

# Konstrukcija i upravljanje uređajem za automatsko unakrsno namatanje kvadratne zavojnice

---

Grljević, Oliver

Undergraduate thesis / Završni rad

2024

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

*Permanent link / Trajna poveznica:* <https://urn.nsk.hr/urn:nbn:hr:235:898432>

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

*Download date / Datum preuzimanja:* **2024-11-24**

*Repository / Repozitorij:*

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



SVEUČILIŠTE U ZAGREBU  
FAKULTET STROJARSTVA I BRODOGRADNJE

# ZAVRŠNI RAD

**Oliver Grljević**

Zagreb, 2024.

SVEUČILIŠTE U ZAGREBU  
FAKULTET STROJARSTVA I BRODOGRADNJE

# ZAVRŠNI RAD

Mentori:

Prof. dr. sc. Mladen Crneković, dipl. ing.

Student:

Oliver Grljević

Zagreb, 2024.

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

Zahvaljujem se prof.dr.sc. Mladenu Crnekoviću na mentorstvu i usmjeravanju u izradi završnog rada.

Oliver Grljević



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

## ZAVRŠNI ZADATAK

Student: **Oliver Grljević** JMBAG: **0035239580**

Naslov rada na hrvatskom jeziku: **Konstrukcija i upravljanje uređajem za automatsko unakrsno namatanje kvadratne zavojnice**

Naslov rada na engleskom jeziku: **Design and control of a device for automatic cross-winding of a square coil**

### Opis zadatka:

Sigurnost rada mnogih postrojenja (posebno nuklearnih) iziskuje nerazorno ispitivanje cijevi, zavarenih spojeva i ostalih elemenata sustava. U mjernim uređajima za tu namjenu nalazi se zavojnica s jezgrom od 1 mm<sup>3</sup> koja ima funkciju odašiljanja i primanja signala iz čega se zaključuje o mogućim nepravilnostima u materijalu. Takve je zavojnice zbog malih dimenzija teško proizvesti, a danas se uglavnom namataju ručno uz pomoć mikroskopa i posebnih naprava.

Potrebno je konstruirati uređaj za automatsko unakrsno namatanje bakrene žice promjera 0,08 mm na kvadratnu jezgru dimenzija 1x1x1 mm.

U radu je potrebno:

- Osmisliti konstrukciju mehaničkog dijela uređaja.
- Odabrati pogonske i mjerne elemente sustava.
- Odabrati upravljački sustav uređaja i predložiti strukturu upravljačkog programa.
- Procijeniti trošak proizvodnje prototipa.

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

Zadatak zadan:

24. 4. 2024.

Datum predaje rada:

**2. rok (izvanredni):** 11. 7. 2024.  
**3. rok:** 19. i 20. 9. 2024.

Predviđeni datumi obrane:

**2. rok (izvanredni):** 15. 7. 2024.  
**3. rok:** 23. 9. – 27. 9. 2024.

Zadatak zadao:

prof. dr. sc. Mladen Cmeković

Predsjednik Povjerenstva:

izv. prof. dr. sc. Petar Čurković

## SADRŽAJ

SADRŽAJ .....	I
POPIS SLIKA .....	II
POPIS TABLICA.....	III
SAŽETAK.....	IV
SUMMARY .....	V
1. UVOD.....	1
1.1. Princip rada .....	2
1.2. Cilj.....	2
2. Konstrukcija uređaja.....	3
2.1. Namatalica .....	3
2.2. Sustav rotacije feritne jezgre.....	4
2.3. Sustav preciznog pozicioniranja žice.....	5
2.4. Sustav promjene osi namatanja.....	7
2.5. Sustav za održavanje napetosti žice .....	9
3. UPRAVLJAČKI PROCES.....	11
3.1. Izbor upravljačke jedinice .....	11
3.2. Rad namatalice .....	12
3.3. Upravljanje motorima .....	14
3.3.1. Regulacija položaja istosmjernog motora .....	15
3.3.2. Upravljanje položajem koračnog motora.....	16
4. PROGRAM UREĐAJA .....	18
5. PROCJENA TROŠKOVA RAZVOJA .....	21
6. ZAKLJUČAK.....	24
LITERATURA.....	25
PRILOZI.....	26

**POPIS SLIKA**

Slika 1	Prikaz namotane zavojnice veličine 1mm <sup>3</sup> .....	1
Slika 2	Feritna jezgra .....	1
Slika 3	Izgled namatalice .....	3
Slika 4	Sustav rotacije feritne jezgre .....	4
Slika 5	Sustav preciznog pozicioniranja žice .....	6
Slika 6	Izgled otvora za vođenje žice .....	6
Slika 7	Sustav promjene osi namatanja .....	7
Slika 8	Izgled prihvatnice .....	7
Slika 9	Karakteristike motora FIT0487 [6] .....	8
Slika 10	Klizni prsten MT1233-S06 [7] .....	8
Slika 11	Sustav za održavanje napetosti žice .....	9
Slika 12	Pakna s magnetom u središtu .....	10
Slika 13	Arduino Mega Rev3 [10] .....	11
Slika 14	Adafruit Motor Shield v2 [11] .....	11
Slika 15	Dijagram toka namatalice .....	13
Slika 16	Dijagram toka sučelja .....	13
Slika 17	Blok dijagram namatalice .....	14
Slika 18	Izgled PWM signala [14] .....	14
Slika 19	H-most [15] .....	15
Slika 20	Primjer odziva PI regulatora na skokovitu pobudu [16] .....	15
Slika 21	Dijagram toka sustava za upravljanje istosmjernim motorom .....	16
Slika 22	Bipolarni koračni motor i način spajanja njegovih namota [17] .....	17
Slika 23	Dijagram toka sustava za upravljanje koračnim motorom .....	17
Slika 24	Arudino kod rada namatalice .....	18
Slika 25	Arduino kod rada sučelja .....	19
Slika 26	Arduino kod rada PI regulatora istosmjernog motora .....	20

---

**POPIS TABLICA**

Tablica 1 Karakteristike motora 103H5210-5140 [1] ..... 5  
Tablica 2 Troškovnik..... 21



**SAŽETAK**

Ovaj rad opisuje razvoj uređaja za automatsko unakrsno namatanje kvadratne zavojnice, ključnog za nerazorno ispitivanje materijala u industrijama poput nuklearne energetike. Uređaj je konstruiran kako bi osigurao visoku preciznost namatanja tanke bakrene žice na feritnu jezgru dimenzija 1mm<sup>3</sup>, koristeći sustav koji se sastoji od tri motora za rotaciju jezgre, translaciju žice i promjenu osi namatanja.

Rad također uključuje odabir odgovarajućih mehaničkih i pogonskih komponenti, poput koračnih i DC motora, te primjenu PI regulatora za precizno upravljanje položajem. Opisana je i konstrukcija podsustava za održavanje napetosti žice, ključnog za stabilnost procesa namatanja. Također je opisan programski kod kojeg uređaj izvršava. Na kraju, procijenjeni su troškovi razvoja prototipa, uzimajući u obzir cijene materijala, izrade i ljudskog rada.

Ovaj rad pruža sveobuhvatan uvid u tehničke izazove i rješenja pri razvoju uređaja za automatsko namatanje visokopreciznih zavojnica.

Ključne riječi: Mehanička konstrukcija; mikroupravljač, programiranje; regulacija; aditivne tehnologije; istosmjerni motori

---

**SUMMARY**

This thesis describes the development of a device for automatic cross-winding of a square coil, essential for non-destructive material testing in industries such as nuclear energy. The device is designed to ensure high precision in winding thin copper wire onto a core measuring 1 mm<sup>3</sup>, using a system consisting of three motors for core rotation, wire translation, and axis change during winding.

The work also involves selecting appropriate mechanical and drive components, such as stepper and DC motors, and implementing a PI controller for precise position control. The construction of a subsystem for maintaining wire tension, crucial for the stability of the winding process, is described as well. Furthermore, the program code that the device executes is described. Finally, the development costs of the prototype are estimated, considering material, manufacturing, and labor expenses.

This thesis provides a comprehensive overview of the technical challenges and solutions in developing a device for automatic winding of high-precision coils.

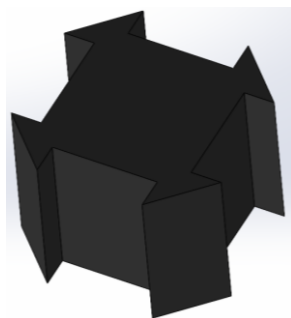
## 1. UVOD

Nerazorno ispitivanje materijala ključno je za održavanje sigurnosti i pouzdanosti brojnih industrijskih postrojenja, uključujući nuklearne elektrane, gdje kvarovi mogu imati ozbiljne posljedice. Ova metoda omogućava detekciju unutarnjih nepravilnosti u materijalima bez njihovog oštećivanja, čime se osigurava dugotrajna funkcionalnost i sigurnost ključnih komponenti. U tom kontekstu, zavojnice igraju važnu ulogu kao temeljni dio sonde koje se koriste za primanje i odašiljanje signala. Njihova sposobnost da induciraju i detektiraju elektromagnetska polja omogućuje precizno mjerenje promjena u materijalu koje ukazuju na prisutnost oštećenja ili nepravilnosti.



**Slika 1 Prikaz namotane zavojnice veličine 1mm<sup>3</sup>**

Zbog male veličine i složenosti, proizvodnja kvadratnih zavojnica s jezgrom od 1 mm<sup>3</sup> predstavlja izazov, osobito kada je riječ o postizanju visoke preciznosti u namatanju žice promjera 0,08 mm. Trenutno se ove zavojnice proizvode ručno uz pomoć mikroskopa, što je dugotrajan proces koji nosi rizik od ljudske pogreške i varijabilnosti u kvaliteti. Elektronički uređaji koji koriste ove zavojnice ovise o visokoj točnosti i stabilnosti, budući da i najmanja odstupanja u namatanju mogu utjecati na njihovu funkcionalnost i pouzdanost.



**Slika 2 Feritna jezgra**

## **1.1. Princip rada**

Ovaj uređaj za automatsko unakrsno namatanje kvadratne zavojnice koristi tri motora kako bi precizno namotao tanku bakrenu žicu na jezgru zavojnice.

Prvi motor rotira feritnu jezgru zavojnice oko jedne osi. Kada se jezgra rotira, žica se namata oko nje. Istovremeno, drugi motor pokreće sustav koji pomiče žicu po duljini jezgre, tako da se ona ravnomjerno namata i formira jedan uredan red namota.

Kada se cijeli red popuni, treći motor dolazi u igru. On rotira jezgru zavojnice za 90 stupnjeva, što znači da se žica sada počinje namatati oko druge osi jezgre. Proces namatanja na ovoj novoj osi funkcionira na isti način.

Nakon što je red na ovoj osi popunjen, treći motor ponovno rotira jezgru za 90 stupnjeva, vraćajući je u prvotni položaj. Ovaj ciklus se ponavlja sve dok se ne postigne željeni broj slojeva namota na jezgri zavojnice.

Na ovaj način uređaj osigurava da žica bude precizno namotana, bez ručne intervencije, što je ključno za postizanje točne i funkcionalne zavojnice.

## **1.2. Cilj**

Cilj ovog završnog rada je razviti konstrukciju uređaja za automatsko unakrsno namatanje zavojnica, čime bi se postigla veća produktivnost procesa proizvodnje istih. Osim konstrukcije mehaničkog dijela uređaja, rad će obuhvatiti odabir pogonskih i mjernih komponenti, te definirati upravljački sustav koji će omogućiti optimalno funkcioniranje uređaja. Na kraju, bit će napravljena procjena troškova proizvodnje prototipa, što će pomoći u evaluaciji komercijalne isplativosti ovakvog rješenja.

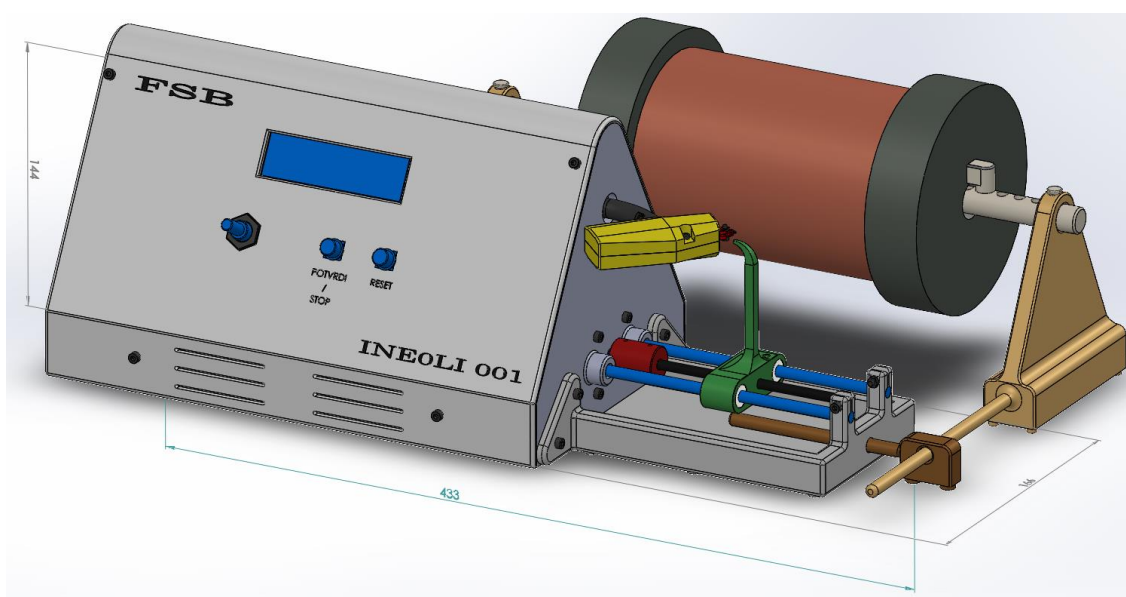
## 2. Konstrukcija uređaja

U ovom poglavlju opisana je konstrukcija namatalice te su odabrani svi potrebni mehanički i pogonski elementi. U uvodnom je dijelu predstavljena ideja i opisan princip rada cijelog uređaja, a ovom poglavlju će biti daljnje opisana izvedba i realizacija podsustava uređaja. Glavni zadatak konstrukcije je osigurati tri stupnja slobode gibanja i težiti nuli u odstupanjima sa svrhom da elektronika može raditi kako je zamišljena. Za potrebe konstrukcije CAD modela namatalice korišten je programski paket SolidWorks.

### 2.1. Namatalica

Konstrukcija namatalice [Slika 3] razvijena je s naglaskom na eliminaciju momenta inercije i njegovih posljedica, poput savijanja pojedinih elemenata i impulsnih sila koje nastaju zbog naglog pokretanja i velikih masa komponenti. Da bi se postigla visoka preciznost i pouzdanost, posebna pažnja posvećena je smanjenju mase svih dijelova sustava. Naime, dizajn je optimiziran kako bi se smanjila inercija i minimizirali neželjeni efekti poput deformacija. Osim toga, određeni dijelovi su konstruirani s visokim faktorom sigurnosti kako bi se dodatno osigurala čvrstoća i stabilnost, smanjujući rizik od savijanja i deformacija pod opterećenjem.

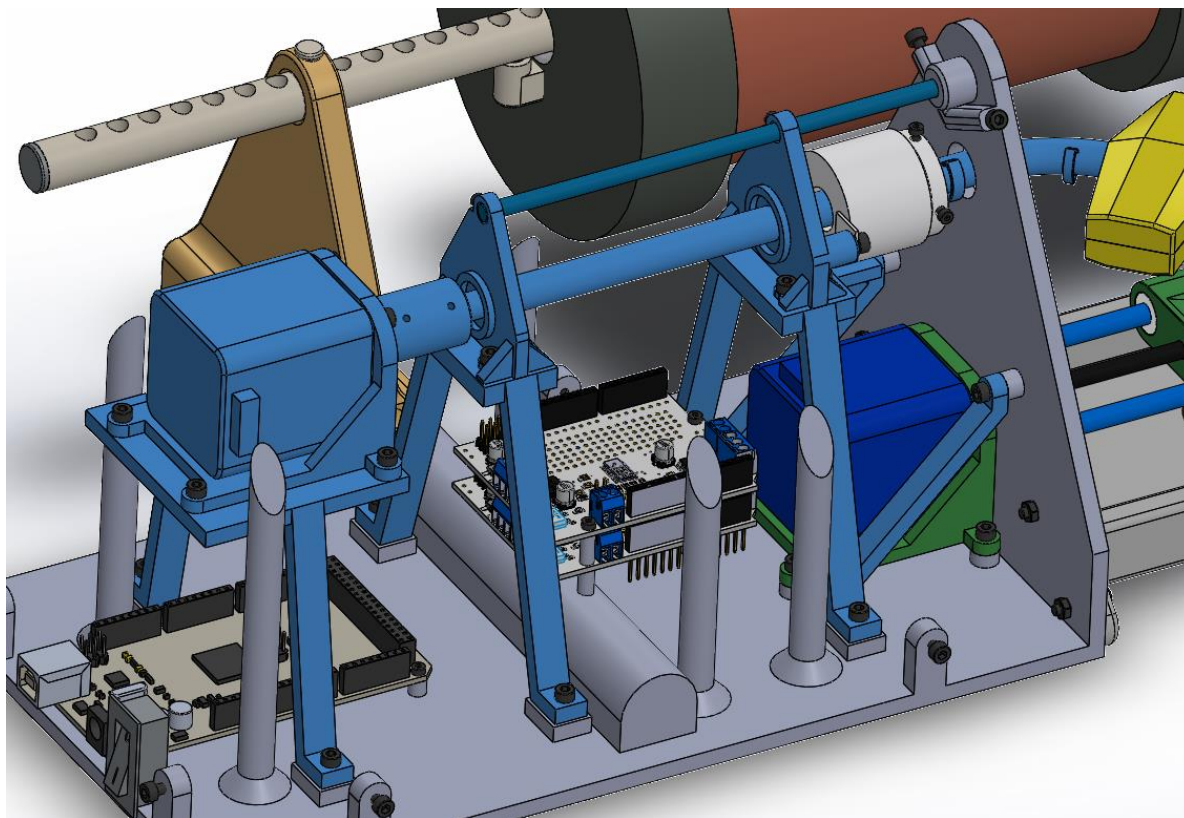
Ovaj pristup omogućuje ne samo postizanje optimalnih performansi već i dugoročnu pouzdanost sustava. Konstrukcija je također uzela u obzir jednostavnost korisničkog sučelja, omogućujući lako postavljanje i praćenje procesa namatanja.



Slika 3 Izgled namatalice

## 2.2. Sustav rotacije feritne jezgre

Primarna funkcija ovog podsustava je rotacija ferita, što omogućuje precizno namatanje bakrene žice oko njega.



Slika 4 Sustav rotacije feritne jezgre

Sustav se sastoji od koračnog motora koji preko elastične spojke prenosi okretni moment na polugu. Koračni motor je odabran zbog svoje sposobnosti za vrlo precizno upravljanje pozicijom, jer se kreće u unaprijed definiranim koracima. Konkretno, izabran je model 103H5210-5140 [Tablica 1] zbog svog velikog okretnog momenta, što je ključno za stabilno održavanje položaja i smanjenje vibracija. Taj motor ima 200 koraka po punom okretaju (što znači da motor radi kutni zakret od  $1,8^\circ$  po koraku) što nudi visoku rezoluciju i preciznost u kontroliranju rotacije, što doprinosi točnosti namatanja. Također, jedan od glavnih preduvjeta za izbor motora je bio da sadrži enkoder kako bi računalo dobivalo povratnu informaciju o realnoj poziciji motora.

**Tablica 1 Karakteristike motora 103H5210-5140 [1]**

Bipolar winding, Lead wire type

Model number		Holding torque at 2-phase energization	Rated current	Wiring resistance	Winding inductance	Rotor inertia	Mass (Weight)	Motor length (L)
Single shaft	Dual shaft	[N·m (oz·in) min.]	A/phase	$\Omega$ /phase	mH/phase	[ $\times 10^{-4}$ kg·m <sup>2</sup> (oz·in <sup>2</sup> )]	[kg (lbs)]	mm (in)
103H5205-5040	103H5205-5010	0.23 (32.57)	0.25	54	78	0.036 (0.20)	0.23 (0.51)	33 (1.25)
103H5205-5140	103H5205-5110	0.25 (35.40)	0.5	13.4	23.4	0.036 (0.20)	0.23 (0.51)	33 (1.25)
103H5205-5240	103H5205-5210	0.265 (37.53)	1	3.4	6.5	0.036 (0.20)	0.23 (0.51)	33 (1.25)
103H5208-5040	103H5208-5010	0.35 (49.56)	0.25	66	116	0.056 (0.31)	0.29 (0.64)	39 (1.54)
103H5208-5140	103H5208-5110	0.38 (53.81)	0.5	16.5	34	0.056 (0.31)	0.29 (0.64)	39 (1.54)
103H5208-5240	103H5208-5210	0.39 (55.23)	1	4.1	9.5	0.056 (0.31)	0.29 (0.64)	39 (1.54)
103H5209-5040	103H5209-5010	0.38 (53.81)	0.25	71.4	133	0.062 (0.34)	0.31 (0.68)	41 (1.61)
103H5209-5140	103H5209-5110	0.41 (58.06)	0.5	18.2	39	0.062 (0.34)	0.31 (0.68)	41 (1.61)
103H5209-5240	103H5209-5210	0.425 (60.18)	1	4.4	11	0.062 (0.34)	0.31 (0.68)	41 (1.61)
103H5210-5040	103H5210-5010	0.465 (65.85)	0.25	80	123.3	0.074 (0.40)	0.37 (0.82)	48 (1.89)
103H5210-5140	103H5210-5110	0.49 (69.39)	0.5	20	35	0.074 (0.40)	0.37 (0.82)	48 (1.89)
103H5210-5240	103H5210-5210	0.51 (72.22)	1	4.8	9.5	0.074 (0.40)	0.37 (0.82)	48 (1.89)

Elastična spojka [2] je uključena u dizajn kako bi se kompenzirala nepoznata kvaliteta izrade motora i eliminirali kutni pomaci rotora motora. Ova komponenta omogućuje glatku i stabilnu rotaciju, čime se poboljšava ukupna učinkovitost sustava.

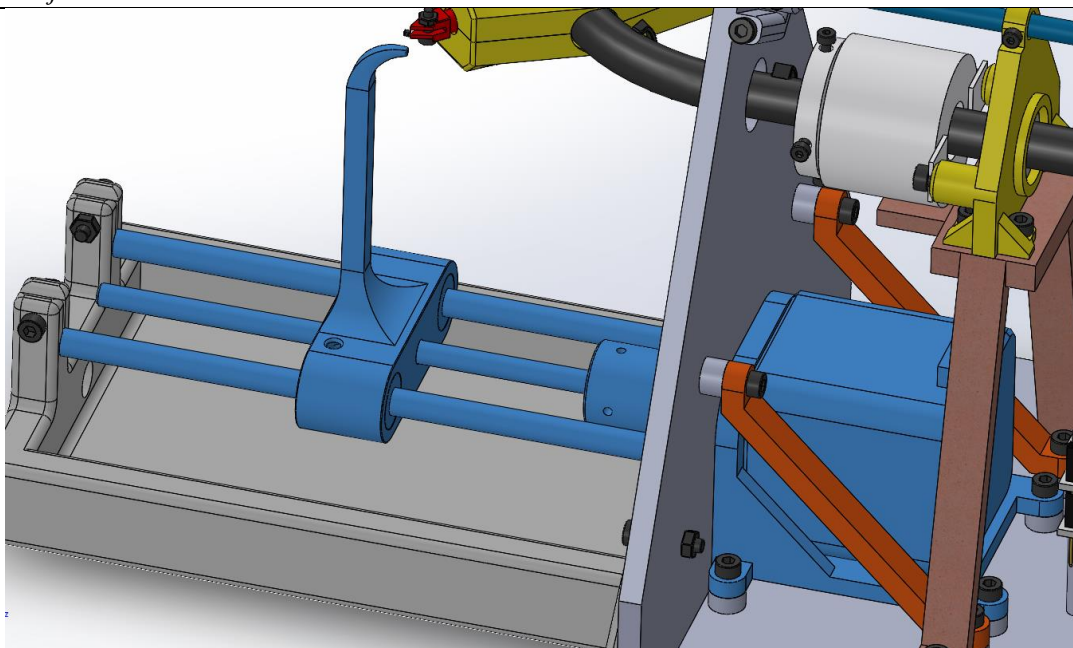
Da bi se spriječili aksijalni, radijalni i kutni pomaci, sustav koristi SKF ležajeve modela 61800 [3] i 618/8 [3], koji su odabrani zbog svoje pouzdanosti i preciznosti. Krutost cijelog podsustava dodatno je osigurana zahvaljujući potporama na konstrukciji, čvrstim držačima ležajeva i stabilizatorima koji omogućuju točno postavljanje i održavanje aksijalne pozicije komponenti.

Detaljnija razrada konstrukcije poluge bit će obrađena u poglavlju 2.4.

### 2.3. Sustav preciznog pozicioniranja žice

Primarna funkcija ovog podsustava [Slika 5] je precizno pozicioniranje žice tako da savršeno prati utore po širini feritne jezgre. Time se osigurava ravnomjerno namatanje žice i točno slaganje namota na jezgru, što je ključno za kvalitetu zavojnice.

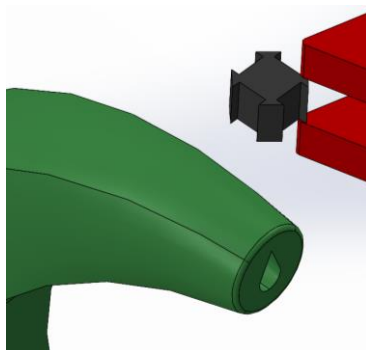
Za pokretanje ovog podsustava koristi se koračni motor 103H5210-5140 s enkoderom [Tablica 1], izabran zbog svoje sposobnosti preciznog upravljanja pozicijom, kao što je već objašnjeno u poglavlju 2.2. Okretni moment se prenosi putem elastične spojke, koja kompenzira eventualne kutne pomake i osigurava glatko djelovanje, a zatim se prenosi na navojno vreteno.



**Slika 5** Sustav preciznog pozicioniranja žice

Sustav koristi navojno vreteno i maticu za pretvorbu rotacijskog gibanja u translacijsko. Odabran je metrički fini navoj M5x0.5mm [4], što znači da svaka puna revolucija koračnog motora rezultira sa 0,5 mm aksijalnog pomaka matice. S obzirom da je jedna puna revolucija koračnog motora podijeljena na 200 koraka, jedan korak motora ostvaruje 0,0025 mm aksijalnog pomaka matice. Znajući da je žica debljine 0,08 mm, motor treba napraviti 32 koraka kako bi pomaknuo žicu u sljedeći utor. Ova razina preciznosti je više nego dovoljna, a također osigurava da u slučaju pogreške ili proklizavanja motora, odstupanje ostane neprimjetno.

Za osiguranje aksijalnog vođenja matice i eliminaciju odstupanja, korišten je sustav s dvije ojačane čelične vodilice [5] i dva linearna ležaja LM6 UU [6]. Ovaj sustav omogućuje stabilno i precizno kretanje matice, čime se dodatno povećava točnost i pouzdanost namatanja žice.

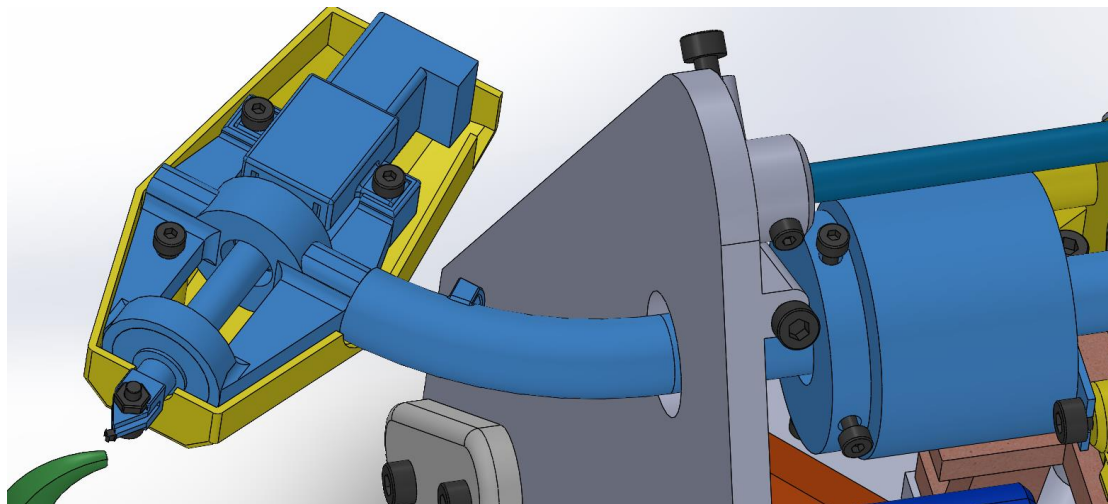


**Slika 6** Izgled otvora za vođenje žice

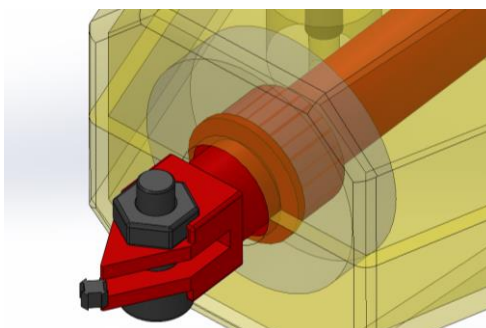


## 2.4. Sustav promjene osi namatanja

Glavna funkcija ovog podsustava [Slika 7] je rotacija feritne jezgre za 90 stupnjeva, čime se mijenja os namatanja žice. Ova rotacija je ključna za stvaranje ravnomjernih slojeva namota na različitim osima jezgre, što omogućuje preciznu i pravilnu izradu zavojnice.



Slika 7 Sustav promjene osi namatanja



Slika 8 Izgled prihvatnice

Za pokretanje ovog podsustava odabran je istosmjerni motor FIT0487 [Slika 9]. Glavni razlog za ovaj izbor je njegova iznimno mala masa od samo 18 grama, što je idealno za ovu primjenu jer minimizira moment inercije i olakšava kontrolu. Važno je napomenuti da je motor opremljen reduktorom, što je ključno za precizno upravljanje. Prijenosni omjer kod ovog motora je 380:1, što znači da za svakih 380 revolucija rotora motora, vratilo napravi samo jednu revoluciju. Motor je također opremljen rotacijskim enkoderom spojenim na rotor, koji šalje 14 impulsnih signala za svaku revoluciju motora. To rezultira s 5320 impulsa za svaku revoluciju vratila, što omogućuje iznimno preciznu kontrolu. Ova preciznost je posebno važna zbog malih dimenzija

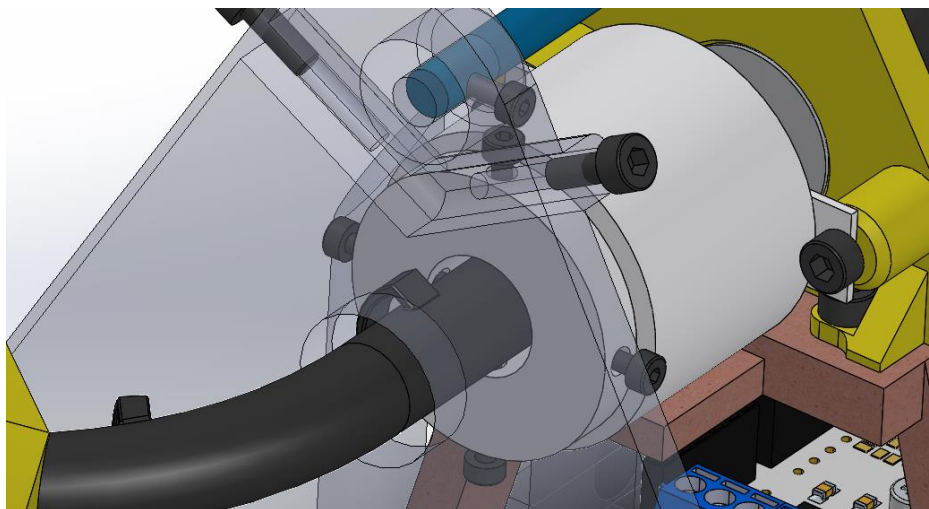
feritne jezgre i žice, kao i potencijalnih nedostataka u kvaliteti izrade motora, enkodera i prijenosa. Čak i ako nije sve savršeno, greške su toliko male da nemaju utjecaja na rad namatalice.

## SPECIFICATION

- Rated Voltage: 6.0 V
- Motor Speed: 15000 RPM
- Gear Reduction Ratio: 380:1
- Reducer Length: 9.0 mm
- No-Load Speed: 41 rpm@6v
- No-Load Current: 60 mA
- Rated Torque: 2.5 kg.cm
- Rated Speed: 24 rpm@6V
- Current Rating: 170 mA
- Instant Torque: <2.8kg.cm
- Hall Feedback Resolution: 5320
- Weight: 18g

**Slika 9** Karakteristike motora FIT0487 [6]

U ovom sustavu također se koristi "through-hole slip ring" (kroz-provodni klizni prsten) [Slika 10]. To je uređaj koji omogućuje prijenos električne energije i signala između rotirajućih i stacionarnih dijelova. U ovom slučaju, koristi se za dovođenje napajanja i signalnih žica za upravljanje motorom, kao i za povratnu vezu enkodera, s obzirom na to da se cijeli motor rotira. Da njega nema, sve žice bi se zapetljale prilikom prvog korištenja.



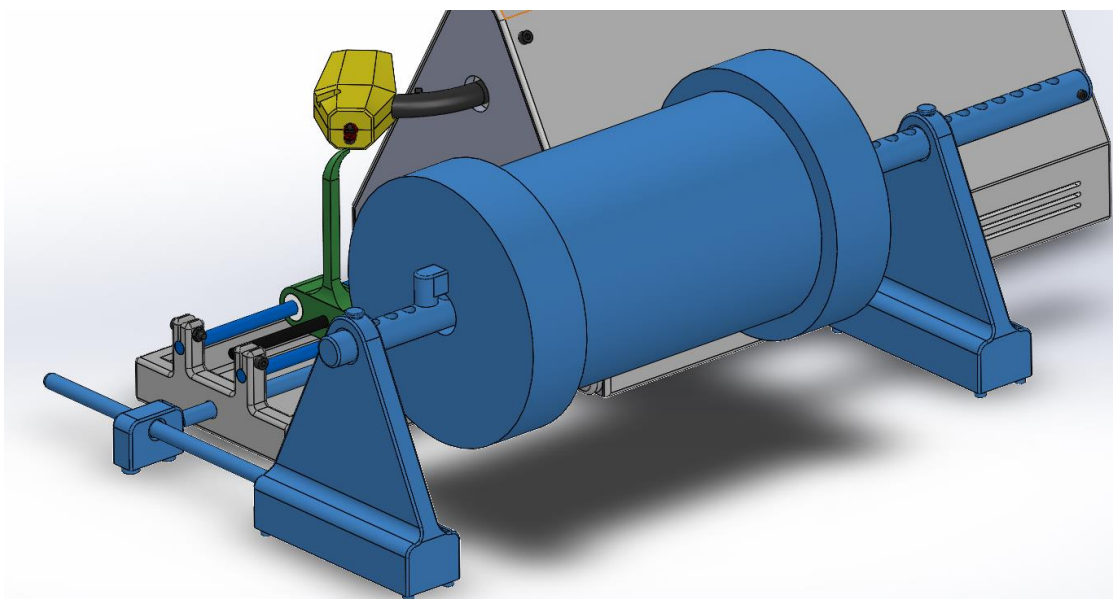
**Slika 10** Klizni prsten MT1233-S06 [7]

Za postizanje potrebne preciznosti u radu prihvatnice koja drži feritnu jezgru, koriste se dva SKF ležaja, 618/6 [3] i 618/7 [3]. Ovi ležajevi osiguravaju glatku i točnu rotaciju, minimizirajući bilo kakve aksijalne ili radijalne pomake.

Na kraju, važno je napomenuti da se cijeli podsustav nalazi na poluzi, zbog čega su masa i krutost sustava od iznimne važnosti za stabilnost i preciznost u radu.

## 2.5. Sustav za održavanje napetosti žice

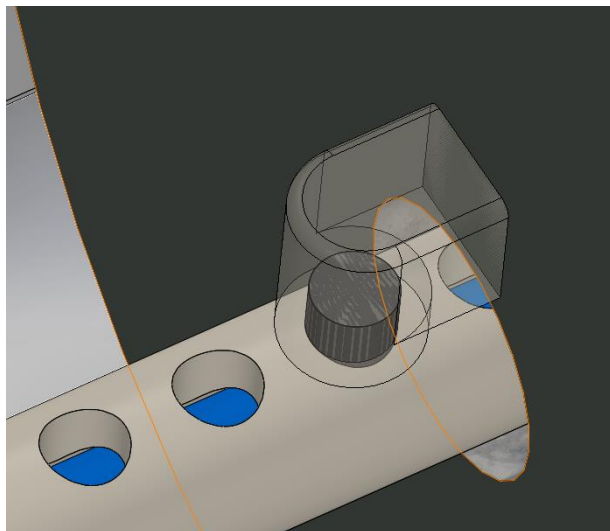
Primarna funkcija ovog podsustava [Slika 11] je održavanje stalne napetosti žice tijekom procesa namatanja, čime se osigurava precizno i ravnomjerno namatanje bez neželjenog odmotavanja. Bez odgovarajuće napetosti, stroj ne bi mogao pravilno funkcionirati, što bi rezultiralo loše namotanom zavojnicom i mogućim oštećenjima žice ili jezgre.



**Slika 11** Sustav za održavanje napetosti žice

Sustav za održavanje napetosti žice sastoji se od šuplje plastične cijevi kroz koju prolazi čelična šipka [5]. Cijev je po cijeloj dužini perforirana s otvorima, a na njoj je postavljena špula s žicom koja se može slobodno rotirati. Napetost žice ostvaruje se trenjem između špule i pakne [Slika 12] koja je pričvršćena za cijev. Pakna sadrži magnet [9] u sredini, koji je drži na mjestu privlačeći se prema čeličnoj osovini. Ovaj sustav je također dizajniran da podržava špule

različitih veličina, što pruža dodatnu fleksibilnost pri izboru dobavljača žice i omogućuje prilagodbu različitim zahtjevima proizvodnje.



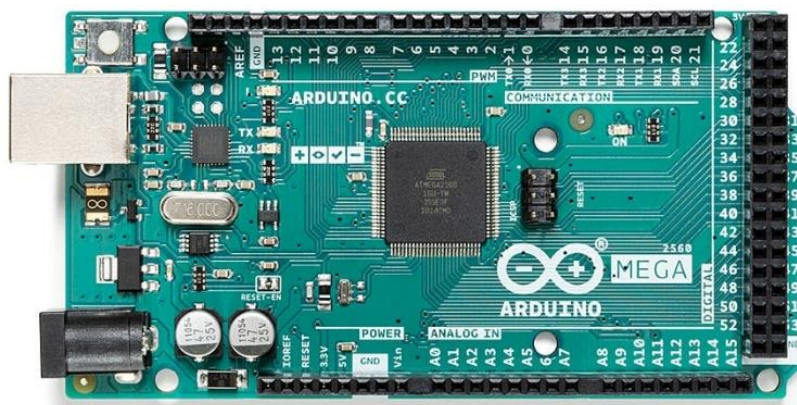
**Slika 12** Pakna s magnetom u središtu

### 3. UPRAVLJAČKI PROCES

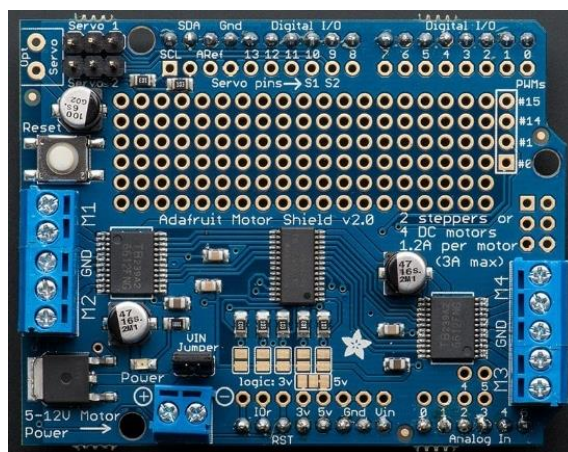
#### 3.1. Izbor upravljačke jedinice

Za upravljanje složenim procesom automatskog unakrsnog namatanja kvadratne zavojnice, odabrana je Arduino Mega razvojna pločica [Slika 13] kao centralna upravljačka jedinica. Ovaj izbor temelji se na jednostavnosti i pouzdanosti Arduino platforme, koja nudi pristupačno i korisnički prijateljsko okruženje za razvoj. Arduino IDE je razvojna platforma koja nije idealna za korištenje u gotovim proizvodima, ali je i više nego odlična za razvoj uređaja jer dopušta da glavni fokus bude na ostatku uređaja. Razvoj i izrada vlastite upravljačke jedinice i motor kontrolera je izvan područja razrade ovog rada.

Glavni razlog za izbor Arduino Mega je potreba za velikim brojem ulazno-izlaznih pinova. Arduino Mega nudi 54 digitalna I/O pina, među kojima se nalazi i šest pinova sa mogućnošću vanjskog prekida, što je ključno za precizno praćenje i upravljanje različitim komponentama sustava, poput enkodera i motora.



Slika 13 Arduino Mega Rev3 [10]



Slika 14 Adafruit Motor Shield v2 [11]

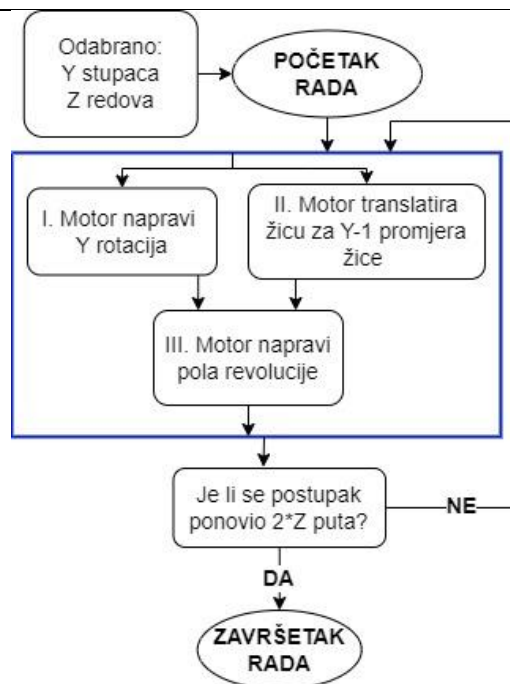
Za upravljanje motorima, korišteni su dva motor kontrolera, proizvođača Adafruit [Slika 14], svaki od kojih omogućuje precizno upravljanje koračnim i DC motorima. Adafruit motor kontroler omogućava jednostavno upravljanje brzinom i smjerom vrtnje motora te pruža zaštitu i pouzdanost. Korištenjem dva kontrolera, omogućava se istovremeno upravljanje više motora, što je neophodno za složene pokrete i koordinaciju u procesu namatanja.

Važan aspekt u ovom sustavu je upotreba prekida za vrlo precizno očitavanje enkodera. Enkoderi mjere rotaciju motora i omogućuju precizno praćenje pozicije i brzine. Prekidači se koriste za generiranje signala u trenutku kada se pojavi promjena u stanju enkodera. Kada enkoder generira signal, prekid odmah zaustavlja trenutno izvođenje koda i obrađuje podatke s enkodera. Ova tehnika omogućuje visoku preciznost u očitavanju i reakciji na promjene u stvarnom vremenu, što je ključno za točno upravljanje motorima.

PWM (Pulse Width Modulation) se koristi za upravljanje brzinom motora. PWM signal se generira kao niz impulsnih signala sa različitim vremenskim intervalima, gdje se duže vrijeme trajanja impulsa koristi za povećanje brzine motora, a kraće vrijeme trajanja za smanjenje brzine. Arduino Mega ima ugrađene PWM izlaze na nekoliko pinova, koji omogućuju precizno kontroliranje brzine i smjera motora. Korištenjem PWM-a, moguće je postići glatko i kontinuirano upravljanje brzinom motora, što je ključno za postizanje željenih performansi u procesu automatskog namatanja.

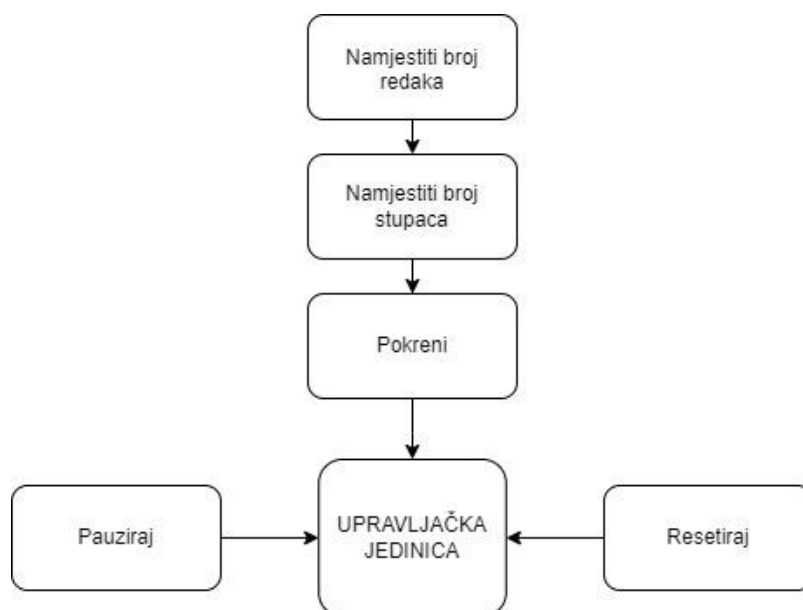
### **3.2. Rad namatalice**

Kao što je već objašnjeno u uvodu, namatalica koristi tri motora kako bi precizno namotala bakrenu žicu na jezgru. Prvi motor rotira jezgru oko jedne osi dok drugi motor pomiče žicu duž jezgre, stvarajući uredan red namota. Nakon što se jedan red popuni, treći motor rotira jezgru za 90 stupnjeva kako bi se namatanje nastavilo oko druge osi. Taj proces se ponavlja sve dok jezgra nije potpuno prekrivena željenim brojem slojeva.



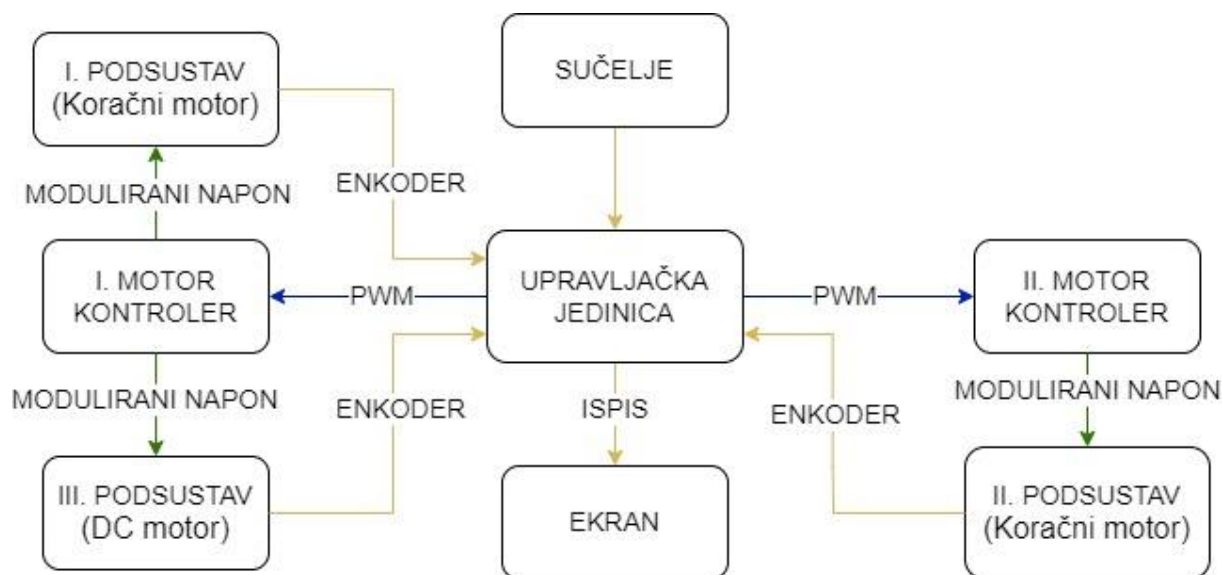
Slika 15 Dijagram toka namatalice

Ovaj postupak sastoji se od nekoliko ključnih cjelina. Prvo, korisnik unosi željene parametre putem jednostavnog sučelja. Sučelje je razvijeno u dogovoru sa tehničarima koji imaju iskustva u izradu zavojnica. Naglašena je potreba za unosom broja redova i stupaca zavojnice kako bi uređaj bio prilagodljiv različitim dimenzijama žice i feritne jezgre. Također, važno je da uređaj sadrži tipkalo kojim operater može zaustaviti rad u slučaju greške pri radu, te sukladno tome vratiti uređaj u početni položaj.



Slika 16 Dijagram toka sučelja

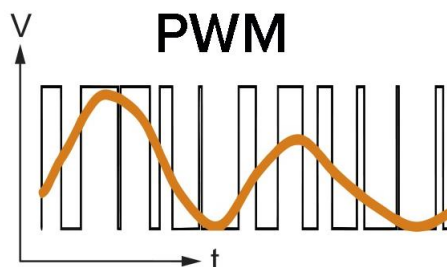
Program te podatke integrira u svoj algoritam, koji zatim, u kontinuiranoj komunikaciji s motor kontrolerima, precizno usmjerava kretnje motora. Enkoderi pružaju povratne informacije o poziciji, što omogućuje precizno upravljanje. Sve ove faze mogu se u realnom vremenu pratiti na OLED ekranu [12], koji prikazuje trenutni status i napredak procesa, pružajući korisniku jasan uvid u rad sustava.



Slika 17 Blok dijagram namatalice

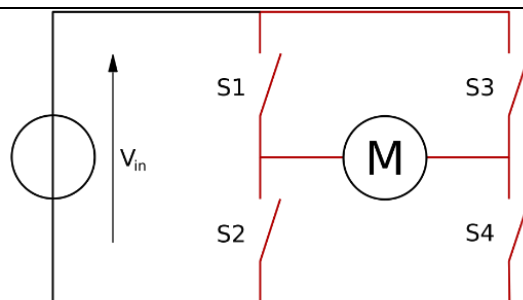
### 3.3. Upravljanje motorima

Upravljanje motorima u ovom sustavu provodi se pomoću PWM-a [Slika 18] kojeg mikroupravljač šalje motor kontroleru. Ovaj signal kontrolira smjer motora putem H mosta [Slika 19], dok se napajanje osigurava iz vanjskog izvora napona. Za precizno upravljanje pozicijom motora koristi se PI regulator, koji omogućuje stabilnu i točnu regulaciju.



Slika 18 Izgled PWM signala [14]

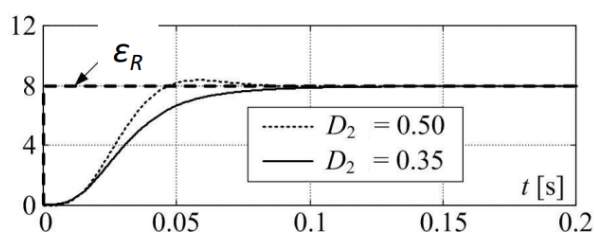




Slika 19 H-most [15]

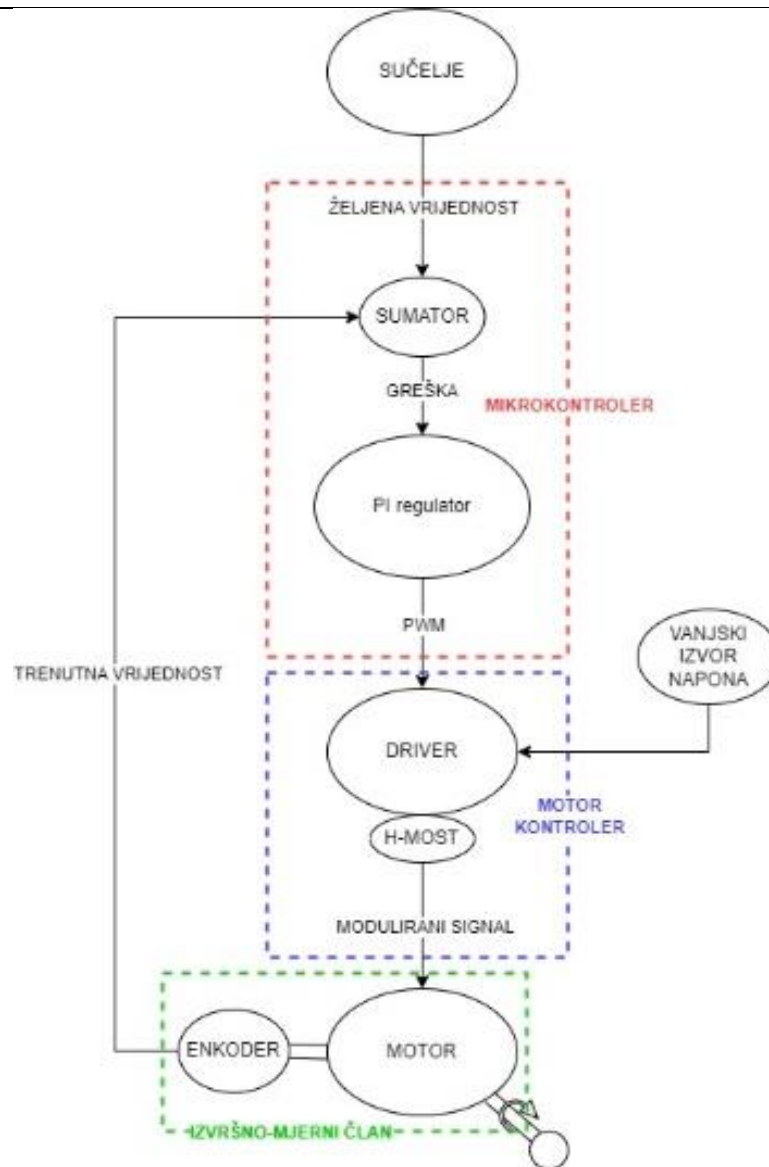
### 3.3.1. Regulacija položaja istosmjernog motora

Za precizno upravljanje pozicijom motora koristi se PI regulator, koji omogućuje stabilnu i točnu regulaciju. P komponenta PI regulatora izračunava proporcionalni odgovor na razliku između željene i trenutne pozicije motora, pružajući brzu reakciju na promjene. Međutim, samostalna P komponenta može uzrokovati oscilacije i nemogućnost postizanja točne željene pozicije. Zbog toga je dodana I komponenta koja integrira grešku tijekom vremena. Uvodeći inerciju u sustav postupno smanjuje razliku između trenutne i željene pozicije, čime osigurava precizniju i stabilniju regulaciju.



Slika 20 Primjer odziva PI regulatora na skokovitu pobudu [16]

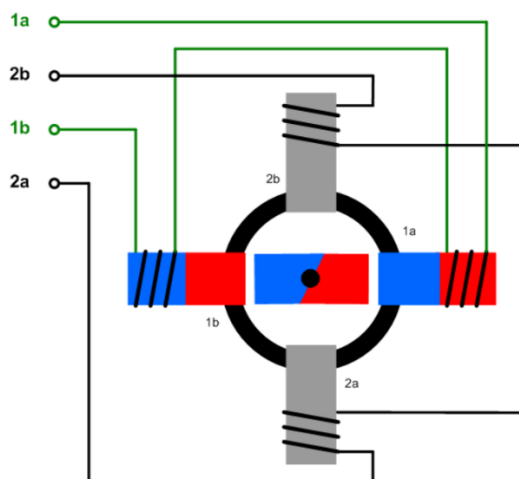
Trenutna pozicija motora očitava se pomoću inkrementalnog enkodera, koji generira prekidni signal svaki put kada motor napravi korak. Ovi signali omogućuju mikroupravljaču praćenje stvarnog položaja motora i usporedbu s željenom pozicijom, što je ključno za rad PI regulatora. H most se koristi za kontrolu smjera vrtnje motora te zaštitu motora i sklopa. Vanjski izvor napona osigurava dovoljnu snagu za rad motora, omogućujući stabilno napajanje bez preopterećenja mikroupravljača.



Slika 21 Dijagram toka sustava za upravljanje istosmjernim motorom

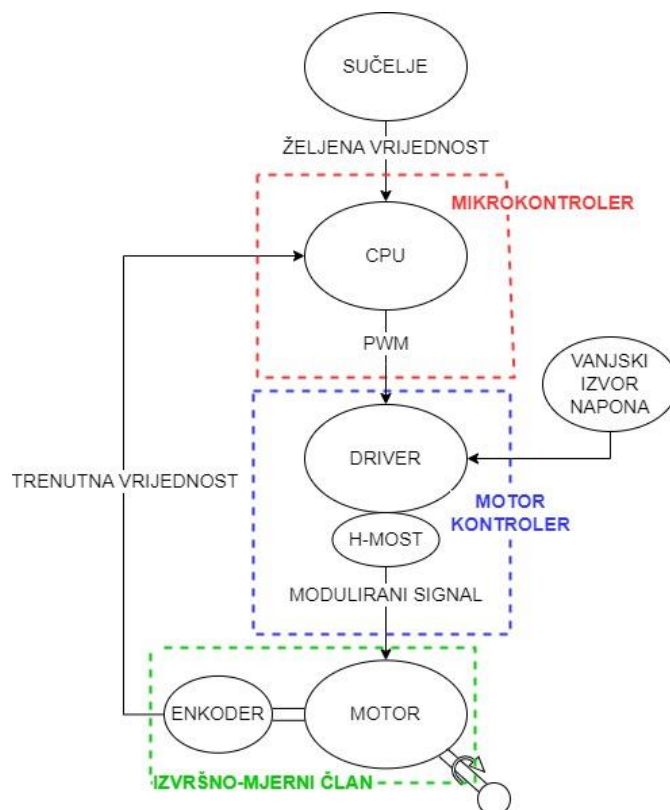
### 3.3.2. Upravljanje položajem koračnog motora

Regulacija položaja istosmjernog motora radi na principu regulacije brzine motora. Regulacija brzine motora radi na principu regulacije struje koju motor prima. Za razliku od običnog istosmjernog motora, koračni motor nema četkice. Koračni motor ima rotaciju podijeljenu u određeni broj koraka koji proizlaze iz konstrukcije motora [Slika 22].



**Slika 22** Bipolarni koračni motor i način spajanja njegovih namota [17]

Razlika u konstrukciji predstavlja i razliku u upravljanju motorom. S obzirom da se motor kreće u diskretnim koracima, upravljanje položajem je moguće napraviti s naredbom da motor napravi X broj koraka. Danas su takvi motori iznimno precizni i u većini slučajeva nije ni potrebno pratiti njegov rad. To ne znači da ne postoji prostor za regulaciju, dapače, regulacija je iznimno poželjnija u slučajevima visokih brzina i inercije. Konkretno u primjeni ovog rada, toga neće biti i time nema potrebe koristiti ikakav regulator za upravljanje motorom.



**Slika 23** Dijagram toka sustava za upravljanje koračnim motorom

## 4. PROGRAM UREĐAJA

Nakon izrade logičke strukture upravljanja uređaja, kao i pojedinih djelova istog, potrebno je napraviti adekvatan program koji će uređaj izvršavati. Program je napisan u Arduino IDE sučelju, čiji je jezik zasnovan na C++ programskom jeziku. Sučelje i jezik su izabrani iz jednostavnosti korištenja i pojednostavljenja procesa programiranja korištenjem ugrađenih biblioteka (za prikaz na ekranu, upravljanja koračnim motorima itd.). Potpuni program nalazi se u prilogu I.

Sukladno dijagramu toka namatalice [Slika 15], dani program [Slika 24] prikazuje rad namatalice. Mikroupravljač, radeći u beskonačnoj petlji, konstantno očitava stanje enkodera koji daje informacije o trenutnoj poziciji motora u danom trenutku. Uspoređujući željene vrijednosti zavojnice i trenutne pozicije motora, mikroupravljač određuje trenutačno stanje rada i diktira iduće. Enkoderi se žičano spajaju na ulaze koji sadrže vanjske prekide na razvojnoj pločici. Time se izvršava prekidni protokol pri svakom pomaku motora što znači da niti jedan impuls neće promaknuti mikroprocesoru.

```
// povratna veza - signal sa enkodera
read_encoder1(&trenutna_pozicija_GL, &brojac_okretaja_GL, &pomak_od_pocetne_pozicije_GL); // GL_stepper
read_encoder2(&trenutna_pozicija_DC, &pomak_od_pocetne_pozicije_DC); // DC_motor
read_encoder3(&trenutna_pozicija_Z, &pomak_od_pocetne_pozicije_Z); // Z_stepper

int prošli_potvrđi_brojac = potvrđi_brojac;
potvrđi(&stanje_rada_tipkala, &potvrđi_brojac); // u funkciji kao STOP tipkalo
if (prošli_potvrđi_brojac != potvrđi_brojac) {
    myMotor1->setSpeed(0);
    myMotor2->setSpeed(0);
    myMotor3->setSpeed(0);
} else {
    if (trenutni_broj_redova != ciljani_broj_redova) {
        if (trenutna_pozicija_GL != cilj_GL) {
            myMotor1->setSpeed(60); // rpm
            brzina_DC = 255; // drži pod punim naponom
            smjer_DC = RELEASE; // DC stoji
            myMotor3->setSpeed(60); // rpm
            myMotor1->step(25, FORWARD, INTERLEAVE);
            // izbor smjera kretanja
            if (((trenutni_broj_redova / 0.5) % 2) == 0) {
                myMotor3->step(4, FORWARD, INTERLEAVE);
            } else {
                myMotor3->step(4, BACKWARD, INTERLEAVE);
            }
        } else {
            myMotor1->setSpeed(0); // GL_stepper stoji
            myMotor3->setSpeed(0); // Z_stepper stoji

            if (trenutna_pozicija_DC < (cilj_DC - tolerancija_DC) || trenutna_pozicija_DC > (cilj_DC + tolerancija_DC)) {
                DC_motor(trenutna_pozicija_DC, &brzina_DC, &smjer_DC); // DC_motor radi
            } else {
                cilj_DC = 1 - cilj_DC; // promjena cilja
                trenutna_pozicija_GL = 0; // resetiranje jedinice
                trenutni_broj_redova += 0.5;
            }
        }
    } else {
        oled.print(" Motanje završeno ");
    }
}
```

Slika 24 Arudino kod rada namatalice

U skladu sa dijagramom toka sučelja, izrađen je program koji prima informacije o željenim svojstvima zavojnice pomoću sučelja [Slika 25]. Sučelje se sastoji od inkrementalnog enkodera prilagođenog ručnoj primjeni koji služi za odabir broja redova i stupaca zavojnice, višenamjenskog tipkala koje potvrđuje, pokreće i pauzira rad uređaja, te tipkala za resetiranje rada uređaja. Procedura unosa podataka je popraćena prikazom na ekranu. Tipkalo detektira samo ulazne bridove, a inkrementalni enkoder je spojen na ulaz sa detekcijom prekida na razvojnoj pločicu.

```
// izvodenje programa
else {
  // Odabir broja redaka pomocu rotacijskog enkodera
  if (potvrdi_brojac == 0) {
    odabran_broj_redova = trenutna_pozicija_enkodera_sucelja;
    if (zaustavnica1 == false) { // zelim da se to samo jednom pokaze na ekranu...
    }
    //oled.set2X() - opcija ovisno o ekranu;
    oled.print(" ");
    oled.setCursor(8 * 7, 5);
    oled.println(odabran_broj_redova);
    oled.clearField(8 * 7, 5, 1);
    ciljani_broj_redova = odabran_broj_redova;
    potvrdi(&stanje_rada_tipkala, &potvrdi_brojac);
  }

  // Odabir broja stupaca pomocu rotacijskog enkodera
  if (potvrdi_brojac == 1) {
    if (zaustavnica2 == false) { // za resetiranje encodera...
    }
    odabran_broj_stupaca = trenutna_pozicija_enkodera_sucelja;
    if (zaustavnica21 == false) { // zelimo da se to samo jednom pokaze na ekranu...
    }
    //oled.set2X();
    oled.print(" ");
    oled.setCursor(8 * 7, 5);
    oled.println(odabran_broj_stupaca);
    oled.clearField(8 * 7, 5, 1);
    cilj_GL = odabran_broj_stupaca * iznos_kruga_u_koracima_GL;
    potvrdi(&stanje_rada_tipkala, &potvrdi_brojac);
  }

  // početak motanja?
  if (potvrdi_brojac == 2) {
    if (zaustavnica3 == false) { // zelim da se to samo jednom pokaze na ekranu...
    }
    oled.clear();
    oled.println();
    oled.print(" Započni motanje?");
    oled.println();
    potvrdi(&stanje_rada_tipkala, &potvrdi_brojac);
  }

  // Motanje - rad
  if (potvrdi_brojac >= 3) { ...

```

**Slika 25** Arduino kod rada sučelja

Iznos brzine i smjer gibanja motor određuje PI regulator. Njegova svrha je opisana u poglavlju 3.1.1., a kompletan proces upravljanja istosmjernim motorom prikazuje dijagram toka sustava za upravljanje istosmjernim motorom [Slika 21].

```
// DC_motor PI regulacija položaja brzinom
void DC_motor(long p2, float* b2, int* s2) {
    long trenutnoT2 = micros();
    float deltaT2 = ((float)(trenutnoT2 - prosloT2)) / (1.0e6);
    prosloT2 = trenutnoT2;

    int greska2 = p2 - cilj_DC;
    float integral2 += greska2 * deltaT2;
    float u2 = kp2 * greska2 + ki2 * integral2;

    *s2 = FORWARD;
    *b2 = fabs(u2); // uvijek pozitivna vrijednost

    if (p2 < (cilj_DC + tolerancija_DC) && p2 > (cilj_DC - tolerancija_DC)) {
        *b2 = 0;
        *s2 = RELEASE;
    } else {
        if (u2 < 0) {
            *s2 = BACKWARD;
        }
    }
}
```

**Slika 26** Arduino kod rada PI regulatora istosmjernog motora

## 5. PROCJENA TROŠKOVA RAZVOJA

Procjena troškova razvoja uključuje nekoliko ključnih aspekata. Prvo, potrebno je uzeti u obzir cijenu materijala potrebnih za izradu komponenata. Nadalje, obuhvaćeni su troškovi proizvodnje dijelova te cijene nabavljenih komponenti od vanjskih dobavljača, s naglaskom na standardizirane proizvode koji su odmah dostupni.

Osim materijalnih troškova, u procjenu ulazi i vrednovanje dosadašnjih ljudskih sati uloženi u razvoj projekta. Za završetak projekta potrebno je osigurati dodatni budžet koji će pokriti preostale resurse i angažman radne snage, kako bi se osigurao uspješan dovršetak i realizacija konačnog proizvoda.

Bazirano je da se većina konstrukcijskih dijelova izrađuju pomoću aditivnih tehnologija, točnije 3D printerima koji printaju na smolu (SLA tehnologija) radi njihove iznimne točnosti. U praksi, tolerancija izrade se kreće oko 25 mikrona u izradi, što je iznimno potrebno za ovaj uređaj. Također, za neke dijelove, poput držača špule, nema potrebe za tolikom preciznošću pa je dovoljno izraditi ih na neki ekonomičniji način koji i dalje dozvoljava unikatnost oblika, poput konvencionalnih 3D printera (FDM tehnologije).

Cijene elektroničkih komponenti su uzete iz ponuda globalnih kompanija Farnell-a i Mouser-a radi njihove pouzdanosti, a mehaničkih komponenti iz lokalnih Trgo-Agencija, PKL te Strojopromet.

Budući da je količina dijelova koja se izrađuje aditivnim tehnologijama velika, tražena je ponuda kompanije 3D Tvornica na cjelokupnu narudžbu.

**Tablica 2 Troškovnik**

Rbr.	Naziv	Količina	Cijena [Euro]
<b>DIJELOVI IZRAĐENI ADITIVNIM TEHNOLOGIJAMA</b>			
1	Dno	1	-
2	Nosač koračnog motora	2	-
3	Poluga	1	-
4	Nosač ležaja I.	1	-
5	Nosač ležaja II.	1	-
6	Postolje za ležaj I.	1	-
7	Postolje za ležaj II.	1	-

8	Postolje za koračni motor	1	-
9	Odvojn timer	3	-
10	Prihvat motora	1	-
11	Prihvatnica	1	-
12	Poklopac prihvatnice donji	1	-
13	Poklopac prihvatnice gornji	1	-
14	Matica	1	-
15	Plitica	1	-
16	Stranica držača	2	-
17	Pomoćni držač	1	-
18	Držač	1	-
19	Osigurač	2	-
20	Čep držača	1	-
21	Pakna (stoper)	2	-
22.	Prednji poklopac	1	-
23	Stražnji poklopac	1	-
24	Potpورا	2	-
25	Stabilizator	1	-
<b>Ukupno:</b>			276
<b>ELEKTRONIČKE KOMPONENTE</b>			
1	Arduino MEGA R3 razvojna pločica [10]	1	41,11
2	Adafruit Motor Shield V2 [11]	2	34,76
3	Koračni motor 103H5210-5140 [1]	2	74,56
4	Istosmjerni motor FIT0487 [7]	1	9,57
5	Klizni prsten MT1233-S06 [8]	1	95
6	OLED ekran MDOB256064B1V-WM [12]	1	44,56
7	Sklopka za paljenje	1	9,43
8	Tipkalo	2	4,64



9	Rotacijski enkoder	1	7,71	
10	Napajanje AC/DC MP001989 [13]	1	21,97	
<b>Ukupno:</b>			343,31	
<b>MEHANIČKE KOMPONENTE</b>				
1	Elastična spojka 5/5mm [2]	2	7,20	
2	Vreteno M5x0,5 [4]	1	8	
3	Vodilica W 06 H6 2000mm CF53 [5]	1	13	
4	SKF Ležaj 61800 [3]	1	7,74	
5	SKF Ležaj 618 8 [3]	1	5,66	
6	SKF Ležaj 618 7 [3]	1	5,57	
7	SKF Ležaj 618 6 [3]	1	5,49	
8	Linearni ležaj LM6 UU THK [6]	2	3,80	
9	Cilindrični magnet sa H6 tolerancijom [9]	2	11,62	
<b>Ukupno:</b>			68,08	
<b>LJUDSKI RAD</b>				
Dosadašnji rad		6 E/h	320h	1920E
Procjena potrebnih sati za razvoj uređaja		6 E/h	9 mjeseci	17280E
Potreban kapital za daljnji razvoj			1000 E	
<b>Ukupno:</b>			20200 E	
<b>UKUPNA VRIJEDNOST PROJEKTA</b>				
<b>20888 Eura</b>				

---

## **6. ZAKLJUČAK**

Ovaj rad predstavlja jedno od mogućih rješenja za automatizirano unakrsno namatanje kvadratne zavojnice, s ciljem istraživanja i prikaza složenosti te količine problematike u razvoju takvog uređaja. Kroz razradu konstrukcije, programa i odabir komponenti, ovaj projekt pruža dublji uvid u tehničke izazove i potrebu za preciznim upravljanjem procesom namatanja. Odabir motora, implementacija PI regulatora i sustav održavanja napetosti žice pokazali su se ključnim segmentima za postizanje zadovoljavajućih rezultata.

Rad također služi kao platforma za osobni profesionalni razvoj, omogućujući autoru da se suoči s izazovima iz različitih područja mehatronike. Kroz ovaj projekt došlo se do realnije slike o kapitalu potrebnom za daljnji razvoj i potencijalnu realizaciju ovog uređaja, uz razumijevanje kompleksnosti koja dolazi s automatizacijom specifičnih procesa. Iako projekt nije završen kao komercijalno rješenje, predstavlja važan korak u autorovom stručnom razvoju i pruža temelj za buduće projekte i istraživanja.

---

**LITERATURA**

- [1] <https://uk.farnell.com/sanyo-denki-sanmotion/103h5208-5240/stepper-motor-2-ph-42mm-bipolar/dp/2610516>
- [2] <https://www.nkxmotor.si/hr/shop/spojka/spiralna-spojka/fleksibilna-aluminijska-spiralna-spojka-2/>
- [3] SKF Group: SKF General Catalogue 6000/I EN, lipanj 2008.
- [4] Kraut, B.: Strojarski priručnik, Tehnička knjiga Zagreb, 1970.
- [5] <https://www.trgo-agencija.hr/lezajevi/linearni-lezajevi/sipke/sipka-w-06-h6-2000mm-cf53-ck55-62-2-hrc.html>
- [6] <https://www.trgo-agencija.hr/lezajevi/linearni-lezajevi/lm/lezaj-linearni-lm-6-uu-mtm-6x12x19.html>
- [7] <https://uk.farnell.com/dfrobot/fit0487/dc-geared-motor-380-1-1500rpm/dp/4308212>
- [8] <https://www.moflon.com/mt1233.html>
- [9] <https://www.selos.hr/cilindricni-alnico-magnetski-elementi/magnetski-element-cilindricni-s-tolerancijom-alnico/>
- [10] <https://store.arduino.cc/en-hr/products/arduino-mega-2560-rev3>
- [11] <https://learn.adafruit.com/adafruit-motor-shield-v2-for-arduino/overview>
- [12] <https://export.farnell.com/midas/mdob256064b1v-wm/oled-graphic-display-cob-256-x/dp/3759033>
- [13] <https://uk.farnell.com/multicomp-pro/mp001989/adapter-ac-dc-12v-5a/dp/3293067>
- [14] <https://www.thomsonlinear.com/en/support/tips/what-is-pwm>
- [15] [https://sr.wikipedia.org/sr-el/%D0%A5\\_%D0%BC%D0%BE%D1%81%D1%82#](https://sr.wikipedia.org/sr-el/%D0%A5_%D0%BC%D0%BE%D1%81%D1%82#)
- [16] J. Deur, B. Škugor : „Mikroprocesorsko upravljanje – digitalna regulacija“, predavanja i auditorne vježbe, Fakultet strojarstva i brodogradnje Sveučilišta u Zagrebu, 2020
- [17] <https://www.tme.eu/hr/news/library-articles/page/41861/korachni-motori-vrste-i-primjeri-primjene-korachnih-motora/>

---

**PRILOZI**

## I. Arduino programski kod namatalice

```
#include <Arduino.h>
#include <Wire.h>
#include <Adafruit_MotorShield.h>
#include "utility/Adafruit_MS_PWM_ServoDriver.h"
#include "SSD1306Ascii.h"
#include "SSD1306AsciiWire.h"
#include <avr/wdt.h>

// OLED display 256 x 64 I2C
#define I2C_ADDRESS 0x3C
SSD1306AsciiWire oled;

// motor
Adafruit_MotorShield AFMS = Adafruit_MotorShield();
Adafruit_DCMotor* myMotor1 = AFMS.getStepper(200, 1); // motor360 -
GL_stepper
Adafruit_DCMotor* myMotor2 = AFMS.getMotor(2); // motor180 - DC_motor
Adafruit_DCMotor* myMotor3 = AFMS.getStepper(200, 3); // Z_stepper

// encoder
#include <Encoder.h>
Encoder Enc1(2, 3); //encoder za GL_stepper
Encoder Enc2(20, 21); //encoder za DC_motor
Encoder Enc3(18, 19); //encoder za Z_stepper

// "RESET" tipkalo
const int reset_Pin = 52;
volatile bool reset_zastavica = false;

// "POTVRDI - ZAPOČNI S MOTANJEM - STOP" tipkalozaustavnica21
const int buttonPin = 6;
int stanje_tipkala = 0;
int proslo_stanje_tipkala = 0;
int stanje_rada_tipkala = 1;
int potvrdi_brojac;

// inkrementalni enkoder - sucelje
const int pinA = 50;
const int pinB = 51;
volatile int trenutna_pozicija_enskodera_sucelja = 0;
volatile bool aproslostanje4;
volatile bool bproslostanje4;
```

```
// sluze da se neki dio koda koji se nalazi u loopu izvrši samo jednom
bool zaustavnica1 = false;
bool zaustavnica2 = false;
bool zaustavnica21 = false;
bool zaustavnica3 = false;
bool zaustavnica4 = false;
bool zaustavnica5 = false;

// za pocetnu poziciju uredaja
int pomak_od_pocetne_pozicije_GL = 0;
int pomak_od_pocetne_pozicije_DC = 0;
int pomak_od_pocetne_pozicije_Z = 0;
int potreban_pomak_za_povratak_GL = 0;
int potreban_pomak_za_povratak_DC = 0;
int potreban_pomak_za_povratak_Z = 0;

// cilj
int iznos_kruga_u_koracima_GL = 200;
int iznos_kruga_u_koracima_Z = 32;
int cilj_GL = 0;
int iznos_kruga_u_impulsima_DC = 5320; // DC_motor je 380:1
int cilj_DC = iznos_kruga_u_impulsima_DC / 2;
float trenutni_broj_redova = 0;
int ciljani_broj_redova = 0;

// postavke motanja
int odabran_broj_redova = 0;
int odabran_broj_stupaca = 0;

// GL_stepper
int brojac_okretaja_GL = 0;
int trenutna_pozicija_GL = -1;

// DC_motor
float kp2 = 2.5;
float ki2 = 0.5;
int tolerancija_DC = 100; // tolerancija poloazaja
int trenutna_pozicija_DC = -1;
long prosloT2 = 0;
float brzina_DC = 0;
int smjer_DC = RELEASE;

// Z_stepper
int trenutna_pozicija_Z = -1;

void setup() {
  Serial.begin(1000000);
  AFMS.begin();
  Wire.begin();
```

```

oled.begin(&SH1106_256x64, I2C_ADDRESS);
oled.setFont(ZevvPeep8x16);
oled.clear();

pinMode(buttonPin, INPUT_PULLUP); // "POTVRDI - ZAPOČNI S MOTANJEM - STOP"
tipkalo
pinMode(reset_Pin, INPUT_PULLUP); // "RESET" tipkalo

pinMode(pinA, INPUT_PULLUP);
pinMode(pinB, INPUT_PULLUP);
aproslostanje4 = digitalRead(pinA);
bproslotanje4 = digitalRead(pinB);

PCICR |= (1 << PCIE2); //Omogućava PCINT2 prekide
PCMSK2 |= 0b00000111; // Aktivira prekid samo na PCINT16, 17 i 18
}

void loop() {
  // RESET procedura
  if (reset_zastavica == true) {
    // potreban pomak za vraćanje u početni položaj
    potreban_pomak_za_povratak_GL = fabs(pomak_od_pocetne_pozicije_GL) %
iznos_kruga_u_koracima_GL;
    potreban_pomak_za_povratak_DC = fabs(pomak_od_pocetne_pozicije_DC) %
(iznos_kruga_u_impulsima_DC / 2);
    potreban_pomak_za_povratak_Z = fabs(pomak_od_pocetne_pozicije_Z) %
iznos_kruga_u_koracima_Z;

    // pomak motora u početni položaj
    myMotor1->setSpeed(60);
    myMotor1->step(potreban_pomak_za_povratak_GL, BACKWARD, INTERLEAVE);
    brzina_DC = 255;
    smjer_DC = BACKWARD;
    cilj_DC = potreban_pomak_za_povratak_DC;
    DC_motor(trenutna_pozicija_DC, &brzina_DC, &smjer_DC);
    myMotor3->setSpeed(60);
    myMotor1->step(potreban_pomak_za_povratak_Z, BACKWARD, INTERLEAVE);

    if (potreban_pomak_za_povratak_GL == 0 && potreban_pomak_za_povratak_DC <=
tolarancija2 && potreban_pomak_za_povratak_Z == 0) {
      resetArduino();
    }
  }
  // izvođenje programa
  else {
    // Odabir broja redaka pomoću rotacijskog enkodera
    if (potvrđi_brojac == 0) {
      odabran_broj_redova = trenutna_pozicija_ekodera_sucelja;

```

```
    if (zaustavnica1 == false) { // zelim da se to samo jednom pokaze na
ekranu
        zaustavnica1 = true;
        oled.clear();
        oled.println();
        oled.print(" Broj redaka:");
        oled.println();
    }
    //oled.set2X() - opcija ovisno o ekranu;
    oled.print(" ");
    oled.setCursor(8 * 7, 5);
    oled.println(odabran_broj_redova);
    oled.clearField(8 * 7, 5, 1);
    ciljani_broj_redova = odabran_broj_redova;
    potvrdi(&stanje_rada_tipkala, &potvrdi_brojac);
}

// Odabir broja stupaca pomocu rotacijskog enkodera
if (potvrdi_brojac == 1) {
    if (zaustavnica2 == false) { // za resetiranje encodera
        zaustavnica2 = true;
        trenutna_pozicija_enkodera_sucelja = 0;
        aproslostanje4 = 0;
        bproslostanje4 = 0;
        oled.clear();
        //oled.set1X();
        oled.println();
        oled.println(" Odabrano redaka:");
        //oled.set2X();
        oled.print(" ");
        oled.print(odabran_broj_redova);
        delay(1000);
        stanje_rada_tipkala = 0;
    }
    odabran_broj_stupaca = trenutna_pozicija_enkodera_sucelja;
    if (zaustavnica21 == false) { // zelimo da se to samo jednom pokaze na
ekranu
        zaustavnica21 = true;
        oled.clear();
        //oled.set2X();
        oled.println();
        oled.print(" Broj stupaca:");
        oled.println();
    }
    //oled.set2X();
    oled.print(" ");
    oled.setCursor(8 * 7, 5);
    oled.println(odabran_broj_stupaca);
    oled.clearField(8 * 7, 5, 1);
```

```
    cilj_GL = odabran_broj_stupaca * iznos_kruga_u_koracima_GL;
    potvrdi(&stanje_rada_tipkala, &potvrdi_brojac);
}

// početak motanja?
if (potvrdi_brojac == 2) {
    if (zaustavnica3 == false) { // zelim da se to samo jednom pokaze na
ekranu
        zaustavnica3 = true;
        stanje_rada_tipkala = 0;
        oled.clear();
        oled.println();
        oled.println(" Odabrano\n stupaca:");
        oled.print("  ");
        oled.print(odabran_broj_stupaca);
        delay(1000);
    }
    oled.clear();
    oled.println();
    oled.print(" Započni motanje?");
    oled.println();
    potvrdi(&stanje_rada_tipkala, &potvrdi_brojac);
}

// Motanje - rad
if (potvrdi_brojac >= 3) {
    if (zaustavnica4 == false) { //jednokratno
        zaustavnica4 = true;
        stanje_rada_tipkala = 0;
        oled.clear();
        oled.println();
        oled.println(" Počinje motanje");
        delay(1000);
    }
    // povratna veza - signal sa enkodera
    read_encoder1(&trenutna_pozicija_GL, &brojac_okretaja_GL,
&pomak_od_pocetne_pozicije_GL); // GL_stepper
    read_encoder2(&trenutna_pozicija_DC,
&pomak_od_pocetne_pozicije_DC); // DC_motor
    read_encoder3(&trenutna_pozicija_Z,
&pomak_od_pocetne_pozicije_Z); // Z_stepper

    int prosli_potvrdi_brojac = potvrdi_brojac;
    potvrdi(&stanje_rada_tipkala, &potvrdi_brojac); // u funkciji kao STOP
tipkalo
    if (prosli_potvrdi_brojac != potvrdi_brojac) {
        myMotor1->setSpeed(0);
        myMotor2->setSpeed(0);
        myMotor3->setSpeed(0);
    }
}
```



```

    } else {
        if (trenutni_broj_redova != ciljani_broj_redova) {
            if (trenutna_pozicija_GL != cilj_GL) {
                myMotor1->setSpeed(60); // rpm
                brzina_DC = 255; // drzi pod punim naponom
                smjer_DC = RELEASE; // DC stoji
                myMotor3->setSpeed(60); // rpm
                myMotor1->step(25, FORWARD, INTERLEAVE);
                // izbor smjera kretanja
                if (((trenutni_broj_redova / 0.5) % 2) == 0) {
                    myMotor3->step(4, FORWARD, INTERLEAVE);
                } else {
                    myMotor3->step(4, BACKWARD, INTERLEAVE);
                }
            } else {
                myMotor1->setSpeed(0); // GL_stepper stoji
                myMotor3->setSpeed(0); // Z_stepper stoji

                if (trenutna_pozicija_DC < (cilj_DC - tolerancija_DC) ||
                    trenutna_pozicija_DC > (cilj_DC + tolerancija_DC)) {
                    DC_motor(trenutna_pozicija_DC, &brzina_DC, &smjer_DC); //
                    DC_motor radi
                } else {
                    cilj_DC = 1 - cilj_DC; // promjena cilja
                    trenutna_pozicija_GL = 0; // resetiranje jedinice
                    trenutni_broj_redova += 0.5;
                }
            }
        } else {
            oled.print(" Motanje završeno ");
        }

        if (zaustavnica5 == false) { // prikaz na ekranu
            zaustavnica5 = true;
            //oled.set1X();
            oled.clear();
            oled.print(" STUPCI: RETCI:");
        }
        oled.setCursor(0, 4);
        oled.clearField(0, 4, 16);
        oled.print(" ");
        oled.print(brojac_okretaja_GL);
        oled.print(" ");
        oled.println(trenutni_broj_redova);
        oled.print();

        myMotor2->setSpeed(b2);
        myMotor2->run(s2);
    }

```

```
    }
  }
}

// "POTVRDI - ZAPOČNI S MOTANJEM - STOP" tipkalo
void potvrdi(int* sr3, int* pbr) {
  stanje_tipkala = digitalRead(buttonPin3);
  if (stanje_tipkala == HIGH && proslo_stanje_tipkala == LOW) {
    *sr3 = !*sr3;
    if (*sr3 == HIGH) {
      oled.clear();
      //oled.set2X();
      oled.println();
      oled.print(" OK");
      *pbr = *pbr + 1;
    }
  }
  proslo_stanje_tipkala = stanje_tipkala; // Ažuriranje prethodnog stanja
  tipke
}

// softverski reset arduino - brze i jednostavnije od ponovnog namjesta
// svake varijable na početnu vrijednost
void resetArduino() {
  wdt_enable(WDTO_15MS); //nakon isteka timera arudino se resetira
  while (true) {
    // Čekaj dok se Arduino ne resetira
  }
}

// za rotacijski enkoder - namjesta
// stupaca i redaka
ISR(PCINT16 || PCINT17) {
  bool aVal4 = digitalRead(pinA);
  bool bVal4 = digitalRead(pinB);

  // Provjeravamo je li došlo do promjene na pinu A
  if (aVal4 != aproslostanje4) {
    aproslostanje4 = aVal4;
    // Provjeravamo je li promjena na pinu B u odnosu na pin A (smjer
    // rotacije)
    if (bVal4 != aVal4) {
      trenutna_pozicija_ekodera_sucelja++;
    } else {
      trenutna_pozicija_ekodera_sucelja--;
    }
  }
}
```

```
// Provjeravamo je li došlo do promjene na pinu B
if (bVal4 != bproslotanje4) {
    bproslotanje4 = bVal4;
    // Provjeravamo je li promjena na pinu A u odnosu na pin B (smjer
rotacije)
    if (aVal4 == bVal4) {
        trenutna_pozicija_ekodera_sucelja++;
    } else {
        trenutna_pozicija_ekodera_sucelja--;
    }
}
if (trenutna_pozicija_ekodera_sucelja < 0) {
    trenutna_pozicija_ekodera_sucelja = 0;
}
}

// za RESET tipkalo
ISR(PCINT18) {
    reset_zastavica = true; // Postavlja zastavicu na true
}

// GL_stepper - 200 koraka po okretaju
void read_encoder1(int* poz1, int* bo1, int* pom1) {
    long newPos1 = Enc1.read();
    if (newPos1 != *poz1) {
        *poz1 = newPos1;
        // za pracenje udaljenosti od pocetne pozicije
        if (newPos1 >= *poz1) {
            *pom1 += 1;
        }
        if (newPos1 <= *poz1) {
            *pom1 -= 1;
        }
    }
    *bo1 = *poz1 / 200;
}

// DC_motor - 380:1 prijenos - 5320 impulsa po okretaju
void read_encoder2(int* poz2, int* pom2) {
    int newPos2 = Enc2.read();
    if (newPos2 != *poz2) {
        *poz2 = newPos2;
        // za pracenje udaljenosti od pocetne pozicije
        if (newPo21 >= *poz2) {
            *pom2 += 1;
        }
        if (newPos2 <= *poz2) {
            *pom2 -= 1;
        }
    }
}
```

```
}  
// Z_stepper - 200 koraka po okretaju  
void read_encoder3(int* poz3, int* pom3) {  
    int newPos3 = Enc3.read();  
    if (newPos3 != *poz3) {  
        *poz3 = newPos3;  
        // za pracenje udaljenosti od pocetne pozicije  
        if (newPos3 >= *poz3) {  
            *pom3 += 3;  
        }  
        if (newPos3 <= *poz3) {  
            *pom3 -= 3;  
        }  
    }  
}  
  
// DC_motor PI regulacija poloazaja brzinom  
void DC_motor(long p2, float* b2, int* s2) {  
    long trenutnoT2 = micros();  
    float deltaT2 = ((float)(trenutnoT2 - prosloT2)) / (1.0e6);  
    prosloT2 = trenutnoT2;  
  
    int greska2 = p2 - cilj_DC;  
    float integral2 += greska2 * deltaT2;  
    float u2 = kp2 * greska2 + ki2 * integral2;  
  
    *s2 = FORWARD;  
    *b2 = fabs(u2); // uvijek pozitivna vrijednost  
  
    if (p2 < (cilj_DC + tolerancija_DC) && p2 > (cilj_DC - tolerancija_DC)) {  
        *b2 = 0;  
        *s2 = RELEASE;  
    } else {  
        if (u2 < 0) {  
            *s2 = BACKWARD;  
        }  
    }  
}
```