora*

4.2. Mehanizam podizanja i spuštanja elektrode

Mehanizam je također modeliran u programu „Solidworks“ te je ,nakon kreiranja potrebnih nacrta, izrađen od željeza. Motor je povezan sa vodećim vijkom koji je preko odgovarajuće matice povezan sa nosačem elektrode. Na nosač elektrode pričvršćena su dva linearna ležaja koji klize po dvije vodilice što osigurava pravocrtno gibanje nosača. Detaljnije slike nalaze se u prilogu 3.

4.3. Simulacija

Za potrebe simulacije izrađen je sustav koji simulira signale poslane sa Mach3 CNC sustava. Shema sustava dana je slikom 4.3. Pritiskom na tipkalo napon iznosa 5V se provodi na optoizolator te se aktivira odgovarajuća interrupt funkcija. Uz prikazani testni sklop korišten je i promjenjivi regulator napona koji simulira mijenjanje napona elektrode plazma rezača. Prilikom simuliranja sustav se ponašao prema očekivanom. Vrijednost napona mjerena voltmetrom iznosila je 9.98 dok je Arduino očitavao 9.961. Iz čega slijedi da je pogreška mjerenja 0.2%.



Slika 4.3. *Sustav za testiranje*

5. Zaključak

Za rad bilo kojeg CNC plazma rezača neophodno je imati sustav za kontrolu visine elektrode. Prilikom rezanja različitih metala, zbog velikih temperatura često dolazi do savijanja metala te ukoliko ne bi imali sustav kontrole visine elektrode moglo bi doći do oštećenja elektrode kao i samog materijala.

Zbog različitih izvedbi plazma rezača, neki od njih imaju izrazito visoke visokofrekvencijske napone što bi predstavljalo mogući problem za sustav izrađen u ovom radu. Preporuka za rješenje tog problema bila bi korištenje analognog optoizolatora čijim bi se korištenjem galvanski odvojili strujni krug plazma rezača te strujni krug Arduina bez narušavanja funkcionalnosti.

Pogreška mjerenja napona od 0.2% je prihvatljiva jer prilikom ugradnje THC sustava svakako je potrebno izvršiti nekoliko probnih rezova da se utvrdi koji napon daje najbolje rezultate. Stoga nije toliko značajna prava vrijednost napona. Pogreška nastaje zbog blagih padova referentnog napona u Arduinu. Ukoliko bi točna vrijednost napona bila potrebna, problem bi se mogao riješiti korištenjem referentne diode kao što je LM4040DIZ koja daje stabilan napon koji bi dalje bio korišten kao Arduino referentni napon.

LITERATURA

[1]https://www.bakersgas.com/weldmyworld/2016/12/12/plasma-cutting-tips-improve-cut/ (21.5.2019.)

[2] http://www.cncplasma.com.my/images/cutting_quality2_img.png (21.5.2019.)

[3] https://home.howstuffworks.com/plasma-cutter2.htm (21.5.2019.)

[4] https://www.lincolnelectric.com/en-us/equipment/plasma-cutters/process-and-theory/Pages/how-a-plasma-cutter-works.aspx (21.5.2019.)

[5] https://images-na.ssl-images-amazon.com/images/I/51wf9gnzpBL._SX425_.jpg (21.5.2019.)

[6] https://www.esabna.com/shared/content/images/how-a-plasma-torch-works.gif (21.5.2019.)

[7] https://i.pinimg.com/originals/63/04/df/6304dfb178fe604160e35f1a5e807905.jpg (22.5.2019.)

[8]https://hr.wikipedia.org/wiki/CNC_upravljanje#/media/File:MCFH_40_CNC_-_control_panel_(1).jpg (22.5.2019.)

[9]https://www.machsupport.com/wp-content/uploads/2013/02/Mach3Mill_Install_Config.pdf (22.5.2019.)

[10] https://i.ytimg.com/vi/tOKhe-vXerM/maxresdefault.jpg (22.5.2019.)

[11]https://github.com/regeg/ArdunioTHC/blob/master/THC%20Manual/Arduino%20THC%20Manual.docx (28.5.2019.)

[12] http://sim.okawa-denshi.jp/en/CRtool.php (28.5.2019.)

[13] https://www.electronicshub.org/passive-low-pass-rc-filters/ (28.5.2019.)

[14] https://www.autodesk.com/products/eagle/blog/how-an-optocoupler-works/ (26.6.2019.)

[15] https://www.arduino.cc/reference/en/language/variables/constants/constants/ (26.6.2019)

[16] https://circuitdigest.com/article/relay-working-types-operation-applications (26.6.2019.)

[17] https://www.youtube.com/watch?v=FHX398piPf0 (26.6.2019.)

[18] https://en.wikipedia.org/wiki/Stepper_motor (26.6.2019.)

[19] https://learn.sparkfun.com/tutorials/easy-driver-hook-up-guide/all (26.6.2019.)

SAŽETAK

Sustav kontrole visine elektrode neophodan je za rad CNC plazma rezača. Ukoliko ne bi imali takav sustav, došlo bi do oštećenja elektrode te rezovi ne bi bili čisti. Cijene takvih sustava često prelaze iznose od nekoliko tisuća kuna. To je bila glavna motivacija za izradu ovog rada.

U radu je opisan način izrade takvog sustava. Postignuta je točnost mjerenja napona elektrode od 99.8%. Rad je u potpunosti realiziran no zbog nedostataka opreme nije testiran na pravom plazma rezaču već je njegov rad simuliran te se sustav ponašao prema očekivanjima.

Ključne riječi: plazma rezač, elektroda, THC, CNC, Mach3, step motor, Arduino, naponsko djelilo

ABSTRACT

Torch height control

An electrode height control system is necessary for the operation of a CNC plasma cutter. Not having such a system would result in electrode damage and unclean cuts. The prices of such systems often exceed the amount of several thousand kunas. That was the main motivation for making this work.

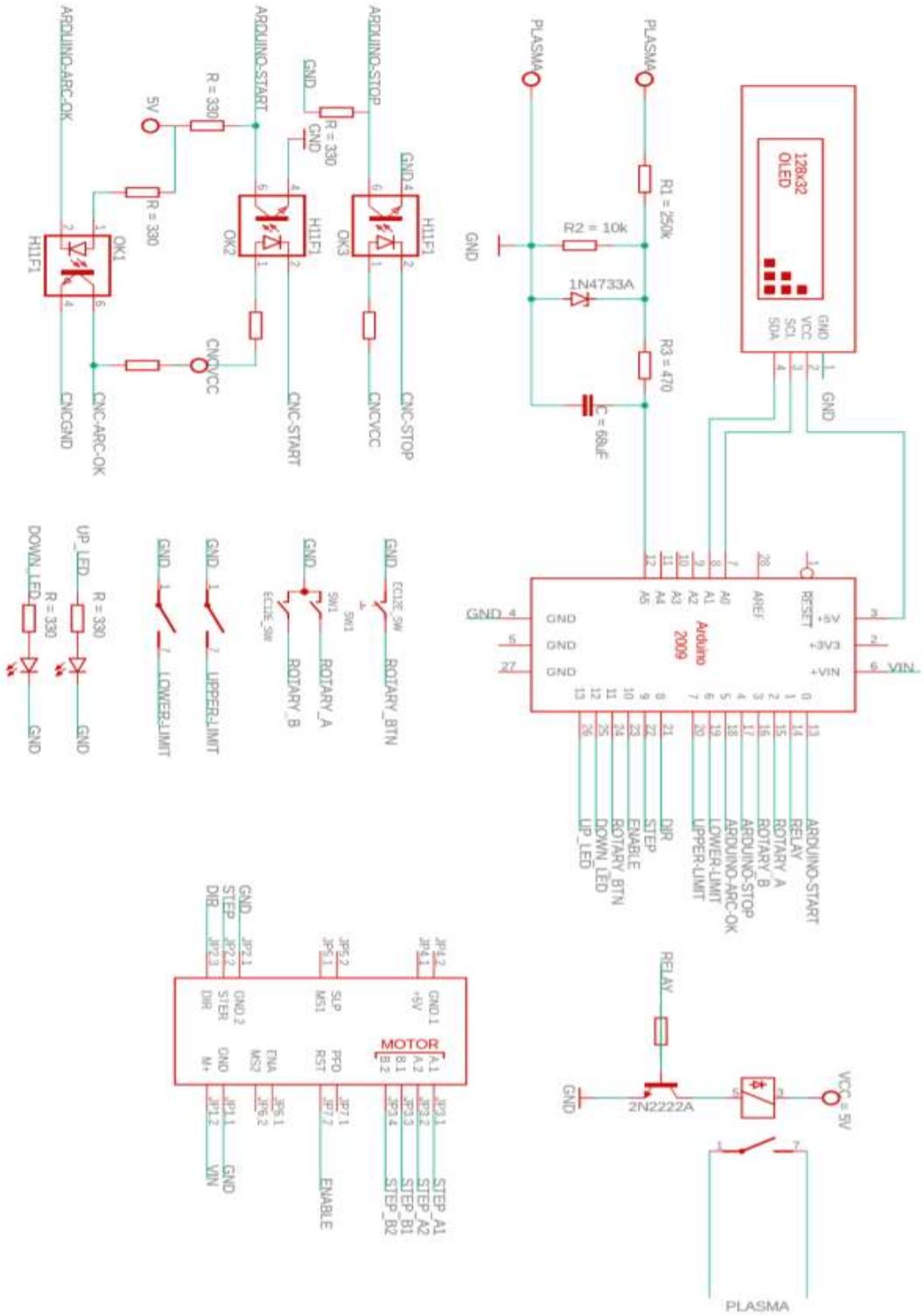
The paper describes how to design such a system. An accuracy of 99.8% electrode voltage measurement was achievet. The assignment was fully realized but due to the lack of equipment it was not tested on a real plasma cutter but its operation was simulated and the system behaved as expected.

Keywords: plasma cutter, electrode, THC, CNC, Mach3 stepper motor, Arduino, voltage divider

ŽIVOTOPIS

Jakov Sudar je rođen 2. srpnja 1997. godine u Vinkovcima. Osnovnu školu završio je 2012. godine u Ivankovu. Nakon toga upisao je srednju školu „Tehnička škola Ruđera Boškovića Vinkovci“ u Vinkovcima. Srednju školu završava 2016. godine te izravnim upisom upisuje preddiplomski studij Računarstva na Fakultetu elektrotehnike, računarstva i informacijskih tehnologija Osijek u Osijeku.

Prilog 1


ARDUINO-ARC-OK
ARDUINO-START
ARDUINO-STOP
5V
R = 330
GND
H11F1
OK1
OK2
OK3
CNCVCC
CNCGND
CNC-ARC-OK
CNC-START
CNC-STOP
PLASMA
R1 = 250k
R2 = 10k
1N4733A
R3 = 470
C = 68uF
128x32 OLED
SCL
SDA
VCC
Arduino 2009
RESET
AREF
+5V
+3V3
+VIN
VIN
ARDUINO-START
RELAY
ROTARY A
ROTARY B
ARDUINO-ARC-OK
LOWER-LIMIT
UPPER-LIMIT
DIR
STEP
ENABLE
ROTARY BTN
DOWN_LED
UP_LED
DOWN_LED
UP_LED
ROTARY_A
ROTARY_B
ROTARY_BTN
LOWER-LIMIT
UPPER-LIMIT
MOTOR
STEP_A1
STEP_A2
STEP_B1
STEP_B2
2N2222A
VCC = 5V

Prilog 2

```
#include <PinChangeInterrupt.h>
#include <LiquidCrystal_I2C.h>
LiquidCrystal_I2C lcd(0x27,20,4);

#define start 0 //Ulazni signal za pokretanje rezanja
#define relay 1 //Izlazni signal za paljenje releja
#define pinA 2  //Ulazni signal sa enkodera
#define pinB 3  //Ulazni signal sa enkodera
#define stop 4  //Ulazni signal za prekidanje rezanja
#define ArcOk 5 //Izlazni signal za uspješno uspostavljen luk
#define down 6  //Ulazni signal sa kranjeg prekidača
#define upper 7 //Ulazni signal sa krajnjeg prekidača
#define dir 8   //Izlazni signal za step driver, određuje smjer vrtnje motora
#define stp 9   //Izlazni signal za step driver, njegova promjena pravi jedan korak motora
#define enable 10    //Izlazni signal za step driver koji omogućuje/onemogućuje kretanje motora
#define buttonPin 11 //Ulazni signal sa tipkala enkodera
#define downLed 12   //Izlazni signal za paljenje LEDice
#define upLed 13     //Izlazni signal za paljenje LEDice

uint8_t arrow[8] = {0x0, 0x04 ,0x06, 0x1f, 0x06, 0x04, 0x00, 0x00}; //znak za pokazivač
//Vrijednosti otpornika
float R1 = 250000;
float  R2 = 10000;
//menu varijable
int menu = 0;
//Varijable u nastavku su tipa byte jer spremaju vrijenosti manje od 255 te nema potrebe deklarirati ih kao int što zauzima dodatnu memoriju
volatile byte aFlag = 0; // Pomoćna varijabla za razlikovanje smjera okretanja enkodera
volatile byte bFlag = 0; // Pomoćna varijabla za razlikovanje smjera okretanja enkodera(suprotan smjer od aFlag)
volatile byte encoderPos = 1; //Varijabla koja sprema poziciju enkodera.
volatile byte oldEncPos = 1; //varijabla koja sprema staru poziciju enkodera. Koristi se za detekciju pomaka enkodera. Smanjuje broj nepotrebnih osvježavanja displaya
volatile byte reading = 0; //Pomoćna varijabla za spremanje stanja pinova enkodera
byte oldButtonState = HIGH;  //Pomoćna varijabla za detekciju promjene stanja tipkala. Korištena zbog debouncinga
unsigned long debounceTime = 30;  //Trajanje debouncinga
unsigned long btnTime;  //Varijabla u koju se sprema vrijeme posljednjeg pritiska tipkala. Također debouncing
boolean buttonPressed = 0; //Zastavica za pritisnuto tipkalo
byte Mode = 0;   //Varijabla koja sprema odabrani meni
const byte modeMax = 5; //broj menija koje imamo
//varijable u nastavku su postavke koje korisnik odabire
byte initialHeight = 3;
```

```
byte setVoltage = 4;
bool enabled = true;
volatile bool started = false; //Zastavica za Start signal s CNC sustava
bool display= false; //Varijabla koja pamti je li displaya za rezanje postavljen ili ne. Uklanja velik broj nepotrebnog osvježavanja displaya
int i;
void setup() {
   pinMode(stp, OUTPUT);
   pinMode(dir, OUTPUT);
   pinMode(down, INPUT_PULLUP);
   pinMode(upper, INPUT_PULLUP);
   pinMode(start, INPUT_PULLUP);
   pinMode(stop, INPUT_PULLUP);
   pinMode(upLed,OUTPUT);
   pinMode(downLed,OUTPUT);
   pinMode(enable, OUTPUT);
   pinMode (buttonPin, INPUT_PULLUP);
   pinMode(pinA, INPUT_PULLUP);
   pinMode(pinB, INPUT_PULLUP);
   //Kod u nastavku postavlja intterupt pozive na pinove
   attachPCINT(digitalPinToPCINT(pinA),PinA,RISING);
   attachPCINT(digitalPinToPCINT(pinB),PinB,RISING);
   attachPCINT(digitalPinToPCINT(start),StartCuttingISR,FALLING);
   attachPCINT(digitalPinToPCINT(stop),StopCuttingISR,FALLING);
   digitalWrite(enable,HIGH); //Onemogućavanje kretanja motora
   lcd.init();
   lcd.backlight();
   lcd.createChar(1, arrow);   //Kreiranje pokazivača
   lcd.home();
   lcd.clear();
   //Splash screen
   lcd.setCursor(5,0);
   lcd.print("THC");
   lcd.setCursor(2,1);
   lcd.print(" Jakov Sudar ");
   delay(3500);
}
void loop() {
   rotaryMenu();
   while(started){
      if(!display){
         digitalWrite(enable,!enabled);
         CuttingDisplay();
         InitialPosition(initialHeight);
      }
      Cutting();
   }
```

```
}
void StartCuttingISR(){
   started= true;
}
void StopCuttingISR(){
   digitalWrite(enable,HIGH);
   digitalWrite(ArcOk,LOW);
   started= false;
   display= false;
   encoderPos=1;
   sei();
}
void Homing(){
   digitalWrite(enable,LOW);
   while(digitalRead(upper)== HIGH){
      Up(700);
   }
   digitalWrite(enable,HIGH);
}
void Up(int speed){
   digitalWrite(upLed,HIGH);
   digitalWrite(dir, HIGH);
   digitalWrite(stp, HIGH);
   delayMicroseconds(speed);
   digitalWrite(stp, LOW);
   delayMicroseconds(speed);
   digitalWrite(upLed,LOW);
}
void Down(int speed){
   digitalWrite(downLed,HIGH);
   digitalWrite(dir, LOW);
   digitalWrite(stp, HIGH);
   delayMicroseconds(speed);
   digitalWrite(stp, LOW);
   delayMicroseconds(speed);
   digitalWrite(downLed,LOW);
}
//Funkcija za očitavanje referentnog napona koji će biti korišten pri ADC
//Preuzeto sa https://www.instructables.com/id/Secret-Arduino-Voltmeter/
long readVcc() {
   long result;
   ADMUX = _BV(REFS0) | _BV(MUX3) | _BV(MUX2) | _BV(MUX1);
   delay(2);
   ADCSRA |= _BV(ADSC);
   while (bit_is_set(ADCSRA,ADSC));
   result = ADCL;
   result |= ADCH<<8;
```

```
  result = 1125300L / result;
  return result;
}
void InitialPosition(int mm){ //mm - pocetna pozicija u milimetrima
  digitalWrite(enable,LOW);
  //Spušta elektrodu dolje dok se ne aktivira prekidač
  while(digitalRead(down)==HIGH){
    Down(700);
  }
  //Napravi početni odmak
  for(i=0;i<mm*25+502;i++){      //ukupan broj koraka; "prazni hod"+ zadana pocetna vrijednost
    Up(700);
  }
  digitalWrite(enable,HIGH);
  //Upali relej
  digitalWrite(relay,HIGH);
}
void Cutting(){
  unsigned int sensorValue;
  double voltage;
  double Vcc;
  digitalWrite(enable,!enabled);
  Vcc = readVcc()/1000.0;
  sensorValue = analogRead(0);
  voltage = (((sensorValue / 1024.0) * Vcc)/(R2/(R1+R2)));
  //Ako je uspostavljen luk šalje se signal CNC sustavu da krene sa radom
  if(voltage>setVoltage*0.8){
    digitalWrite(ArcOk,HIGH);
  }else
  if(voltage<0.7*setVoltage){ //Ako je napon pao ispod te razine znači da se luk prekinuo
    digitalWrite(ArcOk,LOW); //Šalje se signal CNC sustavu da prestane sa radom
    digitalWrite(relay,LOW); //Gasi se elektroda
    InitialPosition(initialHeight);  //Pokušava se uspostaviti luk opet
    delay(700); //Vrijeme da se luk stigne uspostaviti
  }
  if(voltage>setVoltage*1.03){     //Ako je napon prevelik, spusti elektrodu
    for(int steps=0;steps<5;steps++)
    Down(800);
  }else
  if(voltage<setVoltage*0.97){    //Ako je napon premal, podigni elektrodu
    for(int steps=0;steps<5;steps++)
    Up(800);
  }
  lcd.setCursor(8,1);   //Osvježavanje vrijednosti napona na displayu
  lcd.print(String(voltage));
}
```

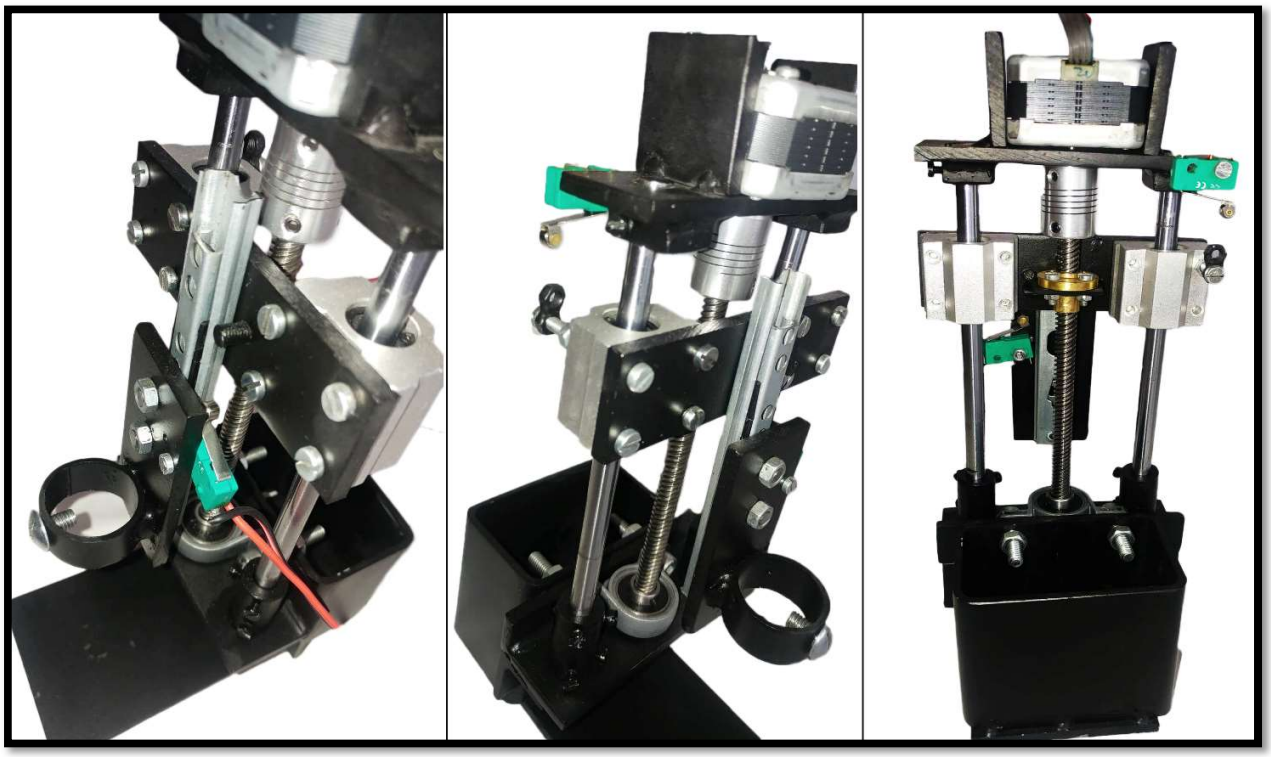
```
void CuttingDisplay(){
   lcd.clear();
   lcd.setCursor(0,0);
   lcd.print("Set voltage:"+String(setVoltage));
   lcd.setCursor(0,1);
   lcd.print("Voltage:");
   display= true;  //varijabla koja označava da je display rezanja postavljen te se više ova funkcija neće
pozivati.
}
void rotaryMenu() { //Funkcija za menue
   if(oldEncPos != encoderPos) {
      //Kod u nastavku služi za osvježavanje postavljene vrijednosti postavke na displayu
      switch (menu){
         case 1:
            lcd.setCursor(12,1);
            lcd.print(String(encoderPos)+"mm");
            if(encoderPos<10){
               lcd.setCursor(15,1);
               lcd.print(" ");
            }
            if(encoderPos>15){
               encoderPos=2;
               lcd.setCursor(15,1);
               lcd.print(" ");
               lcd.setCursor(12,1);
               lcd.print(String(encoderPos)+"mm");
            }else if(encoderPos<2){
               encoderPos=15;
               lcd.setCursor(12,1);
               lcd.print(String(encoderPos)+"mm");
            }
            break;
         case 2:
            lcd.setCursor(12,1);
            lcd.print(String(encoderPos)+"V");
            if(encoderPos<100){
               lcd.setCursor(15,1);
               lcd.print(" ");
            }
            if(encoderPos>=131){
               encoderPos=90;
               lcd.setCursor(15,1);
               lcd.print(" ");
               lcd.setCursor(12,1);
               lcd.print(String(encoderPos)+"V");
            }else if(encoderPos<90){
               encoderPos=130;
```

```
        lcd.setCursor(12,1);
        lcd.print(String(encoderPos)+"V");
    }
    break;
case 3:
    lcd.setCursor(10,1);
    if(encoderPos%2==0){
        enabled=true;
        }else
        {
        enabled=false;
        }
    lcd.print(enabled?"True ":"False");
    break;
case 0:
//Kod u nastavku pomjera pokazivač po menijima
    switch (encoderPos)
    {
        case 1:
            lcd.clear();
            lcd.setCursor(0,0);
            lcd.write(1);   //Crtanje pokazivača
            lcd.print("Initial height");
            lcd.setCursor(1,1);
            lcd.print("Set Voltage");
            break;
        case 2:
            lcd.clear();
            lcd.setCursor(1,0);
            lcd.print("Initial height");
            lcd.setCursor(0,1);
            lcd.write(1);
            lcd.print("Set Voltage");
            break;
        case 3:
            lcd.clear();
            lcd.setCursor(0,0);
            lcd.write(1);
            lcd.print("Enable THC");
            lcd.setCursor(1,1);
            lcd.print("Check wiring");
            break;
        case 4:
            lcd.clear();
            lcd.setCursor(1,0);
            lcd.print("Enable THC");
            lcd.setCursor(0,1);
```

```
                lcd.write(1);
                lcd.print("Check wiring");
                break;
            case 5:
                lcd.clear();
                lcd.setCursor(0,0);
                lcd.write(1);
                lcd.print("Homing");
            default:
                break;
          }
          break;
        default:
          break;
      }
      oldEncPos = encoderPos;
    }
    byte buttonState = digitalRead(buttonPin);
    if (buttonState != oldButtonState){
      if (millis () - btnTime >= debounceTime){ // Provjera je li pritisak tipkala stvaran ili je pritisak
od boncing efekta
        btnTime = millis ();  //pomocna varijabla za debounciranje
        oldButtonState =  buttonState;
        if (buttonState == LOW){
          buttonPressed = 1;
        }
        else {
          buttonPressed = 0;
        }
      }
    }
    //Mode==0 znači da smo u glavnom meniju
    if (Mode == 0) {
      if (encoderPos<1) encoderPos = modeMax; //Provjera pozicije enkodera te ispravljanje ukoliko
je potrebno
      else if (encoderPos > modeMax) encoderPos = 1;
      if (buttonPressed){
        Mode = encoderPos; //Na pritisak tipkala, zapamti koji menu je odabran
        buttonPressed = 0;
        //Početni zasloni unutar opcije
        if (Mode == 1) {
          lcd.clear();
          lcd.setCursor(1,0);
          lcd.print("Initial height");
          lcd.setCursor(2,1);
          lcd.print("Set Value:"+ String(initialHeight)+"mm");
          encoderPos = initialHeight;
```

```
            menu=1;
        }
        if (Mode == 2) {
            lcd.clear();
            lcd.setCursor(3,0);
            lcd.print("Voltage");
            lcd.setCursor(2,1);
            lcd.print("Set Value:"+ String(setVoltage)+"  V");
            encoderPos = setVoltage;
            menu=2;
        }
        if (Mode == 3) {
            lcd.clear();
            lcd.setCursor(3,0);
            lcd.print("Enable THC");
            lcd.setCursor(2,1);
            lcd.print("Enabled:"+ String(enabled? "True":"False"));
            menu=3;
        }
        if(Mode==4){
            lcd.clear();
            lcd.setCursor(3,0);
            lcd.print("Press upper");
            lcd.setCursor(6,1);
            lcd.print("limit");
            while(digitalRead(upper)){ //Dok korisnik ne pritisne gornje tipkalo, ispisuj točkice kao
loader
                lcd.setCursor(11,1);
                lcd.print(".  ");
                delay(250);
                lcd.setCursor(11,1);
                lcd.print("..");
                delay(250);
                lcd.setCursor(11,1);
                lcd.print("...");
                delay(250);
            }
            lcd.clear();
            lcd.setCursor(4,0);
            lcd.print("Upper OK");
            delay(1000);
            lcd.clear();
            lcd.setCursor(3,0);
            lcd.print("Press down");
            lcd.setCursor(6,1);
            lcd.print("limit");
```

```
            while(digitalRead(down)){ //Dok korisnik ne pritisne donje tipkalo, ispisuj točkice kao loader
                lcd.setCursor(11,1);
                lcd.print(".  ");
                delay(250);
                lcd.setCursor(11,1);
                lcd.print("..");
                delay(250);
                lcd.setCursor(11,1);
                lcd.print("...");
                delay(250);
            }
            lcd.clear();
            lcd.setCursor(4,0);
            lcd.print("Down OK");
            delay(1000);
            buttonPressed=1;
          }
          if(Mode==5){
            Homing();
            buttonPressed=1;
          }
        }
    }
  //Kod u nastavku nakon postavljanja vrijednosti postavke te pritiskanja
  //tipkala kao potvrde, sprema postavljenu vrijednost u varijablu te prikazuje glavni meni
    if (Mode == 1 && buttonPressed) {
      initialHeight = encoderPos;
      mainMenu();
    }else
    if (Mode == 2 && buttonPressed) {
      setVoltage = encoderPos;
      mainMenu();
    }else
    if (Mode == 3 && buttonPressed){
      mainMenu();
    }else
    if (Mode == 4 && buttonPressed){
      mainMenu();
    }else
    if (Mode == 5 && buttonPressed){
      mainMenu();
    }
}
void mainMenu(){
    encoderPos = 1; //Postavlja se u 1 kako bi se pokazivač na početnom ekranu postavio na prvi izbornik
```

```
    //resetiranje varijabli
    buttonPressed = 0;
    Mode = 0;
    menu = 0;
    lcd.clear();
    lcd.setCursor(0,0);
    lcd.write(1);
    lcd.print("Initial height");
    lcd.setCursor(1,1);
    lcd.print("Set Voltage");
}
//Funkcija koja se okida na pomak enkodera
void PinA(){
    cli(); //Zaustavlja sve ostale intterupte
    //U nastavku su registarske funkcije. Korištenje registarskih funkcija utječe na brzinu izvođenja koda. Kod intterupt funkcija želimo
    //što prije završiti funkciju
    reading = PIND & 0xC; //PIND su svi pinovi od 0-7. Bitwise naredbom and odbacujemo sve pinove osim pinova 2 i 3. 0xc je 1100 binarno.
    if(reading == B00001100 && aFlag) { //Provjera jesu li oba pina (2 i 3) postavljeni u high
        encoderPos --; //pozicija enkodera se umanjuje
        bFlag = 0; //resetiranje zastavica
        aFlag = 0;
    }
    else if (reading == B00000100) bFlag = 1; //Ako nisu oba pina postavljena u high, znači da očekujemo da se na idući poziv intterupt funkcije
                                              //drugi pin postavi u high.
    sei(); //Ponovno omogućavanje intterupta
}
//Funkcija koja se okida na pomak enkodera
void PinB(){
    //vidi komentare za prethodnu funkciju
    cli();
    reading = PIND & 0xC;
    if (reading == B00001100 && bFlag) {
        encoderPos ++;
        bFlag = 0;
        aFlag = 0;
    }
    else if (reading == B00001000) aFlag = 1;
    sei();
}
```

Prilog 3