Upravljački sklop za BLDC motor

Bačić, Ivan

SVEUČILIŠTE JOSIPA JURJA STROSSMAYERA U OSIJEKU

FAKULTET ELEKTROTEHNIKE, RAČUNARSTVA I INFORMACIJSKIH TEHNOLOGIJA

Sveučilišni studij

Upravljački sklop za BLDC motor

Završni rad

Ivan Bačić

Osijek, 2019.

FERIT
FAKULTET ELEKTROTEHNIKE, RAČUNARSTVA I INFORMACIJSKIH TEHNOLOGIJA OSIJEK

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

Osijek, 03.07.2019.

Odboru za završne i diplomske ispite

Prijedlog ocjene završnog rada

Ime i prezime studenta:	Ivan Bačić
Studij, smjer:	Preddiplomski sveučilišni studij Računarstvo
Mat. br. studenta, godina upisa:	4032, 24.09.2018.
OIB studenta:	72296877290
Mentor:	Izv. prof. dr. sc. Davor Vinko
Sumentor:	
Sumentor iz tvrtke:	
Naslov završnog rada:	Upravljački sklop za BLDC motor
Znanstvena grana rada:	**Elektronika (zn. polje elektrotehnika)**
Predložena ocjena završnog rada:	Izvrstan (5)
Kratko obrazloženje ocjene prema Kriterijima za ocjenjivanje završnih i diplomskih radova:	Primjena znanja stečenih na fakultetu: 3 bod/boda Postignuti rezultati u odnosu na složenost zadatka: 2 bod/boda Jasnoća pismenog izražavanja: 2 bod/boda Razina samostalnosti: 3 razina
Datum prijedloga ocjene mentora:	03.07.2019.
Datum potvrde ocjene Odbora:	10.07.2019.

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

FERIT
FAKULTET ELEKTROTEHNIKE, RAČUNARSTVA
I INFORMACIJSKIH TEHNOLOGIJA OSIJEK

IZJAVA O ORIGINALNOSTI RADA

Osijek, 11.07.2019.

Ime i prezime studenta:	Ivan Bačić
Studij:	Preddiplomski sveučilišni studij Računarstvo
Mat. br. studenta, godina upisa:	4032, 24.09.2018.
Ephorus podudaranje [%]:	7

Ovom izjavom izjavljujem da je rad pod nazivom**: Upravljački sklop za BLDC motor**

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

i sumentora

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

Potpis studenta:

Sadržaj

1. UVOD

Prvo treba definirati što je to „BLDC“ (eng. *Brushless direct current*) motor. Postoje dvije najčešće definicije za istosmjerni motor bez četkica. Jedna od njih govori da je BLDC samo onaj motor koji prima kvadratni signal dok motori koji su pokretani sa sinusnim signalom trebaju zvati sinkroni motor sa trajnim magnetima (PMSM, eng. *Permanent magnet synchronous motor*), dok druga definicija govori da se oba motora mogu zvati BLDC. U ovom radu se koristi BLDC motor s trajnim magnetima, tri faze i upravljan kvadratnim signalom. Motor se sastoji od rotora sa trajnim magnetima i statora s zavojnicama koje predstavljaju elektromagnete. Zavojnice su spojene u zvijezdu, fizičke implementacije često imaju veći broj zavojnica i trajnih magneta, ali uvijek umnožak broja tri, no u teoriji ih se često prikazuje kao tri zavojnice i tri magneta zbog jednostavnosti objašnjavanja rada motora. Zbog toga BLDC ima kompliciran upravljački sklop jer iz istosmjernog izvora mora napraviti trofazni pravokutni izmjenični signal. Razlikuju se dva glavna principa upravljanja BLDC motorom. Prvi princip je upravljanje pomoću senzora za poziciju rotora s obzirom na stator. Takva metoda omogućuje vrlo precizno određivanje i upravljanje brzinom, pozicijom, okretnim momentom i optimalnom potrošnjom. Ovakav dizajn motora ima brojne prednosti kao što su jednostavna struktura, visoka efikasnost, veliki okretni moment, lagana promjena smjera, mali omjer snage za veličinu i masu, dugotrajan, nema četkice koje se istroše s vremenom. Zbog svih tih prednosti danas se BLDC koristi u brojnim područijima [1, 3, 4].

1.1 Zadatak završnog rada

Analiza i usporedba različitih izvedbi upravljačkih sklopova za BLDC motor.

2. ISTOSMJERNI MOTOR BEZ ČETKICA (BLDC)

2.1.Povijest BLDC motora

Praktična upotreba motora s trajnim magnetima nije započeta desetljećima poslije izuma prvih električnih motora. Prvi električni motori koristili su trajne magnete koji su tada bili vrlo loše kvalitete, što je značilo da se takvi motori nisu mogli koristiti u industriji. Jedan od prvih osoba koje su se bavile eksperimentima u području elektriciteta i magnetizma bio je Michael Faraday, koji je također uspješno napravio prvi električni motor. Pomoću ideja Cristiana Oersteda na generiranju magnetskih polja električnom strujom i Williama Wallostana koji je uspio postići okretanje žica kojom protječe električna struja u magnetskom polju trajnog magneta i svojih eksperimenata, Faraday je uspio sagraditi električni uređaj koji je pretvarao električnu energiju u kinetičku energiju (rotiranje). Taj uređaj je imao statične i rotirajuće trajne magnete, posude žive i bateriju. Kada je baterija bila spojena u krug, struja bi stvarala magnetska polja koja bi djelovala na trajne magnete, a rezultat te interakcije je okretanje električnog motora.

Kasnije Dovenport Thomas unaprijedio je dizajn električnog motora od Faradaya i dobio prvi patent za električni motor, no to nije bilo dovoljno za komercijalni uspjeh. U isto vrijeme drugi su zamjenili trajne magnete s elektromagnetima što je bilo puno isplativije, snažnije i učinkovitije. Ti novi DC motori, pogonjeni na istosmjernu struju, ostali su mnogo godina u industriji.

Povratak motora s trajnim magnetima nije bio moguć sve do 1930-tih kada je znanost o trajnim magnetima znatno napredovola. Otkrivanje novih materijala koji su povećavali snagu trajnih magneta je trajalo mnogo godina do 1980. kada je otkriven spoj neodimija, željeza i bora, koji je bio puno jači nego prethodni magneti. Istosmjerni magneti bez četkica s trajnim magnetima (PM BLDC) pojavili su se na tržištu tek 1970-tih, no kvaliteta trajnih magneta nije bila jedini razlog odgode razvitka takvih motora. Bitan element u razvoju bila je zamjena mehaničke komutacije s električnom, što je tek bilo omogućeno naprednom poluvodičkom tehnologijom [1, 3].

Jedan od prvih velikih projekata koji je znatno pridonio razvitku istosmjernog motora bez četkica bila je Apollo svemirska letjelica, točnije sustav za potporu života u letjelici i protok kisika u prijenosnom sustavu za život. Uvjeti u kojima je takav motor trebao funkcionirati bili su sasvim drugačiji od onih na Zemlji. Glavni uvjet bio je da motor ne proizvodi iskre jer se nalazio u čistom kisiku. Također, bilo je potrebno zadovoljiti druge uvjete kao što su visoki

početni okretni moment, visoka efikasnost, kontrola brzine i trajnost. Zbog tih uvjeta izabran je BLDC koji se počeo intenzivno razvijati. Fotodiode su korištene u prvom dizajnu, no zamjenjene su s *Hall*-efekt senzorima. Rezultat rješavanja tih problema bio je puno napredniji BLDC motor za komercijalne svrhe u raznim područijima [2].

2.2. Usporedba BLDC motora i istosmjernog motora

Motor za istosmjernu struju sastoji se od rotora s dva pola (armatura) i statora s trajnim magnetom, komutatora i četkica. DC motor funkcionira tako da se komutator rotira i svakih 180° promjeni smjer struje u zavojnici rotora što znači promjenu smjera generiranog magnetskog polja na koje utječu trajni magneti statora i tako postiže rotaciju. Komutator mijenja smjer struje mehanički dodirujući četkice koje su spojene na izvor električne energije [4, 7].

Tablica 2.1: Usporedba BLDC motora i DC motora

BLDC motor	DC motor
• Vrlo učinkoviti i s manjim magnetima • Veliki raspon maksimalnog okretnog momenta • Puno veća kontrola nad upravljanjem što daje veću preciznost • Dugotrajnost zbog nedostatka četkica • Jednostavno održavanje • Tih • Ne proizvodi iskre	• Maksimalni okretni moment samo na određenoj brzini • Potrebni veći magneti za veći okretni moment • Kontrola je vrlo jednostavna ali manje precizna • Četkice se troše i moraju se mijenjati kao i komutator • Četkice proizvode iskre

2.3. Primjena BLDC motora

BLDC se koristi u mnogim područjima i u različitim uvjetima rada. Autoindustrija je gospodarska grana koja je sve više zainteresirana za BLDC zbog njegove efikasnosti i veličine, ali neki od problema su cijena motora i kompleksno upravljanje. No problemi skupog i kompleksnog upravljanja BLDC-om postaje sve manji zahvaljujući jeftinoj poluvodičkoj tehnologiji i mikrokontrolerima koji omogućuju upotrebu naprednih algoritama za kontrolu BLDC-a. Njihova najčešća upotreba u autoindustriji nije za pogon električnih automobila već za upravljanje drugim elementima, kao što su električna vrata, električni dizači prozora, brisači stakla i klima uređaji. BLDC motor se sve češće upotrebljava u industrijama koje zahtijevaju precizne pokrete, kao što su automatizirane tvornice gdje se koriste robotske ruke. U takvim uvjetima koriste se motori s povratnom vezom za poziciju rotora. Ti motori su skuplji pri izradi, ali zadržavaju sve ostale prednosti kao što su veliki okretni moment, brza reakcija i visoka učinkovitost. Zbog svih pozitivnih i negativnih karakteristika, najčešći oblici BLDC-a su mala veličina sa slabom snagom jer se visoka cijena motora može vrlo brzo isplatiti uštedom energije i izdržljivost u zahtjevnim uvjetima. Vrlo su popularni za pogon dronova i drugih uređaja na daljinu, također se koriste za pumpe, ventilatore i kompresore [4, 5].

3. UPRAVLJANJE BLDC MOTOROM

3.1. Pogonski strujni krug za upravljanje BLDC motorom

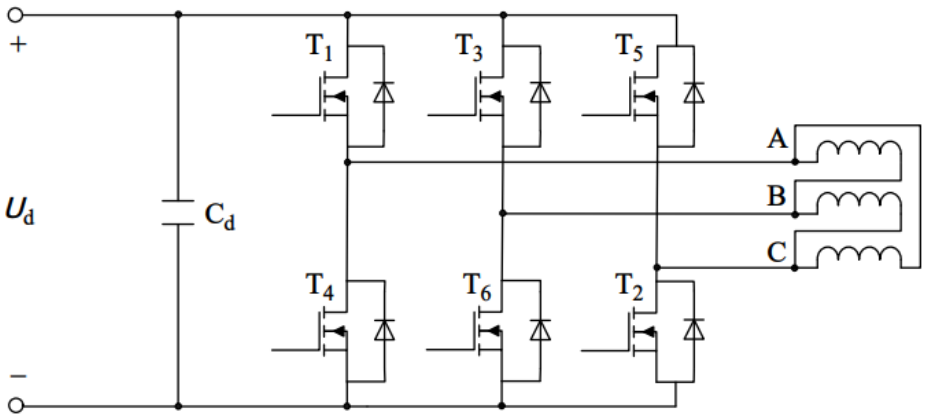
Kod upravljanja trofaznog BLDC motora s trajnim magnetima, prvo se mora izabrati pogonski sklop koji se spaja na tri faze motora. Pogonski sklopovi rade s visokim strujama i visokim frekvencijama paljenja i gašenja. Zbog takvih potreba najčešće se koriste *power* MOSFET-i koji mogu podnijeti velike struje i frekvencije paljenja i gašenja.

3.1.1. Konfiguracija punog mosta

Ovakav sklop sastoji se od šest MOSFET-a koji služe za upravljanje strujama od tri faze motora u spoju zvijezde. Dva najčešća načina korištenja ovoga sklopa su uključene dvije faze i uključene tri faze, što znači da su u svakom trenutku provode dvije ili tri faze.

Kod uključene dvije faze, princip rada je da dvije faze motora uvijek provode struju dok je treća isključena ili slobodna (odspojena). Redoslijed uključivanja faza određuje kut rotora, a promjena koraka se radi svakih 60° okreta rotora, a kroz svaku fazu struja teče 120°. Zato ima šest koraka (promjena) da bi se rotor motora okrenuo za puni krug, zatim se koraci ponavljaju. To znači da je uvijek jedan gornji MOSFET uključen koji dovodi struju u jednu od faza motora i da je jedan donji MOSFET uključen koji odvodi struju iz jedne od drugih faza na kojoj nije već uključen gornji MOSFET.

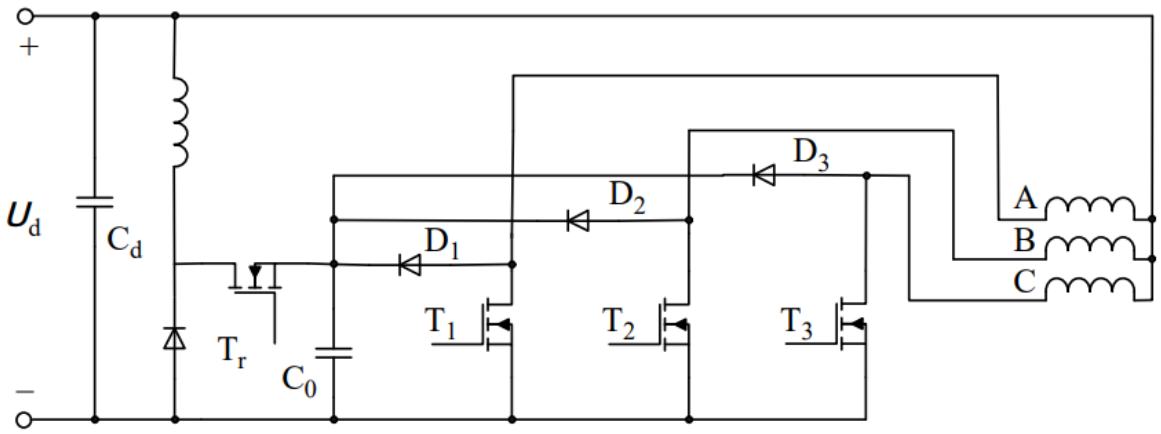
U načinu s uključenim trima fazama rada, uključene su tri faze motora u svakom trenutku. Glavne razlike su: kroz fazu protječe struja 180° okreta motora i drugačiji redoslijed uključivanja faza. To može povećati korištenje svake faze i smanjiti nepoželjne smetnje zakretnog momenta (eng. *torque ripples*), ali može također dovesti do istovremenog uključivanja gornjeg i donjeg MOSFET-a iste faze što je vrlo opasno i može oštetiti strujni krug.



Slika 3.1: konfiguracija punog mosta [4]

3.1.2. „*C-Dump*“ konfiguracija

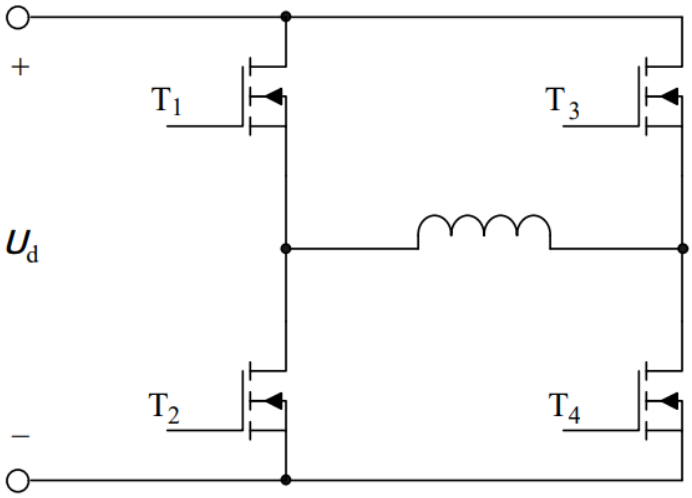
Glavne prednosti ove konfiguracije su niska cijena i manja fizička veličina. Ovaj sklop koristi samo četiri MOSFET-a za pokretanje motora, ima manje gubitke energije nego puni most, ali ima velike valove zakretnog momenta kod komutacije. Štednju energije postiže tako što djelomično usmjerava prikupljenu magnetsku energiju motora u kapacitet C_0.



Slika 3.2: „C dump“ konfiguracija [4]

3.1.3. „H“ most konfiguracija

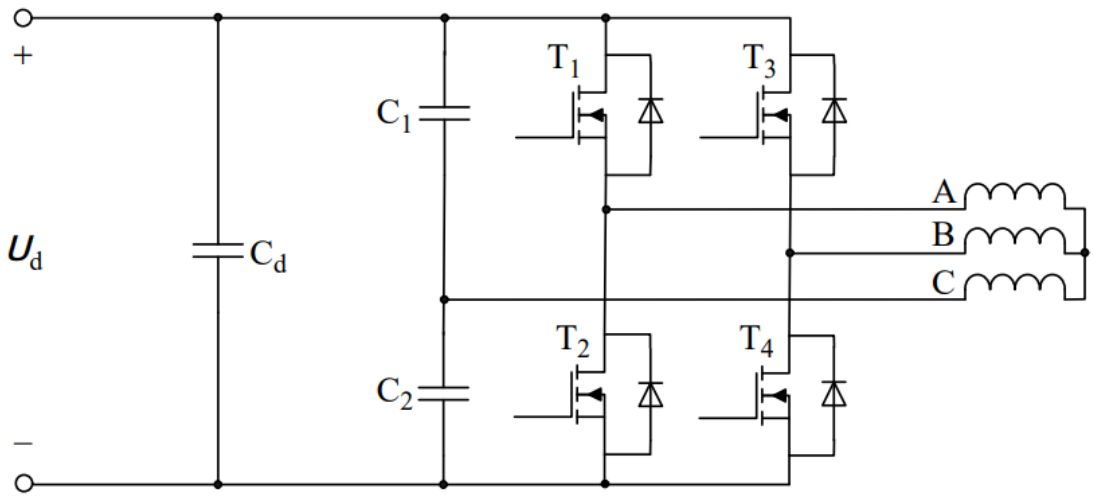
Svaka faza je kontrolirana jednim „H“ mostom što pojednostavljuje upravljanje BLDC-om, ali čini ovakav pristup znatno skupljim jer koristi 12 MOSFET-a za trofazni motor. Iz tog razloga ova se konfiguracija najčešće koristi za jednofazne motore.



Slika 3.3: „H“ most konfiguracija [4]

3.1.4. Konfiguracija s četiri MOSFET-a

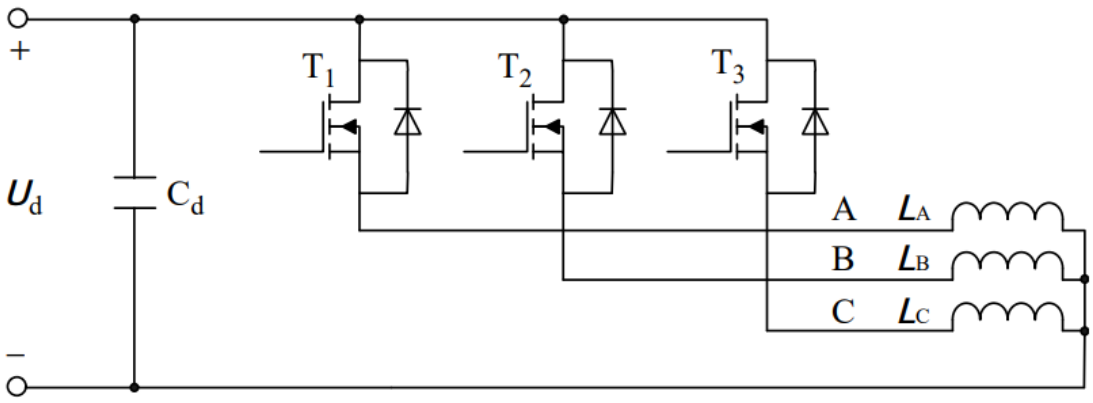
Dva MOSFET-a na jednoj fazi zamjenjuju se sa dva kondezatora tako da je između njih spojena faza C. S ovim se uštedi na MOSFET-ima i gubitcima energije, ali se znatno zakomplicira algoritam s kojim se upravlja BLDC.



Slika 3.4: konfiguracija s četiri mosfeta [4]

3.1.5. Konfiguracija polumosta

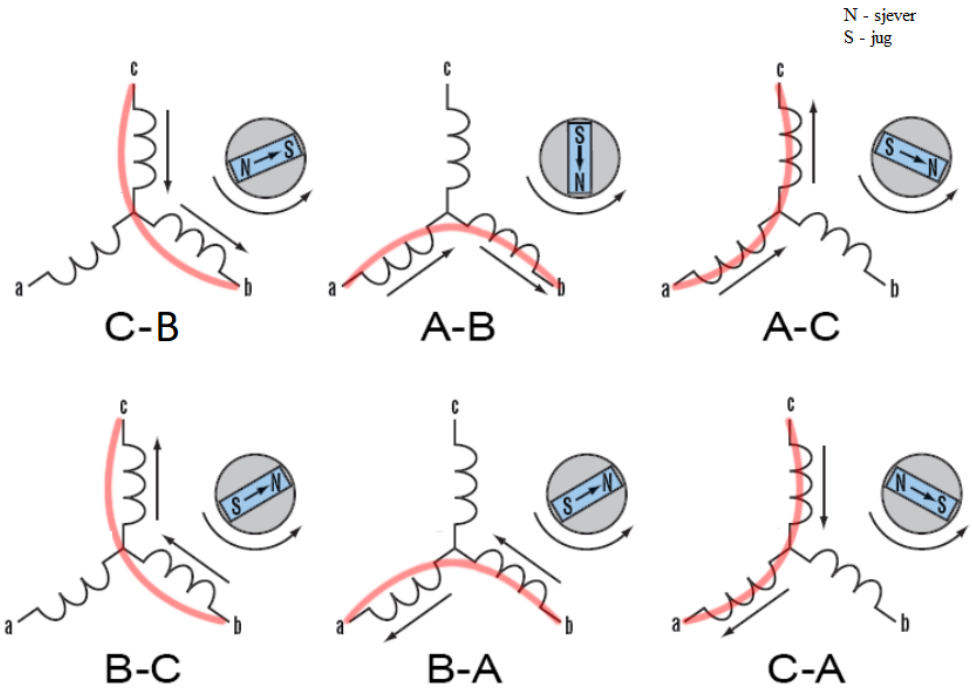
Koriste se tri MOSFET-a, jedan MOSFET za svaku fazu, što čini ovu metodu jeftinijom i jednostavnijom, ali vrlo se rijetko koristi zbog slabe iskoristivosti motora jer svaka faza vodi samo jednu trećinu okreta motora. Također, pojavljuju se velike smetnje u zakretnom momentu kod promjene faze.



Slika 3.5: konfiguracija polumosta [4]

3.2. Logika upravljanja BLDC motorom

Kako bi se moglo objasniti logiku upravljanja BLDC-om prvo se mora odrediti kakav sklop se koristi. U ovom radu koristi se konfiguracija punog mosta ili tri polumosta, koja za svaku fazu ima gornji i donji MOSFET. U svakom trenutku dvije faze moraju voditi, a treća je isključena. Kako bi se motor pravilno rotirao mora se napraviti šest koraka, za svaki korak rotor motora se okrene za 60° a kroz svaku faze teče struja 120° okreta rotora (slika 3.6), ali to samo vrijedi ako motor ima tri fizičke zavojnice, što u ovom radu nije slučaj jer motor ima 12 zavojnica. To samo znači da će motor s 12 zavojnica morati ponoviti ciklus od 6 koraka 4 puta da bi napravio puni krug s rotorom i da se za svaki korak rotor okrene za 15°, a struja teče 30° okretaja rotora [9, 10, 11].



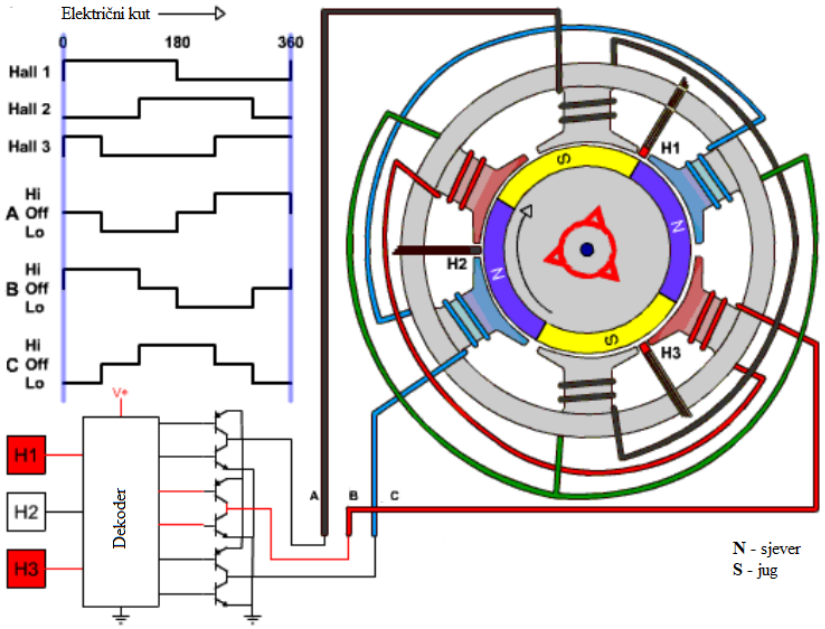
Slika 3.6: smjer struja i magnetskih polja po koracima [11]

Tablica 3.1: Tablica aktivnih faza po koracima

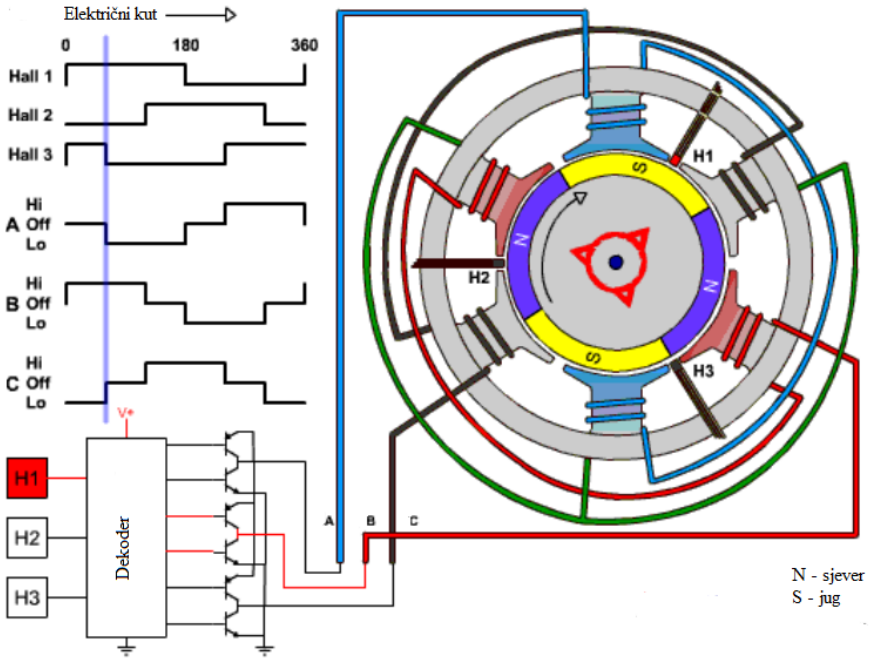
Koraci	1.	2.	3.	4.	5.	6.
Gornji mosfeti	**C**	**A**	**A**	**B**	**B**	**C**
Donji mosfeti	**B**	**B**	**C**	**C**	**A**	**A**

Za pogon motora konstantno se ponavljaju ovi koraci (Tablica 3.1), ako se želi povećati brzina ili smanjiti brzina rotacije motora, treba se regulirati vrijeme između koraka.

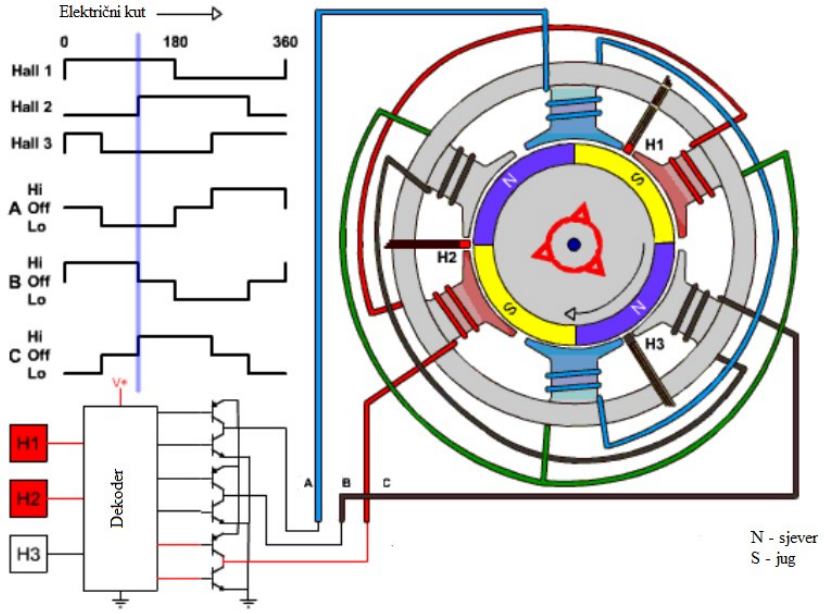
Na slikama od 3.7 do 3.11 ilustrirana je jednostavna kontrola BLDC motorom i prikazane su struje kroz zavojnice motora. U ovom primjeru motor sadrži *Hall* efekt senzor pomoću kojeg se zna u kojem je koraku i kada se treba prijeći u novi korak, signal iz senzora se dekodira pomoću dekodera i spaja se na tranzistore koji upravljaju strujama faza. Crvena boja žice znači da struja ulazi u zavojnicu, plava boja znači da struja izlazi iz zavojnice i crna znači da je faza isključena.



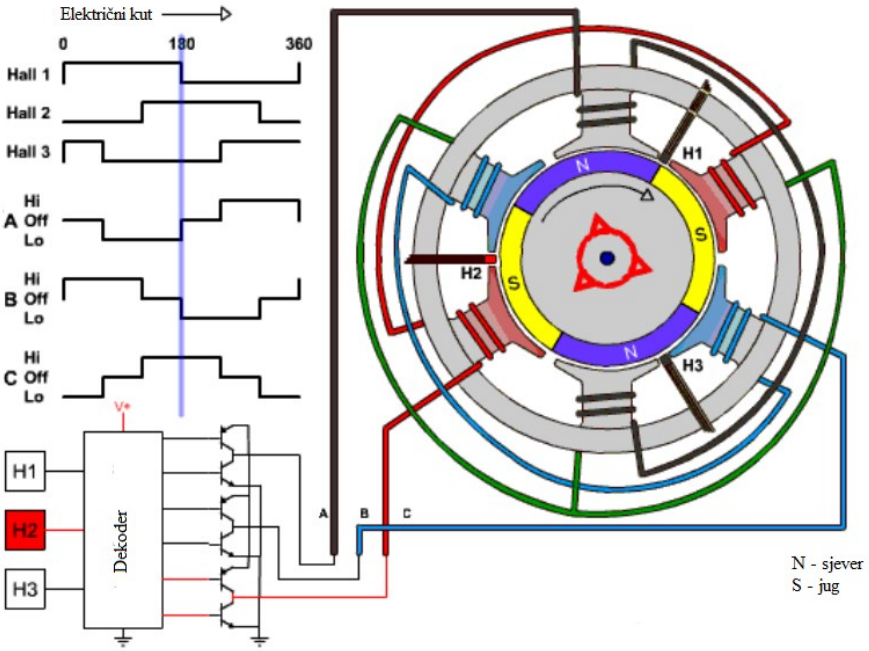
Slika 3.7: Prvi korak [11]



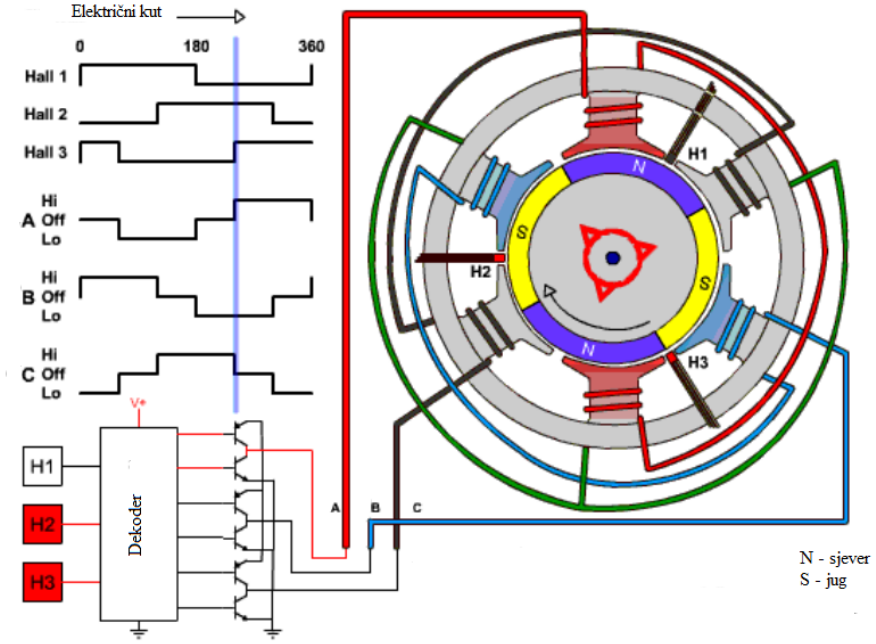
Slika 3.8: Drugi korak [11]



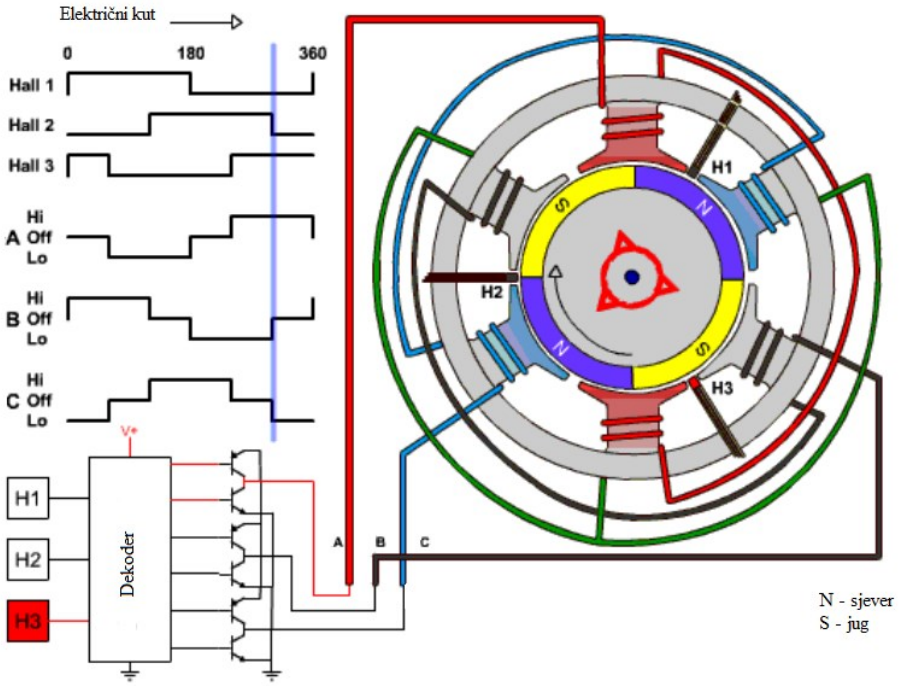
Slika 3.9: Treči korak [11]



Slika 3.10: Četvrti korak [11]



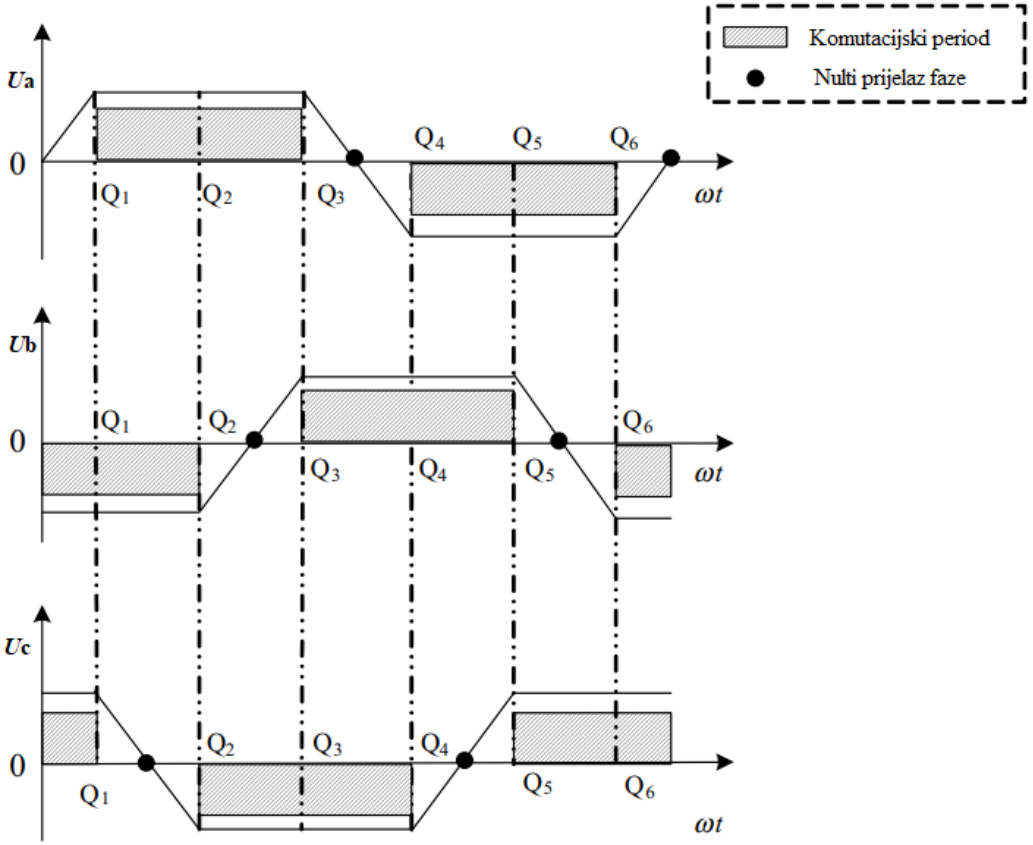
Slika 11: Peti korak [11]



Slika 3.12: Šesti korak [11]

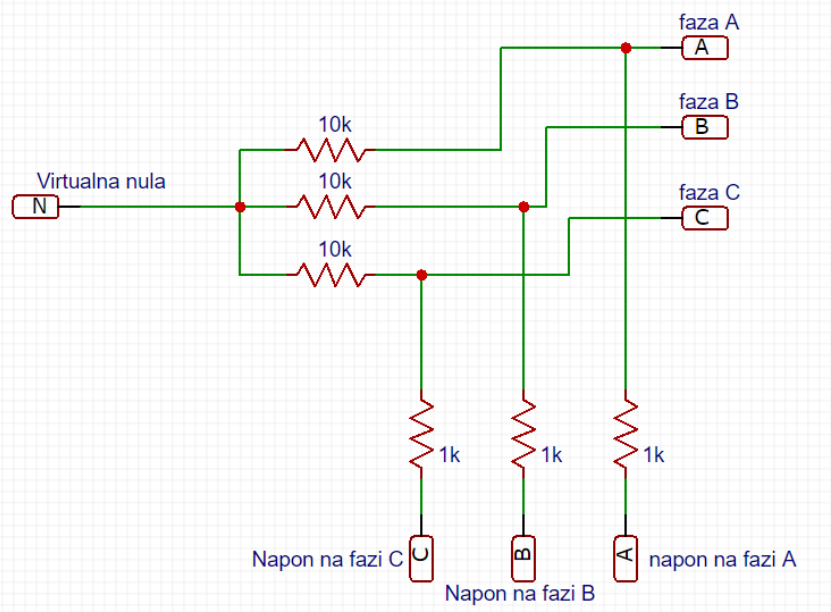
3.3. Kontrola BLDC motora bez senzora pomoću inducirane EMS

Kako se može vidjeti u prošlom poglavlju, kada se izvode koraci za upravljanje, dvije faze su uključene a jedna je isključena. Ta faza je vrlo bitna kod kontrole bez senzora pomoću inducirane EMS (povratna elektromotorna sila) i kako bi se izbjeglo korištenje *Hall* efekt senzora koji povećavaju cijenu, obujam i broj žica iz motora. Kako se vidi na slici 3.13 faza koja ne vodi prolazi preko nulte vrijednosti zbog inducirane EMS. Taj prijelaz na slobodnoj fazi lako se može izmjeriti pomoću neutralne točke motora i napona faze.



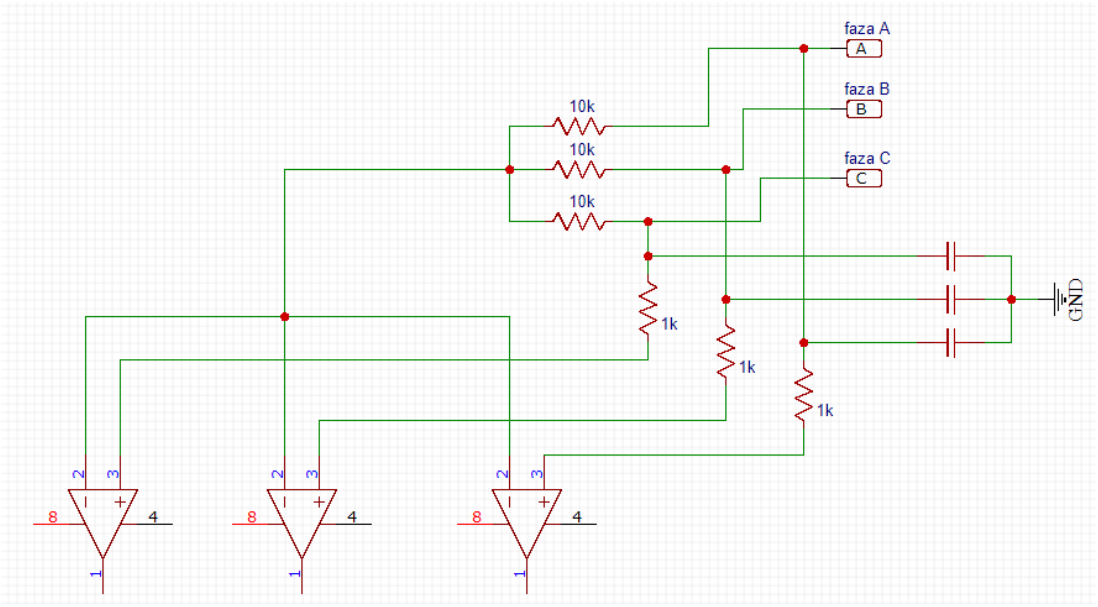
Slika 3.13 Napon na fazama BLDC motora po koracima komutacije [4]

Kako motor ima vodiče samo za tri faze, a nema neutralnu točku, pravi se virtualna neutralna točka s otpornicima spojenim na faze (slika 3.13).



Slika 3.14: Shema za mjerenje napona na fazama

Pomoću ove metode može se precizno i brzo odrediti kad se komutacija prebacuje u sljedeći korak. Glavni nedostatak ove metode je što se ne može detektirati inducirane EMS kad motor stoji i nema nikakve povratne informacije o položaju rotora. Za manje zahtjevne primjene ovaj problem se lako rješava tako što se započnu koraci bez obzira na to što se ne zna pozicija rotora. Motor se stoga može ponašati čudno pri pokretanju pomoću ove metode, postoji mogućnost pokretanja u suprotnom smjeru od namjenjenog. inducirana EMS nastaje tako što se inducira napon u slobodnoj fazi zbog prolaska magneta odnosno magnetskog polja kroz zavojnicu slobodne faze. Kad je magnet točno na zavojnici dobije se prijelaz preko nule što daje znak za sljedeći korak. Sklop za detekciju prijelaza inducirane EMS faze preko nule prikazan je na slici 3.15. Kontrola bez senzora pomoću inducirane EMS još se može implementirati programskom podrškom, čija je implementacija složenija od sklopovlja za detekciju inducirane EMS jer je teže odrediti koja faza ima prijelaz preko nule i potrebne su kompenzacije za kondezatore u sklopovlju za mjerenje napona na fazama i inducirane EMS faza. Iz tog razloga u ovom radu je implementirana metoda detekcije inducirane EMS pomoću sklopovlja.



Slika 3.15: shema sklopa za detekciju prijelaska BEMF-a faza

4. IZRADA UPRAVLJAČA BLDC MOTORA

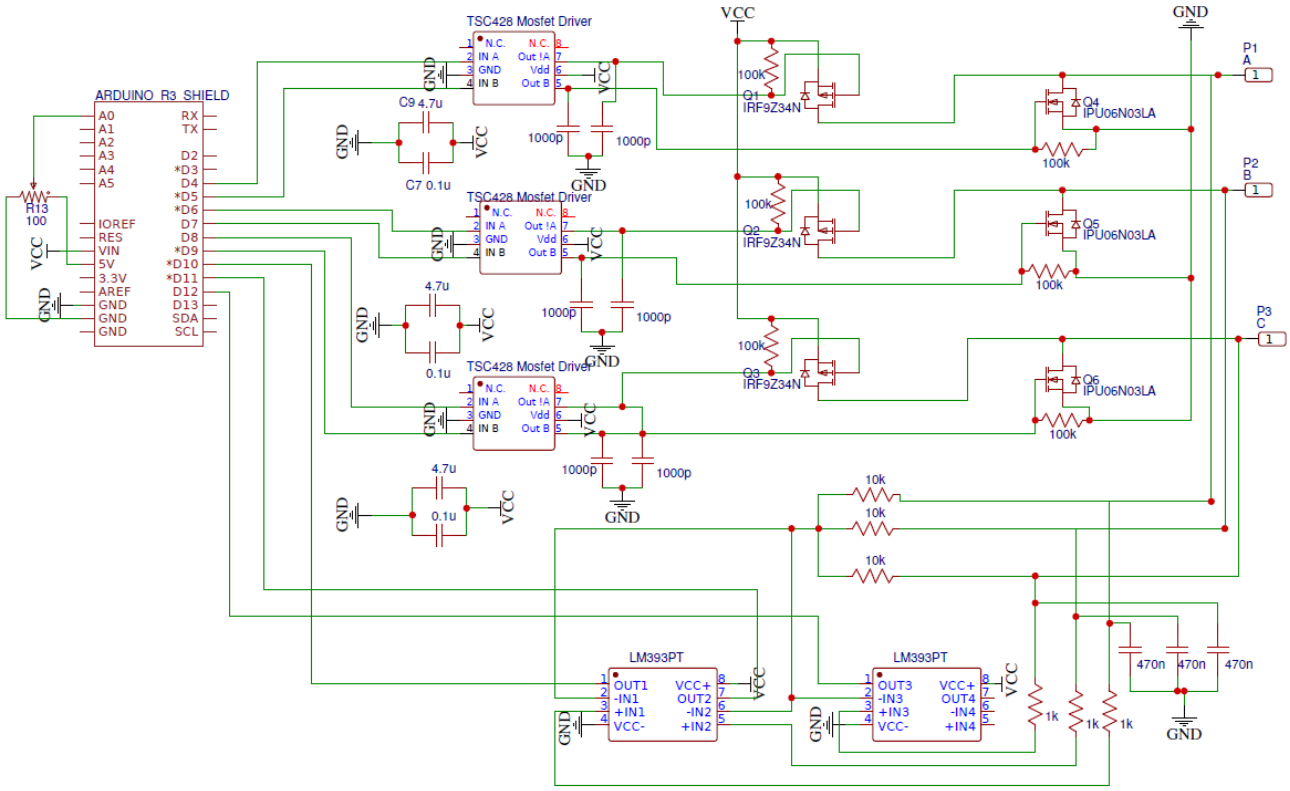
Prvi sklop za upravljanje spojen je po shemi na slici 4.1. Shema je nacrtana u programu *EasyEDA*. Pogonski strujni krug u ovom sklopu je spojen po konfiguraciji punog mosta ili tri polumosta, to znači da ima šest MOSFET-a za upravljanje s tri faze motora. Za svaku fazu koriste se dva MOSFET-a, jedan gornji i jedan donji MOSFET. Gornji MOSFET P tipa služi za dovođenje pozitivnog napona napajanja na fazu, a donji MOSFET N tipa služi za dovođenje negativnog napona napajanja. P MOSFET-i su IRF9Z34N, a N MOSFET-i su IPU06N03LA i njihove karakteristike se nalaze u *datasheet*-u u literaturi. Za gornje MOSFET-e su izabrani MOSFET-i P tipa umjesto N tipa, iako N tip imaja bolje karakteristike od P tipa, zbog pojednostavljenja kontrole jer kada se koriste gornji N MOSFET-i potrebno je dovesti napon veći od pozitivnog napona napajanja kako bi se N MOSFET uključio. U tom slučaju koristi se naponska pumpa koja nam može dati potreban napon za uključenje N MOSFET-a. Iz tog razloga se koriste P MOSFET-i za koje nema potrebe korištenja dodatnih elemenata jer treba dovesti negativni napon s obzirom na pozitivni napon izvora na *gate*. Svi MOSFET-i su obogaćenog tipa kao što je prikazano na shemi (slika 4.1). Dodani otpornici između *gate*-a i source-a služe za pražnjenje kapaciteta koji se pojavljuje u MOSFET-ima kod visokih frekvencija paljenja i gašenja.

Za upravljanje signalima paljenja i gašenja MOSFET-a po prije definiranim koracima koristi se razvojna pločica *Arduino*. Kako je digitalni signal iz *Arduina* 5 V, što nije dovoljno za paljenje MOSFET-a jer je V_{ds} MOSFET-a veći od 5 V, mora se koristiti upravljač za MOSFET-e koji za ulaz prima digitalni signal iz *Arduina*, a na izlazu daje napon jednak naponu napajanja.

Upravljač za MOSFET-e je integrirani strujni krug TSC428; jedno pakiranje sadrži dva upravljača, što za jedno pakiranje daje dva ulaza i dva izlaza, prvi izlaz je invertiran i spaja se na P MOSFET, a drugi izlaz je nepromjenjen i spaja se na N MOSFET. TSC428 kao ulaz prima logički signal, za logičku jedinicu napon mora biti minimalno 2.4 V, a za logičku nulu napon može maksimalno biti 0.8 V. Maksimalni napon napajanja TSC428 je 18 V i minimalni napon 4.5 V, detaljnije karakteristike se nalaze u literaturi.

Sklop za detekciju inducirane EMS sastoji se od dva integrirana strujna kruga LM393, svako od njih u sebi sadrži dva komparatora pomoću kojih se detektira induciranu EMS. Kao referentna vrijednost u komparatoru spaja se virtualna nula, sastavljena od tri otpornika u vrijednosti od 10 kΩ za svaku fazu spojeni u zajedničku točku. Na pozitivni konektor odnosno

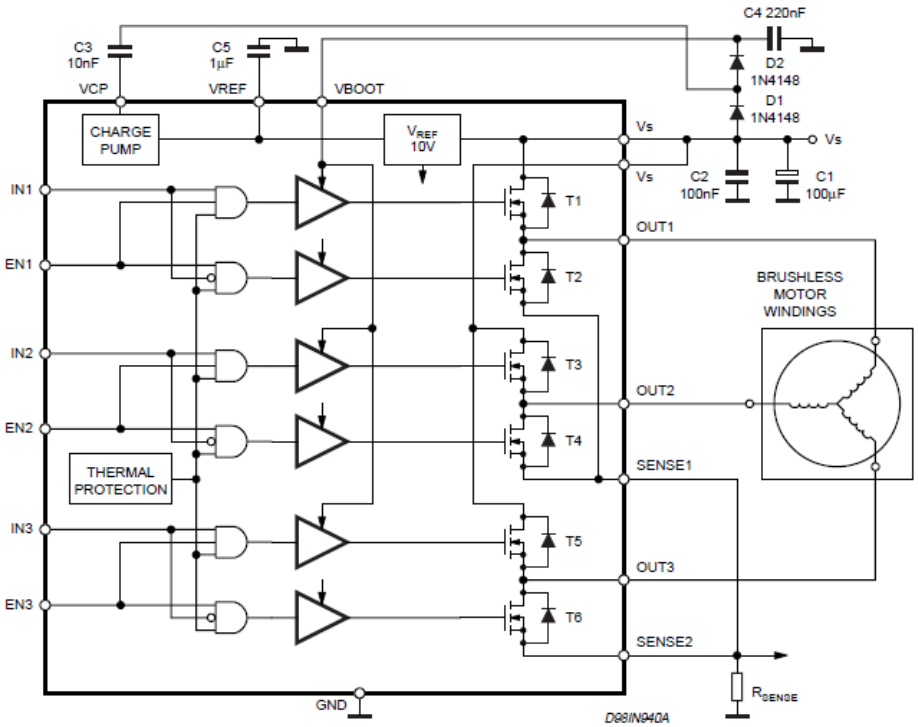
vrijednost koja se uspoređuje dovodi se napon faze pomoću jednog otpornika u vrijednosti od 1 kΩ sa svake faze posebno. Izlaz iz komparatora je digitalan što znači da kada je pozitivni ulaz veće vrijednosti od referentne, na izlazu će biti logička jedinica, a kada je pozitivni ulaz manje vrijednosti od referentne, na izlazu će biti logička nula. Izlazi iz komparatora su spojeni na digitalne ulaze *Arduina*.



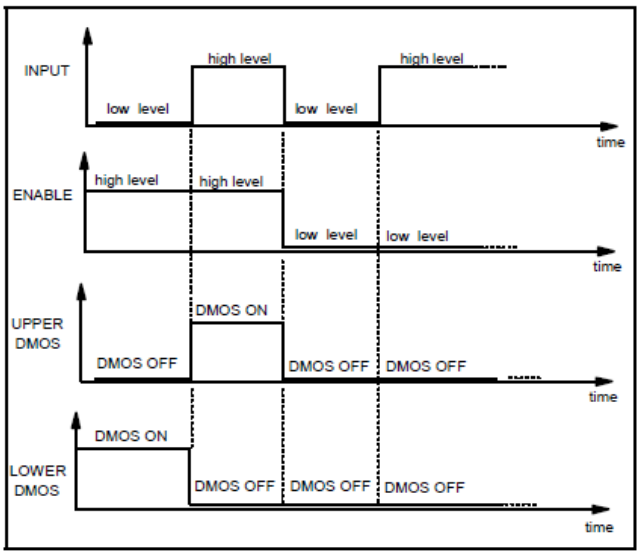
Slika 4.1: shema prvog sklopa za upravljanje BLDC motorom

Na slici 4.4 je shema drugog sklopa za pravljanje BLDC motorom, glavna razlika u odnosu na prvi sklop je ta da se koristi integrirani strujni krug LM6234 (sklia 4.2) namjenjen za pokretanje BLDC motora. LM6234 sadrži šest MOSFET-a u konfiguraciji tri polumosta. U ovoj shemi svih šest MOSFET-a su N tipa što znači da je potrebna naponska pumpa za paljenje gornjih MOSFET-a, sklop sadrži unutarnji oscilator frekvencije 1.2 MHz koji služi za punjenje naponske pumpe. Prvo se puni C3 na $V_{ref} = 10\,V$ kroz diodu D1, tada se puni C4 kroz diodu D2 i nakom par ciklusa oscilatora konektor VBOOT dostiže vrijednost napona on Vs+10V-VD1-VD2. VBOOT je spojen na upravljače gornjih MOSFET-a, što im daje dovoljan napon za paljenje gornjih N MOSFET-a. Konektori za signal kojim se upravlja MOSFET-ima su IN1, EN1, IN2, EN2, IN3, EN3; oni primaju logički signal s maksimalnim naponom od Vs-1V. IN je ulaz koji određuje hoće li biti uključen gornji ili donji MOSFET za određenu fazu, logička

jedinica uključuje gornji MOSFET, a logička nula uključuje donji MOSFET. EN je ulaz koji omogućuje da se upali bilo koji od MOSFET-a na određenoj fazi, što znači ako je EN na logičkoj nuli oba MOSFET-a na toj fazi će biti isključena (slika 4.3). L6234 sadrži logičke sklopove *i* koji osiguravaju da se ne mogu uključiti oba MOSFET-a na jednom polumostu, što bi dovelo do kratkog spoja izvora napajanja sklopa. Također, sklop sadrži senzor za zaštitu od pregrijavanja, senzor je spojen na svaki logički sklop *i* kako bi mogao ugasiti sve MOSFET-e kad temperatura prijeđe vrijednost od 160 °C.

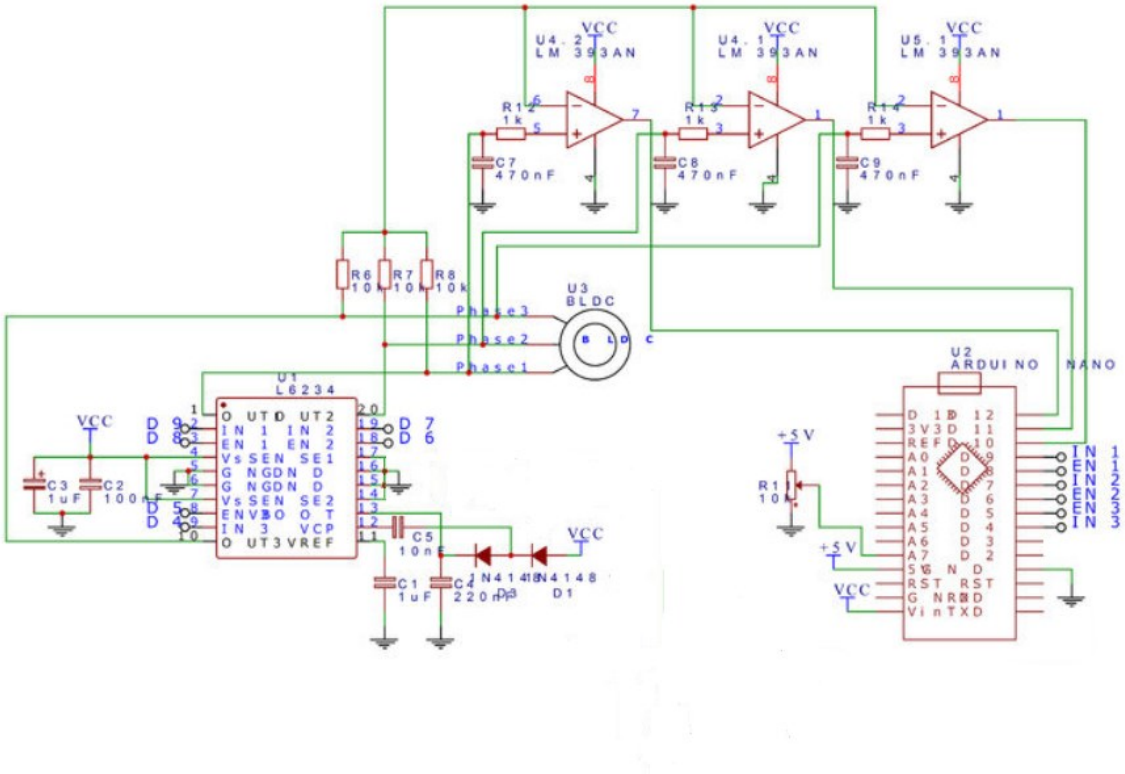


Slika 4.2: Topologija i primjer spajanja integriranog strujnog kruga LM6234



Slika 4.3: Logika upravljanja za svaki polumost

Sklop za detekciju inducirane EMS je isti kao i u prvom sklopu za upravljanje BLDC motorom (slika 4.1 i 4.4), samo su prikazani drugačijim simbolima na shemi.



Slika 4.4: shema drugog sklopa za upravljanje BLDC motorom

4.1. A*rduino* program

Program u ovom projektu je pisan u *Arduino* razvojnom okruženju. Služi za kontrolu upravljača za MOSFET-e, u nastavku su izdvojene i objašnjene glavne funkcije programa, a cijeli program se nalazi u prilogu.

U glavnog funkciji (slika 4.5 i 4.6), koja se nalazi u beskonačnoj petlji, nalazi se *switch case* funkcija koja provjerava koji je sljedeći korak potrebno pokrenuti, vrijednost *step* varijable predstavlja korak. Funkcije za korake su *step0( ), step1( ), step2( ), step3( ), step4( )* i *step5( )*, te funkcije upravljaju digitalnim pinovima 4, 5, 6, 7, 8 i 9 koji su spojeni na ulaze upravljača za MOSFET-e. Dodatne funkcije *step01( ), step11( ), step21( ), step31( ), step41( )* i *step51(),* koriste se za „rezanje“ struje, to se postiže tako što gornji MOSFET koji je uključen u određenom koraku ugasimo i upalimo donji MOSFET te faze svakih 10 µs pomoću ugrađenog 2. brojača u *Arduinu*, to se radi u funkciji ISR(TIMER2_COMPB_vect) koja izaziva prekid svakih 10 µs i poziva program u funkciji koji samo postavlja *wait* variablu na vrijednost 1 (slika 4.7). Druga funkcija ISR(TIMER2_COMPB_vect) samo ponovno poziva glavnu step funkciju svakih 50 µs i ponovno uključuje gornji MOSFET [8].

```
void loop() {
  if (doonce == 0) {
    switch (step) {
      case 0:
        if (wait == 0) {
          step0();
        }
        else {
          step01();
        }
        break;
      case 1:
        if (wait == 0) {
          step1();
        }
        else {
          step11();
        }
        break;
      case 2:
        if (wait == 0) {
          step2();
        }
        else {
          step21();
        }
        break;
```

Slika 4.5: Prvi dio glavne funkcije

```
      case 3:
        if (wait == 0) {
          step3();
        }
        else {
          step31();
        }
        break;
      case 4:
        if (wait == 0) {
          step4();
        }
        else {
          step41();
        }
        break;
      case 5:
        if (wait == 0) {
          step5();
        }
        else {
          step51();
        }
        break;
```

Slika 4.6: Drugi dio glavne funkcije

```
ISR(TIMER2_COMPB_vect) { // timer 2 current choping
  if (wait == 0) {
    wait = 1;
    doonce = 0;
  }
}

ISR(TIMER2_COMPA_vect) { // timer 2 reseting main function
  if (wait == 1) {
    wait = 0;
    doonce = 0;
  }
}
```

Slika 4.7: Funkcije za rezanje struje

Kontrola brzine se upravlja preko varijable *rotation* u koju zapisujemo vrijednost iz analognog ulaza A0 koji je spojen na vanjski potenciometar. *Rotation* ima raspon vrijednosti on 0 do 1023 i mapira se na vrijednost od 5000 do 200, jer tu vrijednost koristimo za određivanje vremena kada se pokreće sljedeći korak (slika 4.8).

$$t = \frac{rotation}{2000000} \tag{4.1}$$

```
ISR(TIMER1_COMPA_vect) {  // timer 1
  rotation = map(analogRead(A0), 0, 1023, 5000, 200);
  OCR1A = rotation;    // sets duration of timer 1 again
  step++;
  doonce = 0;
  if (step == 6) {
    step = 0;
  }
}
```

Slika 4.8: Funkcija za prebacivanje u sljedeći korak i promjena vrijednosti variable rotation

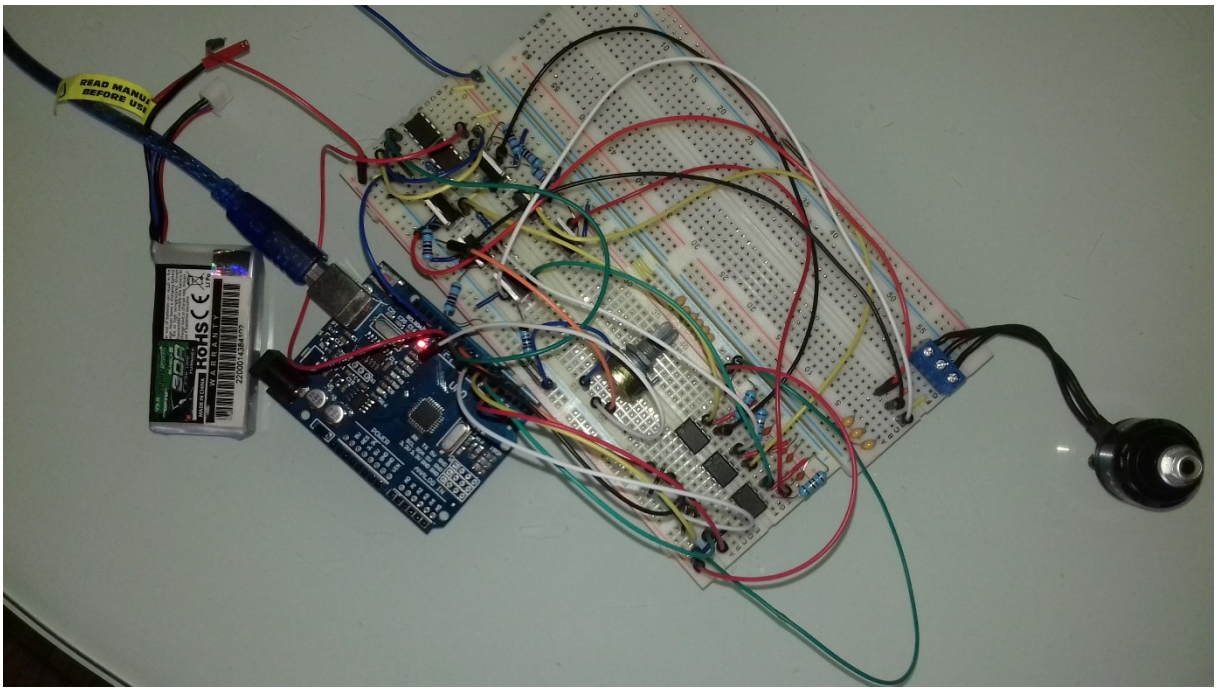
Funkcija ISR (PCINT0_vect) je za detekciju inducirane EMS i određivanje na kojoj fazi se dogodio prijelaz preko nule, služi za precizno određivanje vremena kada treba prebaciti u sljedeći korak. Ova funkcija se poziva kada se dogodi promjena na nekim od digitalnih pinova 10,11,12 na koji su spojeni izlazi komparatora. Funkcija se koristi samo u slučaju kada je *rotation* manje vrijednosti od 1000 što znači samo kada je vrijeme za sljedeći korak manje od 500 µs (slika 4.9).

```
ISR (PCINT0_vect) { // interupt
  if (rotation < 1000) { // if rotation less then 1000 check for chnge in BEMF to change to next step
    current = map(rotation, 1000, 200, 20, 50);  // increases time for current choping from range 10 us to 25 us
    OCR2B = current;
    changedbits = PINB ^ portbhistory;
    portbhistory = PINB;
    if ((changedbits & (1 << PINB2)) && (step == (2 || 5))) //Phase C (3)
    {
      step++;
      wait = 0;
      TCNT1 = 0;
      TCNT2 = 0;
      if (step == 6) {
        step = 0;
      }
      doonce = 0;
    }

    if ((changedbits & (1 << PINB3)) && (step == (0 || 3))) //Phase B (2)
    {
      step++;
      wait = 0;
      TCNT1 = 0;
      TCNT2 = 0;
      doonce = 0;
    }

    if ((changedbits & (1 << PINB4)) && (step == (1 || 4))) //Phase A (1)
    {
      step++;
      wait = 0;
      TCNT1 = 0;
      TCNT2 = 0;
      doonce = 0;
    }
  }
```

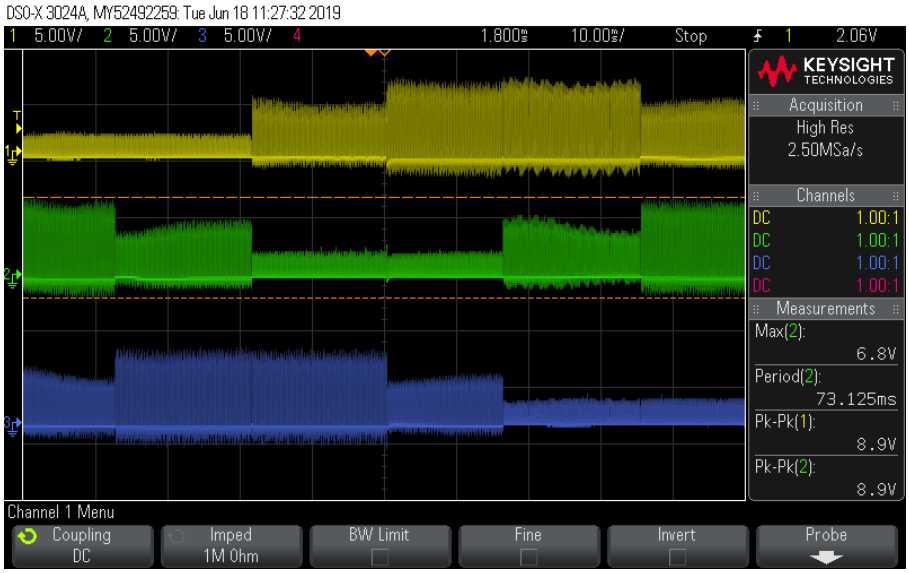
Slika 4.9: Funkcija za prebacivanje koraka s obzirom na BEMF

5. TESTIRANJE UPRAVLJAČKOG SKLOPA ZA BLDC MOTOR

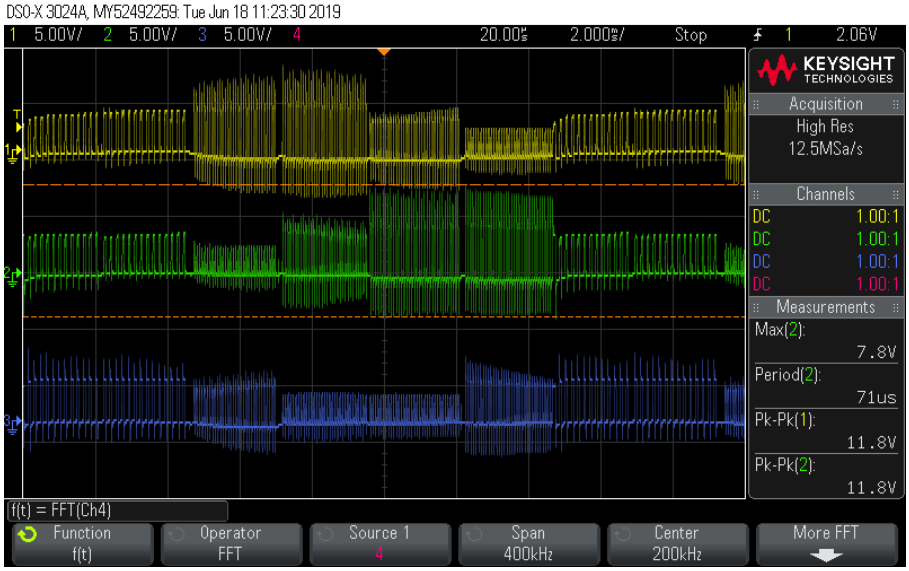
Izrađen je prvi sklop za upravljanje BLDC motorom po shemi na slici 4.1 (slika 5.1). Sklop se napaja sa strujnog izvora, napona 9 V i struje od 1 A, tijekom testiranja. Za testiranje su prikazani naponi faza s obzirom na virtualnu nulu i naponi iz digitalnih izlaza 4, 5, 6, 7, 8, 9 *Arduina*. Za testiranje je korišten motor „Turnigy D2008-2300KV Brushless Motor“. Kao izvor napajanja se još može koristiti baterija „Turnigy Bolt 500mAh 7.6V Lipoly“. Drugi sklop za upravljanje BLDC motorom po shemi na slici 4.4 nije izrađen jer koristi ugrađeni strujni krug sa većinom elemenata koji su potrebni za sklop za upravljanje BLDC motorom, i takvim sklopom se ne bi postigao drugačiji rezultat i ne bi bilo potrebe uspoređivati takva dva sklopa.



Slika 5.1: Izrađeni sklop za upravljanje BLDC motorom



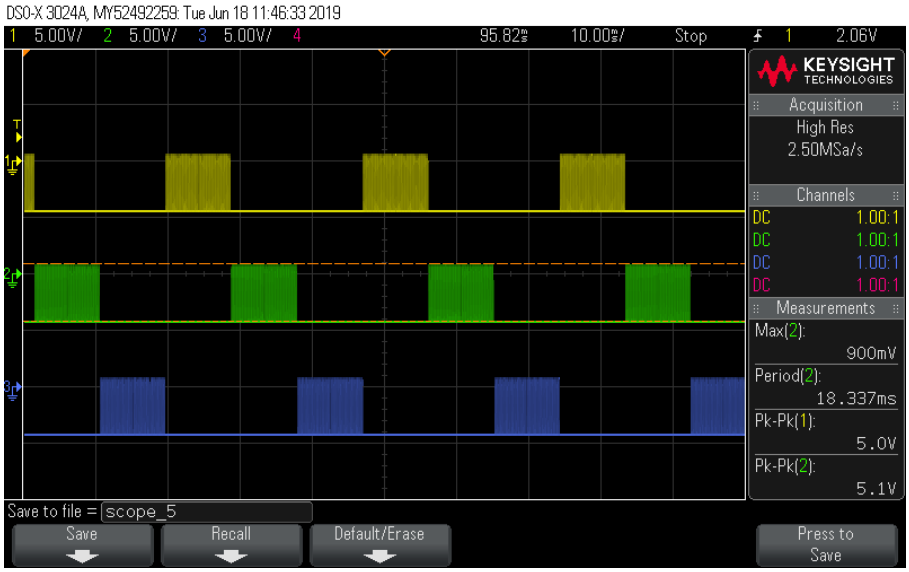
Slika 5.2: Napon na fazama motora



Slika 5.3: Napon na fazama motora u manjem vremenskom intervalu

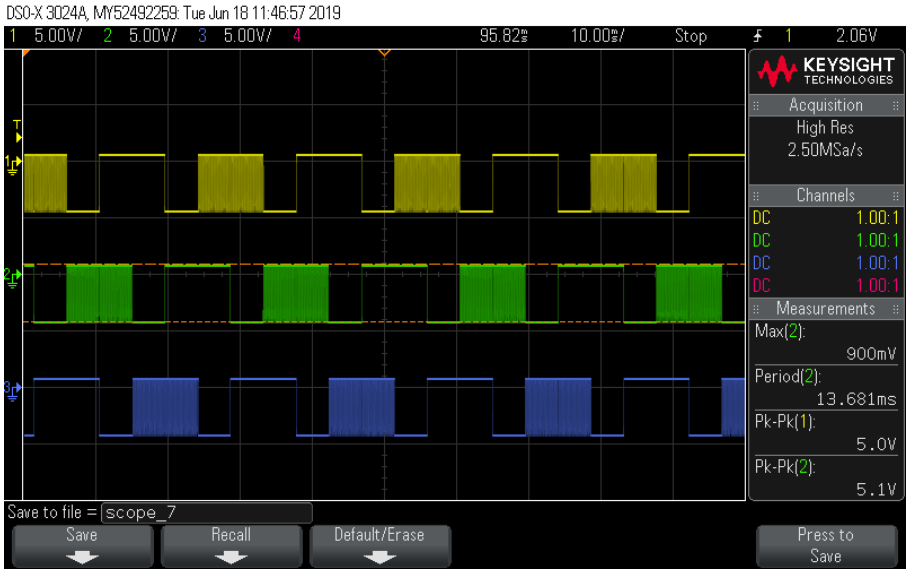
Na slikama od 5.2 i 5.3 nalaze se naponi na fazama motora s obzirom na nulu. Kada se usporede stanja faza na slikama 5.2 i 5.3 i tablicu 3.1 stanja faza prema koracima vidi se kako stanja faza odgovara predviđenim koracima. Kada je napon najveće vrijednosti tada je faza „uključena“, uključen je gornji MOSFET te faze što znači da je faza na pozitivnom naponu izvora. Kada je napon na fazi srednje vrijednosti tada nije uključen niti jedan MOSFET te faze, što znači da je faza slobodna i pojavljuje se inducirana EMS. A kada je napon faze na najmanjoj vrijednosti

faza je isključena što znači da je uključen donji MOSFET koji ju spaja na nulu izvora napajanja. No po tablici 3.1 faze bi trebale biti konstantno uključene za vrijeme jednog koraka, a kako se vidi na slici faze se konstantno pale i gase, približno svakih 50 µs, zbog funkcije „rezanja“ struje koja je implementirana u *Arduino* programu. Takva vrsta kontrole se naziva pulsno-širinska modulacija (eng. PWM). Kontrolom vremena za koje je gornji MOSFET uključen kontrolira se i koliko će snage trošiti motor i hoće li raditi bez vibracija. Kao što se vidi na slici 5.3 kada je na nekoj od faza uključen gornji MOSFET, odnosno na fazu je spojen pozitivni napon napajanja, napon na fazi brzo naraste, ali odmah počne opadati a proteče velika struja jer zavojnica ima vrlo mali otpor a veliki induktivitet. To se događa jer je na strujnom izvoru ograničena maksimalna struja na 1 A pa kada struja u zavojnicama dosegne taj maksimum napon napajanja padne na 6 V, zato se mora isključiti gornji MOSFET i uključiti donji iste faze kako bi se ograničio protok struje kroz zavojnice motora. Ta karakteristika se pojavljuje i kada je na fazi uključen donji MOSFET odnosno faza je spojena na negativni napon izvora zbog toga što je ograničen protok struje kroz obje faze jer kada se isključi gornji MOSFET prekine se dotok struje iz izvora napajanja od motora na kratko vrijeme. Sljedeće slike prikazuju signal iz digitalnih izlaza *Arduina*.



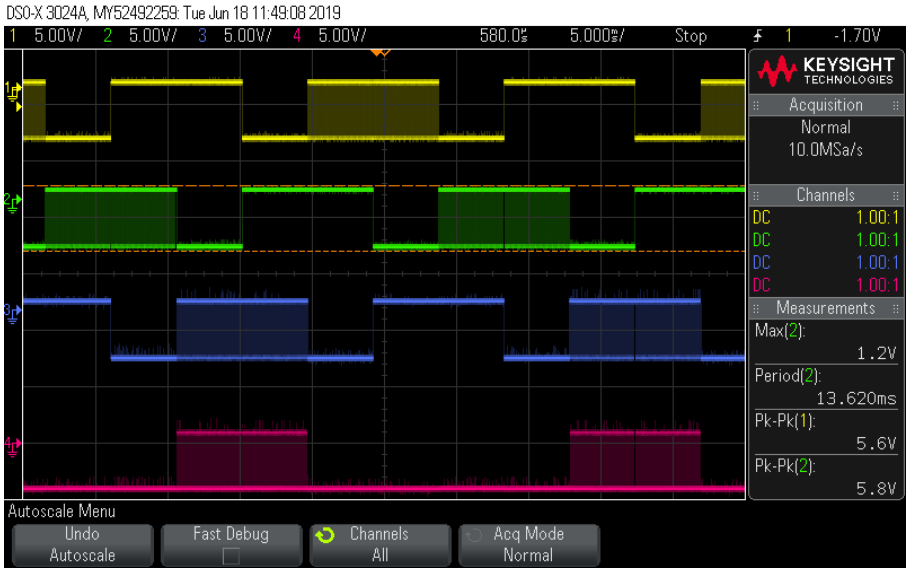
Slika 5.4: Signali iz arduina za kontrolu gornjih MOSFET-a

Na slici 5.4 spojeni su digitalni izlazi iz Arduina za kontrolu gornjih P MOSFET-a, vidi se kako dva ista gornja MOSFET-a nikada nisu uključeni u isto vrijeme.

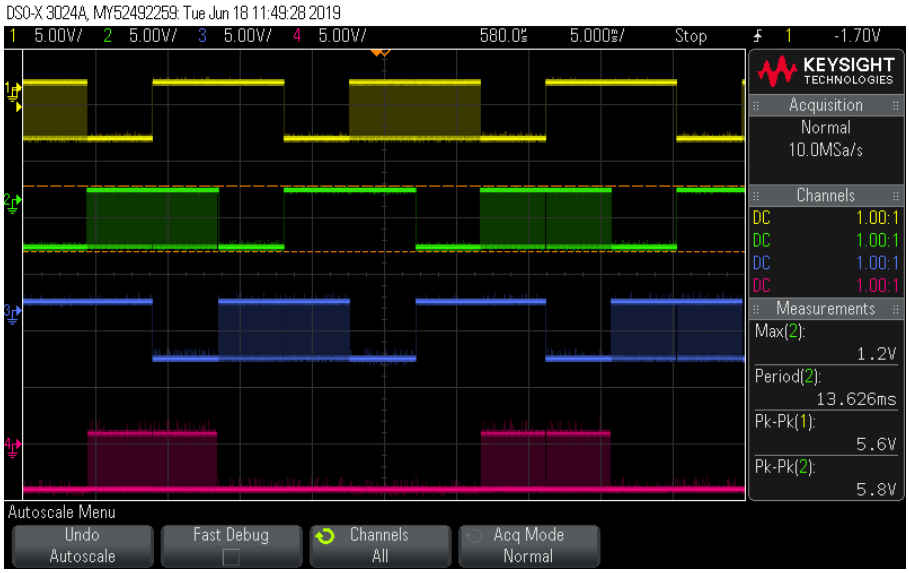


Slika 5.5: Signali iz arduina za kontrolu donjih MOSFET-a

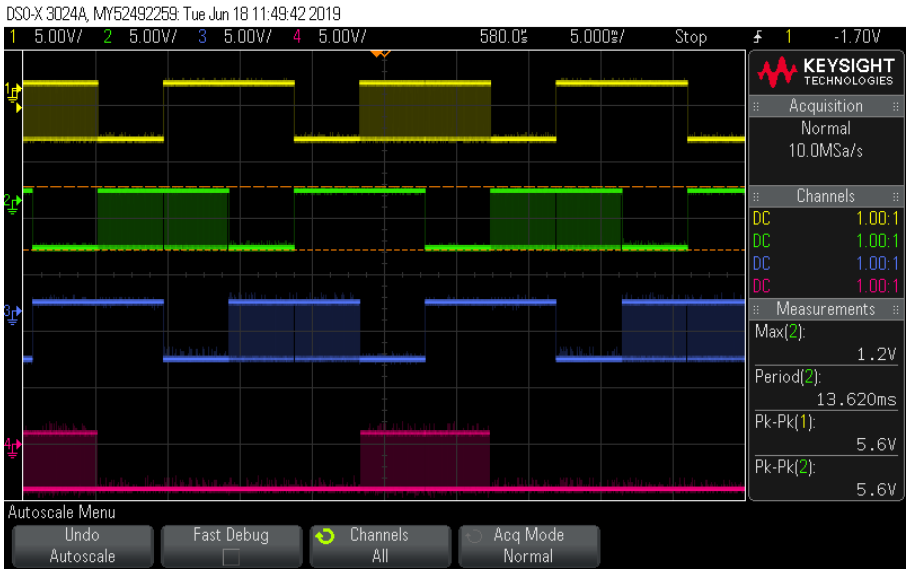
Na slici 5.5 prikazani su signali za upravljanje donjim MOSFET-ima i vidi se kako u isto vrijeme mogu biti upaljena dva MOSFET-a što je posljedica funkcije „rezanja“ struje.



Slika 5.6: Signal iz Arduina za donje MOSFET-e i gornji MOSFET C faze na 4. kanalu

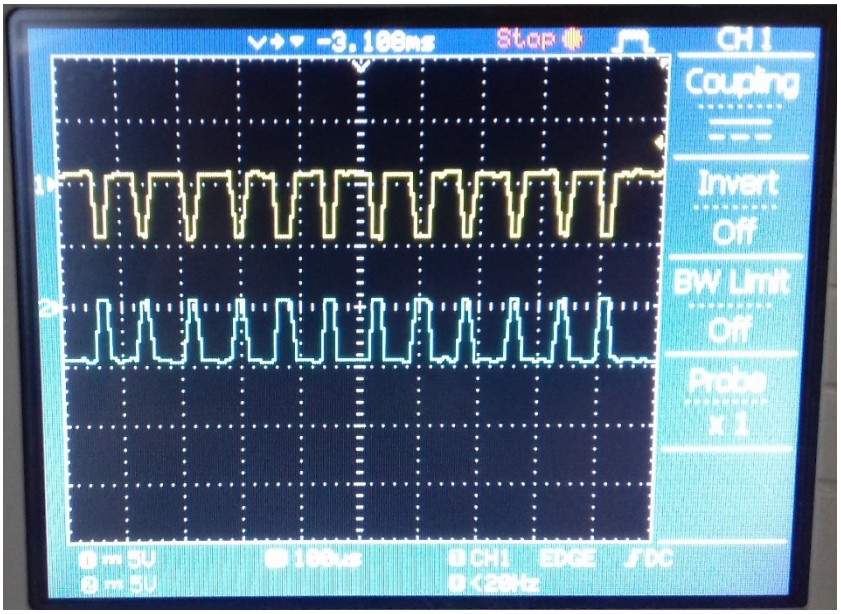


Slika 5.7:Signal iz Arduina za donje MOSFET-e i gornji MOSFET B faze na 4. kanalu



Slika 5.8 Signal iz Arduina za donje MOSFET-e i gornji MOSFET A faze na 4. kanalu

Na slikama 5.6, 5.7 i 5.8 su prikazani signali iz *Arduina* za donje MOSFET-e, i za jedan gornji mosfet na svakoj od slika. Iako se na slikama čini da su gornji i donji MOSFET od iste faze uključeni u isto vrijeme no to je samo posljedica funkcije „rezanja“ struje i vidi se na slici 5.9 što se zapravo događa.



Slika 5.9 Signal iz Arduina za donji MOSFET i gornji MOSFET iste faze

Slika 5.9 prikazuje signal gornjeg i donjeg MOSFET-a iste faze, gornji signal na slici je signal koji upravlja gornjim MOSFET-om a donji signal upravlja donjim MOSFET-om.

6. ZAKLJUČAK

U ovom završnom radu analiziran je način rada BLDC motora i njegove značajke. Potom su analizirana sklopovlja i izabrana je konfiguracija punog mosta koje ima vrlo dobru funkcionalnost, a kontrola nije prekomplicirana. Navedena je tablica stanja aktivnih faza motora po koracima; s povećanjem vremena između koraka se smanjuje brzina rotacije. Analizirana je bez senzorska kontrola pomoću inducirane EMS kojom se implementirala preciznija kontrola BLDC motora.

Izrađen je sklop po shemi koja je rezultat spajanja nekoliko različitih shemi u jednu. Korištena je *Arduino* razvojna pločica, iako ne može direktno upravljati MOSFET-ima jer ima prenizak napon signala, zato se koristi dodatni upravljači za MOSFET-e koji primaju signal iz *Arduina* i povećavaju napon na MOSFET-ima. Za ovakve sklopove najčešće se koriste svih šest N MOSFET-a, ali u ovom sklopu se koriste gornji P MOSFET-i i donji N MOSFET-i kako se ne bi morala koristiti naponska pumpa za pokretanje gornjih N MOSFET-a što bi zakompliciralo sklop.

Testiranje sklopa je rađeno s BLDC motorom i izvorom napajanja na 9 V i 1 A, struja je ograničena relativno nisko za ovakav sklop zbog toga što je spojen na *ProtoBoard* koji se može oštetiti na većim strujama. Mjerenje je vršeno na fazama motora s obzirom na nulu izvora motora i na digitalnim izlazima iz *Arduina*.

7. LITERATURA

[1] A Historicl overview of premament magnet motors - Ohio Electric motors, http://www.ohioelectricmotors.com/2015/07/a-historical-overview-of-permanent-magnet-motors/ (15.6.2019).

[2] Midwest research institute U25 Volker Boulevard Kansas City, MO 6klIO, https://ntrs.nasa.gov/archive/nasa/casi.ntrs.nasa.gov/19750007247.pdf (15.6.2019).

[3] History of brushless dc motors, http://shodhganga.inflibnet.ac.in/bitstream/10603/50968/4/chapter%201.pdf (15.6.2019).

[4] C.L. Xia, Permanent magnet brushless dc motor drives and controls, 2012.

[5] Permanent-magnet DC motors in servo applications, https://www.machinedesign.com/motorsdrives/whats-difference-between-ac-induction-permanent-magnet-and-servomotor-technologies (15.6.2019).

[6] S. Ganguli Analysis of a c-dump converter for switched reluctance motor drive using pspice. https://www.technicaljournalsonline.com/ijeat/VOL%20II/IJAET%20VOL%20II%20ISSUE%20IV%20%20OCTBER%20DECEMBER%202011/ARTICLE%2052%20IJAET%20VOLII%20ISSUE%20IV%20OCT%20DEC%202011.pdf (15.6.2019).

[7] What's The Difference Between Brush DC And Brushless DC Motors, https://www.electronicdesign.com/electromechanical/what-s-difference-between-brush-dc-and-brushless-dc-motors, (15.6.2019).

[8] Arduino 101: Timers and Interrupts, https://www.robotshop.com/community/forum/t/arduino-101-timers-and-interrupts/13072 (15.6.2019).

[9] Sensorless BLDC motor control with Arduino, https://simple-circuit.com/arduino-sensorless-bldc-motor-controller-esc/, (15.6.2019).

[10] GreatScottLab, Make Your Own ESC, https://www.instructables.com/id/Make-Your-Own-ESC/ (15.6.2019).

[11] ElectroNoob, ESC - Electronic speed controller, http://electronoobs.com/eng_circuitos_tut4.php, (15.6.2019).

8. SAŽETAK

U ovom završnom radu je objašnjena teorija o BLDC motoru i uspoređen je s istosmjernim motorom kojeg sve više zamjenjuje. Zatim je obrađena teorija sklopovlja upravljača za BLDC motor; način na koji takav sklop radi s BLDC motorom i teorije upravljanja takvim sklopom. Tada je prikazan jedan primjer izrade takvog sklopa i testiranje istog sa kratkim objašnjenjem rezultata testiranja.

Ključne riječi: upravljački sklop, BLDC motor, BLDC, Arduino

9. ABSTRACT

In this final thesis the theory of BLDC motors is described and it is compared with DC motors which it replaces. Then the theory of hardware for BLDC motor has been studied; the way it works with BLDC motors and theory of controling that circuit. Then one example of such circuit is displayed and tested with short review of the results.

Key words: controller, BLDC motor, BLDC, Arduino

10. ŽIVOTOPIS

Ivan Bačić rođen je 12.1.1997. godine u Đakovu, pohađao Osnovnu školu u Gorjanima i Trgovačku i komercijalnu školu "Davor Milas" u Osijeku. Nakon toga upisuje preddiplomski sveučilišni studij računarstva na Fakultetu elektrotehnike, računarstva i informacijskih tehnologija u Osijeku.

11. PRILOG

Izvorni *Arduino* program.

```
byte step = 0;
bool wait = 0;
bool doonce = 0;
unsigned int current;
const unsigned char PS_128 = (1 << ADPS2) | (1 << ADPS1) | (1 << ADPS0);
uint8_t changedbits;
volatile uint8_t portbhistory = 0xFF;
unsigned int rotation;

void setup() {
  ADCSRA &= ~PS_128;
  ADCSRA |= (1 << ADPS1);
  TCCR1A = 0;
  TCCR1B = 0;
  TCNT1 = 0;
  TCCR1B |= (1 << WGM12) | (1 << CS11) | (1 << CS10);
  TIMSK1 |= (1 << OCIE1A);
  TCCR2A = 0;
  TCCR2B = 0;
  TCNT2 = 0;
  OCR2A = 100;    //50 us
  OCR2B = 20;     //10 us
  TCCR2A |= (1 << WGM21);
  TCCR2B |= (1 << CS21);
  TIMSK2 |= (1 << OCIE2A) | (1 << OCIE2B);
  PCICR |= (1 << PCIE0);    // interupt on pin 10, 11 ,12
  PCMSK0 |= (1 << PCINT0);  // interupt on pin 10, 11 ,12
  sei();
  DDRD = B11110000;    //Pin 4,5,6,7 --> OUTPUT Pin3 --> INPUT
  DDRB = B00100011;       //Pin 8,9 --> OUTPUT, Pin 10,11,12 INPUT
  PORTD = B00000000;
  PORTB = B00011100;
  rotation = map(analogRead(A0), 0, 1023, 5000, 200);
  OCR1A = rotation;
}

ISR(TIMER1_COMPA_vect) {  // timer 1
  rotation = map(analogRead(A0), 0, 1023, 5000, 200);
  OCR1A = rotation;    // sets duration of timer 1 again
  step++;
  doonce = 0;
  if (step == 6) {
    step = 0;
  }
}

ISR(TIMER2_COMPB_vect) { // timer 2 current choping
  if (wait == 0) {
    wait = 1;
    doonce = 0;
  }
}

ISR(TIMER2_COMPA_vect) { // timer 2 reseting main function
  if (wait == 1) {
    wait = 0;
      break;
    case 2:
      if (wait == 0) {
        step2();
      }
      else {
        step21();
      }
      break;
    case 3:
      if (wait == 0) {
        step3();
      }
      else {
        step31();
      }
      break;
    case 4:
      if (wait == 0) {
        step4();
      }
      else {
        step41();
      }
      break;
    case 5:
      if (wait == 0) {
        step5();
      }
      else {
        step51();
      }
      break;
  }
 }
}

void step0() {
  PORTB = B00011101;
  PORTD = B00100000;
  doonce = 1;
}

void step01() {
  PORTB = B00011101;
  PORTD = B00010000;
  doonce = 1;
  TCNT2 = 0;
}

void step1() {
  PORTB = B00011101;
  PORTD = B10000000;
  doonce = 1;
}

void step11() {
  PORTB = B00011101;
  PORTD = B01000000;
  doonce = 1;
```

```
    doonce = 0;
  }
}

ISR (PCINT0_vect) { // interupt
  if (rotation < 1000) { // if rotation less then 1000 check for
chnge in BEMF to change to next step
    current = map(rotation, 1000, 200, 20, 50);   // increases
time for current choping from range 10 us to 25 us
    OCR2B = current;
    changedbits = PINB ^ portbhistory;
    portbhistory = PINB;
    if ((changedbits & (1 << PINB2)) && (step == (2 || 5)))
//Phase C (3)
    {
     step++;
     wait = 0;
     TCNT1 = 0;
     TCNT2 = 0;
     if (step == 6) {
       step = 0;
     }
     doonce = 0;
    }

    if ((changedbits & (1 << PINB3)) && (step == (0 || 3)))
//Phase B (2)
    {
     step++;
     wait = 0;
     TCNT1 = 0;
     TCNT2 = 0;
     doonce = 0;
    }

    if ((changedbits & (1 << PINB4)) && (step == (1 || 4)))
//Phase A (1)
    {
     step++;
     wait = 0;
     TCNT1 = 0;
     TCNT2 = 0;
     doonce = 0;
    }
  }
}

void loop() {
  if (doonce == 0) {
    switch (step) {
      case 0:
        if (wait == 0) {
          step0();
        }
        else {
          step01();
        }
        break;
      case 1:
        if (wait == 0) {
          step1();
        }
        else {
          step11();
        }
```

```
  TCNT2 = 0;
}
void step2() {
  PORTB = B00011100;
  PORTD = B10010000;
  doonce = 1;
}

void step21() {
  PORTB = B00011100;
  PORTD = B01010000;
  doonce = 1;
  TCNT2 = 0;
}

void step3() {
  PORTB = B00011110;
  PORTD = B00010000;
  doonce = 1;
}

void step31() {
  PORTB = B00011101;
  PORTD = B00010000;
  doonce = 1;
  TCNT2 = 0;
}

void step4() {
  PORTB = B00011110;
  PORTD = B01000000;
  doonce = 1;
}

void step41() {
  PORTB = B00011101;
  PORTD = B01000000;
  doonce = 1;
  TCNT2 = 0;
}

void step5() {
  PORTB = B00011100;
  PORTD = B01100000;
  doonce = 1;
}

void step51() {
  PORTB = B00011100;
  PORTD = B01010000;
  doonce = 1;
  TCNT2 = 0;
}
```