

Sklop za upravljanje bipolarnim koračnim motorom temeljen na Arduino platformi

Boić, Marin

Undergraduate thesis / Završni rad

2022

Degree Grantor / Ustanova koja je dodijelila akademski / stručni stupanj: **University of Rijeka, Faculty of Engineering / Sveučilište u Rijeci, Tehnički fakultet**

Permanent link / Trajna poveznica: <https://um.nsk.hr/um:nbn:hr:190:888408>

Rights / Prava: [Attribution 4.0 International](#)/[Imenovanje 4.0 međunarodna](#)

Download date / Datum preuzimanja: **2025-01-15**



Repository / Repozitorij:

[Repository of the University of Rijeka, Faculty of Engineering](#)



SVEUČILIŠTE U RIJECI

TEHNIČKI FAKULTET

Preddiplomski sveučilišni studij elektrotehnike

Završni rad

**SKLOP ZA UPRAVLJANJE BIPOLARNIM KORAČNIM
MOTOROM TEMELJEN NA ARDUINO PLATFORMI**

Rijeka, studeni 2022.

Marin Boić

0069087805

SVEUČILIŠTE U RIJECI

TEHNIČKI FAKULTET

Preddiplomski sveučilišni studij elektrotehnike

Završni rad

**SKLOP ZA UPRAVLJANJE BIPOLARNIM KORAČNIM
MOTOROM TEMELJEN NA ARDUINO PLATFORMI**

Mentor: doc. dr. sc. Ivan Volarić

Rijeka, studeni 2022.

Marin Boić

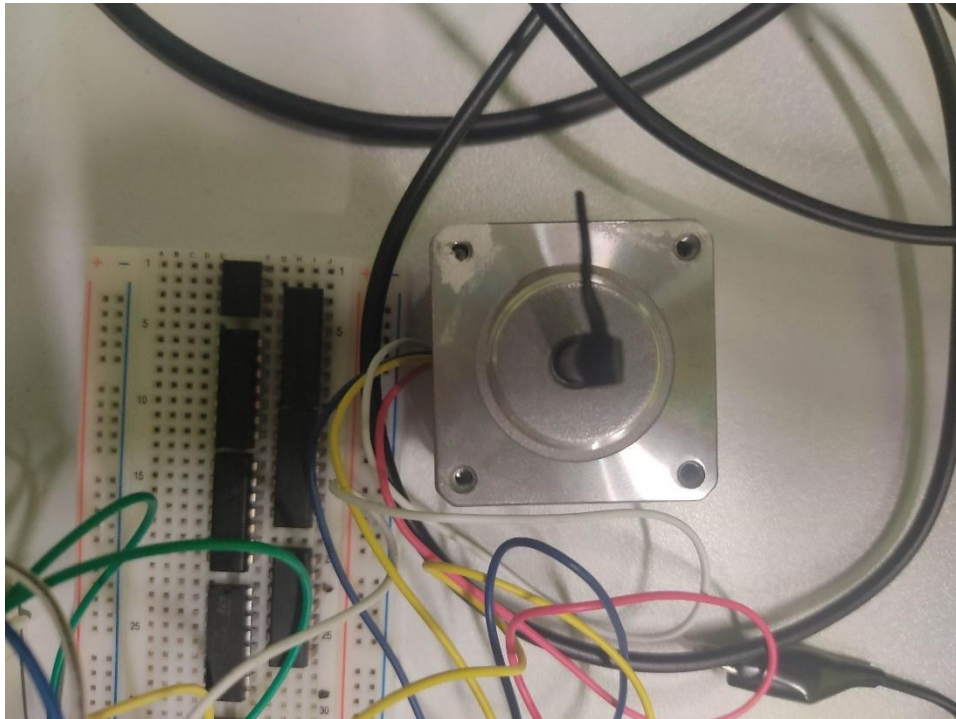
0069087805

Sadržaj

1. Uvod.....	2
2. Bipolarni koračni motor.....	4
2.1. Upravljanje.....	5
2.2. Specifikacija motora.....	7
3. Upravljački krug.....	9
3.1. H – most.....	9
3.2. Full bridge driver.....	11
3.3. HIP4081A.....	12
3.4. Konfiguracija drivera.....	14
3.4.1. Napajanje i bootstrapping.....	15
3.4.2. Propagacijsko kašnjenje.....	18
3.4.3. Ulazna logika.....	20
3.5. Arduino platforma.....	22
4. Zaključak.....	26

1. UVOD

Koračni motori (eng. stepper motors) su specijalna vrsta motora s impulsnim oblikom uzbude, čija glavna karakteristika su precizni mehanički pomaci u intervalu $0.5^\circ - 90^\circ$. Slika broj 1.1. prikazuje izgled koračnog motora.



Slika 1.1: izgled koračnog motora korištenog za potrebe završnog rada.

Brojne su funkcije ostvarive primjenom koračnih motora. Niska energetska zahtjevnost, male dimenzije, visok faktor iskoristivosti te jednostavna struktura dovoljni su razlozi primjene u nekim kućanskim aparatima, kao što su printeri ili pametna rasvjeta. Precizni mehanički pomaci čine ih vrlo korisnima za upotrebu u satovima. Njihovo korištenje gotovo je neizbježno u industriji. Koriste se u robotskim rukama (slika 1.2.), kada je u nekom proizvodnom procesu potrebno imitirati ljudsku ruku, radi obuhvaćanja i prijenosa alata ili opreme ili upravljanjem istim na proizvodnim trakama. U takvim slučajevima, motori dobivaju impulse iz PLC-eva. Neizostavni su dio 3D printera, strojeva za pakiranje ili opreme za pozicioniranje.



Slika 1.2: robotska ruka kao primjer primjene koračnog motora. [1]

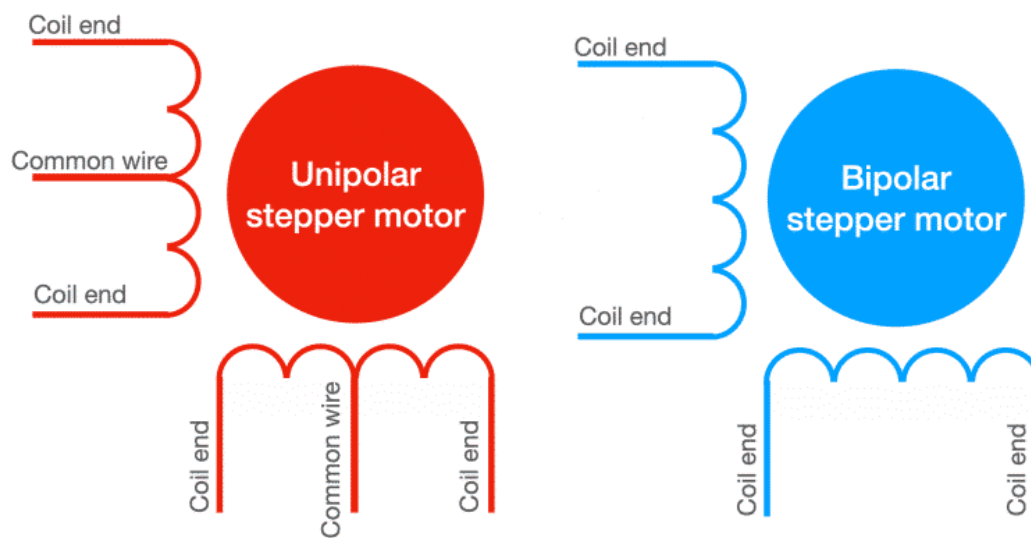
Dodatne prednosti takvih motora su mogućnost rada bez potrebe za povratnom vezom, jer bilo kakva upravljačka jedinica posjeduje informaciju o poziciji rotora na temelju informacije o broju generiranih upravljačkih impulsa, te visoki momenti pri malim brzinama (većina električnih motora ne može bez modifikacija ostvariti velike momente pri malim brzinama). Također se vrlo jednostavno može vršiti upravljanje brzinom i smjerom vrtnje, koja se može kretati od niske i isprekidane (koračni rad - najčešći slučaj) do praktički konstantne (režim okretanja), a brzine koje se pritom mogu ostvariti kreću se i preko 10000 koraka u sekundi. Takva regulacija također će se demonstrirati u sklopu ovog završnog rada. Sve navedeno ih praktički čini neizostavnim dijelom modernih automatiziranih pogona, a vrlo često su upravljani mikroračunalima i programirani za specifičnu svrhu.

2. BIPOLARNI KORAČNI MOTOR

S obzirom na izvedbu statora, glavna podjela koračnih motora vrši se na unipolarne (četvero-fazne) i bipolarne (dvo-fazne) koračne motore (slika 2.1.). Unipolarni motori koriste zasebnu fazu za izvedbu svakog pola statora, dok bipolarni koriste dva namotaja (faze) s kojim mogu realizirati četiri pola i oni će biti predmet upravljanja u ovom završnom radu.

Prednosti bipolarnih koračnih motora u usporedbi s unipolarnim su te što bipolarni motori:

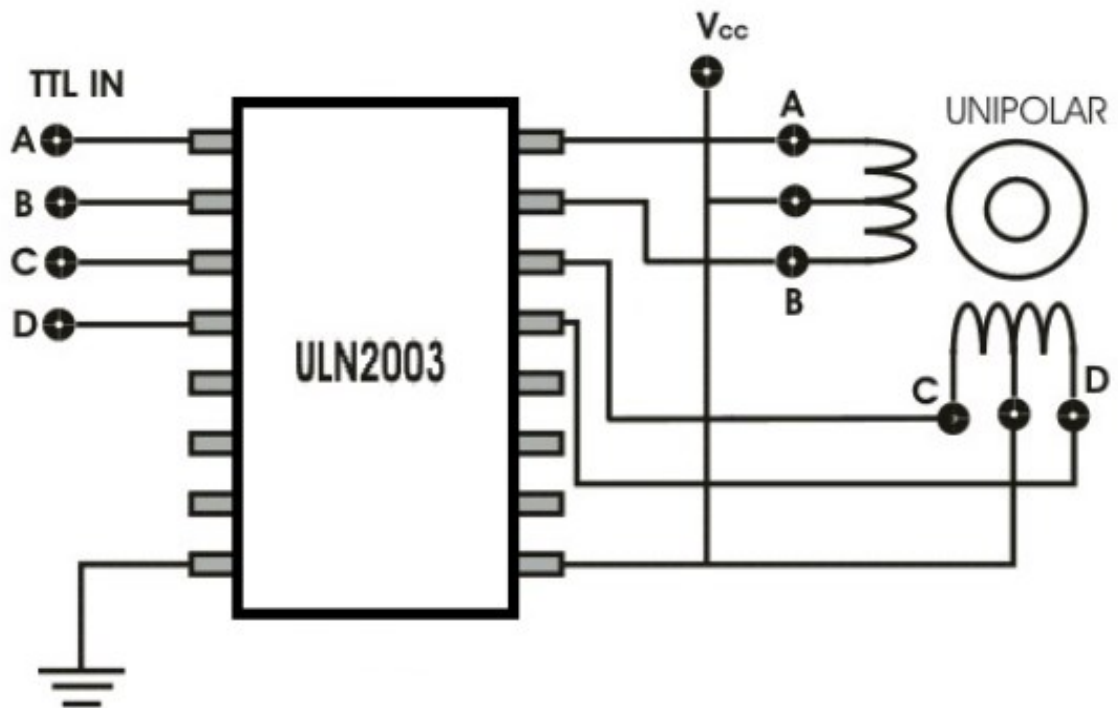
- imaju jednostavniju izvedbu, budući da zahtijevaju manji broj namota (2 namota ukoliko se želi realizirati 4 pola) te
- imaju bolje momentne karakteristike, jer za razliku od unipolarnih motora, struje teku cjelovito kroz namote (namoti unipolarnih motora imaju srednji izvod pa struja teče samo kroz polovinu namotaja).



Slika 2.1: ilustrativni prikaz razlike u građi dvaju koračnih motora. [2]

2.1. Upravljanje

U odnosu na upravljanje unipolarnim koračnim motorm, upravljanje bipolarnim motorom je kompliciranije jer podrazumijeva promjenu smjera struje kroz namotaje te u praksi uključuje korištenje specifičnih drivera, tzv. *stepper drivera*. Oni se koriste i za upravljanje unipolarnim motorima, no u praksi su bipolarni motori češći. Danas na tržištu postoji čitav niz specijaliziranih integriranih krugova koji služe za upravljanje različitim tipovima koračnih motora. Blok shema jednog IC-a (ULN 2003) za upravljanje četvero-faznim koračnim motorom prikazana je na slici 2.2.



Slika 2.2: blok shema za upravljanje unipolarnim koračnim motorom[3]

Dovođenjem impulsa na ulazne pinove, driver slijedi redoslijed uključivanja faza te na izlazne pinove šalje impulse dovoljne jakosti i duljine trajanja da motor može napraviti korak.

U sklopu ovog završnog rada, upravljački krug za upravljanje koračnim motorom složit će se samostalno.

Kut koji će motor pokriti jednim korakom ovisi o broju polova na statoru, a u specifičnim slučajevima i o jakosti struje kroz namote. Ukoliko motor ima 2 namotaja (dvije faze), postoji maksimalno osam sekvenci koje se mijenjaju tijekom jednog potpunog zakreta motora. U tom slučaju, navedeni režim rada se zove osam-koračni ili polu-koračni rad.

Sekvenca od 4 koraka zove se koračni rad. Motor može raditi u koračnom režimu na dva načina:

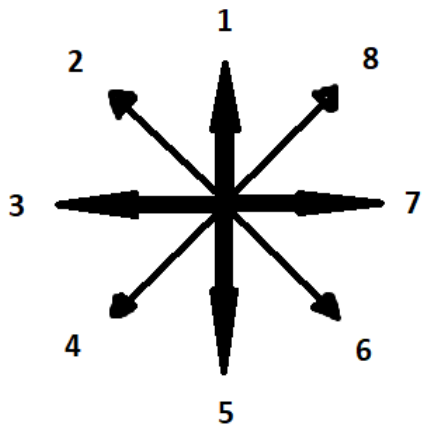
- postizanje koraka uzbuđivanjem jednog namota,
- postizanje koraka uzbuđivanjem oba namota.

Drugi način upravljanja ujedno se zove i koračni rad po polu-koracima. U sklopu ovog rada, upravljati će se motorom na način da radi u svim režimima rada.

Tablica 2.1: raspored polariteta po priključcima faza (AB i CD) za svaki od režima rada.

n	POLU-KORAČNI RAD				KORAČNI RAD				KORAČNI RAD PO POLU-KORACIMA			
	A	B	C	D	A	B	C	D	A	B	C	D
1	+	-	/	/	+	-	/	/	/	/	/	/
2	+	-	+	-	/	/	/	/	+	-	+	-
3	/	/	+	-	/	/	+	-	/	/	/	/
4	-	+	+	-	/	/	/	/	-	+	+	-
5	-	+	/	/	-	+	/	/	/	/	/	/
6	-	+	-	+	/	/	/	/	-	+	-	+
7	/	/	-	+	/	/	-	+	/	/	/	/
8	+	-	-	+	/	/	/	/	+	-	-	+

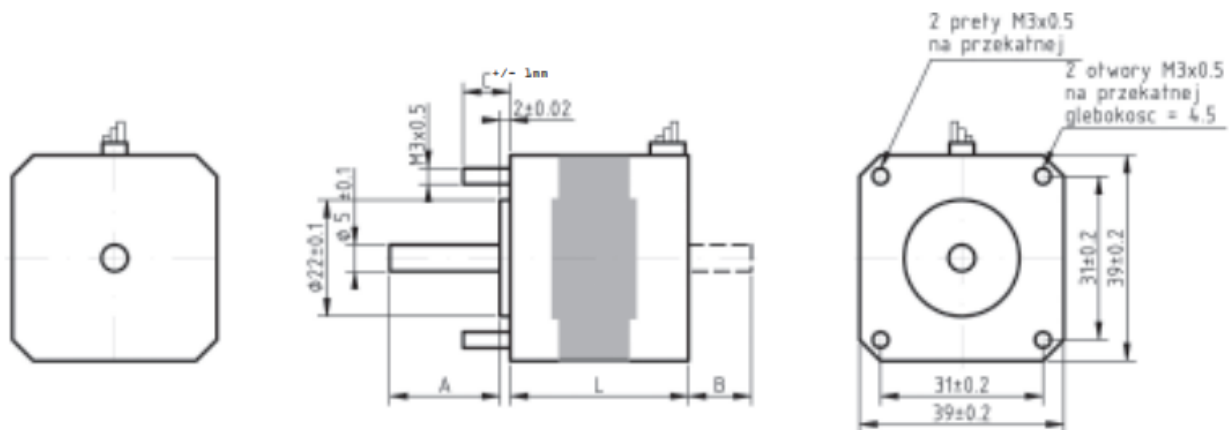
Kao što prikazuje tablica 2.1., potrebno je slijedno polarizirati jedan ili oba namota po određenom redosljedu kako bi rotor sa permanentnim magnetima (najčešći oblik rotora) mogao napraviti željeni korak. Pritom se najčešće koriste naponi iznosa 12, 24 ili 48 volti.



Slika 2.3: ilustracija koraka.

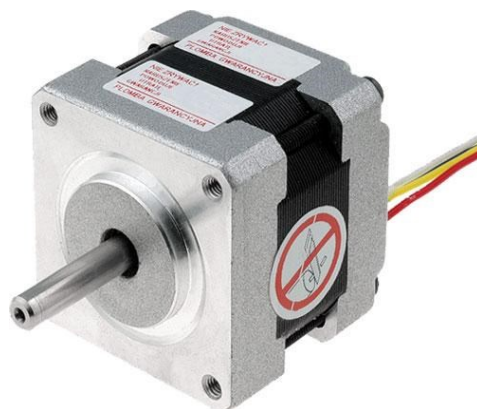
2.2. Specifikacija motora

Za potrebe ovog završnog rada, korišten je dvofazni bipolarni koračni motor iz serije 39BYGH. Kao što je vidljivo iz slike 2.3., motor je malih dimenzija (39*39 mm), osovina mu je promjera 5 mm.



Slika 2.3: dimenzije motora, prema dokumentaciji proizvođača. [4]

Stator motora konstruiran je na način da se motor zakrene za $1,8^\circ$ po koraku, što znači da se uz polukoračno upravljanje, motor može zakrenuti za $0,9^\circ$ stupnjeva. To je maksimalno 400 koraka po okretu. Motor se napaja istosmjernih izvorom od 12 V. Otpor namota je 30Ω , što znači da za rad povlači 0,4 A i ostvaruje, za svoje dimenzije, značajan moment od 0,21 Nm.

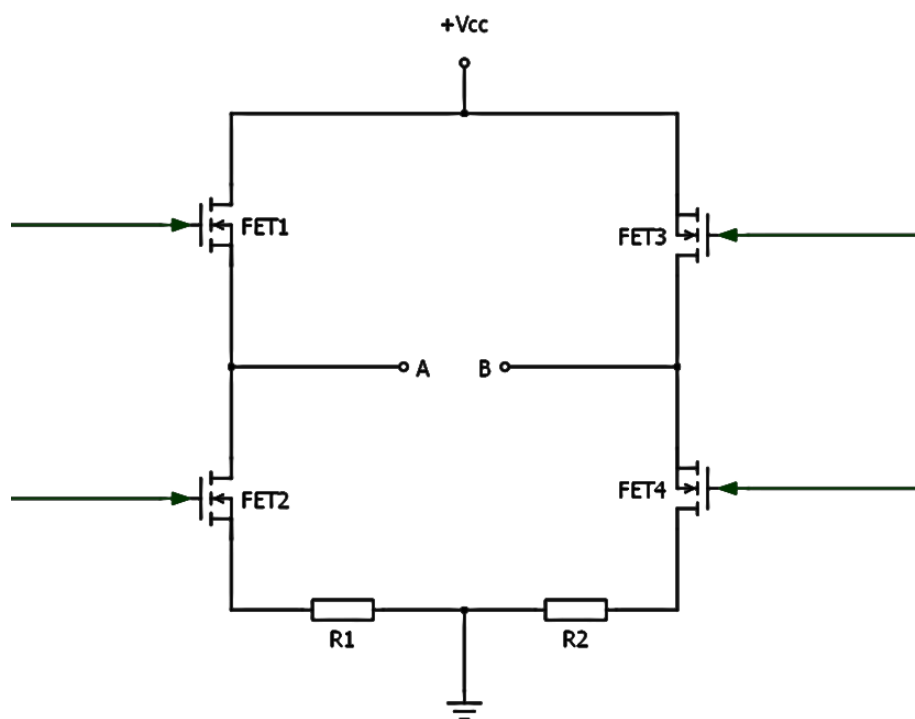


Slika 2.4: koračni motor 39BYGH405B.

3. UPRAVLJAČKI KRUG

3.1. H – most

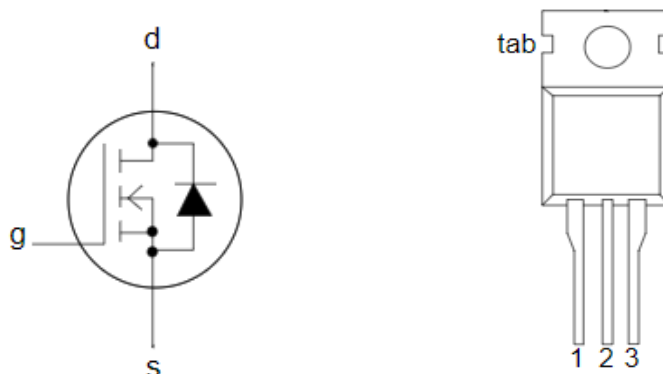
Svaka faza bipolarnog motora napajana je iz zasebnog H – mosta (slika 3.1). H – most je električni krug koji omogućuje promjenu smjera struje (dobivanje izmjeničnog električnog signala iz istosmjernog) prikladnim uključivanjem i isključivanjem tranzistora koji djeluju kao sklopke, a aktiviraju se iz posebnog upravljačkog sklopa koji se zove full-bridge ili half-bridge gate driver, (koji će biti opisan u idućem potpoglavlju) dovođenjem impulsa na gate elektrode MOSFET tranzistora.



Slika 3.1: shema H – mosta za napajanje jedne faze (AB) motora.

Uključivanjem para tranzistora FET1/FET 4 ostvaruje se pozitivan polaritet napona Vcc na namotu motora. Isključivanjem para FET1/FET 4 te uključivanjem para FET2/FET 3 ostvaruje se negativan polaritet napona Vcc, a time i komutacija struje kroz motor.

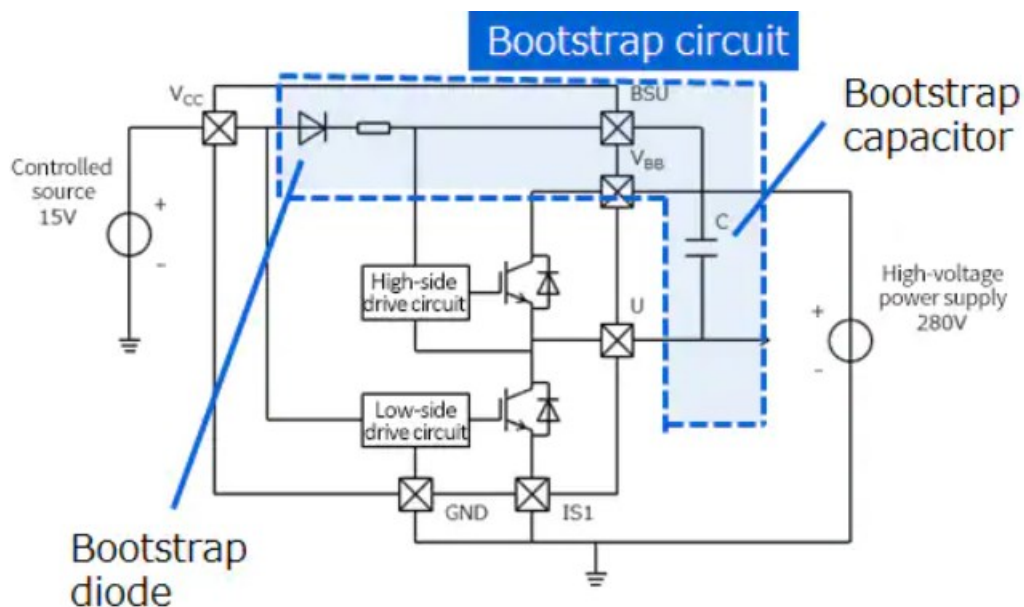
H-most je ostvaren pomoću 8 (2x4) IRF630 tranzistora. U praktičnoj upotrebi, nužno je antiparalelno tranzistorima spojiti diode, jer prilikom komutacije struje, a kao posljedica elektromagnetskih pojava uzrokovanih induktivitetom zavojnice, doći će do prenapona koji će uzrokovati neželjno vođenje a time i uništenje tranzistora (energija koja se akumulirala u induktivitetu tijekom vođenja, mora se prestankom vođenja „isprazniti“). Dioda u tome slučaju omogućavaju održavanje strujnog kruga i pražnjenje akumulirane energije. No, većina današnjih tranzistora ima integrirane diode unutar kućišta, pa tako i IRF630, što se lako može provjeri u datasheetu proizvođača. Maksimalni izlazni napon tranzistora je 200 V, uz izlaznu struju 9 A, stoga odabrani tranzistor ima i više nego zadovoljavajuće specifikacije za upravljanje motorom koji ne povlači više od 1 A. Ulazni napon, odnosno napon praga koji je potrebno dovesti na upravljačku elektrodu tranzistora, je minimalno 2 V, što znači da će driver morati osigurati razliku potencijala od minimalno 2 V između upravljačke elektrode i uvoda, a način na koji će se to ostvariti objasniti će se u idućem potpoglavlju.



Slika 3.2: simbol i raspored pinova IRF630 tranzistora (1 – gate, 2 – drain, 3 – source).

3.2. Full bridge driver

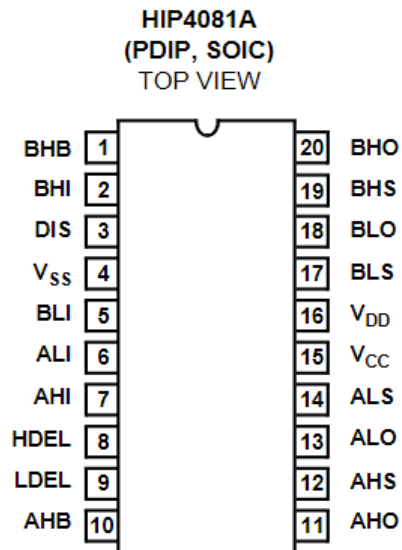
Jedna problematika koja se veže uz H – mostove je nestabilnost u radu u smislu nemogućnosti uključivanja gornjih tranzistora. Naime, kako bi se tranzistor uspješno uključio, tijekom dovođenja impulsa na gate, napon U_{GS} (između upravljačke elektrode i uvoda) mora imati odgovarajući iznos (najčešće oko 1 V) pa su uvodi tranzistora veoma često spojeni na masu. No, uvod gornjeg tranzistora prije samog paljenja je na istom potencijalu kao i odvod donjeg tranzistora na obje strane H – mosta, a taj potencijal je nedefiniran. Čak i ako se postigne paljenje mosta, javlja se problematika održavanja potrebnog napona upravljanja upravljačke elektrode tranzistora. Naime, poželjno je da padovi napona U_{DS} na tranzistorima budu što manji, tj. da je zavojnica motora spojena na napon približno jednak naponu napajanja. To znači da je željeni napon uvod gornjeg tranzistora jednak naponu zavojnice motora, tj. malo manji od napona napajanja. Stoga, da bi gornji tranzistor bio u vođenju, napon upravljačke diode mora biti veći od napona napajanja. Navedena problematika rješava se spajanjem tzv. bootstrap kruga (slika 3.3.), no u praksi se rješava korištenjem tzv. *full-bridge* ili *half-bridge* drivera. Njihov način rada i primjena opisat će se detaljnije u idućim poglavljima. Za potrebe ovog završnog rada, koristio se integrirani krug HIP4081A.



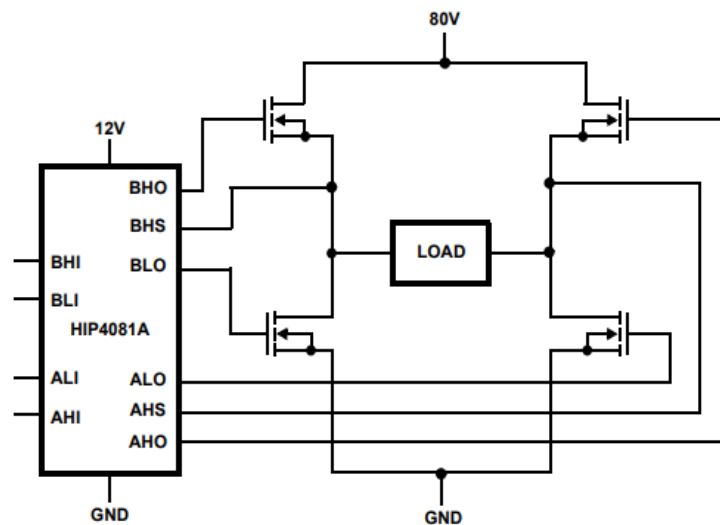
Slika 3.3: primjer rješavanja problematike korištenjem tzv. bootstrap kondenzatora. [5]

3.3. HIP4081A

HIP4081A pripada seriji integriranih krugova HIP4081x čija primarna svrha su visokofrekventna upravljanja H – mostovima (slika 3.5.). Mogu se koristiti za rad pri naponima od 10 V do 80 V, a dolaze u DIP kutiji zbog čega su bili prigodni za potrebe ovog završnog rada, jer se upravljanje motorom vrši preko breadboard pločice. Sadrže 20 pinova, čiji raspored je prikazan na slici 3.4. Opis svakog pina nalazi se u tablici 3.1.



Slika 3.4: raspored pinova HIP4081 drivera. [6]



Slika 3.5: načelna shema H – mosta upravljanog HIP4081A driverom. [6]

Generalna primjena im je u industriji, gdje god se koriste koračni ili istosmjerni motori, zvučna oprema ili za niskonaponske invertore ili DC/DC pretvarače. Sposobni su upravljati mostovima na frekvencijama do 250 kHz, stoga su i više nego primjereni za potrebe upravljanja malim koračnim motorima.

Tablica 3.1: opis pinova.

BHB	Pin za bootstrapping „B strane“ H – mosta; na njega se u paraleli spajaju dioda i kondenzator.
BHI	Input za gornji „B tranzistor“ (<i>B high-side input</i>); Prima logičke signale do max. 15 V i uključuje BHO izlaz.
DIS	Pin za gašenje svih izlaznih pinova (<i>DISable input</i>); Prima logički signal do max. 15 V.
Vss	Negativno napajanje; generalno se spaja na masu.
BLI	Input za donji „B tranzistor“ (<i>B low-side input</i>); Prima logičke signale do max. 15 V i uključuje BLO izlaz.
ALI	Input za donji „A tranzistor“ (<i>A low-side input</i>); Prima logičke signale do max. 15 V i uključuje ALO izlaz.
AHI	Input za gornji „A tranzistor“ (<i>A high-side input</i>); Prima logičke signale do max. 15 V i uključuje AHO izlaz.
HDEL	Vremenska odgoda za uključivanje gornje strane (<i>high-side turn on DELay</i>);
LDEL	Vremenska odgoda za uključivanje donje strane (<i>low-side turn on DELay</i>);
AHB	Pin za bootstrapping „A strane“ H – mosta; na njega se u paraleli spajaju dioda i kondenzator.
AHO	Izlaz za gornji „A tranzistor“ (<i>A high-side output</i>) – pin preko kojeg se šalje impuls na gate tranzistora.
AHS	Priključak za uvod (<i>source</i>) gornjeg „A tranzistora“; spaja se uz korištenje bootstrap kondenzatora.
ALO	Izlaz za donji „A tranzistor“ (<i>A low-side output</i>) – pin preko kojeg se šalje impuls na gate tranzistora.
ALS	Priključak za uvod (<i>source</i>) donjeg „A tranzistora“.
Vcc	Napajanje za gate drivere.
V _{DD}	Napajanje za donje gate drivere. Mora biti na istom potencijalu kao i Vcc.
BLS	Priključak za uvod (<i>source</i>) donjeg „B tranzistora“.
BLO	Izlaz za donji „B tranzistor“ (<i>B low-side output</i>) – pin preko kojeg se šalje impuls na gate tranzistora.
BHS	Priključak za uvod (<i>source</i>) gornjeg B tranzistora; spaja se uz korištenje bootstrap kondenzatora.
BHO	Izlaz za gornji „B tranzistor“ (<i>B high-side output</i>) – pin preko kojeg se šalje impuls na gate tranzistora.

3.4. Konfiguracija drivera

3.4.1. Napajanje i bootstrapping

Za uspješan rad drivera, prilikom dizajna je nužno voditi računa o slijedećim problematikama:

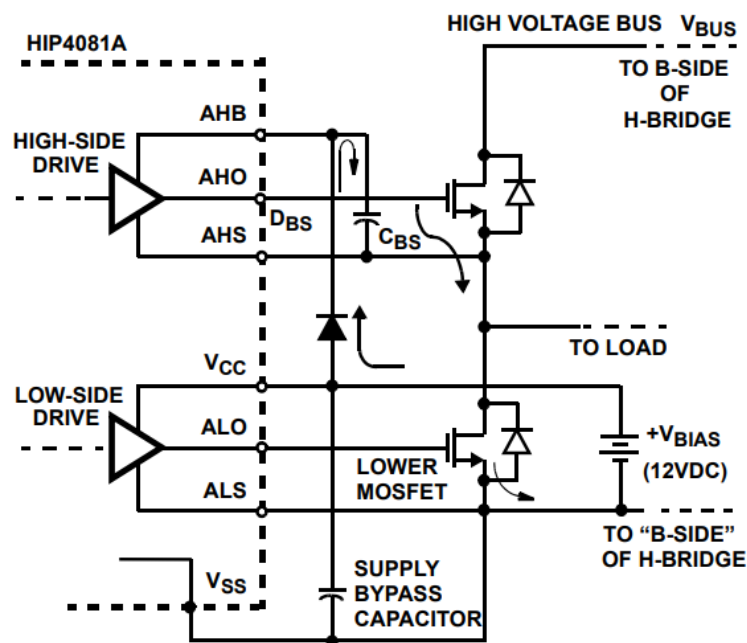
- Općenita naponska opskrba tranzistora
- Opskrbni dizajn upravljačkog kruga donjih tranzistora
- Dizajn bootstrap kruga

Općenita naponska opskrba tranzistora se odnosi na potrebu da se tranzistorima osigura potreban napon praga za uključivanje tranzistora, odnosno, potpuno otvaranje kanala kako bi se minimizirali gubici vođenja tranzistora, a da se pritom vodi računa o gate-source kapacitetu (C_{GS}), odnosno o količini naboja kojom se nabija upravljačka elektroda što utječe na vrijeme gašenja tranzistora. Rad mosta na velikim frekvencijama nosi rizik da se izazove istodobno vođenje gornjeg i donjeg tranzistora (ako je period sklapanja kraći od vremena gašenja tranzistora odnosno vremena potrebnog da se u potpunosti zatvori kanal) čime se kratko spaja izvor s masom. Vrijednost napona koji napaja čip (V_{DD} i V_{CC}) od oko 12 V trebala bi biti optimalna da se izbjegne navedena problematika.

Opskrbni dizajn upravljačkog kruga za donje tranzistore bitan je i za normalno funkcioniranje gornjih tranzistora, jer se upravljački krug (bootstrap krug) napaja upravo tijekom vođenja donjih tranzistora, stoga napajanje donjeg upravljačkog kruga mora pružati dovoljno struje za normalan rad logičkih sklopova i strujnih krugova drivera koji upravljaju gornjim i donjim dijelom H – mosta. Zato se zbog održavanja stabilnog napona između zajednički spojenih pinova V_{CC} i V_{DD} , te pina V_{SS} spaja kondenzator vrijednosti kapaciteta između 0,22 μF i 0,5 μF .

Dizajn kruga za opskrbu gornjih tranzistora se zapravo odnosi na proces poznat kao *bootstrapping*. Bootstrapping i naponska pumpa dvije su metode kojima se napaja sklop za upravljanje gornjim dijelom H – mosta. Bootstrap tehnika zapravo je odgovorna za brzo paljenje MOSFET-a, stoga ju se može preskočiti ako nema potrebe za brzim komutacijama struje kroz trošilo. Naponska pumpa je ta koja je odgovorna za uključivanje gornjih tranzistora, tako što osigurava dovoljno veliki napon između upravljačke elektrode (*gate*) i uvoda (*source*). Ona je integrirana u čip, a ukoliko nam je važno ostvariti značajne brzine uklapanja/isklapanja, važno je preko određenih pinova (u ovom

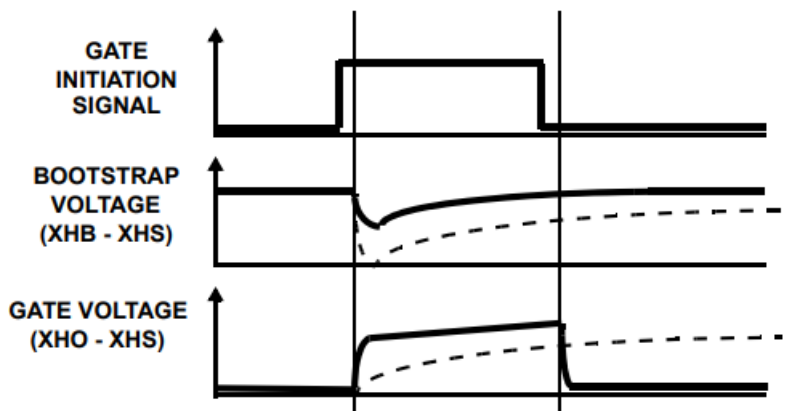
slučaju AHS, BHS, AHB i BHB pinove) realizirati sklop za bootstrap. Način realizacije prikazan je na slici 3.4. a uključuje korištenje bootstrap diode i kondenzatora. Dioda je zapravo opcionalna, no izostavljanje diode bi značilo znatno sporije punjenje kondenzatora, jer bi se u tom slučaju kondenzator punio iz naponske pumpe, koja ne pruža struju veću od $50 \mu\text{A}$, a vrijeme potrebno da se napuni kondenzator bi bilo otprilike jednako dugo ekvivalentnom vremenu punjenja kapaciteta C_{GS} . Spajanjem kondenzatora preko diode na izvor V_{CC} ga se može napuniti gotovo trenutno, neposredno nakon uključivanja donjeg strujnog kruga. Strelice na slici 3.4 prikazuju protok naboja kojime se puni kondenzator.



Slika 3.6: shema sklopa za bootstrap (shema je prikazana samo za „A-stranu“ H – mosta). [7]

Kako se bootstrap kondenzator koristi da u relativno kratkom razdoblju pruži upravljačkoj elektrodi dovoljno naboja da se stvore uvjeti za potpuno otvaranje kanala, mora biti otprilike 10 puta većeg iznosa ekvivalentnog kapaciteta C_{GS} :

$$C_{BS} \approx 10 * C_{GS}. \quad (3.1)$$



Slika 3.7: naponi na ulaznom pinu, bootstrap kondenzatoru i upravljačkoj elektrodi. [7]

Slika 3.7. prikazuje valne oblike napona na bootstrap kondenzatoru i između upravljačke elektrode i odvoda za granični slučaj (crtkane linije), u kojem je $C_{BS} = C_{GS}$, te kada je bootstrap kondenzator prikladno dimenzioniran (pune linije).

Za nastavak proračuna potrebno je poznavati nekoliko stvari. Iz datasheet-a odabranog tranzistora potrebno je isčitati naboj koji se treba dovesti upravljačkoj elektrodi za potpuno formiranje kanala. IRF630 tranzistorima treba dovesti količinu naboja od 6,3 nC u ekvivalentnom C_{GS} kapacitetu. Kako ističu proizvođači čipa, nužno je uzeti u obzir gubitak naboja koji je rezultat reverznog polariziranja diode. Naime, isključivanjem donjeg i uključivanjem gornjeg tranzistora, bootstrap dioda se reverzno polarizira i pritom „povuče“ dio naboja sa sobom takozvanom reverznom strujom, pa izgublenu količinu naboja treba uzeti u obzir pri dimenzioniranju bootstrap kondenzatora. U navedenom slučaju, korištena dioda je 1N4007 te joj vrijeme oporavka iznosi 30μs.

Iz datasheeta se isčita vršna vrijednost reverzne struje koja iznosi 5 μA. Kako je valni oblik reverzne struje otprilike trokutasti, količina reverznog naboja iznosi:

$$Q_R = \frac{30 \cdot 10^{-6} \text{s} \cdot 5 \cdot 10^{-6} \text{A}}{2} = 75 \text{ nC}. \quad (3.2)$$

Sada se može izračunati vrijednost kapaciteta bootstrap kondenzatora:

$$C_{BS} = \frac{Q_{GS} + Q_R}{\Delta V}, \quad (3.3.)$$

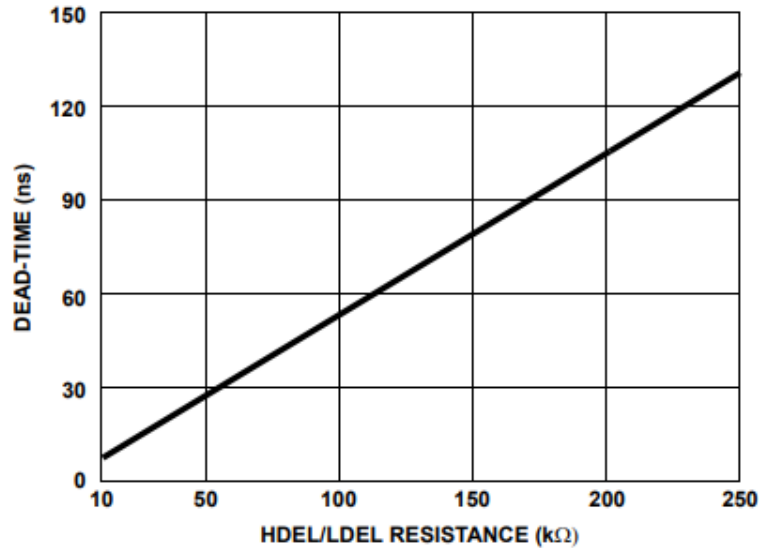
pri čemu je C_{BS} kapacitet bootstrap kondenzatora, Q_{gs} naboj koji se formiranjem kanala nakuplja u spoju gate-source, Q_{gs} količina reverznog naboja diode dok je ΔV razlika u padovima napona prije i nakon uključivanja gornjeg MOSFET-a. Uvrštavanjem poznatih vrijednosti dobivamo:

$$C_{BS} = \frac{6.3nC + 75nC}{1V} = 81,3 \text{ nF}. \quad (3.4)$$

Zaključno tomu, za ispravno funkcioniranje sklopa, potrebno je koristiti kondenzator od minimalno 80nF kako bi se na upravljačku elektrodu tranzistora dovelo dovoljno naboja za potpuno formiranje kanala, odnosno kako pad napona tijekom uključivanja gornjeg tranzistora ne bi bio veći od 1 V. Prema datasheet-u, maksimalna propusna struja diode iznosi 1 A, pa treba voditi računa o tome da u strujnom krugu ima dovoljno otpora, jer prevelika struja može izazvati pregaranje diode.

3.4.2. Propagacijsko kašnjenje

Za potpuno punjenje bootstrap kondenzatora potrebno je neko vrijeme. Pinovi HDEL i LDEL služe za regulaciju propagacijskog kašnjenja. Na njih se spajaju odgovarajući otpornici čije vrijednosti su proporcionalne sa željenim propagacijskim kašnjenjem. Graf na slici 3.8 prikazuje ovisnost kašnjenja o spojenim otporima.



Slika 3.8: ovisnost propagacijskog kašnjenja (dead-time) o spojenim otporima. [6]

U navedenom slučaju, kondenzator ima relativno mali kapacitet, ali za veće bi kapacitete idealno bilo spojiti što veći otpor kako bi se bootstrap kondenzatoru omogućilo što više vremena za punjenje, jer je važno osigurati dovoljno vremena da se bootstrap izvor opskrbi punim naponom. Ipak, nije preporučeno da se spajaju otpori veći od 250 kΩ. Spajanjem otpornika od otprilike 115 kΩ na HDEL i LDEL pinove, odgodilo bi paljenje gornjih i donjih tranzistora za otprilike 60 ns, odnosno, ukupno propagacijsko kašnjenje za period jedne komutacije struje bi iznosilo 120 ns. Prema datasheetu, naboj kojime je potrebno nabiti C_{GS} da bi se u potpunosti formirao kanal iznosi 6,3 nC. Uz napon upravljačke elektrode 2 V, kapacitet iznosi:

$$C_{GS} = \frac{Q_{GS}}{U_{GS}} = \frac{6.3 \cdot 10^{-9} \text{As}}{2\text{V}} = 3,15 \text{ nF}. \quad (3.5)$$

Pretpostavimo da je realni otpor na putu do upravljačke diode 1Ω, vremenska konstanta iznosi:

$$\tau = R * C = 1 \Omega * 3.15 * 10^{-9} \text{F} = 3,15 \text{ ns}, \quad (3.6)$$

što znači da će se kapacitet C_{GS} u potpunosti napuniti za otprilike 5 tau, što iznosi 16 ns.

Zbrojeno s propagacijskim kašnjenjem (60 ns), te prijenosnim vremenom između aktivacije input-output pinova za gornje tranzistore ($\approx 60\text{ns}$) dobiva se ukupno vrijeme 136 ns od dolaska signala na

ulazne pinove integriranog kruga do potpunog otvaranja kanala i ulaska u stanje vođenja. Naponska pumpa HIP4081 čipa pruža struju od maksimalno 50 μA . Ako uzmemo potrebnu količinu naboja za potpuno formiranje kanala, dobivamo da će naponska pumpa moći uključiti tranzistor tek nakon:

$$t = \frac{6.3 \cdot 10^{-9} \text{C}}{50 \cdot 10^{-6} \text{A}} = 126 \mu\text{s}. \quad (3.7)$$

Iz navedenog je vidljiv razlog zbog kojeg se koristi bootstrap tehnika. Kada ne bi bilo diode, bootstrap kondenzator bi punila samo naponska pumpa, što bi bilo vremenski zahtjevno i neizvedivo u primjenama s brzim komutacijama tranzistora, jer bi se cjelokupni proces u tom slučaju usporio skoro 1000 puta. Upravo zbog režima rada na visokim frekvencijama, preporučljivo je u bootstrap sklopu koristiti Schottky diode, jer su znatno brže od običnih silicijskih dioda, te imaju manji pad napona u stanju vođenja.

3.4.3. Ulazna logika (*input logic*)

Ulazna logika čipa je vrlo jednostavna. Minimalni napon koji je potrebno dovesti na ulazne pinove integriranog kruga iznosi 2,5 V i to se smatra logičkom jedinicom. Isto tako, svi naponi koji padaju ispod 1 V smatraju se logičkom nulom. Preporučena vrijednost napona ipak je nešto viša, 5 V. Uz poštivanje gore navedenih procedura za konfiguraciju, dovođenjem logičke jedinice na ulazne pinove integriranog kruga, odnosno, najčešće na parove ulaznih pinova AHI-BLI ili BHI-ALI, trebalo bi aktivirati odgovarajuće izlazne pinove, ili parove AHO-BLO ili BHO-ALO. Vrijedno je spomena da ukoliko istovremeno dovedemo ulazne signale na istovrsne ulazne pinove, npr. AHI i ALI, čip je konstruiran na način da se izbjegne istovremeno paljenje tranzistora koji pripadaju istoj strani mosta (što bi rezultiralo kratkim spojem). Kao što prikazuje slika 3.9, ulazna logika je zamišljena na način da se paljenjem donjeg tranzistora automatski isključuje pripadajući xHO izlaz. Ako na primjer dovedemo logičku jedinicu na ALI, čip će držati AHO isključenim bez obzira na to je li na AHI prisutan ulazni signal ili ne. Dakle, *low input* pinovi su prioritetizirani nad *high input* pinovima.

INPUT				OUTPUT	
ALI, BLI	AHI, BHI	U/V	DIS	ALO, BLO	AHO, BHO
X	X	X	1	0	0
1	X	0	0	1	0
0	1	0	0	0	1
0	0	0	0	0	0
X	X	1	X	0	0

Tablica 3.2: tablica istine za ulazne pinove. [7]

U određenim primjenama to znači da na high input pinove možemo konstantno održavati stanje logičke jedinice (mogu konstantno biti spojeni na 5V), što može pojednostaviti upravljački krug jer se upravljački signali mogu dovoditi samo na low input pinove, ali u slučaju upravljanja koračnog motora bi to dovelo do nestabilnosti u radu, jer bi izgubili mogućnost koračnog upravljanja. Drugim riječima, s takvom ulaznom konfiguracijom integriranog čipa, struja bi uvijek prolazila kroz namotaj motora, bilo u jednom ili drugom smjeru, ali je nikada ne bi mogli zaustaviti, što je osnovni preduvjet koračnog načina rada. Pin „DIS“ ima najveći prioritet, jer dovođenjem signala na taj pin se, bez obzira na logičko stanje ostalih input pinova, gase svi output pinovi.

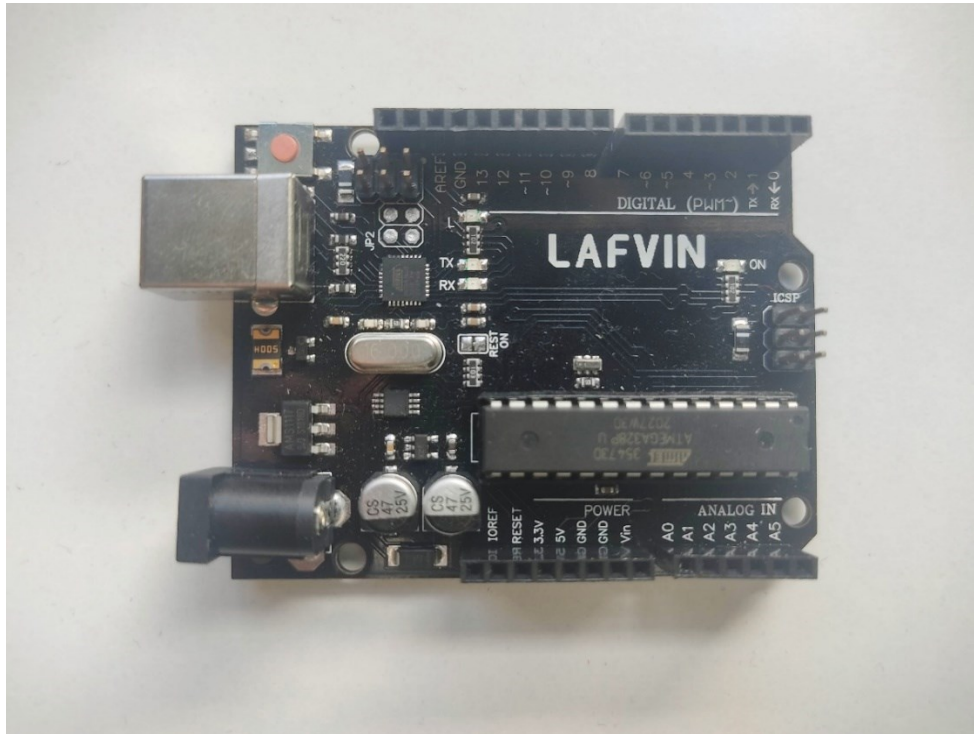
U skladu s gornjim proračunima, u tablici 3.3 navedena je detaljna specifikacija nabavljenih komponenti.

Tablica 3.3: Specifikacija komponenti.

Komponenta	Naziv/vrijednost
MOSFET	IRF630
Gate driver	HIP4081A
Bootstrap kondenzator	80 nF
Dioda	1N4007
Stabilizirajući kondenzator	0,5 μ F
Koračni motor	39BYGH405B

3.5. Arduino platforma

Arduino Uno (slika 3.9) naziv je za razvojnu pločicu čiji rad se bazira na mikrokontroleru. Mikrokontroleri su mala računala sadržana na jednom integriranom sklopu. Arduino najčešće koristi 8 bitne mikrokontrolere, koje proizvodi tvrtka ATMEL. U skladu s time, model koji se koristi na Arduinou naziva se ATMEGA328P.



Slika 3.9: izgled Arduina korištenog u završnom radu.

Sam Arduino je jednostavna pločica s 20 ulazno/izlaznih upravljačkih pinova, od kojih je 14 digitalnih, pri čemu se 6 pinova može koristiti za PWM modulaciju, a 6 analognih. Uz to, postoji i V_{IN} , pin preko kojeg se može dovesti vanjsko napajanje, različito od 5 V. Iako mu je nazivni napon napajanja 5 V, može raditi na naponima iznosa od 7 do 20 V. Sadrži i priključak za vanjsko napajanje, te ukoliko se isti koristi, na pinu V_{IN} će se pojaviti odgovarajući napon napajanja. Postoje pinovi koji kontinuirano pružaju napajanje od 5 V i 3,3 V, te pinovi za masu. Sadrži i USB priključak (USB tip B), te gumb za *reset* odnosno ponovno izvršavanje programa. Možemo ju implementirati u razne električne i elektroničke krugove, s ciljem upravljanja elektroničkim komponentama.

U slučaju upravljanja koračnim motorom, Arduino će se koristiti za slanje digitalnih impulsa na ulazne pinove MOSFET drivera. Programira se u vlastitom razvojnom sučelju (*Arduino IDE*), a kod se bazira na programskom jeziku C/C++. Kao što je vidljivo iz slike 3.10, radno okruženje Arduina vrlo je jednostavno. Sastoji se od izbornika, alatne trake, prostora za pisanje programskog koda te konzole koja otkriva informacije vezane za izvršavanje programa. Preko izbornika se vrši odabir Arduino pločice s kojom se radi i serijski port za komunikaciju s pločicom.



Slika 3.10: radno okruženje Arduina

Preko izbornika pristupamo i bibliotekama koje, između ostalog, sadrže unaprijed napisane programe za pokretanje motora. Iako je u skladu sa zadatkom završnog rada, za upravljanje koračnim motorom napisan poseban program, učinkovitost upravljačkog kruga prvotno se ispitivala

jednostavnijim kodom iz biblioteke. U nastavku je izlistan primjer koda iz biblioteke `Stepper_oneStepAtATime`.

```
#include <Stepper.h>

const int stepsPerRevolution = 400;

Stepper myStepper(stepsPerRevolution, 8, 9, 10, 11);

int stepCount = 0;

void setup() {
  // initialize the serial port:
  Serial.begin(9600);
}

void loop() {

  myStepper.step(1);
  Serial.print("steps:");
  Serial.println(stepCount);
  stepCount++;
  delay(500);

}
```

Pokretanjem programa, pinovi 8, 9, 10 i 11 slati će impulse na *low input* ulazne pinove drivera, prema tablici 2.1., dok će se high input ulazne pinove održavati na logičkoj jedinici. Takav način omogućuje isključivo polukoračni rad motora, kao što je već napomenuto, ali za potrebe testiranja je takav režim dovoljno dobar. Ako se motor okreće, znamo da upravljački krug radi i da smo točno spojili žice motora na pripadajuće pinove. *Stepper.h* naziv je biblioteke u kojoj su unaprijed definirane naredbe i funkcije za upravljanje koračnim motorima, što olakšava pisanje koda. Nakon toga je potrebno ustanoviti koliko koraka motor napravi pri jednom punom okretu, što je u ovom slučaju 400. U varijablu *stepCount* pohranjuje se broj koraka koje je poduzeo motor. Nakon inicijalizacije serijskog porta (omogućuje komunikaciju između računala i pločice), dolazi dio programa koji omogućuje

okretanje motora. Naredba *myStepper.step* je funkcija koja je zaslužna za slanje signala na izlazne pinove pločice, a *Serijal.print* ispisuje podatke na našem monitoru, da vidimo koliko je koraka napravljeno. Nakon svakog koraka, varijabla *stepCount* povišuje vrijednost, a *delay* je kratki vremenski interval tijekom kojeg je se uvodi pauza prije izvođenja iduće naredbe. Promjenom vrijednosti u zagradama zapravo određujemo koliko će se brzo motor okretati. Funkcija *void loop* se ponavlja i tako neograničen broj puta. Kod koji je korišten za potrebe rada naveden je u dodacima.

4. Zaključak

Temu upravljanja koračnim motorom upisao sam zbog zainteresiranosti za načinom funkcioniranja i primjenama istih. Kroz proces izrade završnog rada, samostalno sam proučavao literaturu i naučio kako riješiti problematiku uključivanja tranzistora u H – mostu i pravilno dimenzionirati komponente za upravljački krug H – mosta, tzv. *gate drivere* i njihov način rada. Također sam samostalno vršio nabavu komponenata i uvjerio se u važnu ulogu koju imaju proračuni, naročito u procesu odabira prikladnih dijelova za izradu sklopa. Zbog nedovoljno preciznih proračuna, nabavku sam vršio u više navrata što je ispalo neekonomičnije nego da sam uložio malo više truda i vremena da sve detaljno i pravovremeno specificiram pa da mogu potrebne komponente kupiti u jednom navratu i da pritom zadovoljim kriterije kvalitete i ekonomičnosti. Metodom pokušaja i neuspjeha i uz pomoć mentora sam naučio o nekim tipičnim pogreškama prilikom slaganja sklopa (npr. pravilno spajanje pinova, pravilno korištenje ulaznih upravljačkih pinova), kako iskorisiti mjerne instrumente za otkrivanje nekih manje očitih pogrešaka te samostalnim pisanjem koda nadogradio svoje iskustvo u području programiranja automatiziranih postrojenja. Sada u najmanje detalje znam opisati kako rade koračni motori te što sve stoji iza tih procesa.

5. Literatura

- [1] ATO, „Stepper Motor Applications: in Clock, Industrial Robot & Packaging Machine“, <https://www.ato.com/stepper-motor-applications>, 4.11.2022.
- [2] Tech Explorations, „Unipolar vs bipolar stepper motors“, <https://techexplorations.com/guides/arduino/motors/unipolar-vs-bipolar-stepper-motors/> , 2.6.2022.
- [3] Elprocus, „What is a stepper motor driver: types and its applications“, <https://www.elprocus.com/what-is-a-stepper-motor-driver-types-and-its-applications/>, 4.11.2022.
- [4] Dokumentacija proizvođača: 2-PHASE STEPPER MOTOR https://www.tme.eu/Document/288203d2e4a4157569d1fb637b7f00e9/39bygh_032011_EN-1.pdf , 4.11.2022.
- [5] Toshiba, „What us the bootstrap circuit in high-voltage IPDs?“, https://toshiba.semicon-storage.com/ap-en/semiconductor/knowledge/faq/linear_high-voltage-ipds/high-voltage06.html , 2.6.2022.
- [6] Renesas, „HIP4081 Datasheet“, <https://www.renesas.com/eu/en/document/dst/hip4081a-datasheet> , 5.6.2022.
- [7] Renesas, „AN9405“, <https://www.renesas.com/eu/en/document/apn/an9405-hip4081a-80v-high-frequency-h-bridge-driver-hip4080a81aevalz?language=en> , 5.6.2022.

Sažetak

Osnovna ideja završnog rada bila je upoznati se sa konstrukcijom i načinom rada koračnih motora, izrada sklopa i razvoj programa za regulaciju rada koračnog motora. Izrađena su dva H – mosta korištenjem MOSFET tranzistora i full bridge gate drivera koji omogućuju promjenu smjera struje kroz namotaje motora. Samim sklopom upravlja Arduino razvojna pločica, na kojoj je razvijen jednostavan program koji omogućuje promjenu načina rada, te upravljanje brzinom koračnog motora.

Ključne riječi: koračni motor, full bridge driver, bootstrap metoda, Arduino.

Abstract

The main idea of this undergraduate thesis is the better introduction to the construction and the working principle of stepper motors. Two H – bridges were created using MOSFET transistors and a full bridge gate driver which enable alternate current through the windings of the motor. The driver circuit is operated by the Arduino board, which was used to develop a simple program that enables the regulation of speed and working mode of the motor.

Key words: stepper motors, full bridge driver, bootstrap technique, Arduino.

6. Dodatak A – kod za upravljanje motorom

```
#include <stdio.h>

#define AH 4
#define AL 5
#define BH 6
#define BL 7
#define CH 8
#define CL 9
#define DH 10
#define DL 11

#define DLY 200

int n=0;

/* Preko naredbi define, program će za za neku konstantnu
vrijednost dobiti željeno ime. Odabir imena je po slijedećem
principu: prvo slovo označava stranu H - mosta (A i B za
jedan H - most, C i D za drugi), a drugo slovo označava
položaj tranzistora ( H je gornji, L je donji tranzistor).
DLY predstavlja temeljnu pauzu koja iznosi 200 ms. Definira
se i varijabla n koja predstavlja broj koraka, te joj se
dodijeljuje početna vrijednost 0.
*/
```

```
void setup()
{
  Serial.begin(9600);
  pinMode(AH, OUTPUT);
  pinMode(AL, OUTPUT);
  pinMode(BH, OUTPUT);
  pinMode(BL, OUTPUT);
  pinMode(CH, OUTPUT);
  pinMode(CL, OUTPUT);
  pinMode(DH, OUTPUT);
  pinMode(DL, OUTPUT);

  digitalWrite(AH, LOW);
  digitalWrite(BH, LOW);
  digitalWrite(CH, LOW);
  digitalWrite(DH, LOW);

  digitalWrite(AL, LOW);
  digitalWrite(BL, LOW);
  digitalWrite(CL, LOW);
  digitalWrite(DL, LOW);
}
/* U funkciji void setup određuje se da će definirani pinovi
biti izlazni, te im se pripisuje početno stanje 0.
*/
```

```

void loop()
{
  int m,b;
  Serial.println("Unesite željeni način rada: 1 za polukoračni, 2
za koračni rad po polukoracima, 3 za koračni rad. ");
  while(Serial.available() == 0)
  {
    delay(500);
    m = Serial.read();
    m = m - 48;
  }
  Serial.read();

  if(m<1 || m>3)
  {
    Serial.println("Pogrešan unos.");
    return;
  }

  /*U funkciji void loop definiramo varijable m i b koje
predstavljaju način rada i brzinu, te preko serijske
komunikacije prvo omogućavamo odabir načina rada. Funkcija
while(Serial.available() == 0) se izvodi sve dok korisnik ne
unese traženi broj. Serial.available provjerava serijski
buffer i ukoliko je prazan vraća vrijednost 0. Treba imati
na umu da se svaki unos preko tipkovice registrira u serial
buffer, pa tako i pritisak tipke enter, o čemu se mora
voditi računa i isprazniti serial buffer običnom naredbom
Serial.read(). Kako vrijednosti varijable predstavljaju
način rada, svaki unos koji je izvan domene 1-3 smatra se
pogrešnim unosom. Identičan postupak ponavlja se za odabir
brzine vrtnje, te se preko Serial.printova ispisuje odabrana
brzina i način rada.
*/

```

```
Serial.println("Unesite željenu brzinu vrtnje: 1 za sporo, 2 za
brzo, 3 za jako brzo.");
while(Serial.available() == 0)
{
    delay(500);
    b = Serial.read();
//Čekanje za unos
    b = b - 48;
//'0'
}
Serial.read();
//Pražnjenje buffera (' ')
Serial.print("Odabran je mode: ");Serial.print(m);Serial.print("
Brzina: ");Serial.println(b);
Serial.println("Unesite 0 za zaustavljanje
pogona");Serial.println("");.println("Pogrešan unos.");
return;
}
```

```

while(m>=0 && m<=3)
{
//////////KOD ZA POLUKORAČNI RAD//////////
  if (m==1)
  {
    n++;
    if (n==9)
    {
      n = 1;          //n se mora resetirati nakon
svakog ciklusa
    }
    korak(n,HIGH);    //funkcija s dva inputa za poduzimanje
koraka
    Serial.println(n); //ispis koraka
    delay(DLY/b);     //pauza između koraka, ovisna o
odabranoj brzini
    korak(n,LOW);
    if (Serial.available() > 0) //provjera serial buffera, u
slučaju da korisnik želi zaustaviti pogon
    {
      m = Serial.read();
      m = m - 48;
    }
    Serial.read();
  }

/* Kod za polukoračni rad započinje ukoliko je ispunjena petlja
if (m==1). Potrebno je voditi računa da se broj n održava u
rasponu od 1 do 8 (jer to je ukupan broj koraka koje motor može
ostvariti) stoga ga se provjerava i ako je potrebno, ispravlja
još jednom if petljom. Potom se izvodi ručno napisana funkcija
korak, a nova if petlja provjerava je li korisnik u međuvremenu
unio drukčiji unos. Ako je, naredba Serial.available vraća broj
znakova koji su uneseni.
*/

```

```

//////////////////////////////////KOD ZA KORAČNI RAD PO POLUKORACIMA//////////////////////////////////
  if (m==2)
  {
  n = n+2;
  if ( n%2 != 0)
  {
    n = n-1;
  }
  if (n==10)
  {
    n = 2;
  }
  korak(n,HIGH);
  Serial.println(n);
  delay(DLY/b);
  korak(n,LOW);
  if (Serial.available() > 0)
  {
  m = Serial.read();
  m = m - 48;
  }
  Serial.read();
  }

/* Kod za koračni rad po polukoracima je gotovo identičan, ali sadrži još jednu
if pretlju koja provjerava ako se n ima vrijednost koja nije karakteristična za
položaj motora u tom režimu. Npr. korak n = 3 nije karakterističan za navedeni
režim, pa se smanjuje na najbliži susjedni korak.
*/

```

```

//////////////////////////////////KOD ZA KORAČNI RAD//////////////////////////////////
  if (m==3)
  {
    n = n+2;
    if ( n%2 == 0)
    {
      n = n-1;
    }
    if (n>=9)
    {
      n = 1;
    }
    korak(n,HIGH);
    Serial.println(n);
    delay(DLY/b);
    korak(n,LOW);
    if (Serial.available() > 0)
    {
      m = Serial.read();
      m = m - 48;
    }
    Serial.read();
  }
  if (m==0)
  {
    break;           //ako je koisnik unio 0, iziđi iz petlje
  }
}
}

```

```

void korak(int a, bool stanje) // a=n , stanje može biti HIGH ili LOW
{ switch(a) {
    case 1:
        //AB i CD: A = +; B = -; C = 0; D = 0;
        digitalWrite(AH, stanje);
        digitalWrite(BL, stanje);
        break;
    case 2:
        //AB i CD: A = +; B = -; C = +; D = -;
        digitalWrite(AH, stanje); digitalWrite(BL, stanje);
        digitalWrite(CH, stanje); digitalWrite(DL, stanje);
        break;
    case 3:
        //AB i CD: A = 0; B = 0; C = +; D = -;
        digitalWrite(CH, stanje);
        digitalWrite(DL, stanje);
        break;
    case 4:
        //AB i CD: A = -; B = +; C = +; D = -;
        digitalWrite(AL, stanje); digitalWrite(BH, stanje);
        digitalWrite(CH, stanje); digitalWrite(DL, stanje);
        break;
    case 5:
        //AB i CD: A = -; B = +; C = 0; D = 0;
        digitalWrite(AL, stanje);
        digitalWrite(BH, stanje);
        break;
    case 6:
        //AB i CD: A = -; B = +; C = -; D = +;
        digitalWrite(AL, stanje); digitalWrite(BH, stanje);
        digitalWrite(CL, stanje); digitalWrite(DH, stanje);
        break;
    case 7:
        //AB i CD: A = 0; B = 0; C = -; D = +;
        digitalWrite(CL, stanje);
        digitalWrite(DH, stanje);
        break;
    case 8:
        //AB i CD: A = +; B = -; C = -; D = +;
        digitalWrite(AH, stanje); digitalWrite(BL, stanje);
        digitalWrite(CL, stanje); digitalWrite(DH, stanje);
        break;
}}
/* ovdje je ispisana switch-case petlja pomoću koje se, ovisno o stanju varijable n,
aktiviraju odgovarajući pinovi što omogućava okretanje motora.
*/

```