Upravljačka jedinica DC motora

Skender, Tomislav

Undergraduate thesis / Završni rad

2019

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:235:548424

Rights / Prava: In copyright / Zaštićeno autorskim pravom.

Download date / Datum preuzimanja: **2024-04-25**

Repository / Repozitorij:

Repository of Faculty of Mechanical Engineering and Naval Architecture University of Zagreb

SVEUČILIŠTE U ZAGREBU
FAKULTET STROJARSTVA I BRODOGRADNJE

ZAVRŠNI RAD

Tomislav Skender

Zagreb, 2019.

SVEUČILIŠTE U ZAGREBU
FAKULTET STROJARSTVA I BRODOGRADNJE

ZAVRŠNI RAD

Mentor:

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

Student:

Tomislav Skender

Zagreb, 2019.

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

Zahvaljujem se prof. dr. sc. Mladenu Crnekoviću na korisnim savjetima, pomoći i znanju koje mi je prenio. Također, zahvaljujem se svojoj obitelji, djevojci i prijateljima na podršci i razumijevanju. Na kraju, zahvaljujem se dragom Bogu na svemu što mi je darovao.

Tomislav Skender

SVEUČILIŠTE U ZAGREBU
FAKULTET STROJARSTVA I BRODOGRADNJE
Središnje povjerenstvo za završne i diplomske ispite
Povjerenstvo za završne ispite studija strojarstva za smjerove:
proizvodno inženjerstvo, računalno inženjerstvo, industrijsko inženjerstvo i menadžment, inženjerstvo materijala i mehatronika i robotika

Sveučilište u Zagrebu Fakultet strojarstva i brodogradnje	
Datum	Prilog
Klasa:	
Ur.broj:	

ZAVRŠNI ZADATAK

Student: **TOMISLAV SKENDER** Mat. br.: 0035200005

Naslov rada na hrvatskom jeziku: **UPRAVLJAČKA JEDINICA DC MOTORA**

Naslov rada na engleskom jeziku: **DC MOTOR CONTROLLER**

Opis zadatka:

Zbog niske cijene istosmjerni motori s četkicama kao komutatorom i dalje imaju svoju primjenu u procesima koji nisu ekstremno zahtjevni. Ključni element u karakteristici toga motora je njegova upravljačka jedinica koja ima funkciju usmjeravanja energije iz vanjskog izvora. Kako se današnji kompleksni sustavi (npr. roboti) sastoje od više motora koje bi bilo teško kontrolirati s jednog centralnog mjesta, koristi se decentralizirana kontrola motora. U tu svrhu motori se spajaju u lokalnu mrežu preko koje dobivaju naredbe, a odrađuju ih lokalni kontroleri motora.
U radu je potrebno:

- za struje do 5A odabrati energetski usmjerivač struje motora,
- odabrati upravljački kontroler, komunikaciju i senzore,
- za odabrane elemente projektirati tiskanu pločicu,
- definirati protokol komunikacije.

Potrebno je navesti korištenu literaturu i ostale izvore informacija, te eventualno dobivenu pomoć.

Zadatak zadan:
29. studenog 2018.

Rok predaje rada:
1. rok: 22. veljače 2019.
2. rok (izvanredni): 28. lipnja 2019.
3. rok: 20. rujna 2019.

Predviđeni datumi obrane:
1. rok: 25.2. - 1.3. 2019.
2. rok (izvanredni): 2.7. 2019.
3. rok: 23.9. - 27.9. 2019.

Zadatak zadao:

Prof.dr.sc. Mladen Crneković

Predsjednik Povjerenstva:

Prof. dr. sc. Branko Bauer

SADRŽAJ

SADRŽAJ I
POPIS SLIKA II
POPIS TABLICA IV
POPIS OZNAKA V
SAŽETAK VI
SUMMARY VII
1. UVOD 1
1.1. Općenito o upravljačkim jedinicama DC motora 2
1.2. Cilj 4
2. DIJELOVI UPRAVLJAČKE JEDINICE 5
2.1. Energetski usmjerivač struje motora 5
2.2. Mikrokontroler 8
2.3. komunikacija upravljačke jedinice istosmjernog motora 10
2.4. Stabilizator napona 13
2.5. Senzor brzine vrtnje i pozicije – enkoder 14
2.6. Senzor za inicijalizaciju početnog položaja 15
2.7. Senzor struje 16
2.8. Istosmjerni elektromotor 18
3. PROJEKTIRANJE UPRAVLJAČE JEDINICE ISTOSMJERNOG MOTORA 19
3.1. Općenito o projektiranju u *Altium Designer*-u 19
3.2. Izrada shematskog dokumenta upravljačke jedinice istosmjernog motora 20
3.3. Tiskana pločica upravljače jedinice istosmjernog motora 26
4. PROGRAMIRANJE UPRAVLJAČKE JEDINICE ISTOSMJERNOG MOTORA 30
4.1. Mjerenje kuta zakreta i brzine vrtnje istosmjernog motora 30
4.2. Regulacija kuta zakreta istosmjernog motora 31
4.3. Programiranje mikrokontrolera upravljačke jedinice istosmjernog motora 32
4.4. Rezultati odziva kuta zakreta na step pobudu 36
5. KONSTRUKCIJA MAKETE 38
6. ZAKLJUČAK 39
LITERATURA 40
PRILOZI 41

POPIS SLIKA

Slika 1.	Industrijski robot marke Fanuc	1
Slika 2.	*TReX* kontroler DC motora marke *Pololulu*	2
Slika 3.	*MCP233* kontroler DC motora marke *Pololulu*	3
Slika 4.	*DCS303* kontroler proizvođaća *Ocean Controls*	3
Slika 5.	Pojednostavljeni koncept upravljanja	4
Slika 6.	Prikaz napona i struje istosmjernog motora u PWM načinu rada	6
Slika 7.	Shema tranzistorskog frekvencijskog pretvarača [1]	7
Slika 8.	*LMD18200* shema (lijevo), izvedba (desno)	8
Slika 9.	Atmega2560 mikroprocesor (lijevo) i Arduino MEGA platforma (desno)	9
Slika 10.	Arhitektura Atmega2560 mikrokontrolera [5]	10
Slika 11.	UART serijska komunikacija	11
Slika 12.	Funkcionalni dijagram FT232RL čipa	12
Slika 13.	FT232RL čip	12
Slika 14.	LM2940 – 5 stabilizator napona, shema kućišta (lijevo) i kućište (desno)	13
Slika 15.	Shema stabilizatora napona *LM2940 – 5*	13
Slika 16.	Princip rada optičkog inkrementalnog enkodera	14
Slika 17.	Princip rada magnetskog inkrementalnog enkodera	14
Slika 18.	Prikaz signala s inkrementalnog magnetskog enkodera	15
Slika 19.	*TCST2103* optički senzor za inicijalizaciju početnog položaja	16
Slika 20.	*ACS7145-05B-T* senzor struje, shema (lijevo) i čip (desno)	17
Slika 21.	Funkcionalni prikaz *ACS714-05B-T* čipa	17
Slika 22.	IG-32PGM motor s enkoderom	18
Slika 23.	Shema spajanja kristala i mikrokontrolera	20
Slika 24.	Shema spajanja stabilizatora napona *LM2940-5*	20
Slika 25.	Prikaz spajanja napajanja i uzemljenja mikrokontrolera	21
Slika 26.	Shema spajanja tipke za resetiranje mikrokontrolera	21
Slika 27.	Shema spajanja ISP konektora	22
Slika 28.	Shema spajanja *FT232RL* čipa	23
Slika 29.	Shema spajanja konektora za komunikaciju i mikrokontrolera	23
Slika 30.	Shema spajanja *LMD18200* čipa i mikrokontrolera	24
Slika 31.	Shema spajanja senzora struje *ACS714ELCTR-05B* i mikrokontrolera	24
Slika 32.	Shema spajanja enkodera i mikrokontrolera	25
Slika 33.	Gornji sloj s vodovima ili *Top Layer*	26
Slika 34.	Donji sloj s vodovima ili *Bottom Layer*	27
Slika 35.	Gornja maska za lemljenje ili *Top Solder*	27
Slika 36.	Gornji sloj s oznakama ili *Top Overlay*	28
Slika 37.	3D model upravljačke jedinice istosmjernog motora	28
Slika 38.	Prednja strana PCB-a upravljačke jedinice istosmjernog motora	29
Slika 39.	Stražnja strana PCB-a upravljačke jedinice istosmjernog motora	29
Slika 40.	Struktura regulacije kuta zakreta istosmjernog motora	31
Slika 41.	*void setup()* dio programa	32
Slika 42.	Funkcija *chA_Interrupt()* koju aktivira *A* kanal enkodera	33
Slika 43.	Funkcija *TCST1203_interrupt()* za inicijalizaciju referente točke	34
Slika 44.	Funkcija unutarnje prekidne rutine	34
Slika 45.	Program zadužen za komunikaciju s korisnikom	35
Slika 46.	Odziv kuta zakreta na step pobudu, K_P = 1,5	36
Slika 47.	Odziv kuta zakreta na step pobudu, K_P = 2,2	37
Slika 48.	Konstrukcija makete	38

Slika 49. Prikaz gotove makete i upravljačke jedinice istosmjernog motora 38

POPIS TABLICA

Tablica 1. Karakteristike IG-32PGM motora 18

POPIS OZNAKA

Oznaka	Jedinica	Opis
d	%	Faktor popunjenosti (eng. Duty cycle)
K_P	/	Proporcionalno pojačanje *P* regulatora pozicije
P	/	Trenutni broj impulsa koje daje enkoder
PPR	/	Broj impulsa koje daje enkoder po okretaju
t_C	s	Vrijeme trajanja jednog ciklusa *PWM* signala
t_U	s	Vrijeme trajanja visokog stanja *PWM* signala
u_b	V	Napon izvora napajanja *H*-mosta
U_{sr}	V	Srednji napon na armaturi motora
α	°	Trenutu kut zakreta
α_REF	°	Referentni ili željeni kut zakreta
ω	rad/s	Kutna brzina vrtnje
ΔT	s	Vrijeme uzorkovanja

SAŽETAK

U ovome je radu projektirana upravljačka jedinica istosmjernog motora s četkicama. Rad se može podijeliti u tri glavna dijela. Prvi dio je odabir potrebnih komponenti na temelju zahtjeva upravljačke jedinice. Tako je u tome dijelu odabran usmjerivač struje motora, odnosno tranzistorski frekvencijski pretvarač, odabran je mikrokontroler *Atmega2560* koji upravlja radom upravljačke jedinice. Nadalje, odabran je senzor struje, pozicije i brzine vrtnje, te svi ostali dijelovi potrebni za rad upravljačke jedinice istosmjernog motora. Drugi dio rada odnosi se na projektiranje tiskane pločice upravljačke jedinice u *Altium Designer* softveru. U tome je dijelu najprije objašnjena električna shema povezivanja komponenti u cjelinu, a nakon toga projektirana je i izrađena tiskana pločica. Treći dio rada, odnosi se softverski dio, odnosno na programiranje mikrokontrolera *Atmega2560* u *Arduino* softveru za programiranje. No, prije samog opisa programiranja, objašnjena je regulacija kuta zakreta istosmjernog motora. Na kraju je prikazana maketa koja se koristi za testiranje upravljačke jedinice.

Ključne riječi: upravljačka jedinica istosmjernog motora, projektiranje tiskane pločice, regulacija pozicije istosmjernog motora, mikrokontroler

SUMMARY

In this paper, a DC (**D**irect **C**urrent) motor controller is designed. The work can be divided into three main parts. The first part is selection of the required components based on the request of the DC motor controller. Thus, in this section, the motor current router, or the transistor frequency converter, is selected. It was also selected Atmega2560 microcontroller which controls the operation of the DC motor controller. Furthermore, it is selected the current sensor, position and angle velocity sensor and all other parts required fot the operation of DC motor controller. The second parts of the paper deals with the design of the DC motor controller PCB (**P**rinted **C**ircuit **B**ord). This section first explains the elctrical scheme of linking the components together and after that PCB was designed and made. Part Three refers to the software part, respectively to the programming of the Atmega2560 microcontroller in Arduino's programming software. However, before the description of the programming microcontroller, the regulation of DC motor is explained. Finally, for the testing purpose, the model is shown.

Key words: DC motor controller, PCB design, position control of DC motor, microcontroller

1. UVOD

U današnje vrijeme, koje se može ubrojiti u eru automatizacije, odnosno u vrijeme u kojem se nastoji umni rad čovjek zamijeniti radom informacijskih strojeva, potreba za računalnim upravljanjem i regulacijom elektromotora je od iznimne važnosti, jer su upravo elektromotorni pogoni najčešće prisutni u modernim mehatroničkim sustavima, kao što su CNC strojevi, razni industrijski roboti, bespilotne letjelice i mnogi drugi proizvodi kojima se danas ljudi koriste. Jedan od takvih mehatroničkih sustava je upravo industrijski robot prikazan na sljedećoj slici.

Slika 1. Industrijski robot marke Fanuc

Kod odabira vrste elektromotornih pogona u mehatroničkim sustavima često su korišteni istosmjerni motori s četkicama kao komutatorom, zbog svoje niske cijene, jednostavnog načina upravljanja brzinom i pozicijom te zbog jeftinijih pretvarača koji se koriste pri upravljanju. No, da bi se dobio mehatronički sustav, potrebna je također upravljačka jedinica koja ima ulogu upravljanja energijom iz vanjskog izvora te na taj način predstavlja spoj mehaničkog (elektromotor) i električnog (električni izvor energije) sustava koji onda ima mogućnost računalnog upravljanja i regulacije.

1.1. Općenito o upravljačkim jedinicama DC motora

Kao što je već gore spomenuto, upravljačke jedinice DC motora, odnosno kontroleri DC motora, imaju ulogu usmjeravanja energije iz vanjskog izvora u mehanički sustav, motor. Naime, kontroleri DC motora također služe kako bi signale male snage pretvorili u signale velike snage koji su potrebni za upravljanje motorom, budući da računalo (mikrokontroler) ne može generirati signale velikih snaga. Zbog velike potrebe za takvim upravljačkim uređajima, danas se na tržištu mogu naći različite varijante kontrolera čije kvaliteta ovisi, ponajprije o cijeni. Tako je na slici 2. prikazan kontroler proizvođača *Pololulu*.

Slika 2. *TReX* kontroler DC motora marke *Pololulu*

TReX kontroler ima mogućnost upravljanja brzinom vrtnje motora s četkicama. Nadalje, može pogoniti dva motora te ima mogućnost mjerenja struje motora. Ima četiri načina na koja može komunicirati s drugim mikrokontrolerom ili računalom, a to su TTL serijska komunikacija, RS-232 komunikacija, RC (Radio control) komunikacija te komunikacija putem analognih signala napona. Maksimalna struja koju može dati je 13 A, a maksimalni napon napajanja motora je 16 V. Nedostatak ovog kontrolera jest u tome što nema mogućnost mjerenja brzine vrtnje i pozicije, što znači da nema mogućnosti za regulaciju pozicije DC motora. Osim ovog kontrolera, isti proizvođač nudi kontrolere boljih karakteristika, koji imaju mogućnost mjerenja pozicije i brzine vrtnje, što znači da mogu regulirati poziciju i brzinu DC motora. Jedan takav kontroler s mogućnošću regulacije pozicije DC motora prikazan je na sljedećoj slici.

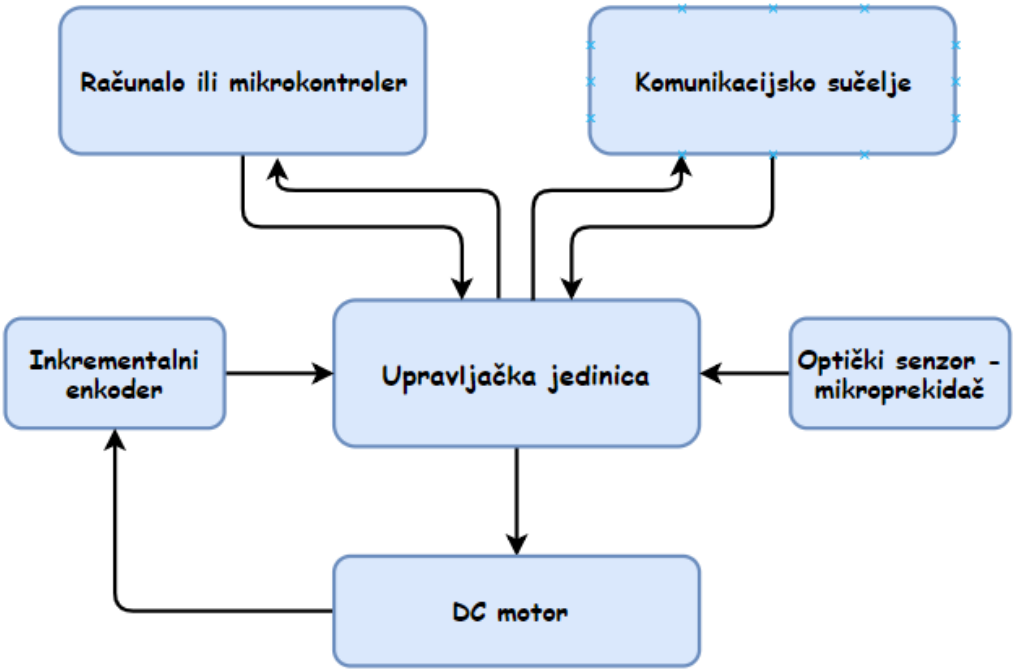
Slika 3. ***MCP233*** **kontroler DC motora marke** ***Pololulu***

MCP233 kontroler uz sve karakteristike kao i prethodno opisani kontroler ima mogućnost USB serijske komunikacije, što znači da može biti programiran direktno preko računala. Također može upravljati akceleracijom, brzinom, pozicijom i strujom DC motora, što je velika prednost u odnosu na prethodni *TReX* kontroler. Na tržištu se mogu također pronaći kontroleri koji imaju implementirane sustave regulacije te uz koje dolaze gotovi softveri s mogućnošću podešavanja parametara regulacije, pri čemu ih nije potrebno programirati. Među takve kontrolere ubraja se *DCS303* kontroler proizvođaća *Ocean Controls* koji je prikazan na slici 4.

Slika 4. ***DCS303*** **kontroler proizvođaća** ***Ocean Controls***

1.2. Cilj

Cilj je ovoga rada napraviti upravljačku jedinicu odnosno kontroler istosmjernog motora s četkicama. Prvi je korak u tome odabir energetskog usmjerivača struje motora, odnosno drajvera. Zatim je potrebno odrediti mikrokontroler koji obrađuje informacije pristigle sa senzora i elektromotora te prema tome šalje informacije u drajver motora i ostale periferne dijelove. Isto tako, potrebno je odabrati senzore koji će komunicirati s mikrokontrolerom i tako mu davati potrebne informacije za uspješno izvršavanje regulacije DC motora. Osim toga, potrebno je također odabrati način komunikacije između mikrokontrolera upravljačke jedinice i računala ili drugog mikrokontrolera koji šalju naredbe upravljačkoj jedinici. Nakon što se odrede sve potrebne komponente, može se preći na drugi dio, a to je projektiranje tiskane pločice na kojoj će biti sve te komponente smisleno povezane i spojene. Zadnji korak je definirati protokol komunikacije, odnosno napraviti program koji će izvršavati komunikaciju s računalom, preko kojega korisnik zadaje naredbe, koje se komunikacijom šalju na mikrokontroler upravljačke jedinice i koji onda izvršava zadane naredbe. Također, u radu je konstruirana i napravljena maketa za testiranje upravljačke jedinice, a shematski prikaz rada upravljačke jedinice DC motora prikazan je na slici 4.



Slika 5. Pojednostavljeni koncept upravljanja

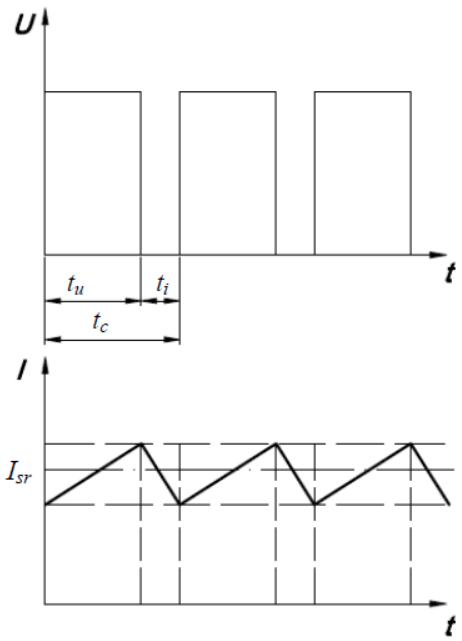
2. DIJELOVI UPRAVLJAČKE JEDINICE

Kao što je u uvodu spomenuto, glavna zadaća upravljačke jedinice istosmjernog motora je upravljanje snagom električnog motora, odnosno naponom i strujom koji pokreću motor. Budući da upravljačka jedinica mora imati mogućnost upravljanja naponom i strujom, potreban je elektronički sklop koji se naziva tranzistorski frekvencijski pretvarač (eng. chopper) odnosno energetski usmjerivač struje motora. Tranzistorski frekvencijski pretvarač napaja motor naponom čiji je iznos određen upravljačkim signalima. Nadalje, da bi se moglo upravljati frekvencijskim pretvaračem potreban je upravljački elektronički sklop ili uređaj koji će slati upravljačke signale na tranzistorski frekvencijski pretvarač. Budući da je potrebno napraviti upravljačku jedinicu koja će biti upravljana preko računala ili mikrokontrolera, kao uređaj za upravljanje signalima tranzistorskog pretvarača koristi se mikrokontroler zbog mogućnosti komunikacije s računalom, programiranja, reprogramiranja te mogućnosti prikupljanja podataka sa senzora i obradu istih kako bi se postiglo precizno upravljanje i regulacija istosmjernog motora. Također, kao što je već spomenuto, potrebni su senzori koji mjere fizikalne veličine (brzinu vrtnje, kut zakreta i struju) i pretvaraju ih u signale pogodne za daljnju obradu, a to su najčešće naponski signali. U nastavku će biti prikazan odabir potrebnih komponenti za projektiranje upravljačke jedinice istosmjernog motora i njihov detaljniji opis i funkcija koju obavljaju.

2.1. Energetski usmjerivač struje motora

Prije nego što se opiše rad energetskog usmjerivača struje, potrebno je objasniti način upravljanja motora s energetskog aspekta. Naime, kod istosmjernog motora s nezavisnom uzbudom brzina vrtnje proporcionalno ovisi o naponu na armaturi motora, što znači da se promjenom napona mijenja brzina vrtnje. Dakle, da bi se moglo upravljati naponom, a time i brzinom vrtnje potreban je elektronički sklop koji ima mogućnost brze i precizne promjene napona. Takav se elektronički sklop naziva tranzistorski frekvencijski pretvarač (eng. chopper) odnosno energetski usmjerivač struje motora. Njegov rad zasnovan je na prekidačkom režimu rada (eng. switching mode) odnosno napajanje armature motora je isprekidano. Ovisno o tome koliko je armatura motora dugo uključena, odnosno isključena, mijenjat će se napon na njoj, a time i brzina vrtnje samog motora.[1] Takvo upravljanje naponom, u konačnici brzinom vrtnje istosmjernog elektromotora, naziva se pulsno širinska modulacija. (eng. Pulse Width Modulation), pri čemu se duljinom trajanja visokog stanja, odnosno niskog, mijenja srednja vrijednost napona. Budući da je rotor istosmjernog motora napravljen od namotaja žice koji se

ponašaju kao zavojnica i otpor u strujnom krugu, to će uzrokovati kašnjenje struje za naponom, kao što je prikazano na sljedećoj slici.



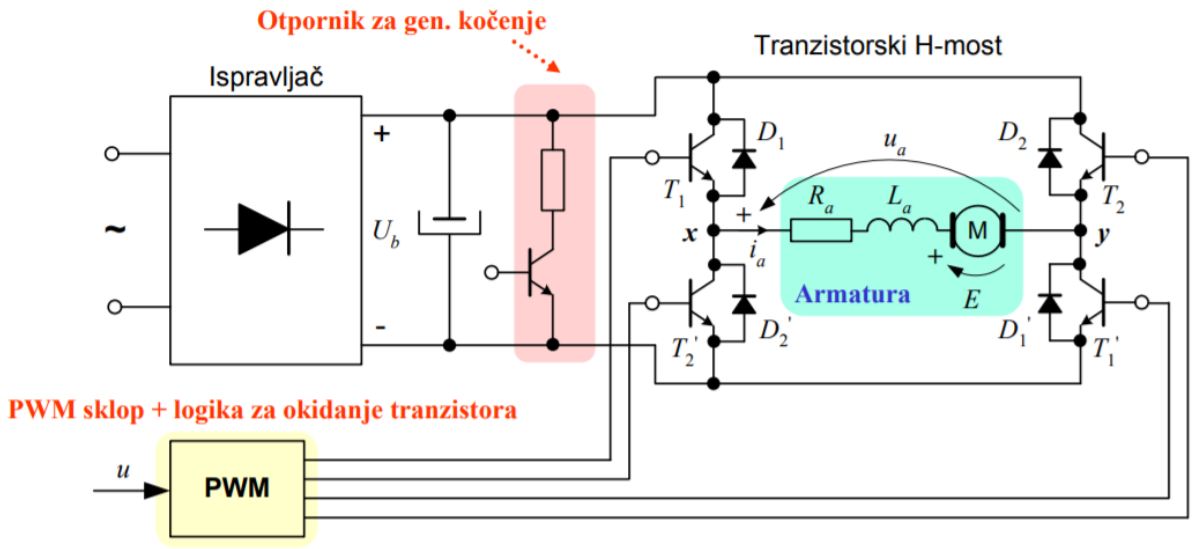
Slika 6. Prikaz napona i struje istosmjernog motora u PWM načinu rada

Isprekidani napon armature uzrokuje oscilacije u struji, što bi značilo da će dolaziti i do oscilacija u momentu motora. Zbog toga je važno da frekvencija uključivanja i isključivanja napona armature bude velika (reda veličine nekoliko kHz) kako bi oscilacije u momentu bile neprimjetne. Srednja vrijednost napona armature računa se prema sljedećem izrazu:

$$U_{sr} = \frac{1}{t_c}\int_0^{t_c} u(t)dt = \frac{u_b \cdot t_u}{t_c} = d \cdot u_b \tag{1}$$

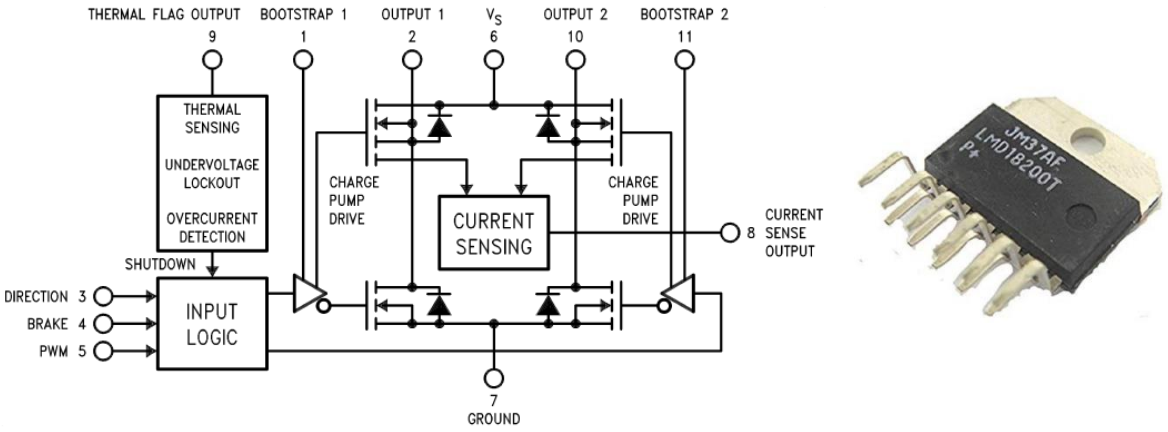
gdje je t_c vrijeme ciklusa sa slike 5, t_u vrijeme uključenja sa slike 5, U_b je napon izvora električne energije koji napaja strujni krug. Omjer vremena uključenja t_u i vremena ciklusa t_c označava se s d, a naziva se faktor popunjenosti (eng. duty cycle), a izražava se u postocima [2]. Nakon što je opisan najčešće primjenjivani način na koji se upravlja naponom armature, potrebno je objasniti princip rada samog tranzistorskog pretvarača. Naime, jezgra pretvarača je tranzistorski *H* – most čija je principijelna shema prikazana na slici 6. Okidanjem tranzistora upravlja generator *PWM* signala, a to je u ovome radu mikrokontroler. Važno je napomenuti da postoje dvije vrste *PWM* modulacije za upravljanje istosmjernim motorom. Prvi način *PWM* modulacije je da se armatura motora isprekidano napaja pozitivnim, odnosno negativnim

naponom strujnog izvora. Stoga iznos faktora popunjenosti od 50 % znači srednji napon jednak nuli, a iznos faktora popunjenosti veći od 50 % rezultira vrtnjom motora u jednom smjeru, dok se za vrijednosti faktora popunjenosti manje od 50 % motor vrti u suprotnom smjeru. Kod drugog načina *PWM* modulacije, koji je korišten u ovome radu, napon armature ide u rasponu od nula do pozitivne vrijednosti napona izvora za jedan smjer vrtnje, odnosno od nula do negativne vrijednosti napona izvora za drugi smjer vrtnje. No, budući da se iznosom faktora popunjenosti ne može utjecati na smjer vrtnje motora, potreban je dodatni signal koji će određivati željeni smjer vrtnje.



Slika 7. Shema tranzistorskog frekvencijskog pretvarača [1]

Danas su na tržištu dostupni integrirani elektronički sklopovi koji u sebi imaju H – most. Upravo jedna takva izvedba, korištena u ovome radu, je *LMD18200*. To je elektronički sklop koji ima integriran *H* – most. Može davati kontinuiranu maksimalnu struju do 3 A te pogoniti motore od 12 V do 55 V. Ima 3 ulazna pina preko kojih se upravlja signalima koje sklop daje na svom izlazu. Također ima izlazni signal koji daje informaciju ako dolazi do prevelikog zagrijavanja sklopa te senzor struje koji daje naponske signale o iznosu struje koja teče kroz armaturu motora. Shema *LMD18200* sklopa te izgled samog sklopa prikazani su na slici 7.



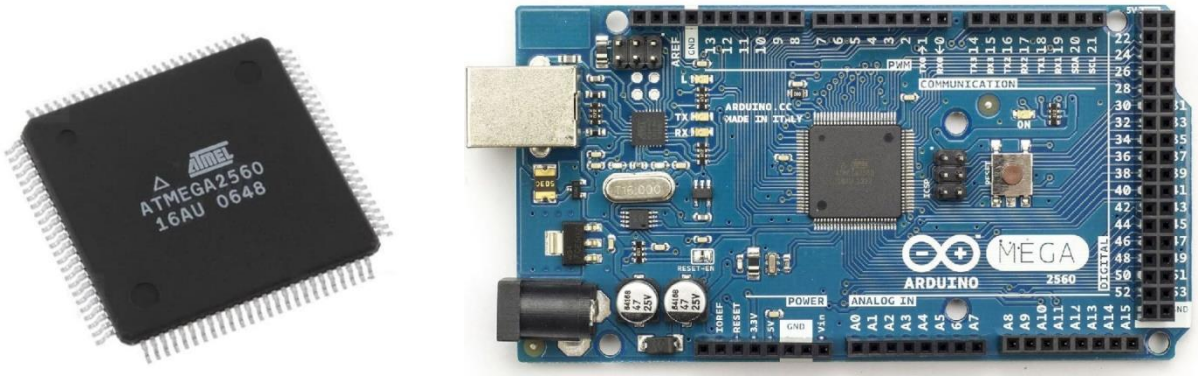
Slika 8. ***LMD18200*** **shema (lijevo), izvedba (desno)**

2.2. Mikrokontroler

Mikrokontroler je digitalni automat sposoban za izvršenje računskih operacija, a na osnovu programa pohranjenog u memoriji. Izveden je u obliku jednog integriranog kruga [4]. Mikrokontroler sekvencijalno izvršava instrukcije koje u konačnici rezultiraju izvršavanjem nekog zadatka. Sastoji se od mikroprocesora koji je vođen vremenskim taktom i izvršava aritmetičke i logičke operacije, *RAM* (eng. **R**andom **A**ccess **M**emory) memorije koja služi za pohranjivanje rezultata operacija koje izvodi mikroprocesor, *ROM* (eng. **R**ead **O**nly **M**emory) memorije u kojoj su pohranjene instrukcije za mikroprocesor, ulazno – izlaznih registara koji služe za komunikaciju s vanjskim uređajima te podatkovne sabirnice putem koje pojedini moduli komuniciraju i razmjenjuju podatke [2].

Kao što je spomenuto u prethodnom poglavlju, *LMD18200* elektronički sklop ima 3 ulazna upravljačka pina preko kojih se definiraju signali na njegovom izlazu. Za zadavanje ulaznih upravljačkih signala na *LMD18200* sklopu, koristi se mikrokontroler. No, to nije jedina njegova zadaća koju obavlja u upravljačkoj jedinici istosmjernog motora. Naime, njegova je uloga također, primati i obrađivati signale sa senzora koji se koriste u ovome radu i koji će biti u sljedećim poglavljima detaljno opisani. Također mikrokontroler je zadužen za komunikaciju između računala ili drugog mikrokontrolera, ovisno o primjeni upravljačke jedinice. Zbog tih razloga potrebno je odabrati mikrokontroler koji ima dovoljno *FLASH* memorije u koju se sprema program koji se treba izvoditi te rezultati matematičkih operacija izvršenih od strane aritmetičke logičke jedinice (eng. **A**ritmetic **L**ogic **U**nit). Nadalje, osim toga mikrokontroler treba imati dovoljan broj brojača. Naime, potreban je jedan 16 – bitni brojač koji će ostvarivati

prekidnu rutinu odnosno program za koji je važno da se izvršava nakon točno određenog vremenskog perioda. Zatim, potreban je također 16 – bitni brojač koji će generirati *PWM* signale. Također potreban je 8 – bitni brojač koji će brojati impulse pristigle sa enkodera koji se koristi kao senzor brzine vrtnje i kuta zakreta motora. Osim toga, potreban je jedan digitalni ulaz s prekidnom rutinom koji se koristi za inicijalizaciju početnog položaja. Kako bi se uspostavila komunikacija s računalom ili drugim mikrokontrolerima potrebno je da mikrokontroler upravljačke jedinice istosmjernog motora ima mogućnost serijske komunikacije – *UART*. Isto tako, kako bi se mogli očitati analogni naponski signali sa senzora struje, zahtjeva se da da mikrokontroler ima analogno – digitalni pretvornik minimalne rezolucije 10 bita. Uzimajući u obzir sve te zahtjeve odabran je mikrokontroler proizvođača *Atmel* pod nazivom *Atmega2560*, koji je danas vrlo rasprostranjen i najčešće korišten na *Arduino* platformi, poznat kao *Arduino MEGA*.



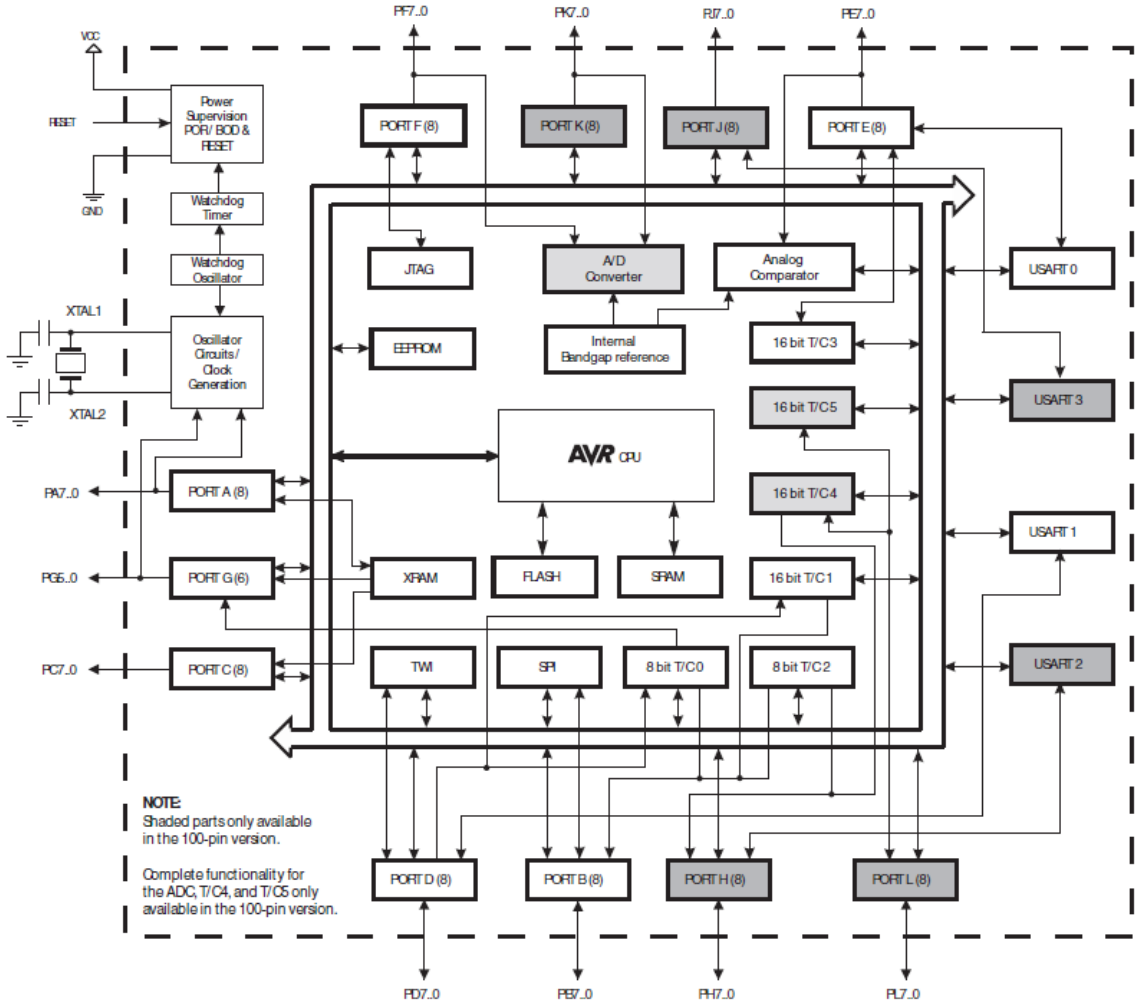
Slika 9. Atmega2560 mikroprocesor (lijevo) i Arduino MEGA platforma (desno)

Atmega2560 mikrokontroler je 8 – bitni mikrokontroler temeljen na poboljšanoj AVR – ovoj *RISC* arhitekturi. Budući da radi na frekvenciji 16 MHz, izuzetno je brz i pouzdan, a neke od njegovih najvažnijih karakteristika su [5]:

- 256 kB *FLASH* memorije
- 8 kB *RAM* memorije
- 4 kB *EEPROM* memorije
- 183 *SFR* registara
- 4 *UART* komunikacije
- 86 ulazno/izlaznih pinova
- 4 16 bitna te 2 8 – bitna brojača

- 16 kanalni 10 -bitni analogno – digitalni pretvornik
- 12 *PWM* izlaza

Zbog ovih odličnih karakteristika, *Atmega2560* mikrokontroler zadovoljava sve tražene zahtjeve za pouzdan i siguran rad upravljačke jedinice istosmjernog motora. Na slici 9. prikazana je arhitektura mikrokontrolera.

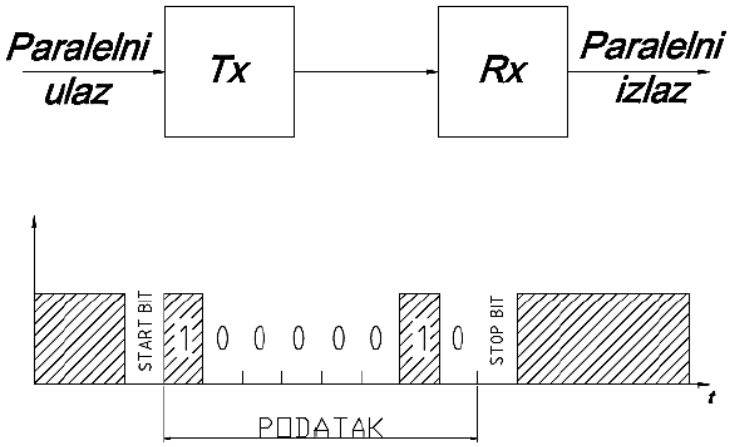


Slika 10. Arhitektura Atmega2560 mikrokontrolera [5]

2.3. komunikacija upravljačke jedinice istosmjernog motora

Nadalje, nakon što je odabran mikrokontroler, potrebno je odabrati komunikaciju koja se koristi za razmjenu informacija između računala ili nekog drugog mikrokontrolera. Stoga će u ovom poglavlju biti navedene neke od najpoznatijih vrsta komunikacije te će naglasak biti na komunikaciji koja se koristi u ovome radu.

Danas je jedan od najčešće upotrebljavanih vrsta komunikacije serijska komunikacija. Tako razlikujemo različite izvedbe serijske komunikacije kao što su *UART* serijska komunikacija, *TTL*, *RS232* , *RS485* (najčešće korišten oblik komunikacije u industrijskom okruženju) te *RS422* koji se od *RS485* razlikuje samo u tome što je moguć istovremeni prijem i predaja podatka. U ovome radu kao komunikacija između mikrokontrolera upravljačke jedinicie istosmjernog motora i računala, odnosno drugog mikrokontrolera koristi se *UART* serijska komunikacija. *UART* (eng **U**niversal **A**synchronous **R**eciver/**T**ransmitter) je osnova svake serijske komunikacije bilo na razini računala ili mikrokontrolera i koristi se za pretvorbu paralenog podatka (najčešće 8 – bitnog podatka) u serijski oblik s ciljem smanjenja potrebnih linija za komunikaciju. U predajniku se paralelni podatak pretvara u serijski niz bitova dok ga prijemnik ponovno vraća u paralelni oblik. Za komunikaciju se koriste dva pina, *Tx* – predajni i *Rx* – prijemni. Na razini *UART* – a komunikacija je dvosmjerna (eng. full duplex) što znači da je moguć istovremeni prijem i predaja podataka. Na sljedećoj slici prikazan je komunikacijski protokol *UART* serijske komunikacije. [6]

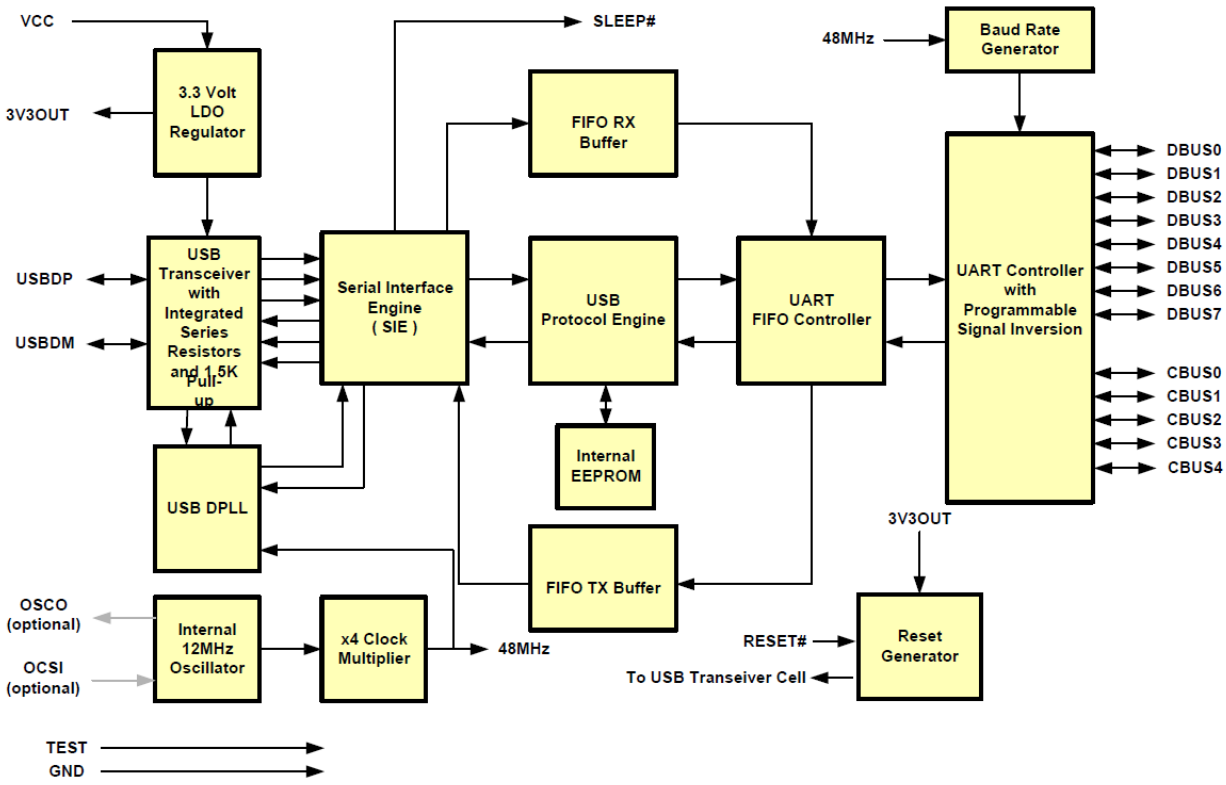


Slika 11. UART serijska komunikacija

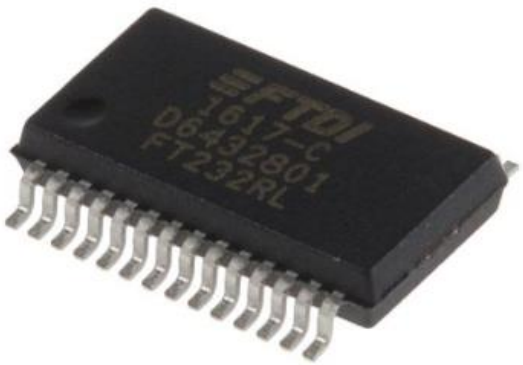
Budući da je *UART* komunikacija asinkrona, da bi prijemnik mogao ispravno dekodirati poslani niz bitova predajnika potrebno je uz 8 -bitni zapis podatka koji se šalje, poslati i dva bita jedan za početak komunikacije a jedan za kraj komunikacije, kako bi prijemnik mogao znati kada treba početi komunikacija, a kada ona završava. Tako se kod *UART* serijske komunikacije kao *START* bit uvijek koristi digitalna nula, nakon kojega počinje slanje 8 – bitnog podatka na prijemnik, počevši od najniže vrijednosti (eng. **L**east **S**ignificant **B**it). Slanje podatka završava *STOP* bitom koji je uvijek digitalna jedinica te koji vraća komunikaciju u neaktivno stanje.

Trajanje svakog bita definirano je brzinom komunikacije u broju bitova u sekundi (eng. Baud) i mora biti jednako postavljeno na prijemniku i predajniku.

Budući da je potrebno uspostaviti komunikaciju između *USB* porta računala i mikrokontrolera upravljačke jedinice istosmjernog motora, potreban je uređaj koji će signale serijske *UART* komunikacije pretvarati u *USB* signale i obrnuto. Zbog toga se koristi *FT232R* elektronički uređaj odnosno čip koji je vrlo jednostavan za korištenje, jer sve što je potrebno za pretvorbu signala jest integrirano unutar samog čipa te ne treba nikakva dodatna programiranja unutar memorije čipa.



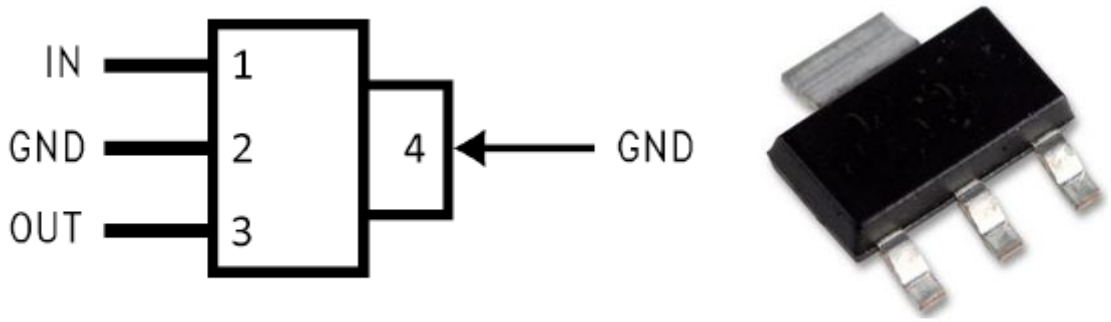
Slika 12. Funkcionalni dijagram FT232RL čipa



Slika 13. FT232RL čip

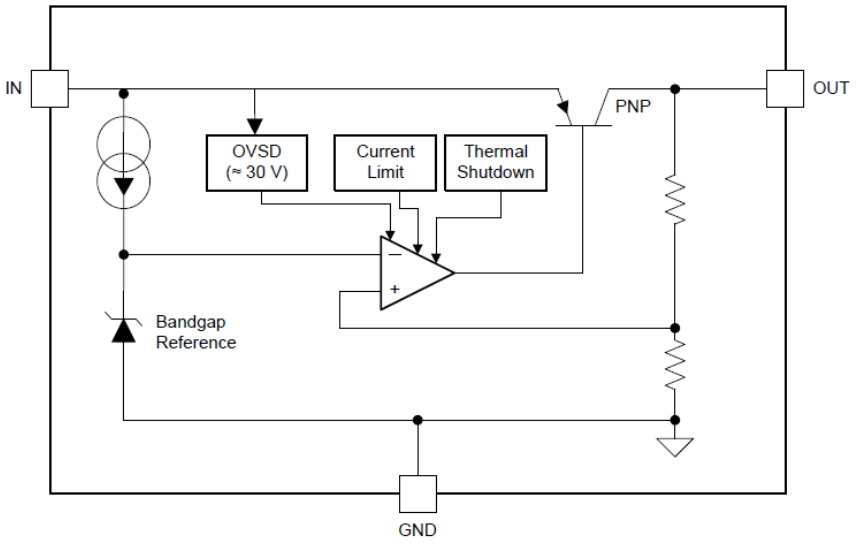
2.4. Stabilizator napona

Budući da mikrokontroler za pouzdan i točan rad zahtjeva stabilan izvor napajanja od 5 V potrebno je osigurati da izvor iz kojega se napaja mikrokontorler bude konstantnog iznosa. Zbog toga je potreban stabilizator napona koji će davati stalan iznos napona od 5 V. Takav elektornički sklop, koji je korišten u ovome radu jest stabilizator napona *LM2940 – 5*. Dostupan je u različitim varijantama kućišta, a u ovome radu je odabrano *SMD* (eng. **S**urface **M**ount **D**esign) kućište *SOT – 223* koje je prikazano na sljedećoj slici (desno).



Slika 14. LM2940 – 5 stabilizator napona, shema kućišta (lijevo) i kućište (desno)

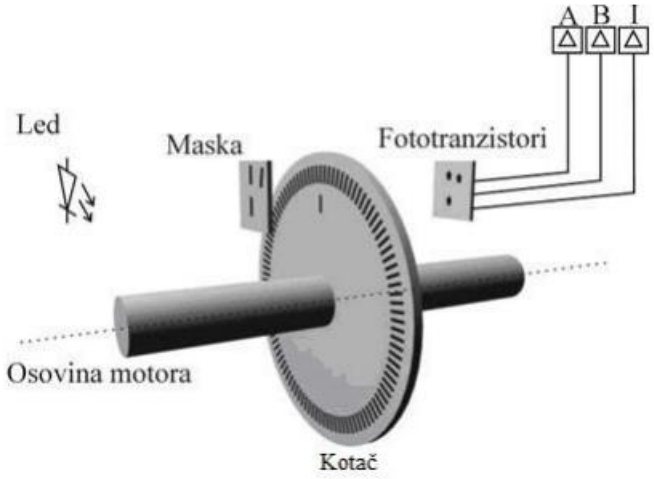
Kao što se vidi na slici 13., *LM2940 -5* ima 4 pina, gdje se pin 1 (*IN*) spaja na izvor napajanja koji može biti za ovaj čip u rasponu od 6,25 V do 26 V, pinovi 2 i 4 (*GND*) spajanju se na uzemljenje odnosno na iznos napona od 0 V, a pin 3 (*OUT*) je izlazni pin na kojemu se dobiva stabilan iznos napajanja od 5 V. Princip rada stabilizatora *LM2940 – 5* prikazan je na sljedećoj slici.



Slika 15. Shema stabilizatora napona *LM2940 – 5*

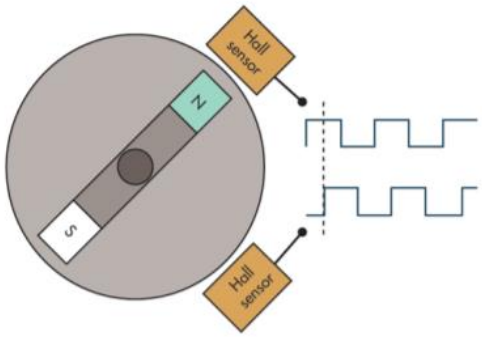
2.5. Senzor brzine vrtnje i pozicije – enkoder

Kako je za regulaciju pozicije i brzine vrtnje motora potrebno poznavanje informacije o tim veličinama i njihovim promjenama u svakom trenutku, potreban je mjerni uređaj koji će mjeriti te veličine i slati informacije o njihovom iznosu na mikrokontroler. Takav mjerni uređaj odnosno senzor koji se koristi u ovome radu naziva se inkrementalni enkoder. Inkrementalni enkoderi se montiraju na osovinu motora te ovisno o zakretu osovine, daju na svojem izlazu niz impulsa. Broj impulsa koje enkoder generira za jedan puni zakret osovine motora, naziva se rezolucija enkodera. Inkrementalni enkoderi mogu biti izvedeni kao optički ili magnetski. Optički enkoderi u pravilu imaju veću rezoluciju, a time i točnost. Naime, oni generiraju impulse na temelju presijecanja svjetlosnog snopa između izvora svijetlosti i fotoprijemnika.



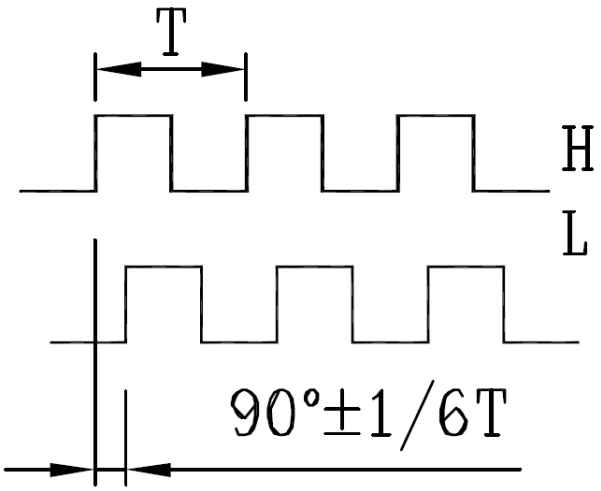
Slika 16. Princip rada optičkog inkrementalnog enkodera

Princip rada magnetskih enkodera temelji se na promjeni magnetskog polja koje uzrokuje okidanje Hall – ovog senzora kako je to prikazano na sljedećoj slici.



Slika 17. Princip rada magnetskog inkrementalnog enkodera

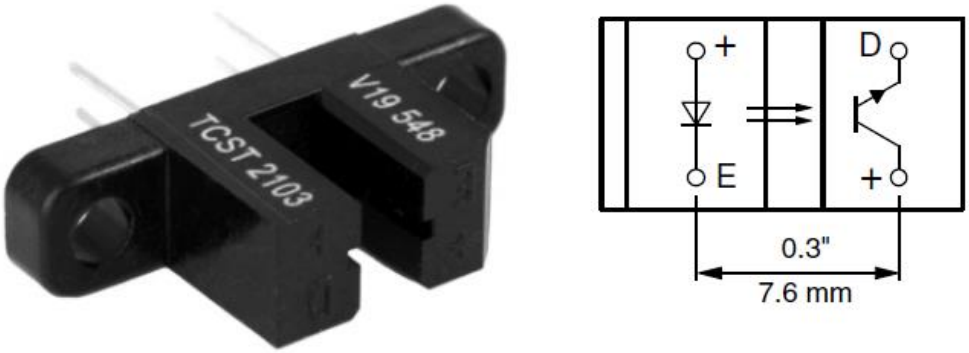
Kao što se može vidjeti na slici 16. inkrementalni enkoder daje dva signala razmaknuta za određeni kut. Na temelju faznog pomaka između tih dvaju signala može se odrediti smjer vrtnje motora, što je i glavni razlog zbog kojega je većina enkodera dvokanalana. Neki enkoderi imaju uz dva osnovna kanala i dodatni kanal koji daje jedan impuls po okretaju i koji se koristi za inicijalizaciju položaja. U ovome radu korišten je magnetski inkrementalni enkoder koji je ugrađen na motor na kojemu će se provesti testiranje upravljačke jedinice, a prikaz enkodera i elektromotora dan je na slici 21. Korišteni inkrementalni enkoder ima rezoluciju 14 impulsa po okretaju, odnosno 7 impulsa po svakom kanalu što je dosta niska rezolucija. No, zbog relativno velikog prijenosnog omjera elektromotora koji je naveden u tablici 1. na strani 18, rezolucija enkodera raste. Kako u ovome radu nije toliko naglasak na preciznosti i točnosti same regulacije pozicije elektromotora, ukupna rezolucija enkodera biti će dovoljna za ispravno testiranje upravljačke jedinice istosmjernog motora. Za precizno pozicioniranje elektromotora bilo bi potrebno koristiti enkodere velikih rezolucija.



Slika 18. Prikaz signala s inkrementalnog magnetskog enkodera

2.6. Senzor za inicijalizaciju početnog položaja

Budući da se pozicija odnosno kut zakreta elektromotora mjeri prema broju impulsa pristiglih s enkodera, da bi se dobila točna informacija o kutu zakreta potrebna je neka referentna točka od koje se mjere impulsi odnosno kut zakreta. No, kako tu referentnu točku nije moguće odrediti na temelju dva signala s enkodera, potreban je dodatni signal koji će davati informaciju o referentnoj točki od koje se mjeri kut. Takav će signal generirati optički senzor *TCST2103* čiji je shematski prikaz te izgled samog senzora prikazan na slici 18.

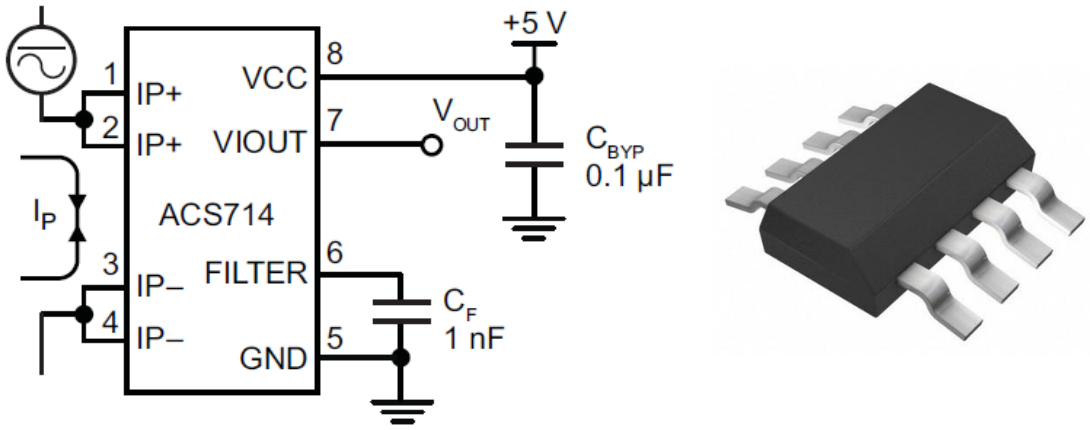


Slika 19. *TCST2103* optički senzor za inicijalizaciju početnog položaja

TCST2103 optički senzor kao što se može vidjeti na slici 18. u sebi ima integriranu *LED* diodu koja emitira infracrvenu svijetlost prema fototranzistoru koji je propusno polariziran sve dok prima infracrvene zrake od emitera. Kada se fototranzistoru prekine infracrvena svijetlost, on prestaje voditi odnosno napon između na pinu *D* prelazi u nisko stanje. Na taj način se može registrirati kada je neki objekt između optičkih vrata te se tu toj točki uzme referentna točka od koje se počinju brojati impulsi. Konkretna primjena *TCST2103* optičkog senzora biti će prikazana na maketi koja se koristi za testiranje upravljačke jedinice istosmjernog motora.

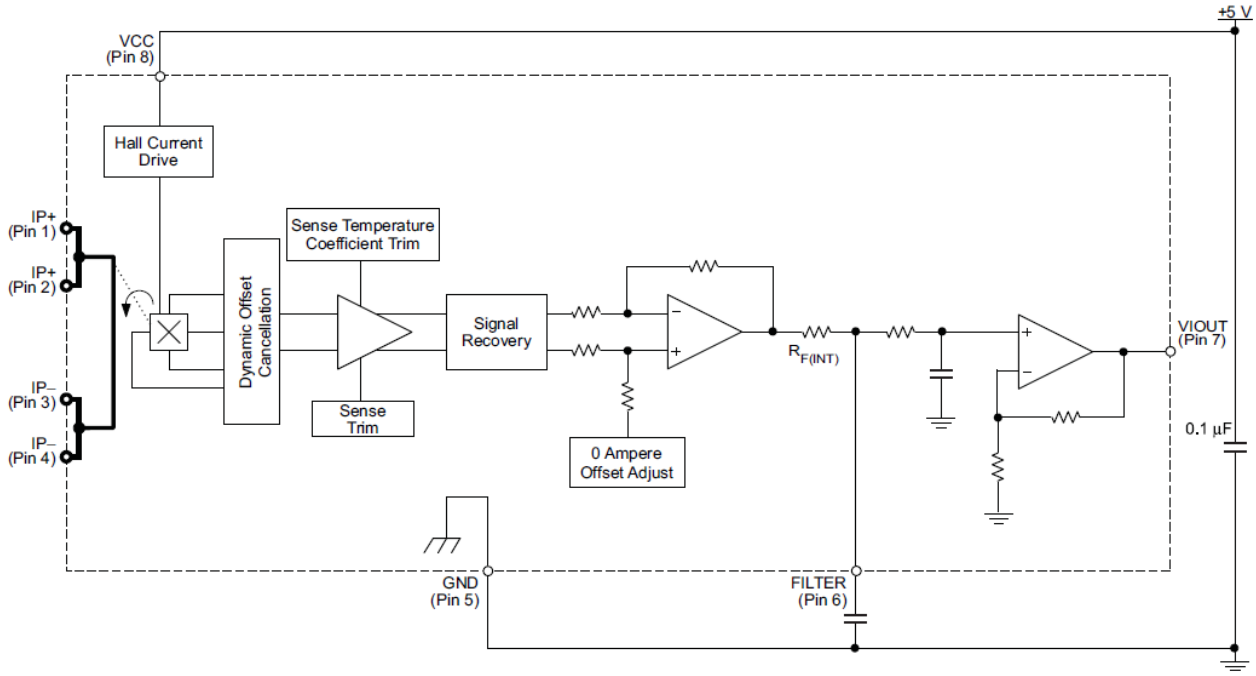
2.7. Senzor struje

Kako maksimalna dopuštena struja *LMD18200* sklopa iznosi 3 A te veće struje od toga mogu uništiti sklop, potrebno je mjeriti struju koja teče kroz armaturu motora te ukoliko struja preraste maksimalni dopušteni iznos isključiti napajanje motora kako ne bi došlo do oštećenja *LMD18200* sklopa. Za mjerenje struje kroz armaturu motora koristi se elektronički ueđaj *ACS714ELCTR -05B-T* čiji je proizvođač *Allegro*. *ACS7145-05B-T* mjeri struju na temelju Hall-ovog efekta, odnosno struja koja protječe kroz vodič stvara magnetsko polje koje je proporcionalno iznosu struje, promjena magnetskog polja također uzrokuje proporcionalnu promjenu napona. Stoga, može se zaključiti da je struja koja teče kroz vodič proporcionalna naponu na izlazu. *ACS7145-05B-T* ima osjetljivost 185 mV/A te za struju od 0 A na izlazu daje napon od 2,5 V a za struju od 1 A na izlazu daje 2,685 V. *ACS7145-05B-T* čip dolazi u *SOIC8* kućištu te potreban napon napajanja za ispravan rad senzora je 5 V. Nadalje ima propusnost od 80kHz te izuzetno stabilan iznos napona na izlazu iz samog čipa.



Slika 20. ***ACS7145-05B-T* senzor struje, shema (lijevo) i čip (desno)**

Kao što se može vidjeti na slici 19. čip ima pin (6) preko kojega se može podešavati propusnost, promjenom iznosa kapaciteta na kondenzatoru C_F, preporučena vrijednost kapaciteta od strane proizvođača je 1 nF te je ta vrijednost i uzeta u ovome radu. Nadalje, funkcionalni prikaz čipa *ACS7145-05B-T* dan je na slici 20.



Slika 21. **Funkcionalni prikaz *ACS714-05B-T* čipa**

2.8. Istosmjerni elektromotor

Kako bi se upravljačka jedinica istosmjernog motora odnosno kontroler testirao, koristi se istosmjerni ili *DC* (eng. **D**irect **C**urrent) motor *IG – 32PGM*. Karakteristike motora prikazane su u tablici 1.

Tablica 1. Karakteristike IG-32PGM motora

Nazivni napon U_N	12 V
Nazivna struja I_N	530 mA
Nazivni moment M_N	$0{,}74 \cdot 10^{-2}$ Nm
Nazivna snaga motora P_N	4,22 W
Nazivna brzina vrtnje n_N	5290 min^{-1}
Prijenosni omjer reduktora i	27
Nazivna brzina na izlazu iz reduktora n_R	195 min^{-1}
Nazivni moment na izlazu iz reduktora T_R	0,207 Nm

Kako je već spomenuto u prethodnim poglavljima, *IG – 32PGM* motor ima na sebi već montiran inkrementalni enkoder, što je prikazano na sljedećoj slici.

Slika 22. IG-32PGM motor s enkoderom

3. PROJEKTIRANJE UPRAVLJAČE JEDINICE ISTOSMJERNOG MOTORA

Projektiranje tiskane pločice upravljače jedinice istosmjernog motora radi se u softveru koji se naziva *Altium Designer*. *Altium Designer* je softver australske kompanije *Altium Limited* te je danas jedan od najpoznatijih softvera za projektiranje tiskanih pločica. Postupak projektiranja u *Altium Designeru* biti će opisan u nastavku ovoga poglavlja te će također biti opisani načini spajanja pojedinih elektroničkih komponenata i sve korištene komponente za izradu tiskane pločice upravljačke jedinice istosmjernog motora.

3.1. Općenito o projektiranju u *Altium Designer*-u

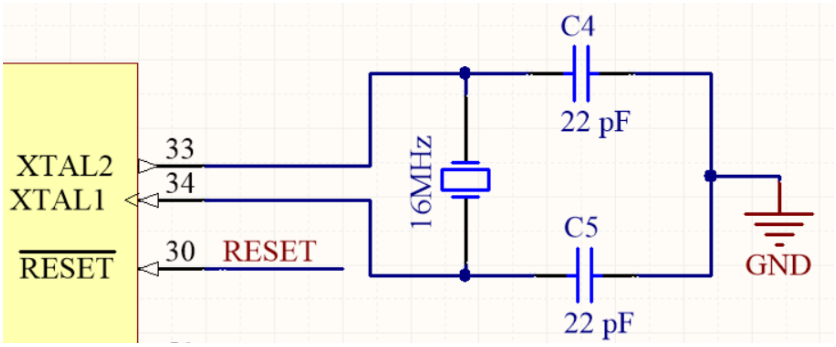
Projektiranje tiskanih pločica u *Altium Designer*-u, ali i u ostalim softverima za projektiranje tiskanih pločica, sastoji se od dva glavna dijela, a to su izrada shematskog dokumenta *SchDoc* i izrada *PCB* (eng. **P**rinted **C**ircuit **B**oard) dokumenta *PcbDoc*. Uz te dokumente potrebno je napraviti i *PrjDoc* odnosno projektni dokument unutar kojega se pohranjuju *SchDoc* i *PcbDoc* dokumenti te svi ostali dokumenti korišteni za izradu projekta kao sto su dokumenti biblioteka (eng. Library) odnosno *PcbLib* i *SchLib* dokumenti. Na kraju kada je projekt izrađen, odnosno kada je napravljen željeni *PCB*, da bi se mogao poslati u proizvodnju potrebno je napraviti gerber dokumente (eng. Gerber files) prema kojima onda proizvođač *PCB*-a izrađuje tiskanu pločicu. Gerber dokumenti također se spremaju unutar projektnog dokumenta. Opisani postupak izrade projekta proveden je i u ovome radu prilikom projektiranja upravljačke jedinice istosmjernog motora.

Budući da su za izradu shematskog i *PCB* dokumenta potrebne shematske oznake komponenata koje se koriste za izradu projekta te stvaran raspored pinova svake komponente na tiskanoj pločici (eng. footprint), potrebno je izraditi biblioteke u kojima će biti sve shematske oznake komponenti te njihove realne dimenzije odnosno stvaran raspored pinova. Naime, *Altium* ima već ugrađene neke od biblioteka za korištenje, no u njima nema svih komponenti koje su korištene za izradu upravljačke jedinice, stoga je potrebno izraditi vlastite biblioteke. Za izradu biblioteke sa shematskim oznakama komponenti potrebno je u *Altium*-u odabrati *File* → *New* → *Library* → *Schematics Library*, nakon čega se otvara sučelje za izradu shematskog prikaza komponente. Postupak izrade biblioteke s *PCB* komponentama je jednak, samo što se umjesto *SchematicsLibrary* odabere *PCB Library*, nakon čega se otvara sučelje za crtanje stvarnog rasporeda pinova pojedine komponente u kojemu je moguće svakoj komponenti izraditi 3D model ili preuzeti već gotovi 3D model u *Step* formatu i konvertirati ga unutar

Altium-a. U ovome radu korištene su biblioteke *MiR.SchLib* i *MiR.PcbLib* koje su izrađene od strane mentora ovoga rada, prof.dr.sc. Mladena Crnekovića, te koja su proširene dodatnim komponentama potrebnim za projektiranje upravljačke jedinice istosmjernog motora.

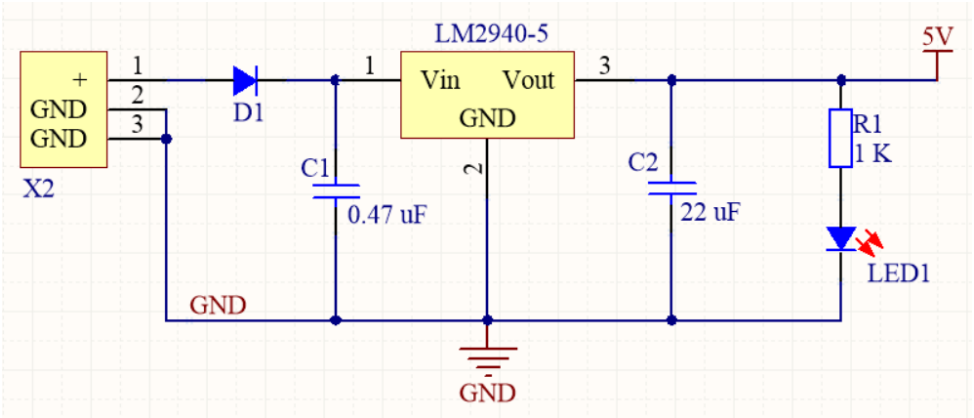
3.2. Izrada shematskog dokumenta upravljačke jedinice istosmjernog motora

Kod izrade sheme upravljačke jedinice istosmjernog motora najprije se kreće od mikrokontrolera. Budući da *Atmega2560* mikrokontroler radi na temelju frekvencije koju proizvodi 16MHz-ni kristal potrebno ga je povezati s kristalom. Način spajanja kristala i mikrokontrolera može se pronaći u tehničkoj dokumentaciji koju daje proizvođač mikrokontrolera. Na slici 22. prikazan je način spajanja mikrokontrolera i kristala kako je to propisano od strane proizvođača.



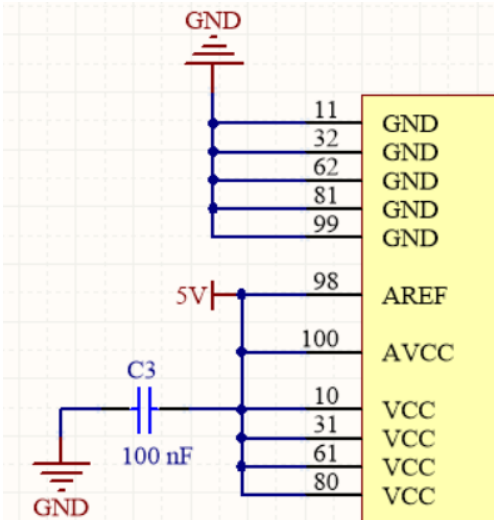
Slika 23. Shema spajanja kristala i mikrokontrolera

Nadalje, kako je u prethodnim poglavljima spomenuto da mikrokontroler zahtjeva stabilan izvor napajanja od 5 V, potrebno je stabilitazor napona također implementirati unutar upravljačke jedinice istosmjernog motora kako bi motor dobivao konstantan izvor napajanja. Odabrani stabilizator napona spaja se na način prikazan slikom 23.



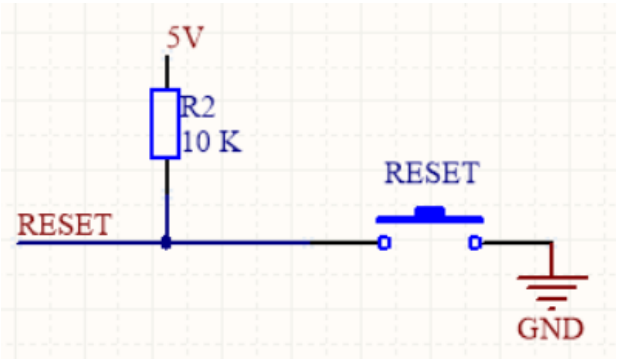
Slika 24. Shema spajanja stabilizatora napona *LM2940-5*

Odabir kapaciteta kondenzatora u spoju stabilizatora *LM2940-5* odabran je prema preporuci proizvođača koja se nalazi u tehničkoj dokumentaciji stabilizatora. Sada kada je osiguran stabilan napon na izlazu iz stabilizatora, potrebno je taj izlaz povezati s pinovima mikrokontrolera preko kojih on dobiva napajanje. Budući da *Atmega2560* mikrokontroler ima više pinova za napajanje, potrebno ih je sve povezati na isti izvor napajanja, isto tako potrebno je sva uzemljenja spojiti kako bi svi bili na jednakom potencijalu od 0 V, što je prikazano na slici 24.



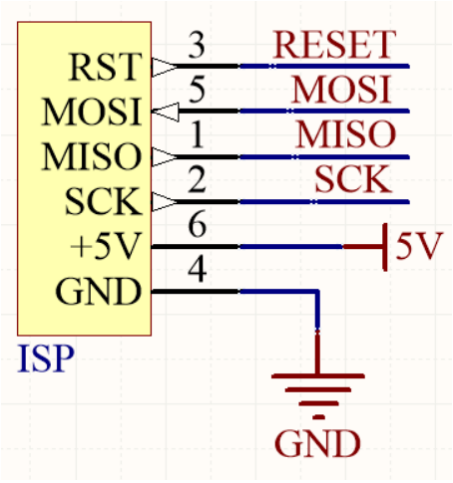
Slika 25. Prikaz spajanja napajanja i uzemljenja mikrokontrolera

Nadalje, kako bi se mikrokontroler mogao resetirati od strane korisnika, potrebno je na njegov pin pod nazivom *RESET* spojiti tipku koju kada korisnik pritisne pokrene vanjsko resetiranje mikrokontrolera. Shema spajanja tipke na pin *RESET* mikrokontrolera prikazana je na slici 25.



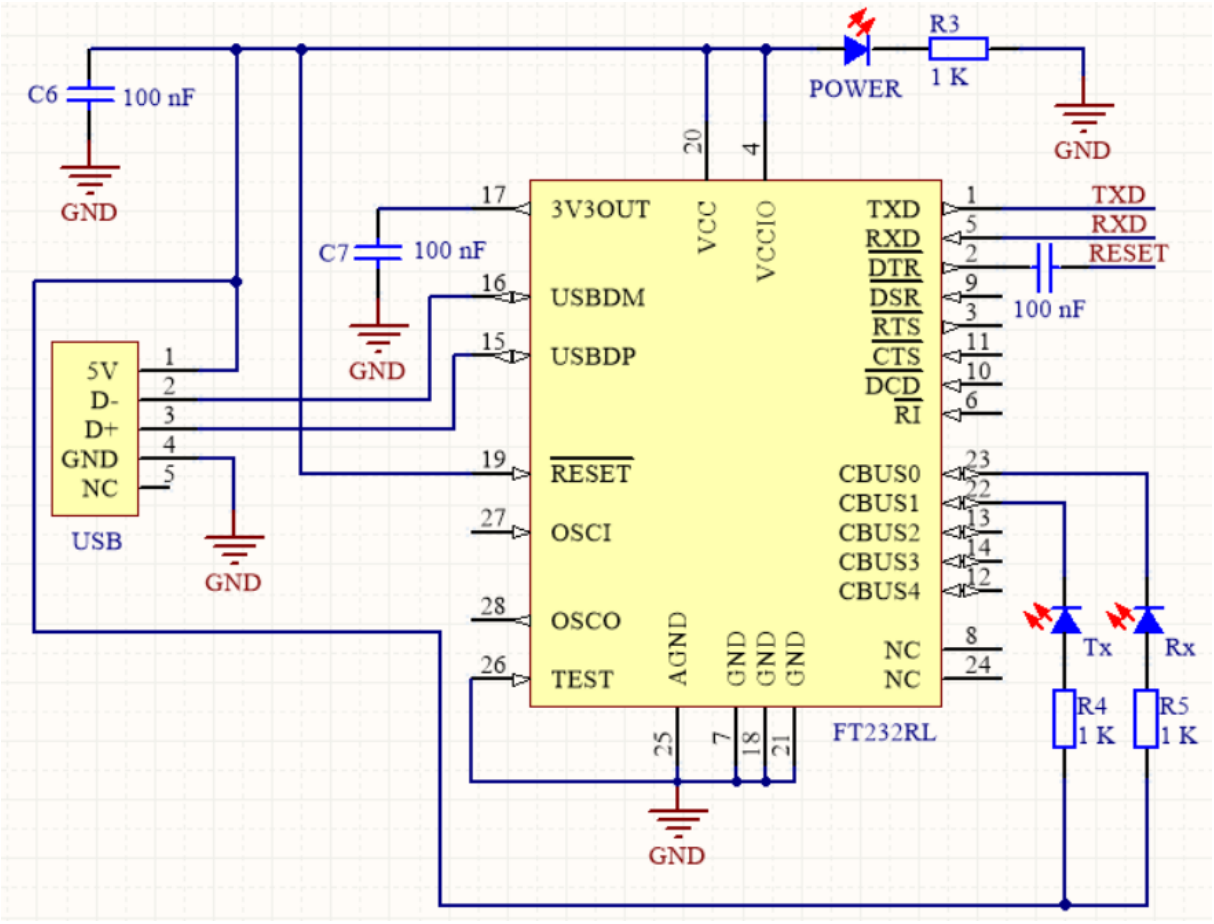
Slika 26. Shema spajanja tipke za resetiranje mikrokontrolera

Kako bi se *Atmega2560* mikrokontroler mogao programirati u različitim softverima kao što su *WinAVR, AtmelStudio, CodeVisionAVR* ili *Arduino* softver, potrebno je u njegovu *FLASH* memoriju ubaciti *Boot Loader* program koji čita kod napisan u jednom od spomenutih softvera i prevodi ga u *FLASH* memoriju mikrokontrolera. Da bi se *Boot Loader* program ubacio potrebno je na mikrokontroler spojiti konektor (*ISP* konektor) preko kojega se ubacuje *Boot Loader* program prema shemi prikazanoj na sljedećoj slici.

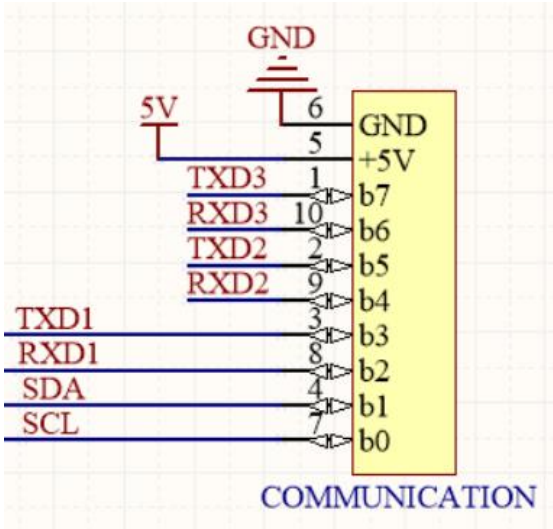


Slika 27. Shema spajanja ISP konektora

Budući da se mikrokontroler programira preko računala te kako bi mogao komunicirati serijskom *UART* vezom sa samim računalom potrebno je uspostaviti komunikaciju između računala i mikrokontrolera za što je zadužen *FT232RL* čip koji je detaljno opisan u poglavlju 2.3. Shema spajanja *FT232RL* čipa i mikrokontrolera prikazana je na slici 27. Osim komunikacije s računalom, mikrokontroler također treba imati mogućnost komunikacije s drugim mikrokontrolerima, stoga je potrebno pinove za komunikaciju na mikrokontroleru (*TXD* i *RXD* pinovi) spojiti na konektor na koji će se moći onda spojiti neki drugi mikrokontroler ili računalo s kojim se želi uspostaviti komunikacija na način prikazan na slici 28. Naime, *Atmega2560* ima 4 *TXD* i *RXD* pina, što znači da može uspostaviti 4 serijske *UART* komunikacije. Kako je u poglavlju 2.3. opisano, za serijsku su komunikaciju potrebna dva pina, a to su *TXD* i *RXD* pinovi. *TXD* pin je pin zadužen za slanje podataka na računalo ili mikrokontroler, dok je *RXD* prijemni pin, odnsono njegova je uloga da prima podatke pristigle serijskom vezom. Stoga je kod spajanja tih pinova važno obratiti pozornost da se uvijek *TXD* pin mikrokontrolera spaja na *RXD* pin uređaja i obrnuto.

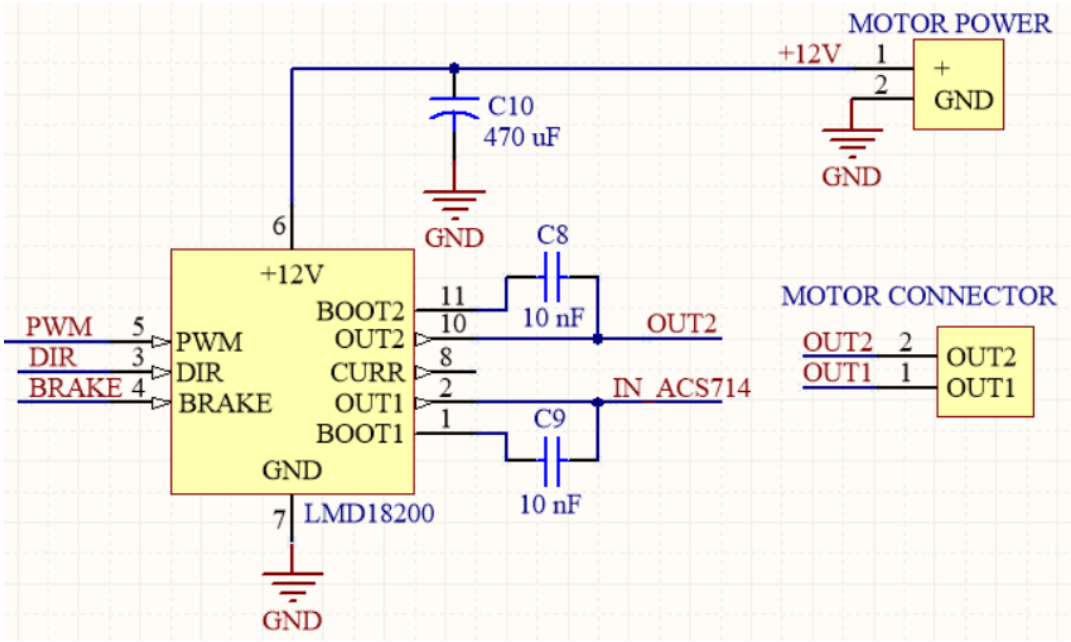


Slika 28. Shema spajanja *FT232RL* čipa



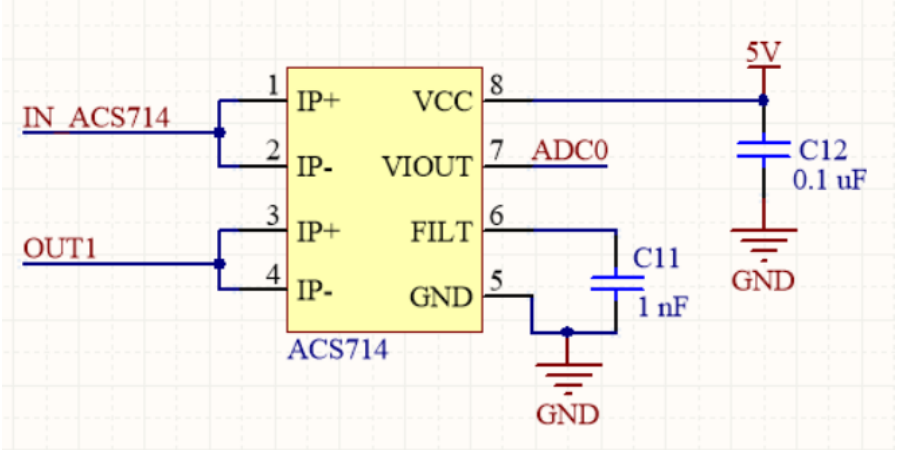
Slika 29. Shema spajanja konektora za komunikaciju i mikrokontrolera

Sljedeće što je potrebno spojiti na mikrokontroler jest energetski usmjerivač struje odnosno *LMD18200* čip s integriranim *H*-mostom. Taj čip ima tri upravljačka pina kojima se određuju signali na njegovu izlazu. Budući da je pin 5 *LMD18200* čipa *PWM* pin, na njega se spaja *PB5* pin mikrokontrolera koji je zapravo 16-bitni *PWM* pin. Na *DIR* pin *LMD18200* čipa spaja se *PB4* pin mikrokontrolera koji je ujedno i 8-bitmi *PWM* pin. Dok se na *BRAKE* pin *LMD18200* čipa povezuje *PB7* pin mikrokontrolera, odnosno 16-bitni *PWM* pin. Na sljedećoj je slici prikazana shema spajanja *LMD18200* čipa i mikrokontrolera.



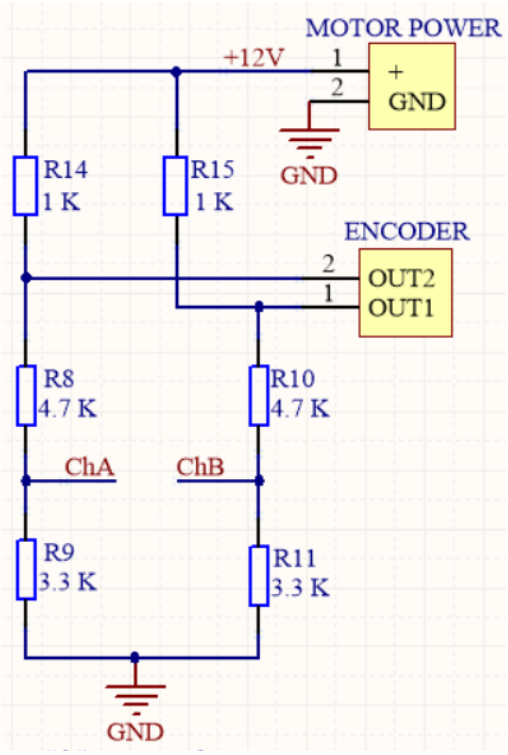
Slika 30. Shema spajanja *LMD18200* čipa i mikrokontrolera

Nadalje, u seriju s *OUT1* pinom *LMD18200* čipa povezuje se senzor struje *ACS714* kao što je prikazano na sljedećoj slici.



Slika 31. Shema spajanja senzora struje *ACS714ELCTR-05B* i mikrokontrolera

Zadnje što je potrebno spojiti na mikrokontroler jest konektor od enkodera. No, budući da enkoder koristi napajanje iz istog izvora kao i motor, što znači 12 V, potrebno je napraviti djelitelj napona na svakom kanalu enkodera kako bi se dobio napon od 5 V koji se onda može spojiti na mikrokontroler. U protivnom, ako bi se enkoder direktno spajao na mikrokontroler došlo bi do uništenja pinova mikrokontrolera, jer je maksimalni napon koji smije doći na pinove 6 V. Nadalje, pinovi mikrokontrolera na koje se spaja enkoder su izabrani tako da imaju prekidnu rutinu (eng. interrupt). Stoga je za spajanje *A* kanala enkodera (*chA*) odabran pin *PE7* sa prekidnom rutinom 7 (*INT7*), dok je *B* kanal enkodera (*chB*) spojen na *PE6* pin mikrokontrolera s prekidnom rutinom 6 (*INT6*). Na sljedećoj je slici prikazana shema spajanja enkodera sa mikrokontrolerom.

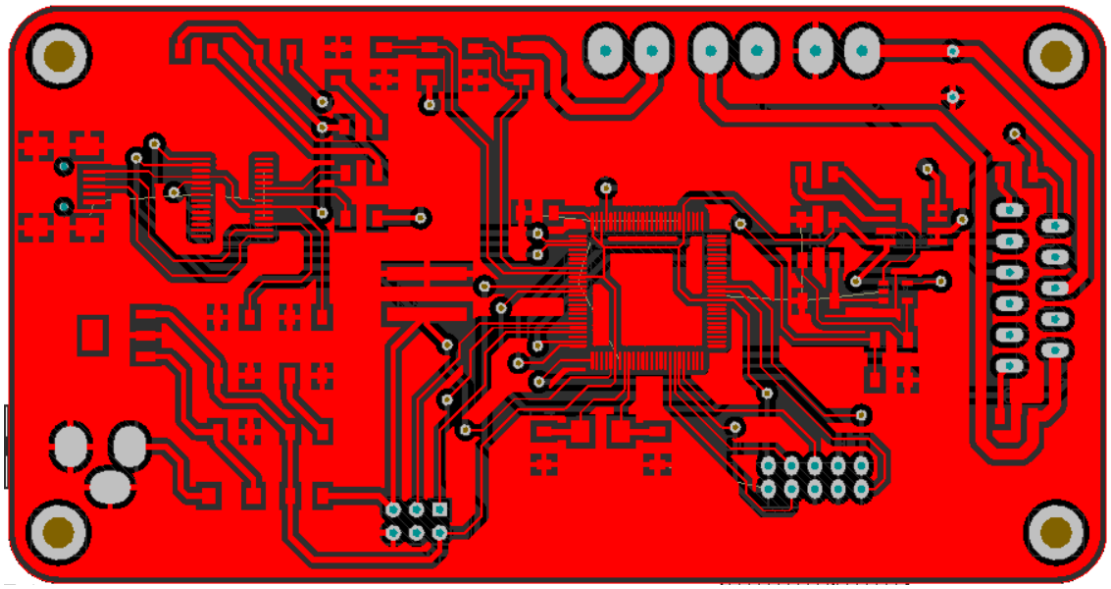


Slika 32. Shema spajanja enkodera i mikrokontrolera

3.3. Tiskana pločica upravljače jedinice istosmjernog motora

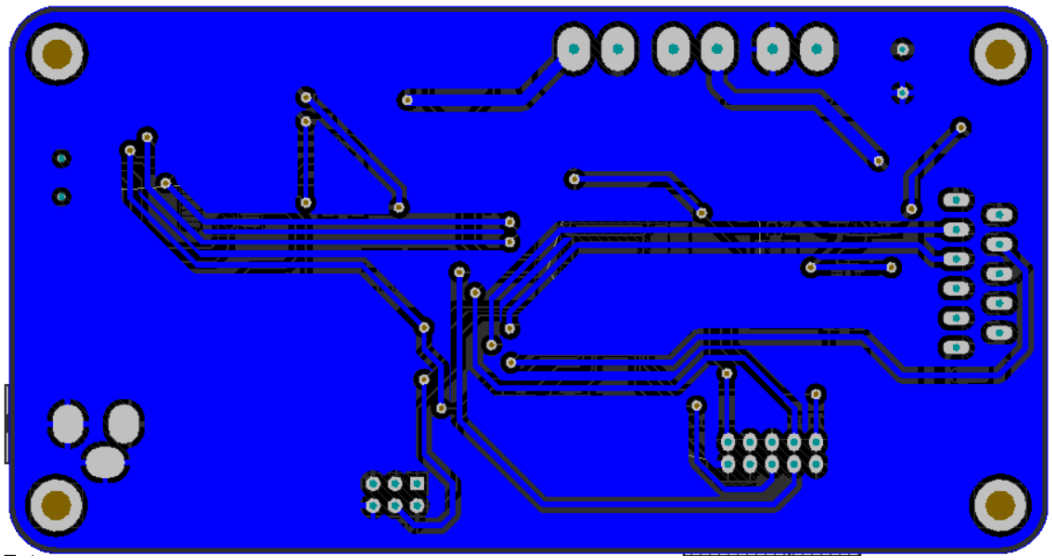
Nakon što je napravljena shema upravljače jedinice u *Altium Designer*-u može se prijeći na izradu *PCB* dokumenta. Naime, *Altium Designer* ima mogućnost da se iz shematskog dokumenta automatski ubace sve korištene komponente u *PCB* dokument u kojemu je onda potrebno najprije razmjestiti komponente obraćajući pažnju na funkcionalnost, duljinu vodova između komponenata (potrebno je da su što kraći) i na estetiku same tiskane pločice. Kada se napravi razmještaj komponenata, slijedi izrada vodova između komponenata prema shemi. No, budući da *Altium* u *PCB* dokumentu prikazuje koje je komponente potrebno međusobno povezati, nije potrebno gledati samu shemu, što izuzetno olakšava posao. Nadalje, kako se tiskana pločica sastoji od više slojeva (eng. Layer), u nastavku će biti prikazani slojevi korišteni za izradu tiskane pločice upravljačke jedinice istosmjernog motora.

Gornji sloj u kojemu se projektiraju vodovi naziva se u *Top Layer* i prikazan je na slici 32.

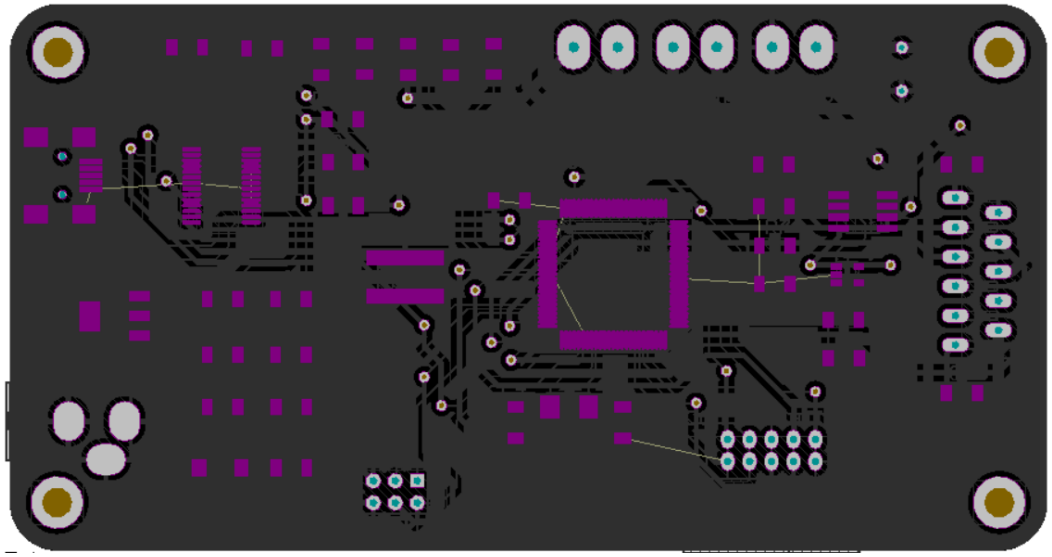


Slika 33. Gornji sloj s vodovima ili *Top Layer*

Na slici 33. prikazan je donji sloj ili *Bottom Layer* na kojemu se također projektiraju vodovi. Razlog zbog kojega se koriste gornji i donji sloj za izradu vodova je u tome što je na taj način moguće projektirati pločice manjih dimenzija. Spajanje gornjeg i donjeg sloja vrši se pomoću rupe (eng. via) koja je iznutra obložena bakrom ili drugim vodljivim materijalom koji se koristi za izradu vodova. Prikaz rupa i bakrenih površina na koje se lemljenjem povezuju komponente (eng. Pad) nalazi se na sloju koji se naziva gornja maska za lemljenje ili *Top Solder Layer* a dan je na slici 34.

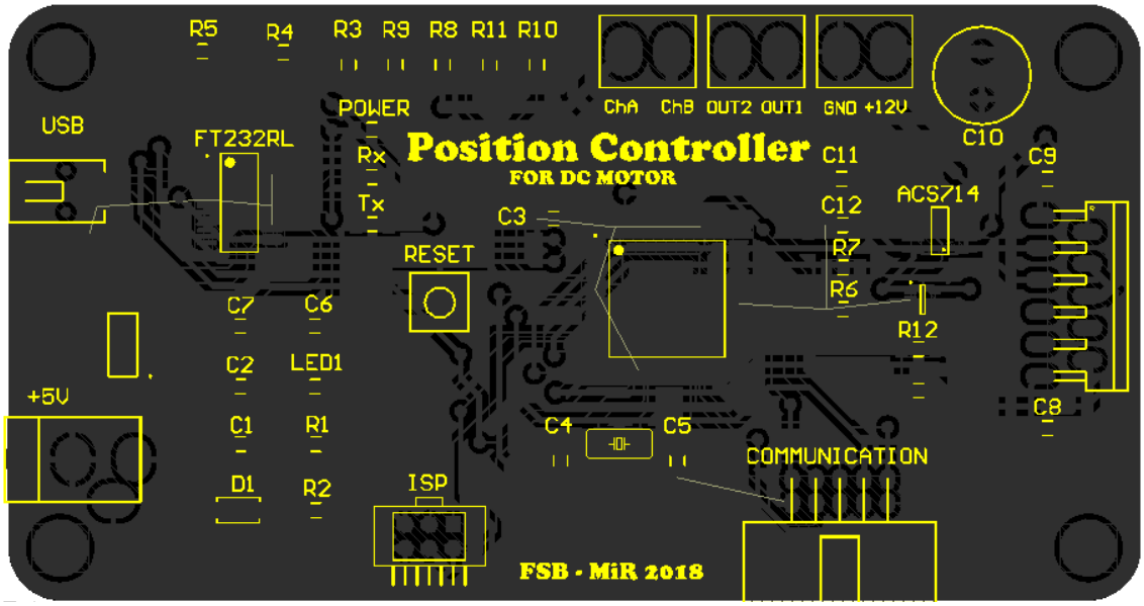


Slika 34. Donji sloj s vodovima ili *Bottom Layer*

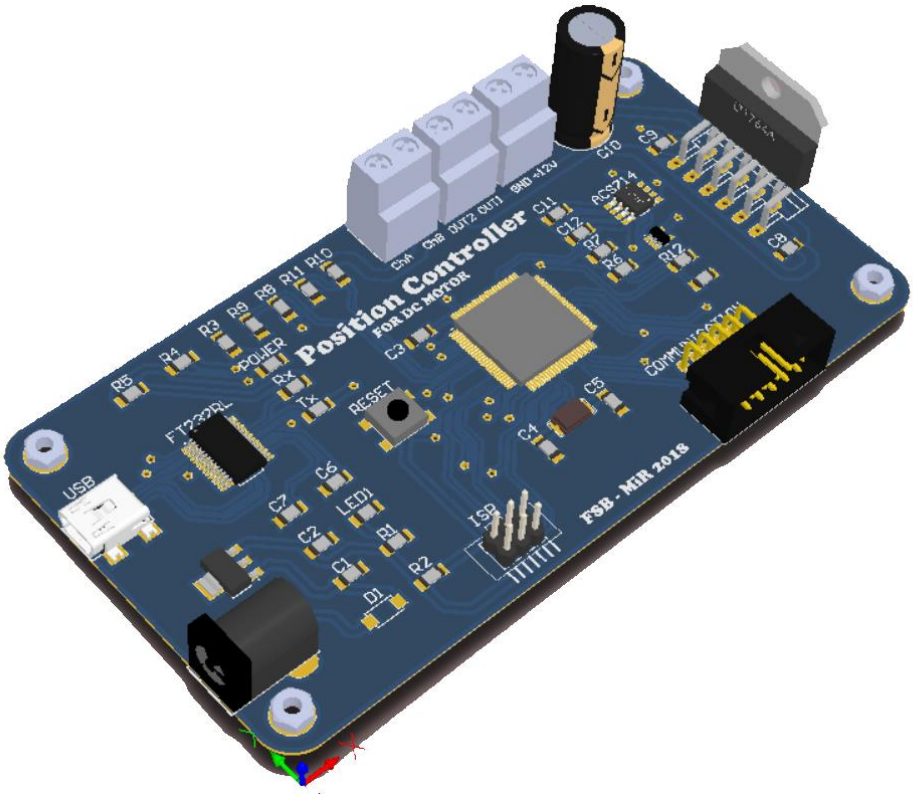


Slika 35. Gornja maska za lemljenje ili *Top Solder*

Gornji sloj s oznaka ili *Top Overlay* koristi se za pisanje naziva ili oznaka pojedinih komponenti koje se koriste na tiskanoj pločici kao i za označavanje njihovog smještaja i orijentacije te za sve ostale potrebne oznake i natpise. Njegov prikaz dan je na slici 34., a na slici 35. prikazan je 3D model gotove *PCB* tiskane pločice upravljačke jedinice istosmjernog motora.

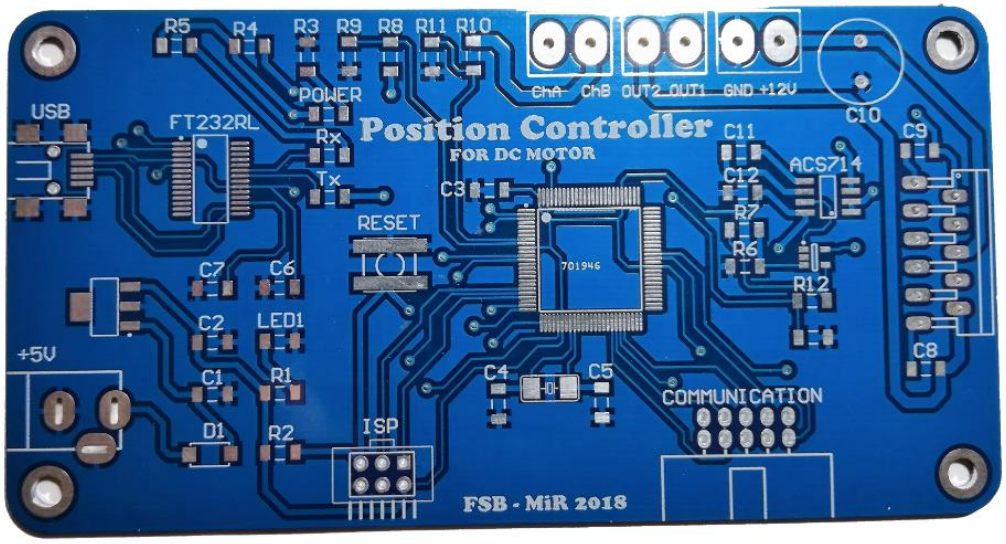


Slika 36. Gornji sloj s oznakama ili *Top Overlay*

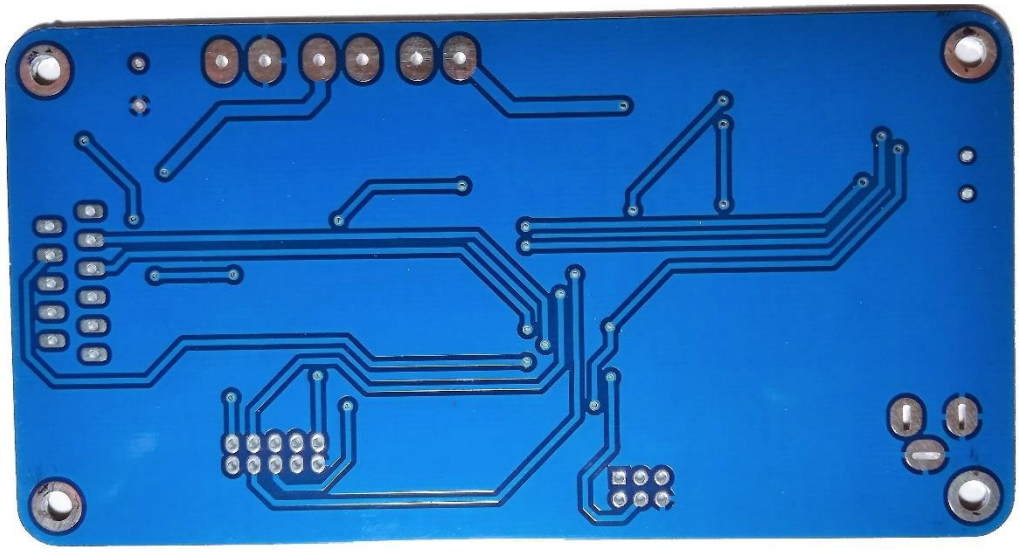


Slika 37. 3D model upravljačke jedinice istosmjernog motora

Nakon što je izrade *PCB* -a u - u, za potrebe proizvodnje tiskane pločice potrebno je napraviti Gerber datoteke prema kojima se onda izrađuje tiskana pločica. Izgled gotove tiskane pločice napravljene od strane kineskog poduzeća *SeedsStudio* prikazan je na slici 37.



Slika 38. Prednja strana PCB-a upravljačke jedinice istosmjernog motora



Slika 39. Stražnja strana PCB-a upravljačke jedinice istosmjernog motora

4. PROGRAMIRANJE UPRAVLJAČKE JEDINICE ISTOSMJERNOG MOTORA

Nakon izrade tiskane pločice i lemljenja svih komponenata, slijedi softverski dio odnosno potrebno je programirati mikrokontroler upravljače jedinice istosmjernog motora kako bi ona mogla upravljati brzinom vrtnje ili kutom zakreta motora ovisno o njenoj primjeni. U ovome radu upravljačka jednica biti će korištena za regulaciju pozicije istosmjernog motora odnosno kuta zakreta osovine motora. Stoga će u ovome poglavlju biti opisana sama regulacija pozicije istosmjernog motora te program koji se implementira u mikrokontroler prema kojemu onda on obavlja sve potrebne zadatke za uspješno pozicioniranje osovine motora. Također biti će objašnjeni principi mjerenja kuta zakreta pomoću enkodera te komunikacija s korisnikom, a kod u cijelosti dan je u prilogu. Za potrebe testiranja upravljačke jedinice konstruirana je maketa čiji je izgled prikazan na slikama 48. i 49.

4.1. Mjerenje kuta zakreta i brzine vrtnje istosmjernog motora

Kako bi regulacija pozicije istosmjernog motora bila moguća, potrebno je poznavati informaciju o trenutnoj poziciji osovine motora odnosno potrebno je poznavati kut u svakom trenutku. Informaciju o kutu zakreta i brzini vrtnje moguće je odrediti na temelju impulsa koji pristižu s enkodera. Točnost informacije o kutu zakreta koja pristiže s enkodera ovisi o rezoluciji samog enkodera, odnosno o broju impulsa koji daju po okretu osovine motora. Inkrementalni enkoder korišten u ovome radu daje 14 impulsa po okretaju, odnosno svaki kanal daje 7 impulsa. No, kako je prijenosi omjer istosmjernog motora, korištenog u ovome radu, jednak 27, to će značiti da će svaki kanal enkodera davati 189 impulsa po okretaju, što je dosta loša razina točnosti. No, u za potrebe testiranja upravljačke jedinice, rezolucija enkodera biti će dovoljna. Nadalje, trenutni kut zakreta osovine motora na temelju impulsa pristiglih s enkodera računa se prema sljedećem izrazu.

$$\alpha(°) = \frac{P}{PPR} \cdot 360, \tag{4.1}$$

gdje je α kut zakreta osovine motora, P je broj impulsa koje daje enkoder i PPR je broj impulsa enkodera po okretaju osovine motora (eng. **P**ulses **P**er **R**otation).

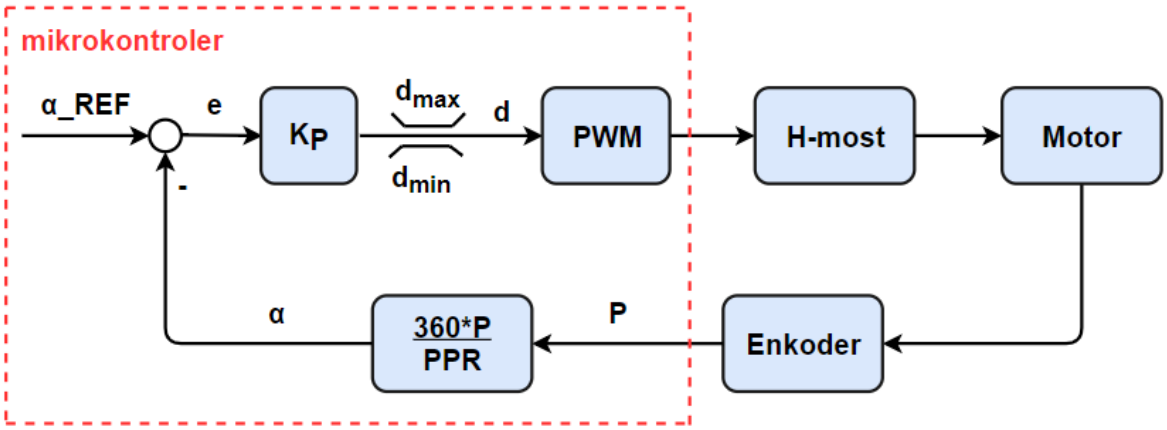
Brzina vrtnje motora na temelju impulsa pristiglih s enkodera računa se kao promjena kuta zakreta u vremenu. Stoga se računanje brzine vrtnje provodi prema sljedećem izrazu.

$$\omega\left(\frac{rad}{s}\right) = \frac{2\pi \cdot P}{PPR \cdot \Delta T}, \tag{4.2}$$

gdje je ω kutna brzina izražena u radijanima po sekundi, P je broj impulsa koje daje enkoder, PPR je broj impulsa enkodera po okretaju te ΔT vrijeme uzorkovanja, odnosno vremenski period nakon kojega se ponovno uzima informacija o kutnoj brzini.

4.2. Regulacija kuta zakreta istosmjernog motora

Prije nego što se krene s opisivanjem samog programiranja upravljačke jedinice, potrebno je objasniti princip regulacije kuta zakreta istosmjernog motora. Naime, u ovome radu biti će implementiran digitalni *P* regulator pozicije odnosno kuta zakreta. Blokovski dijagram regulacije pozicije istosmjernog motora s *P* regulatorom prikazan je na slici 39.



Slika 40. Struktura regulacije kuta zakreta istosmjernog motora

Na slici 39. *a_REF* označava referentnu ili željenu vrijednost koja se zadaje putem komunikacije upravljačke jedinice sa drugog mikrokontrolera ili računala. Oduzimanjem referentne vrijednosti i informacije pristigle s inkrementalnog enkodera *a*, dobiva se greška regulacije *e*. Iznos greške regulacije *e* ulazi u *P* regulator u kojemu se ta greška množi s iznosom proporcionalnog pojačanja K_P te se dobiva varijabla *d*. Naime, varijabla *d* zapravo predstavlja faktor popunjenosti (eng. Duty cycle) koji onda upravlja *PWM* signalima na *H*-mostu te time određuje iznos napona na armaturi motora, a time i brzinu vrtnje samog motora kao što je

opisano u poglavlju 2.1. Budući da u ovome radu nije toliko naglasak na samoj regulaciji pozicije istosmjernog motora, za određivanje proporcionalnog pojačanja K_P koristila se metoda pokušaja i promašaja, dok se nije postigao zadovoljavajući iznos pojačanja.

4.3. Programiranje mikrokontrolera upravljačke jedinice istosmjernog motora

Programiranje mikrokontrolera upravljačke jedinice Atmega2560 moguće je u nekoliko različitih softvera kao što su *WinAVR, AtmelStudio, CodeVisionAVR* ili *Arduino* softver. Zbog velike zajednice korisnika, brzog, jednostavnog i besplatnog pristupa informacijama izabran je *Arduino* softver za programiranje. Stoga će u nastavku biti opisani najvažniji dijelovi programa, a program u cijelosti nalazi se u prilogu ovoga rada.

Struktura programa u *Arduino* softveru dijeli se na dva glavna dijela, a to su *void setup()* i *void loop()*. Dio programa *void setup()* predstavlja zapravo pripremu za glavni dio programa. Stoga se u void setup() dijelu programa postavljaju i definiraju pinovi kao ulazi odnosno izlazi, postavljaju se na neke početne vrijednosti. Postavlja se komunikacija i svi ostali dijelovi potrebni za inicijalizaciju. U *void loop()* dio programa stavljaju se one zadaće i funkcije za koje se želi da se odvijaju neprestano dok je mikrokontroler uključen. Osim ta dva glavna dijela na samom početku se uključuju biblioteke koje će se koristiti, te se definiraju varijable i njihovi tipovi podataka. Na slici 40. prikazan je *void setup()* dio programa.

```
void setup(){

  // POSTAVLJANJE SERIJSKE UART KOMUNIKACIJE
  Serial.begin(9600);
  Serial.println("Inicijalizacija");
  Serial.println("--------------------------------------");
  Serial.println("Za početak pozicioniranja motora upisite '+' ili '-' za prekid rada!");
  Serial.println("--------------------------------------");
  Serial.println("Upisite zeljeni kut: ");
  Serial.println("--------------------------------------");

  // POSTAVLJANJE PREKIDNIH RUTINA
  attachInterrupt(7,chA_Interrupt,RISING);
  attachInterrupt(digitalPinToInterrupt(TCST1203_pin),TCST1203_interrupt,FALLING);

  // POSTAVLJANJE BROJAČA VREMENA DA IZVRSAVA PREKIDNU RUTINU SVAKIH deltaT ms
  timerID = timer.setInterval(deltaT, RepeatTask);

  // INICIJALIZACIJA VARIJABLI
  timerEnabled = false;
  first = true;
  angle_REF = 0;
}
```

Slika 41. *void setup()* dio programa

Kao što se može vidjeti na slici 40. u *void setup()* dijelu programa na početku se definira brzina prijenosa podataka serijskom komunikacijom te se u terminalu za komunikaciju između upravljače jedinice i računala ispisuju početne upute korisniku koji komunicira s upravljačkom jedinicom. Naime, u ovome radu u svrhu testiranja upravljačke jedinice željenu vrijednost kuta zakreta zadaje korisnik preko terminala za serijsku komunikaciju, a kasnije se u svrhu primjene upravljačke jedinice može napraviti program u kojemu bi bilo zadano da ona komunicira sa nekim drugim mikrokontrolerom koji bi joj zadavao referentne vrijednosti. Također u *void setup()* dijelu programa definiraju se vanjske prekidne rutine (eng. external interrupt) pomoću naredbe *attachInterrupt(interrupt, function, mode)*. Za regulaciju kuta zakreta istosmjernog motora potrebne su dvije vanjske prekidne rutine, jedna koju će aktivirati *A* kanal enkodera kada prijeđe iz niskog stanja u visoko te optički senzor za inicijalizaciju početnog položaja kada prijeđe iz visokog u nisko stanje. Aktiviranjem vanjskih prekidnih rutina aktivira se i funkcija koja onda kreće sa izvršavanjem. Funkcija koju aktivira *A* kanal enkodera prelaskom iz niskog u visoko stanje prikazana je na slici 41.

```
// FUNKCIJA KOJU AKTIVIRA A KANAL ENKODERA
void chA_Interrupt(){
   if(digitalRead(chB_pin) == LOW){ anglePulse += 1; }
   else { anglePulse -= 1; }
}
```

Slika 42. Funkcija *chA_Interrupt()* koju aktivira *A* kanal enkodera

Funkcija *chA_Interrupt()* ima zadatak da unutar varijable *anglePulse* zbraja odnosno oduzima impulse pristigle s *A* kanala enkodera. Zbrajanje ili oduzimanje impulsa ovisi o smjeru vrtnje motora, a smjer vrtnje određuje se prema stanju *B* kanala enkodera. Ako je *B* kanal u niskom stanju u trenutku prelaska *A* kanala enkodera iz niskog u visoko stanje, impulsi se zbrajaju te se motor vrti u jednom smjeru, ako bi se dogodila promjena u smjeru vrtnje motora ili bi se motor od početka krenuo vrtjeti u suprotnom smjeru, stanje *B* kanala enkodera bi bilo visoko, što bi imalo za posljedicu oduzimanje impulsa. Nadalje, funkcija koju aktivira optički senzor *TCST1203* prikazana je na slici 42., a njen je zadatak inicijalizacija početne vrijednosti kuta zakreta odnosno definiranje referentne točke od koje se mjeri kut zakreta. Stoga je prije pokretanja sustava, potrebno ručno dovesti osovinu motora na mjesto referentne točke, kako bi se aktivirala prekidna rutina odnosno kako bi se aktivirala funkcija *TCST1203_interrupt()*. U ovome radu za inicijalizaciju referentne točke potrebno je na maketi prikazanoj na slici () ručno

dovesti kazaljku između optičkih vrata *TCST1203* senzora, pri čemu se kut zakreta postavlja na vrijednost nula i od te točke kreće dalje mjerenje kuta zakreta.

```
// FUNKCIJA KOJU AKTIVIRA TCST2103 OPTICKI SENZOR
void TCST1203_interrupt(){
  if (first == true){
   anglePulse = 0;
   first = false;
   Serial.println("TCST1203_interrupt");
  }
}
```

Slika 43. Funkcija *TCST1203_interrupt()* za inicijalizaciju referente točke

Osim vanjskih prekidnih rutina, postoje prekidne rutine koje se izvode nakon točno određenog vremenskog intervala i koje se također definiraju unutar *void setup()* dijela programa. Takve prekidne rutine nazivaju se unutarnje prekidne rutine (eng. internal interrupts), a aktivirane su od strane brojača koji je postavljen da broji neko definirano vrijeme, nakon kojega onda pokrene funkciju definiranu unutar prekidne rutine. Unutarnja je prekidna rutina u regulaciji i od iznimne važnosti, jer je potrebno da se dio programa zaduženog za regulaciju kuta zakreta odvija u točno određenim vremenskim razmacima kako bi bila moguća kvalitetna regulacija. Stoga je i u ovome zadatku korištena unutarnja prekidna rutina koja će svakih *deltaT* ms aktivirati funkciju *RepeatTask()* unutar koje je pohranjen kod za regulaciju kuta zakreta koji se izvršava prema strukturi regulacije prikazanoj na slici 39. Na slici 43. prikazana je funkcija *RepeatTask()*.

```
// FUNKCIJA UNUTARNJE PREKIDNE RUTINE
void RepeatTask(){
  if(anglePulse < 0){anglePulse = (-1)*anglePulse;}
  angle = ((360*(long)anglePulse) / PPR); // Racunanje kuta zakreta
  e_angle = angle_REF - angle; // Greska regulacijee
  d = Kp*e_angle; // Izlaz iz P regulatora
  if(d > 0){
    if(d > 255) {d = 255;} // Ogranicavanje iznosa faktora popunjenosti
    analogWrite(PWM_pin, d); // PWM izlazni signali
    digitalWrite(DIR_pin, HIGH); // Smjer vrtnje - clockwise
  }
  else if(d < 0){
    d = d*(-1);
    if(d > 255){ d = 255;} // Ogranicavanje iznosa faktora popunjenosti
    digitalWrite(DIR_pin, LOW); // Smejr vrtnje - counterclokwise
    analogWrite(PWM_pin, d); // PWM izlazni signali
  }
}
```

Slika 44. Funkcija unutarnje prekidne rutine

Na kraju, potrebno je još objasniti *void loop()* dio programa. Naime, u *void loop()* dijelu programa vrši se komunikacija s korisnikom. Najprije se provjerava jeli uopće serijska veza dostupna te ako je, u Terminalu se traži od korisnika da upiše kut zakreta za koji želi da se motor zakrene. Također kako bi se sustav pokrenuo, potrebno je pritisnuti na tipku ' + ' odnosno na tipku ' - ' ako se želi prekinuti rad. Na slici 44. prikazan je *void loop()* dio programa.

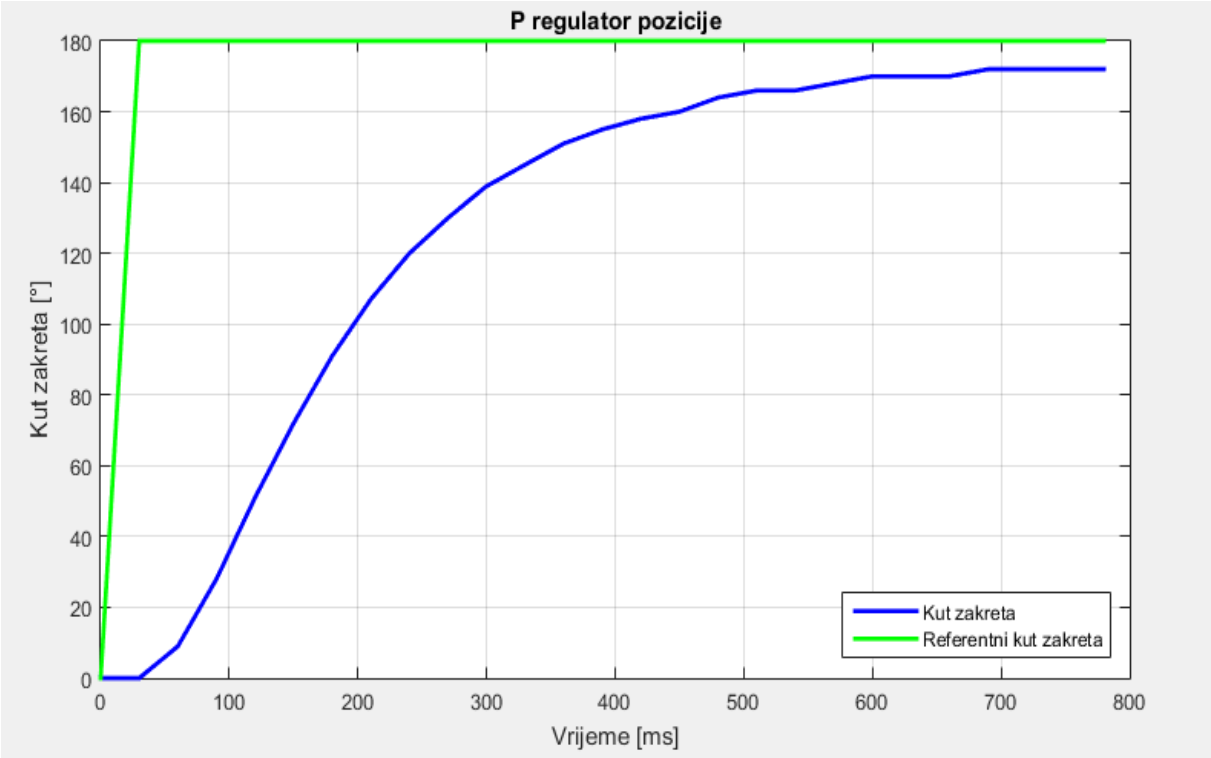
```
void loop(){
  timer.run(); // Pokretanje brojaca
  if(timerEnabled == false){
    timer.disable(timerID);
    analogWrite(PWM_pin, 0);
  }
  if(timerEnabled == true){
    timer.enable(timerID);
  }
  if (Serial.available()){
    charInput = Serial.read();
    if(isDigit(charInput) == true){
      stringNumber += charInput;
    }
    if(isDigit(charInput) == false){
      stringWord +=charInput;
    }
    if(charInput == '\n'){
      intInputNumber = stringNumber.toInt();
      angle_REF = intInputNumber;
      Serial.println("-------------------------------------");
      Serial.print("Zeljeni kut zakreta: ");
      Serial.println(angle_REF);
      Serial.println("-------------------------------------");
      Serial.println("Upisite zeljeni kut: ");
      Serial.println("-------------------------------------");
      stringNumber = "";
    }
    if(charInput == '+'){
      timerEnabled = true;
      Serial.println("DC motor kontroler zapocinje s radom!");
      Serial.println("-------------------------------------");

    }
    if(charInput == '-'){
      timerEnabled = false;
      Serial.println("-------------------------------------");
      Serial.println("DC motor kontroler je ugasen");
    }
  }
}
```

Slika 45. Program zadužen za komunikaciju s korisnikom

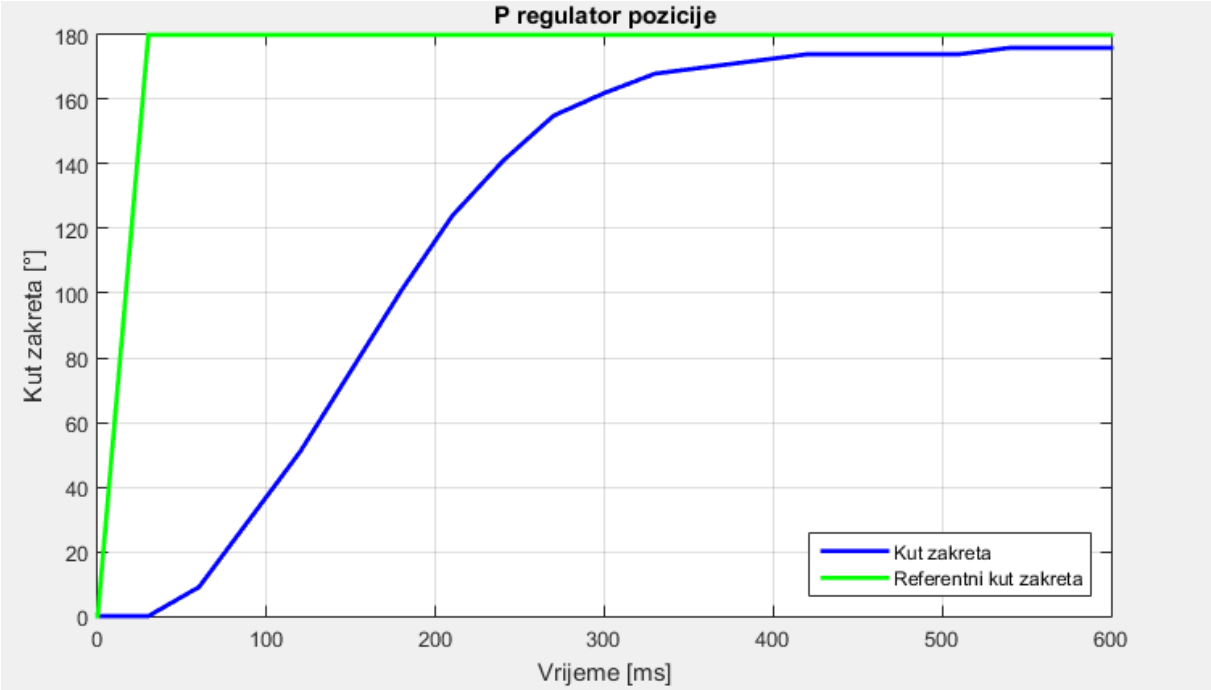
4.4. Rezultati odziva kuta zakreta na step pobudu

Na 45. prikazani su rezultati regulacije kuta zakreta istosmjernog motora koristeći *P* regulator pozicije.



Slika 46. Odziv kuta zakreta na step pobudu, K_P = 1,5

Kao što se može vidjeti na slici 45., prisutna je trajna regulacijska pogreška, što je i očekivano, zbog nedovoljno velike rezolucija enkodera. Također na slici se vidi da je vrijeme porasta dosta veliko, otprilike iznosi 450 ms. Stoga je potrebno povećati pojačanje K_P kako bi se smanjilo vrijeme porasta te kako bi kut zakreta u kraćem vremenu postigao iznos reference, naravnu uz trajnu regulacijsku pogrešku koju zbog nedovoljne točnosti mjernih uređaja nije moguće minimizirati. Kao što je već spomenuto, odabir parametara regulatora, u ovom slučaju samo pojačanja K_P odabire se na temelju metode pokušaja i promašaja, tako da se pojačanje povećava dok se ne dođe do prebačaja, odnosno dok se ne pojave prve oscilacije. Kako je za regulaciju kuta zakreta zahtjev da ne dolazi do prebačaja, odnosno da odziv kuta zakreta na step pobudu bude aperiodski, pojavom prvih oscilacija, pojačanje se smanjuje tako da u odzivu nema prebačaja. Na taj se način odabire pojačanje koje će imati najkraće vrijeme porasta te aperiodski odziv. Na slici 47. prikazan je odziv kuta zakreta na step pobudu za pojačanje odabrano na prethodno objašnjen način.

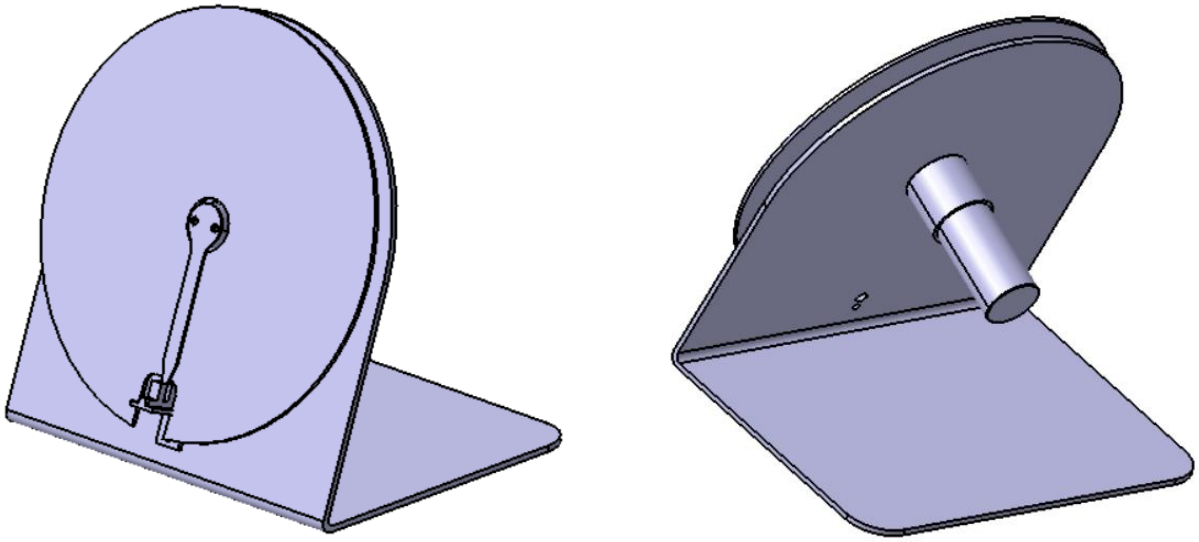


Slika 47. Odziv kuta zakreta na step pobudu, K_P = 2,2

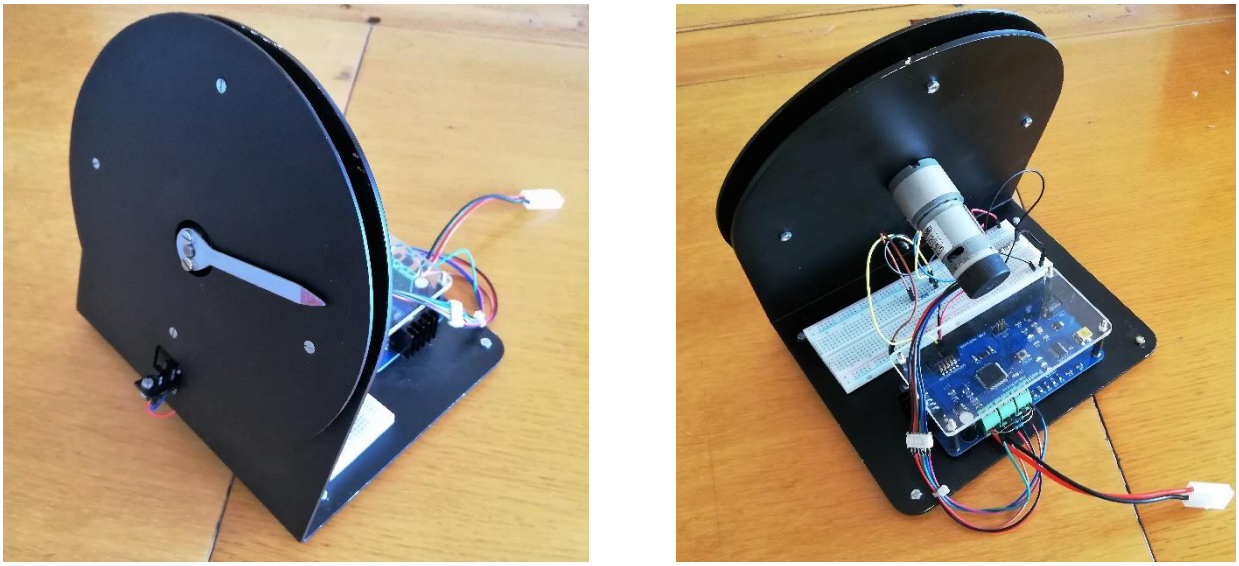
Na slici 47. može se vidjeti da je vrijeme porasta dosta kraće od vremena porasta na slici 46., a samim time je i kvalitetnija i bolja regulacija. Razlog tome je, kako je već spomenuto ranije, pojačanje K_P.

5. KONSTRUKCIJA MAKETE

Kako je već spomenuto, u svrhu testiranja upravljačke jedinice istosmjernog motora, konstruirana je maketa, a njen izgled prikazan je na sljedećim slikama.



Slika 48. Konstrukcija makete



Slika 49. Prikaz gotove makete i upravljačke jedinice istosmjernog motora

6. ZAKLJUČAK

U sklopu ovoga rada projektirana je i izrađena upravljačka jedinica istosmjernog motora s četkicama. Najprije su odabrani svi elementi potrebni za rad upravljačke jedinice. Tako je u 2. poglavlju najprije odabran energetski usmjerivač struje motora zadužen za napajanje armature motora. Budući da odabrani usmjerivač struje motora zahtjeva upravljačke signale, odabran je mikrokontroler koji ispunjava sve parametre potrebne za uspješan i točan rad upravljačke jedinice. Kako bi upravljačka jedinica imala mogućnost regulacije kuta zakreta i brzine vrtnje, odabrani su senzori koji imaju mogućnost mjerenja ovih fizikalnih veličina te ih pretvaraju u signale razumljive mikrokontroleru. Nadalje, kako bi upravljačka jedinica mogla komunicirati s računalom (korisnikom) ili drugim mikrokontrolerom odabrana je komunikacija i definiran njen protokol. Nakon odabira svih potrebnih komponenti, projektirana je tiskana pločica upravljačke jedinice istosmjernog motora. U 3. je poglavlju najprije shemtski opisan način povezivanja odabranih elemenata u cjelinu, nakon čega je opisan način izrade tiskane pločice. Završetkom hardverskog dijela slijedi softverski dio, odnosno programiranje mikrokontrolera upravljačke jedinice. Stoga je u 4. poglavlju najprije opisana regulacija kuta zakreta i teorija koja stoji iza toga, a nakon toga je napravljen kod u Arduino softveru koji je također detaljno objašnjen u istom poglavlju. Na kraju, kako bi se mogao testirati rad upravljačke jedinice istosmjernog motora, konstruirana je i napravljena maketa.

U konačnici, može se zaključiti da su na temelju dobivenih rezultata testiranja, rezultati rada same upravljačke jedinice zadovoljavajući, a greška prisutna tijekom pozicioniranja motora je uvelike posljedica ne dovoljno preciznih mjernih uređaja, a ne toliko same upravljačke jedinice. Naravno, postoje i nedostatci, kao što su nemogućnost regulacije struje motora zbog premale osjetljivosti senzora struje. Isto tako, upravljačka jedinica nema mogućnost *RS232* i *RS485* komunikacije koje su često primjenjivane u industriji.

LITERATURA

[1] Pavković D.: Elektromotorni servopogoni (auditorne vježbe), FSB, 2016.

[2] Buzjak D.: Konstrukcija i vođenje rotacijskog crtala [diplomski rad]. Fakultet strojarstva i brodogradnje, Zagreb, 2017.

[3] *LMD18200* tehnička dokumentacija: http://www.ti.com/lit/ds/symlink/lmd18200.pdf

[4] Zorc D.: Mikroprocesorsko upravljanje (podloga uz predavanja), FSB, 2016.

[5] *Atmega2560* tehnička dokumentacija: http://ww1.microchip.com/downloads/en/devicedoc/atmel-2549-8-bit-avr-microcontroller-atmega640-1280-1281-2560-2561_datasheet.pdf

[6] Serijska komunikacija mikrokontrolera [Na internetu]: http://www.micromatic.hr/download/Serijska%20komunikacija%20mikrokontrolera.pdf

[7] *LM2940-5* tehnička dokumentacija: http://www.ti.com/lit/ds/symlink/lm2940c.pdf

[8] *FT232R* tehnička dokumentacija: https://www.ftdichip.com/Support/Documents/DataSheets/ICs/DS_FT232R.pdf

[9] *TCST2103* tehnička dokumentacija: https://www.vishay.com/docs/81147/tcst2103.pdf

[10] *ACS714* tehnička dokumentacija: https://www.allegromicro.com/~/media/Files/Datasheets/ACS714-Datasheet.ashx

[11] Crneković M.: Projektiranje mikroprocesorskih sustava (podloge uz predavanja), FSB, 2016.

[12] Crneković M.: Mikroprocesorsko upravljanje (auditorne vježbe), FSB, 2016.

PRILOZI

1. Arduino programski kod

```
/*
 * Project: Zavrsni rad
 * Author: Tomislav Skender
 * Version: 1.0
 * Date: 20.2.2019.
 * Chip type: Atmega2560
 */
#include <SimpleTimer.h>
#define PWM_pin 11 // PWM_pin za pokretanje motora je digitalni/PWM pin 11
#define DIR_pin 12 // Pin za odredivanje smjera vrtnje DC motora
#define BRAKE_pin 13 // Pin za kocenje motora
#define chA_pin 82 // A signal enkodera - interrupt 7
#define chB_pin 81 // B signal enkodera - interrupt 6
#define TCST1203_pin 21 // Signal sa optickog senzora - interrupt 0
#define currentValue_pin 54 // Analogni pin A0 ili digitalni D54

SimpleTimer timer;
char charInput;
String stringNumber = "";
String stringWord = "";
int intInputNumber;
int deltaT = 30;
int d = 0;
int timerID;
int anglePulse = 0;
int angle = 0;
int e_angle = 0;
int angle_REF = 0;
float Kp = 2.2;
```

```
int PPR = 188;
boolean timerEnabled;
boolean first;

void setup(){

  // POSTAVLJANJE SERIJSKE UART KOMUNIKACIJE
  Serial.begin(9600);
  Serial.println("Inicijalizacija");
  Serial.println("-------------------------------------");
  Serial.println("Za početak pozicioniranja motora upisite '+' ili '-' za prekid rada!");
  Serial.println("-------------------------------------");
  Serial.println("Upisite zeljeni kut: ");
  Serial.println("-------------------------------------");

  // POSTAVLJANJE PREKIDNIH RUTINA
  attachInterrupt(7,chA_Interrupt,RISING);
  attachInterrupt(digitalPinToInterrupt(TCST1203_pin),TCST1203_interrupt,FALLING);

  // POSTAVLJANJE BROJAČA VREMENA DA IZVRSAVA PREKIDNU RUTINU SVAKIH deltaT ms
  timerID = timer.setInterval(deltaT, RepeatTask);

  // INICIJALIZACIJA VARIJABLI
  timerEnabled = false;
  first = true;
  angle_REF = 0;
}

void loop(){
```

```
 timer.run(); // Pokretanje brojaca
 if(timerEnabled == false){
  timer.disable(timerID);
  analogWrite(PWM_pin, 0);
 }
 if(timerEnabled == true){
  timer.enable(timerID);
 }
 if (Serial.available()){
  charInput = Serial.read();
  if(isDigit(charInput) == true){
   stringNumber += charInput;
  }
  if(isDigit(charInput) == false){
   stringWord +=charInput;
  }
  if(charInput == '\n'){
   intInputNumber = stringNumber.toInt();
   angle_REF = intInputNumber;
   Serial.println("-------------------------------------");
   Serial.print("Zeljeni kut zakreta: ");
   Serial.println(angle_REF);
   Serial.println("-------------------------------------");
   Serial.println("Upisite zeljeni kut: ");
   Serial.println("-------------------------------------");
   stringNumber = "";
  }
  if(charInput == '+'){
   timerEnabled = true;
```

```
      Serial.println("DC motor kontroler zapocinje s radom!");
      Serial.println("-------------------------------------");


    }
    if(charInput == '-'){
      timerEnabled = false;
      Serial.println("-------------------------------------");
      Serial.println("DC motor kontroler je ugasen");
    }
  }
}
// FUNKCIJA UNUTARNJE PREKIDNE RUTINE
void RepeatTask(){
  //if(anglePulse < 0){anglePulse = (-1)*anglePulse;}
  angle = ((360*(long)anglePulse) / PPR); // Racunanje kuta zakreta
  e_angle = angle_REF - angle; // Greska regulacijee
  d = Kp*e_angle; // Izlaz iz P regulatora
  if(d > 0){
    if(d > 255) {d = 255;} // Ogranicavanje iznosa faktora popunjenosti
    analogWrite(PWM_pin, d); // PWM izlazni signali
    digitalWrite(DIR_pin, HIGH); // Smjer vrtnje - clockwise
  }
  else if(d < 0){
    d = d*(-1);
    if(d > 255){ d = 255;} // Ogranicavanje iznosa faktora popunjenosti
    digitalWrite(DIR_pin, LOW); // Smejr vrtnje - counterclokwise
    analogWrite(PWM_pin, d); // PWM izlazni signali
  }
}
```

```
// FUNKCIJA KOJU AKTIVIRA A KANAL ENKODERA
void chA_Interrupt(){
  if(digitalRead(chB_pin) == LOW){ anglePulse += 1; }
  else { anglePulse -= 1; }
}

// FUNKCIJA KOJU AKTIVIRA TCST2103 OPTICKI SENZOR
void TCST1203_interrupt(){
 if (first == true){
  anglePulse = 0;
  first = false;
  Serial.println("TCST1203_interrupt");
 }
}
```