



UNIVERZITET U SARAJEVU
ELEKTROTEHNIČKI FAKULTET
ODSJEK ZA AUTOMATIKU I ELEKTRONIKU

Sistem za upravljanje dvoosnim manipulatorom baziranom na step motorima

ZAVRŠNI RAD
- PRVI CIKLUS STUDIJA -

Student:
Adi Subašić

Mentor:
Doc.dr Emir Sokić, dipl.ing.el.

Sarajevo,
septembar 2018.

Tema za završni rad

Sistem za upravljanje dvoosnim manipulatorom baziranom na step motorima

Sažetak:

U industrijskom okruženju je često potrebno ostvariti dvoosno linijsko kretanje manipulatora. Za precizna pozicioniranja se tipično koriste step motori. U ovom završnom radu je potrebno dizajnirati i implementirati upravljački sistem za dva step motora koji se može upravljati putem PC računara u realnom vremenu, nekim od uobičajenih vidova komunikacije (USB, Ethernet, RS485, Modbus i sl.)

Koncept i metode rješavanja:

Osnovni dijelovi rada trebaju biti:

- pregled literature i postojećih komercijalnih rješenja,
- osmišljavanje optimalne forme sistema sa aspekta cijene, kvaliteta i efikasnosti elektroničkih struktura, korištenog vida komunikacije, mogućnosti rada u realnom vremenu, pri čemu je za baznu upravljačku komponentu potrebno razmotriti mogućnost korištenja mikrokontrolera i PLC-a,
- implementacija idejnog rješenja na matadoru i/ili simulacionom okruženju,
- dizajniranje odgovarajuće elektroničke strukture i projektne dokumentacije,
- integracija sistema u prototip ETFCam v.1.0.
- eksperimentalni rezultati (mjerenje brzine kretanja, vremena odziva, te određivanja preciznosti i upotrebljivosti sistema za rad u realnom okruženju),

Polazna literatura:

- [1] Batchelor, Bruce G., and Paul F. Whelan. Intelligent vision systems for industry. Springer Science Business Media, 2012.
- [2] Demant, Christian, C. Demant, and Bernd Streicher-Abel. Industrial image processing. Springer-Verlag, 1999.
- [3] Szeliski, Richard. Computer vision: algorithms and applications. Springer Science Business Media, 2010.

- [4] Russ, John C. Image analysis of food microstructure. CRC press, 2004.
- [5] McRoberts, Michael, Brad Levy, and Cliff Wootton. Beginning Arduino. New York.: Apress, 2010.
- [6] Oser, Jonathan, and Hugh Blemings. Practical Arduino: cool projects for open source hardware. Apress, 2011.
- [7] Perrin, Bob. "The art and science of RS-485." (2015).
- [8] L. A. Bryan, E. A. Bryan "Programmable Controllers", An Industrial Text Company Publication, USA, 1998.
- [9] J.Hugh "Automating manufacturing systems with PLCs", 2008.,
- [10] T.Kuphald "Lessons in Industrial automation", 2017

Doc. dr Emir Sokić, dipl. ing. el.

Izjava o autentičnosti radova

Završni rad I ciklusa studija

Ime i prezime: Adi Subašić

Naslov rada: Sistem za upravljanje dvoosnim manipulatorom baziranom na step motorima

Vrsta rada: Završni rad 1. ciklusa studija

Broj stranica: 103

Potvrđujem:

- da sam pročitao dokumente koji se odnose na plagijarizam, kako je to definirano Statutom Univerziteta u Sarajevu, Etičkim kodeksom Univerziteta u Sarajevu i pravilima studiranja koja se odnose na I i II ciklus studija, integrirani studijski program I i II ciklusa i III ciklus studija na Univerzitetu u Sarajevu, kao i uputama o plagijarizmu navedenim na web stranici Univerziteta u Sarajevu;
- da sam svjestan univerzitetskih disciplinskih pravila koja se tiču plagijarizma;
- da je rad koji predajem potpuno moj, samostalni rad, osim u dijelovima gdje je to naznačeno;
- da rad nije predat, u cjelini ili djelimično, za stjecanje zvanja na Univerzitetu u Sarajevu ili nekoj drugoj visokoškolskoj ustanovi;
- da sam jasno naznačio prisustvo citiranog ili parafraziranog materijala i da sam se referirao na sve izvore;
- da sam dosljedno naveo korištene i citirane izvore ili bibliografiju po nekom od preporučenih stilova citiranja, sa navođenjem potpune reference koja obuhvata potpuni bibliografski opis korištenog i citiranog izvora;
- da sam odgovarajuće naznačio svaku pomoć koju sam dobio pored pomoći mentora i akademskih tutora/ica.

Sarajevo, datum 5.9.2018

Potpis:

Adi Subašić

Sažetak

Kroz rad je analiziran način rada i podjela koračnih motora, te mogućnosti njihovog upravljanja. Realizovano je upravljanje dvoosnog manipulatora na bazi koračnih motora putem PLC-a. Korisniku je data mogućnost manualnog pomjeranja vrha manipulatora u željenom smjeru, kao i snimanje, te reproducirane sekvence odabirom odgovarajućih tastera na panelu. Sistem je proširen da omogući upravljanje pomoću računara uz korištenje serijske komunikacije i MODBUS protokola. Razvijen je prateći kod u C-u koji ima mogućnost upravljanja sistemom. Opisan je rad sistema i njegove mogućnosti, te su ponuđena odgovarajuća softverska implementacija.

Abstract

In this work the operation and types of step motors are analysed, as well as the options for control. A system for controlling a two-axis manipulator based on step motors was realized using a PLC. The user is able to manually move the gripper of the manipulator in a given direction, as well as recording and reproducing a sequence of movements using the buttons on a panel. The system was expanded to enable computer control using a serial connection and the MODBUS protocol. The required C code was developed to achieve that purpose. In the paper the operation and capabilities of the system are described, and the accompanying software implementation is included.

Sadržaj

Popis slika	8
Popis tabela	9
Indeks pojmova	9
1 Uvod	10
1.1 Tehnički opis sistema	10
2 Opis načina rada i podjele koračnih motora	13
2.1 Podjela koračnih motora	13
2.1.1 Motori sa varijabilnom reluktansom	13
2.1.2 Motori sa permanentnim magnetima	13
2.1.3 Hibridni motori	14
2.2 Podjela koračnih motora prema namotaju	14
2.2.1 Unipolarni motori	14
2.2.2 Bipolarni motori	14
2.2.3 Dvonitni motori	16
2.3 Kratak opis funkcije sistema za upravljanje dvoosnim manipulatorom	16
3 Upravljanje koračnim motorima	17
3.1 Načini upravljanja	18
3.1.1 Upravljanje s punim korakom	18
3.1.2 Upravljanje sa polu-korakom	19
3.1.3 Microstepping	19
3.2 Drajveri	22
3.2.1 Ubrzavanje porasta struje u koračnom motoru	23
3.2.2 Mosni drajver	25
3.2.3 PWM drajver	26
4 Opis sistema za upravljanje	28
4.1 Korištena oprema	28
4.1.1 Popis (specifikacija) opreme	28
4.1.2 Kratak opis najznačajnijih komponenti	28
4.1.3 Proračun dodatnih korištenih komponenti	31
4.2 Principijelna shema	33
4.3 Ormar unutar kojega su smještene komponente	33

5	Realizacija sistema za upravljanje	37
5.1	Generatori pulsa i SoMachine blokovi za njihovo upravljanje	39
5.1.1	ErrorStop i onemogućeno stanje, te prelazak u mirovanje	41
5.1.2	Zaustavljanje i stanje mirovanja	42
5.1.3	Traženje početne pozicije	42
5.1.4	Diskretno i kontinualno kretanje	43
5.2	Implementacija PTO blokova i realizacija programa upravljanja	45
6	Korištenje serijske komunikacije i MODBUS protokola	46
6.1	Uspostavljanje serijske komunikacije i korištenje MODBUS protokola za čitanje markera	46
6.2	Protokol slanja komandi na PLC	48
6.2.1	Podjela komandi i njihov odabir	49
6.2.2	Procedure pozivanja pojedinih komandi	49
6.2.3	Kreiranje proizvoljnih funkcionalnosti u SoMachine-u korištenjem prethodnog protokola	50
6.2.4	Korištenje implementiranih funkcija u C kodu	51
	Prilozi	53
A	Uputstvo za rukovanje panelom sa tasterima	54
B	C kod korišten kao osnova za uspostavljanje serijske komunikacije	56
B.1	C kod za pisanje podataka na serijski port	56
B.2	C kod za čitanje podataka sa serijskog porta	59
C	SoMachine kod	64
D	C kod	92
	Literatura	103

Popis slika

1.1	Sistem ETFcam	11
1.2	Blok dijagram sistema dvoosnog manipulatora	12
2.1	Shematski prikaz trofaznog motora sa varijabilnom reluktansom [1]	14
2.2	Shematski prikaz motora sa permanentnim magnetom [2]	14
2.3	Shematski prikaz hibridnog motora sa strane i poprečnog presjeka [2]	15
2.4	Shematski prikaz unipolarnog motora [3]	15
2.5	Shematski prikaz bipolarnog motora [3]	15
2.6	Shematski prikaz dvonitnog motora [2]	16
3.1	Blok dijagram upravljačkog sistema koračnog motora [4]	17
3.2	Rezonancija kod upravljanja sa punim korakom i polu-korakom [5]	20
3.3	Princip microstepping-a [1]	20
3.4	Ekvivalentno kolo namotaja koračnog motora [4]	22
3.5	Prigušenje diodom [4]	22
3.6	Prigušenje diodom i otpornikom[4]	23
3.7	Prigušenje Zener diodom [4]	23
3.8	Poređenje efekata različitih vrsta prigušenja [4]	23
3.9	Ubrzavanje porasta struje dodavanjem otpornika R_e u seriju s namotajem i povećanjem napona napajanja E [4]	24
3.10	Ubrzavanje porasta struje uz korištenje drajvera dvostrukog napajanja [4]	24
3.11	Karakteristika napona i struje prilikom korištenja drajvera dvostrukog napajanja [4]	25
3.12	Mosni drajver za bipolarno upravljanje [4]	25
3.13	PWM drajver [4]	26
3.14	Strujna i naponska karakteristika PWM drajvera [4]	27
4.1	Blok dijagram SMC64 drajvera[6]	29
4.2	Izgled konektora drajvera[6]	30
4.3	Shema za proračun otpornika na EN ulazima drajvera	32
4.4	Shema za proračun otpornika na ulazima drajvera CLK i DIR	32
4.5	Shema spajanja fotoelektričnog senzora OPB 981	32
4.6	Principijelna shema sistema	34
4.7	Izgled vrata ormara	35
4.8	Izgled unutrašnjosti ormara	36
5.1	Modul za prilagođenje fotoelektričnog senzora - gornja strana	38
5.2	Modul za prilagođenje fotoelektričnog senzora - donja strana	38
5.3	Panel sa tasterima za upravljanje	38

5.4	Podešavanje generatora pulsa	39
5.5	Stanja generatora pulsa	40
5.6	Trapezni profil brzine	40
5.7	Izgled MC_Reset_PTO bloka	41
5.8	Izgled MC_Power_PTO bloka	41
5.9	Izgled MC_Stop_PTO bloka	42
5.10	Izgled MC_Home_PTO bloka	43
5.11	Izgled MC_MoveAbsolute_PTO bloka	44
5.12	Izgled MC_MoveRelative_PTO bloka	44
5.13	Izgled MC_MoveVelocity_PTO bloka	44
A.1	Panel za upravljanje CNC mašinom	55

Popis tabela

3.1	Sekvenca upravljanja unipolarnog motora sa po jednom energiziranim namotajem [3]	18
3.2	Sekvenca upravljanja bipolarnog motora sa po jednom energiziranim namotajem [3]	18
3.3	Sekvenca upravljanja unipolarnog motora sa dva energizirana namotaja	19
3.4	Sekvenca upravljanja bipolarnog motora sa dva energizirana namotaja	19
3.5	Sekvenca polu-koračnog upravljanja bipolarnog motora [3]	19
3.6	Sekvenca polu-koračnog upravljanja unipolarnog motora [3]	19
3.7	Primjer microstepping-a gdje je među-korak jednak osmini punog koraka [1]	21
3.8	Sekvenca sin/cos generatora koja se treba primijeniti za pomjeranje kroz sve kvadrante [1]	21
4.1	Tabela za odabir nominalne struje izborom otpornika R_A i R_B [6]	31
6.1	Tabela sa rezervisanim komandama za PLC	49

Poglavlje 1

Uvod

Za rad sistema vizuelne inspekcije često je potrebno izdvojiti neke proizvode, bilo zbog pronalaska oštećenja, vršenja klasifikacije ili sličnih zahtjeva. Kako bi se taj problem riješio za proizvode koji se kreću pokretnom trakom ovaj rad analizira način rada koračnih motora, te ih primjenjuje za realizaciju upravljanja dvoosnim manipulatorom.

1.1 Tehnički opis sistema

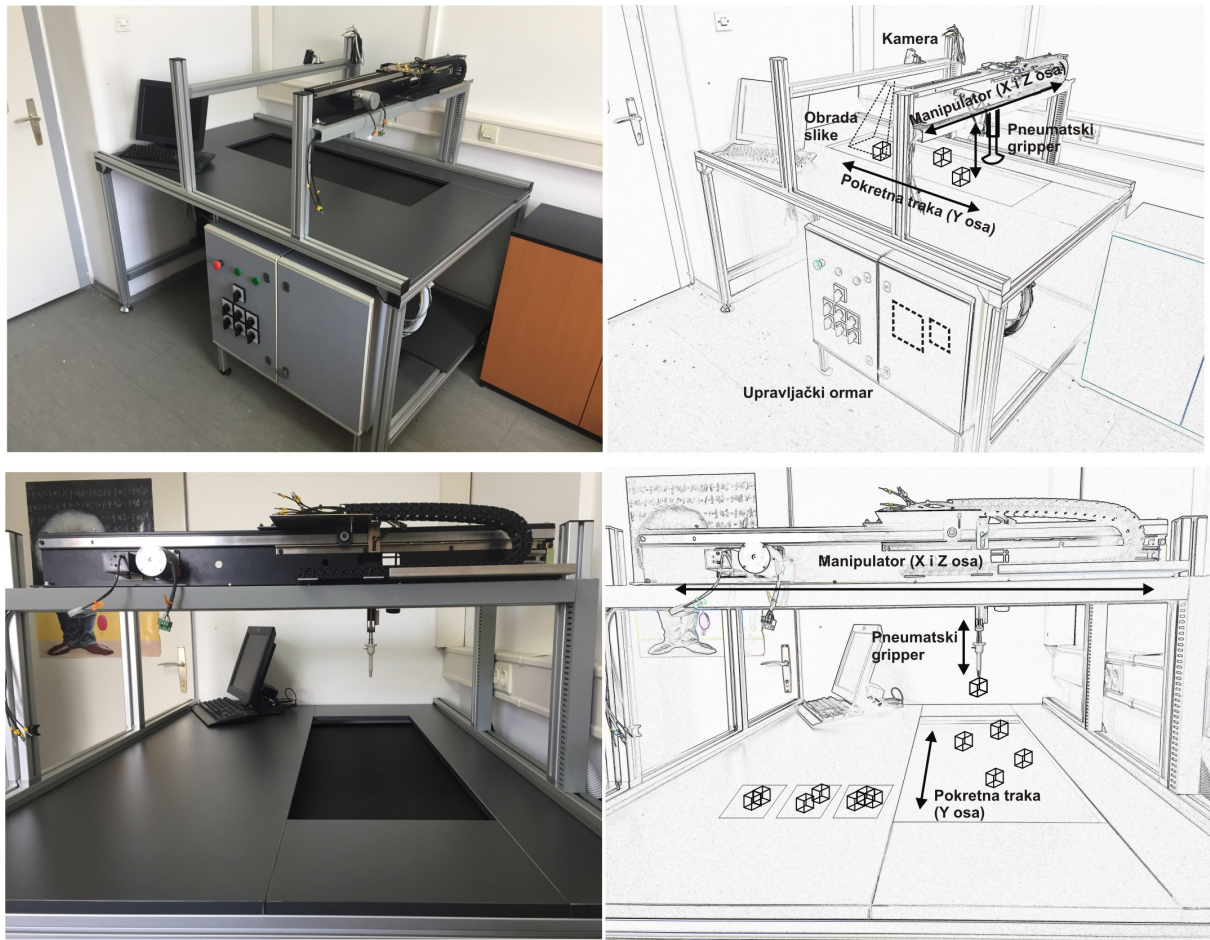
Sam rad predstavlja dio većeg sistema koji je razvijen upravo zbog potreba vizuelne inspekcije proizvoda na pokretnoj traci i naziva se ETFCam. Na slici 1.1 je prikazan fizički izgled sistema ETFCam odakle se nazire njegova funkcionalnost. Desno na slici prikazane su očekivane funkcionalnosti sistema nakon njegovog potpunog osposobljavanja gdje se objekti kreću pokretnom trakom (Y osa), dok sistem za računarsku viziju u realnom vremenu obavlja procesiranje i koordinira manipulatorom (X i Z osa) koji željene objekte preuzima sa trake i izmješta na drugu poziciju. U cjelini, sistem se sastoji od dijela koji obavlja procesiranje signala sa kamera i senzora i dijela koji upravlja kretanjem. U ovom radu riješen je dio upravljanja kretanjem za dva step motora koji upravljaju osama dvoosnog manipulatora (na slici 1.1 Manipulator (X i Z osa)).

Kao upravljačka jedinica korišten je Schneider PLC TM241CEC24R, a za interakciju sa korisnikom prvobitno je planiran Magelis HMISCU6A5, ali je u tu svrhu iskorišten ručno napravljeni panel sa tasterima koji su dalje spojeni na digitalne ulaze i izlaze PLC-a. U buduće svrhe za ovaj sistem će se vjerovatno koristiti spomenuti HMI za koji je predviđeno mjesto i na napravljenom panelu.

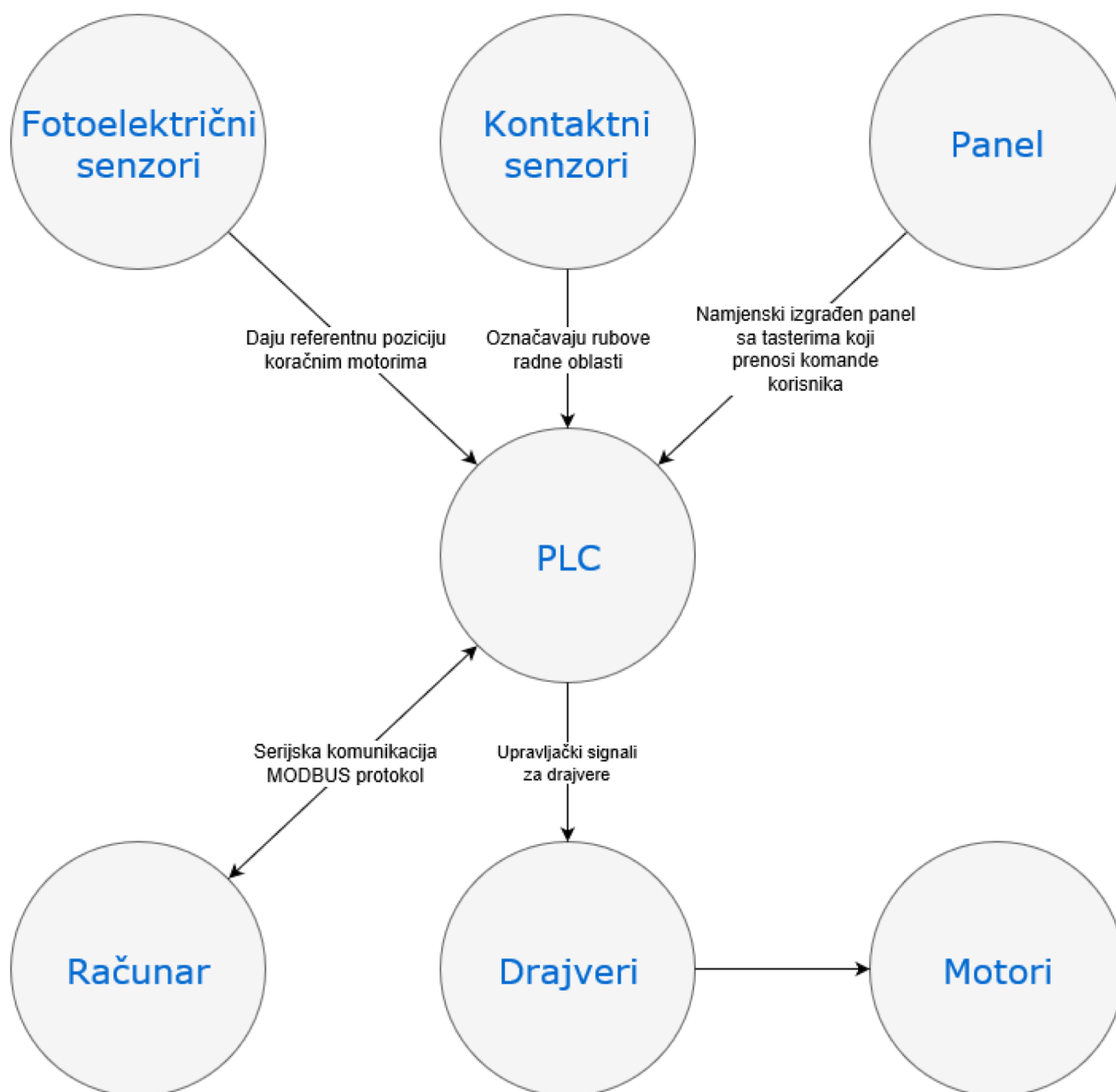
Nakon pokretanja same mašine korisnik je u mogućnosti izabrati jedan od dva moda za rad mašine. Prvi mod je manuelni, gdje korisnik određenim tasterom na panelu može odabrati smjer kretanja pojedine ose (gore-dolje po Z osi i desno-lijevo po X osi). Drugi mod je automatski, tzv. "record"- "repeat" mod, gdje se prvo zadaje sekvenca kretanja tasterima na panelu te se nakon toga željena sekvenca pohranjuje i po želji ponovo pokreće putem panela.

Mehaničko upravljanje osama vrši se pomoću step motora dok su sami motori upravljani drajverima SMC64 povezanim na PLC. Da bi se odredile granice kretanja CNC mašine, za X i Z osu postavljene su fizičke granice koje će zaustavljati daljnje kretanje. Radi sigurnosti i zaštite postavljene su dvostruke granice. Prvenstveno su granice određene sa po jednim fotoelektričnim senzorom za svaku osu, a kao dodatna zaštita koriste se krajnji prekidači, također po jedan za svaku osu. Ujedno, fotoelektrični senzori su iskorišteni za postavljanje referentne pozicije.

Radi lakšeg uklapanja u ostatak ETFCam sistema potrebno je ostvariti komunikaciju samog sistema i eksternog računara, te time omogućiti eksterno upravljanje i promatranje rada sistema dvoosnog manipulatora. Za tu svrhu koristi se serijska komunikacija i MODBUS protokol,



Slika 1.1: Sistem ETFcam



Slika 1.2: Blok dijagram sistema dvoosnog manipulatora

te time ostvaruje komunikacija između PLC-a i računara. Računarski program je pisan u C-u, te podržava sve mogućnosti upravljanja kao i panel sa tasterima, također sadrži i dodatne mogućnosti kratanje do zadate tačke ili kroz skup zadatih tačaka. Na slici 1.2 prikazan je blok dijagram koji prikazuje veze između pojedinih elemenata sistema.

Poglavlje 2

Opis načina rada i podjele koračnih motora

Definicija koračnih motora navedena je u knjizi čiji je autor A. A. Athani i nalazi se u Britanskom standardu specifikacija [1], te kaže:

"Koračni motor je DC motor bez četkica čiji se rotor okreće u diskretnim ugaonim inkrementima kada su njegovi statorski navoji pobuđeni na programiran način. Rotacija se javlja zbog magnetne interakcije između polova rotora i polova sekvencijalno pobuđenih statorskih navoja. Rotor nema električnih navoja, već tekuće i/ili magnetisane polove."

Oni se mogu podijeliti na tri osnovna tipa [3]: motori sa permanentnim magnetima, varijabilnom reluktansom i hibridni. Motori sa permanentnim magnetom, kao što ime sugerise, imaju rotor sačinjen od permanentnog magneta, dok je oni s varijabilnom reluktansom nemaju. Hibridni motori kombiniraju aspekte od prethodna dva tipa motora. Motori sa permanentnim magnetom kao i hibridni motori mogu imati [3]: unipolarne, bipolarne ili dvonitne navoje.

2.1 Podjela koračnih motora

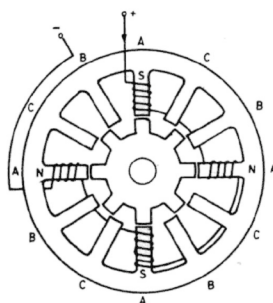
2.1.1 Motori sa varijabilnom reluktansom

Motori sa varijabilnom reluktansom ne sadrže permanentne magnete, kao ni namotaje na rotoru. Rotor je tipa tekućeg pola i sačinjen je u potpunosti od mehkog željeza, kao i stator na kome se nalaze namotaji. Broj polova statora je paran umnožak od broja faza.

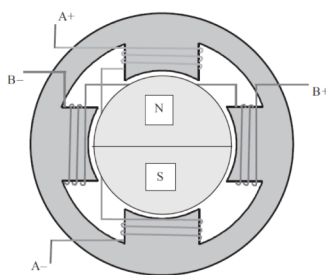
Na slici 2.1 vidimo primjer trofaznog VR motora sa 12 polova na statoru i 8 na rotoru. Kada je faza A energizirana motor se poravnava s tom fazom. Ako se onda ona isključi, a energizira faza B rotor će se poravnati s njom i okrenuti za 30° u smjeru obrnutom od kazaljke na satu.

2.1.2 Motori sa permanentnim magnetima

Motori sa permanentnim magnetima imaju sličnu konstrukciju kao VR motori, ali njihov rotor nema oblik zupčanika, već oblik cilindra i sačinjen je od permanentnog magneta. Na slici 2.2 vidimo primjer takvog motora gdje rotor ima dva magnetna pola koji se poravnavaju prema statoru. Promjena pobuđenosti namotaja prouzrokuje korak od 90° .



Slika 2.1: Shematski prikaz trofaznog motora sa varijabilnom reluktansom [1]



Slika 2.2: Shematski prikaz motora sa permanentnim magnetom [2]

2.1.3 Hibridni motori

Hibridni motori kombiniraju neke elemente od prethodna dva tipa motora. Njihov rotor je nazubljen kao onaj kod motora sa varijabilnom reluktansom, ali sadrži i permanentni magnet na osovini. Zubi na rotoru obezbjeđuju put koji pomaže za navođenje magnetnog fluksa, dok permanentni magnet poboljšava zaustavni, zadržni moment, te statički u dinamičku karakteristiku momenta motora [3]. Na slici 2.3 je prikazan takav motor.

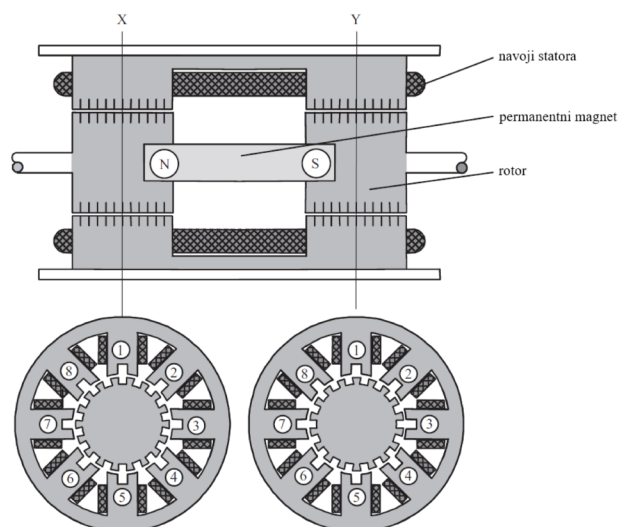
2.2 Podjela koračnih motora prema namotaju

2.2.1 Unipolarni motori

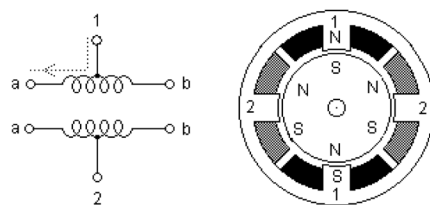
Unipolarni motori se sastoje od dva namotaja oba s izvodom na sredini namotaja. Središnji izvodi se mogu voditi na ulaz motora odvojeno ili zajedno kao jedan izvod tako da ovi motori tipično imaju 5 ili 6 žica. Oni rade tako što se sjeverni i južni polovi na permanentnom magnetu rotora privlače ka polovima statora [3]. Na slici 2.4 je prikazan presjek unipolarnog motora s korakom od 30° .

2.2.2 Bipolarni motori

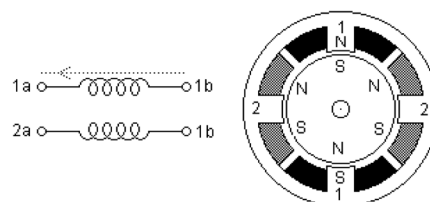
Bipolarni motori su sačinjeni od dva namotaja i imaju četiri žice. Oni nemaju središnje izvode kao unipolarni motori. Zbog toga struja mora proći kroz čitav namotaj što izaziva pojavu većeg momenta. Međutim oni zahtijevaju kompleksnija upravljačka kola od unipolarnih motora [3]. Na sici 2.5 je prikazan presjek unipolarnog motora s korakom od 30° .



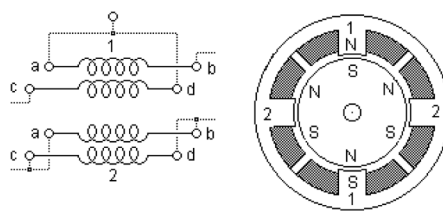
Slika 2.3: Shematski prikaz hibridnog motora sa strane i poprečnog presjeka [2]



Slika 2.4: Shematski prikaz unipolarnog motora [3]



Slika 2.5: Shematski prikaz bipolarnog motora [3]



Slika 2.6: Shematski prikaz dvonitnog motora [2]

2.2.3 Dvonitni motori

Dvonitni motori su identični u rotoru i statoru bipolarnim motorima s tim da je svaki namotaj sačinjen od dvije žice namotanim paralelno jedna drugoj. Prema tome oni imaju osam žica na ulazu. Njima se može upravljati kao s unipolarnim ili bipolarnim motorima. Da bi se koristion kao unipolarni motor potrebno je namotaje koje odgovaraju jednoj fazi spojiti serijski, te koristiti središnju konekciju kao izvod, dok za bipolarnu operaciju namotaji se mogu spajati serijski ili paralelno. Serijska konekcija omogućuje rad na većim naponima, dok paralelna rad na većim strujama [3]. Na slici 2.6 je prikazan presjek dvonitnog motora s korakom od 30°.

2.3 Kratak opis funkcije sistema za upravljanje dvoosnim manipulatorom

Ovaj završni rad kroz analizu rada koračnih motora ima za cilj omogućiti upravljanje kretanjem step motora u dvodimenzionalnom prostoru na dva načina: manuelno i automatski. Manuelni način rada podrazumijeva da na HMI-ju postoje 4 "tastera" (gore, dolje, lijevo, desno) kojima se može ostvariti kretanje dvije ose, te izvršiti podešavanje brzine kretanja duž osa. Automatski način predstavlja tzv. "record"- "repeat" funkcionalnost, u kojem se nakon pritiska na "record" putem tastera pomjeraju ose i snima željena sekvenca kretanja, a nakon pritiska na "repeat" sekvenca ponavlja.

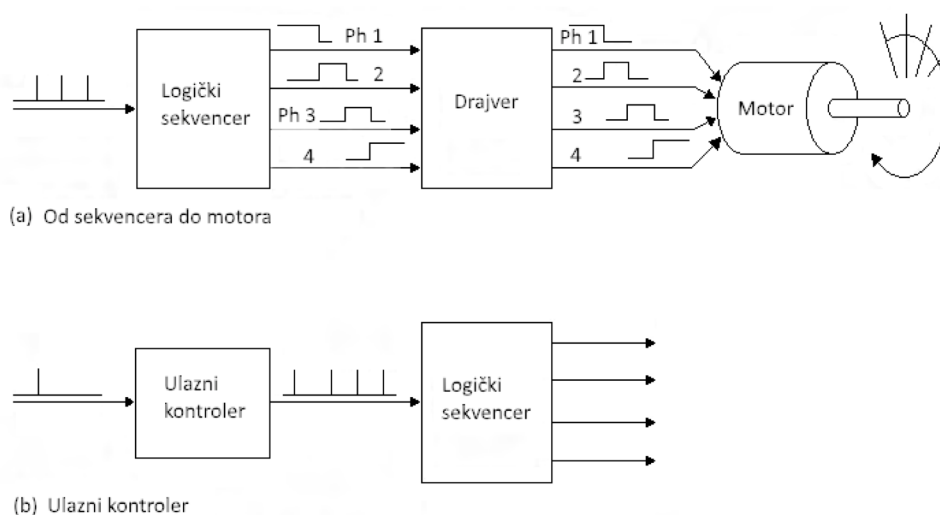
Također je potrebno omogućiti upravljanje sistemom izvana, tako da se pomoću programa na računaru može izvršiti pomjeranje step motora. U nastavku će detaljnije biti analizirano upravljanje koračnim motorima, dok će u poglavlju 4 detaljnije biti opisan sam sistem za upravljanje dvoosnim manipulatorom.

Poglavlje 3

Upravljanje koračnim motorima

Koračni motori se mogu upravljati u otvorenom ili sa zatvorenom povratnom spregom. Kako se rotacija vrši korak po korak moguće je tačno podesiti poziciju i brzinu rotacije koračnog motora bez potrebe korištenja enkodera ili drugih mjernih instrumenata. Za ispravan rad ovakvog sistema potrebno je dobro poznavanje uslova rada i radne oblasti motora. U protivnom, ukoliko se motor izloži prevelikom teretu ili se zahtijeva veća brzina od ostvarive, očekivano ponašanje motora neće biti obezbijeđeno, te postoji mogućnost oštećenja motora.

Na slici 3.1 je prikazan blok dijagram upravljačkog sistema u otvorenom. Sistem je moguće prilagoditi radu sa povratnom spregom dovođenjem odgovarajućih informacija sa senzora na ulazni kontroler i modifikacijom upravljačkog algoritma. Sistem se sastoji od ulaznog kontrolera, logičkog sekvencera-a, drajvera, te samog motora. Kontroler generiše povorku impulsa, gdje svaki impuls zahtijeva pomjeranje motora u određenom smjeru za jedan korak. Logički sekvencer pobuđuje faze motora potrebne za održavanje pozicije ili koračanje motora. Drajver prenosi pobudu logičkog sekvencera na faze motora, te pojačava snagu dobivenog signala. Ukoliko logički sekvencer može obezbijediti dovoljno veliku struju za napajanje motora on se može direktno spojiti sa fazama motora.



Slika 3.1: Blok dijagram upravljačkog sistema koračnog motora [4]

(a) Blok dijagram koji prikazuje vezu od sekvencera do motora

(b) Blok dijagram koji pokazuje vezu ulaznog kontrolera i logičkog sekvencera

Korak	0	1	2	3	4	5	6	7	8	9	10	11	12
Namotaj 1a	1	0	0	0	1	0	0	0	1	0	0	0	1
Namotaj 1b	0	0	1	0	0	0	1	0	0	0	1	0	0
Namotaj 2a	0	1	0	0	0	1	0	0	0	1	0	0	0
Namotaj 2b	0	0	0	1	0	0	0	1	0	0	0	1	0

Tabela 3.1: Sekvenca upravljanja unipolarnog motora sa po jednom energiziranim namotajem [3]

Korak	0	1	2	3	4	5	6	7	8	9	10	11	12
Terminal 1a	+	0	-	0	+	0	-	0	+	0	-	0	+
Terminal 1b	-	0	+	0	-	0	+	0	-	0	+	0	-
Terminal 2a	0	+	0	-	0	+	0	-	0	+	0	-	0
Terminal 2b	0	-	0	+	0	-	0	+	0	-	0	+	0

Tabela 3.2: Sekvenca upravljanja bipolarnog motora sa po jednom energiziranim namotajem [3]

3.1 Načini upravljanja

Način upravljanja je određen logičkim sekvencerom, odnosno veličinom koraka koji sekvencer obezbjeđuje. Moguće je ostvariti upravljanje sa punim korakom, polu-korakom, te proizvoljno malim korakom koje se naziva microstepping.

3.1.1 Upravljanje s punim korakom

Upravljanjem s punim korakom motor se pomjera sa osnovnim korakom definisanim u specifikacijama.

Unipolarni motori se upravljaju tako što se središnji izvod(i) vezuju sa napajanjem dok se krajevi namotaja alternativno vezuju za masu. Prema tome u ovim motorima smjer struje kroz namotaje statora određuje koji će rotorski polovi biti privučeni kojim statorskim polovima, smjer struje ovisi o tome koja polovina namotaja je energizirana [3]. U tabeli 3.1 prikazana je logička sekvenca upravljanja unipolarnog motora na slici 2.4 koji ima korak od 30° . Sa brojem jedan (1) označen je kraj namotaja kroz koji teče struja, odnosno kraj koji je vezan na masu. Kroz krajeve namotaja označene sa nulom (0) ne teče struja, te su oni odspojeni (ili također spojeni na napon napajanja kao i središnji izvod). Navedena sekvenca vrši rotaciju motora za punih 360° .

Struja kroz namotaj bipolarnog motora može teći u oba smjera, tako da je za njegovo upravljanje potrebno promijeniti polaritet na svakom kraju namotaja. Kao što se vidi na slici 2.5 struja će teći od lijeva ka desno u namotaju 1 kada se na kraj 1a dovede pozitivan, a na kraj 1b negativan napon. Struja će teći u obrnutom smjeru kada se napon na oba kraja zamijeni. Kako bi se ovo postiglo koristi se H-most, te kako su izvedena dva namotaja potrebna su dva takva kola [3]. U tabeli 3.1 prikazana je logička sekvenca upravljanja bipolarnog motora na slici 2.5, također sa korakom od 30° . U tabeli plus (+) i minus (-) označavaju polaritet napajanja, dok 0 označava da navedeni kraj nije pod napajanjem. Ova sekvenca rotira motor za 12 koraka odnosno punih 360° .

Ukoliko su istovremeno energizirana dva namotaja pol će se poravnati na sredini između njih. Prema istom principu moguće je istovremeno uključivanje po dva namotaja, te time postizanje istog efekta kao i ranije. Na slikama 3.3 i 3.4 je prikazana upravo takva sekvenca za unipolarne i bipolarne motore respektivno. Oznake u tabeli nose isto značenje kao što je prethodno objašnjeno. Kako su u ovome slučaju uvijek dva namota energizirana omogućeno je postizanje većeg momenta, ali je ujedno povećana i potrošnja motora.

Korak	0	1	2	3	4	5	6	7	8	9	10	11	12
Namotaj 1a	1	1	0	0	1	1	0	0	1	1	0	0	1
Namotaj 1b	0	0	1	1	0	0	1	1	0	0	1	1	0
Namotaj 2a	0	1	1	0	0	1	1	0	0	1	1	0	0
Namotaj 2b	1	0	0	1	1	0	0	1	1	0	0	1	1

Tabela 3.3: Sekvenca upravljanja unipolarnog motora sa dva energizirana namotaja

Korak	0	1	2	3	4	5	6	7	8	9	10	11	12
Terminal 1a	+	+	-	-	+	+	-	-	+	+	-	-	+
Terminal 1b	-	-	+	+	-	-	+	+	-	-	+	+	-
Terminal 2a	-	+	+	-	-	+	+	-	-	+	+	-	-
Terminal 2b	+	-	-	+	+	-	-	+	+	-	-	+	+

Tabela 3.4: Sekvenca upravljanja bipolarnog motora sa dva energizirana namotaja

3.1.2 Upravljanje sa polu-korakom

Prilikom upravljanja sa polu-korakom motor se pomjera sa osnovnim korakom upola manjim od predviđenog. Ovo je moguće zato što dvije prethodno navedene sekvence upravljanja s punim korakom generišu međusobno neovisne pozicije motora, tako da jednostavnim kombinovanjem dvije prehodne sekvece dobivamo sekvencu koja vrši upravljanje sa polu-korakom. Takva sekvenca za unipolarne i bipolarne je prikazana na slikama 3.6 i 3.5 respektivno.

Korak	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24
Terminal 1a	+	+	0	-	-	-	0	+	+	+	0	-	-	-	0	+	+	+	0	-	-	-	0	+	+
Terminal 1b	-	-	0	+	+	+	0	-	-	-	0	+	+	+	0	-	-	-	0	+	+	+	0	-	-
Terminal 2a	0	+	+	+	0	-	-	-	0	+	+	+	0	-	-	-	0	+	+	+	0	-	-	-	0
Terminal 2b	0	-	-	-	0	+	+	+	0	-	-	-	0	+	+	+	0	-	-	-	0	+	+	+	0

Tabela 3.5: Sekvenca polu-koračnog upravljanja bipolarnog motora [3]

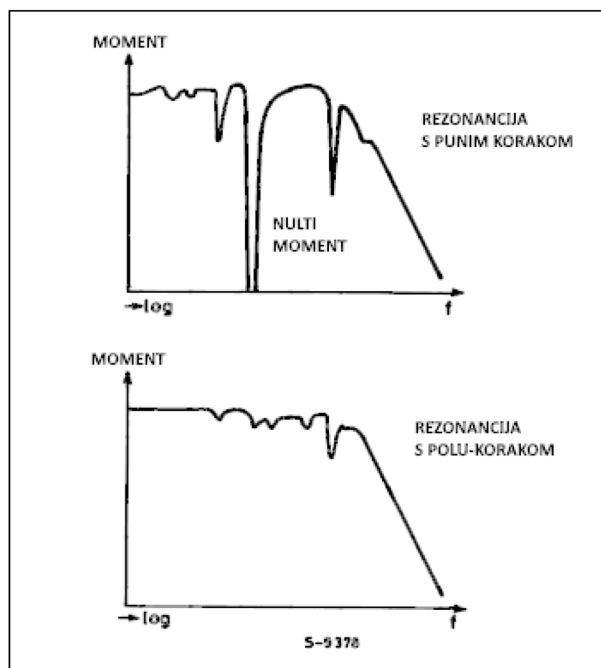
Korak	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24
Namotaj 1a	1	1	0	0	0	0	0	1	1	1	0	0	0	0	0	1	1	1	0	0	0	0	0	1	1
Namotaj 1b	0	0	0	1	1	1	0	0	0	0	0	1	1	1	0	0	0	0	0	1	1	1	0	0	0
Namotaj 2a	0	1	1	1	0	0	0	0	0	1	1	1	0	0	0	0	0	1	1	1	0	0	0	0	0
Namotaj 2b	0	0	0	0	0	1	1	1	0	0	0	0	0	1	1	1	0	0	0	0	0	1	1	1	0

Tabela 3.6: Sekvenca polu-koračnog upravljanja unipolarnog motora [3]

Glavna prednost ovoga načina upravljanja je dvostruko povećana rezolucija. Ova opcija je bolja i zbog pojave rezonancije motora. Naime stator i rotor prilikom rada se mogu posmatrati kao sistem mase i opruge, koji može biti stimuliran ka vibraciji. Ta pojava može dovesti do situacija da pri određenim brzinama motor prestane pružati bilo kakav moment. Upravljanje sa polu-korakom pomaže jer je put između koraka duplo manji, a i sistem je manje stimuliran. Glavni nedostaci ovakvog sistema su postojanje potrebe za duplo više impulsa radi vršenja jedne pune rotacije, te manji moment u odnosu na upravljanje s punim korakom [5].

3.1.3 Microstepping

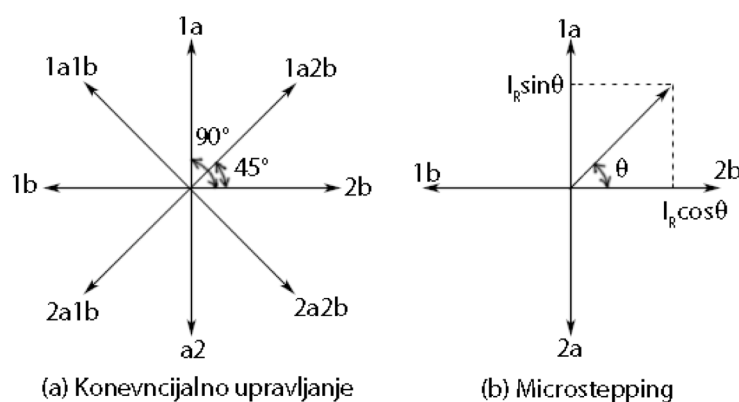
Pored upravljanja sa punim ili polu-korakom, koračni motor se može pomjerati za proizvoljno mali među-korak. Ovakav način upravljanja se naziva microstepping. On omogućava zadržavanje



Slika 3.2: Rezonancija kod upravljanja sa punim korakom i polu-korakom [5]

proizvoljno male pozicije između punog ili polu-koraka. Također uveliko povećava glatkoću kretanja pri malim brzinama, šum na srednjim brzinama, te umanjuje probleme rezonancije [7].

Princip microstepping-a ili među-koračnog upravljanja se može objasniti uz osvrtanje na upravljanje s punim korakom. Neka se koristi sekvenca prikazana Tabelom 3.1, gdje je u svakom trenutku uključena samo jedna faza. Neka je u početnom trenutku samo namotaj 2b pod napajanjem i kroz njega teče struja $I_{2b} = I_R$, nominalna fazna struja. Svi ostali namotaji u isključeni, te statorsko magnetno polje djeluje u pozitivnom smjeru duž realne ose. Rotor će također biti u poziciji $\theta = 0$. S dolaskom novog impulsa na ulaz namotaj B2 se gasi, dok se A1 uključuje. U ovom slučaju je struja $I_{2a} = I_R$, dok su ostale struje jednake nuli, te kao rezultat stator magnetnog polja rotira za 90° (električki) u smjeru suprotnom od kazaljke na satu. Rotor ga prati rotirajući se za 90° (električki) ili θ_S (mehanički). Prilikom konvencionalnog upravljanja s punim korakom stator se rotira za 90° (električno), što izaziva pomjeranje rotora za puni korak



Slika 3.3: Princip microstepping-a [1]

- (a) - Fazni dijagram statorskog magnetnog polja prilikom upravljanja s punim ili polu-korakom
 (b) - Fazni dijagram statorskog magnetnog polja prilikom microstepping-a

Međukorak	Struja kroz fazu A [%]	Struja kroz fazu B [%]
0	0	100
1	19.51	98.08
2	38.27	92.39
3	55.56	83.15
4	70.71	70.71
5	83.15	55.56
6	92.39	38.27
7	98.08	19.51
8	109	0

Tabela 3.7: Primjer microstepping-a gdje je među-korak jednak osmini punog koraka [1]

Smjer rotacije	Kvadrant za smjer	Kvadrant po standardnoj konvenciji	Namotaj motora			
			1a	2a	1b	2b
Obrnuto od smjera kazaljke na satu	I	I	sin			cos
	II	II	sin		cos	
	III	III		sin	cos	
	IV	IV		sin		cos
U smjeru kazaljke na satu	I	IV		sin		cos
	II	III		sin	cos	
	III	II	sin		cos	
	IV	I	sin			cos

Tabela 3.8: Sekvenca sin/cos generatora koja se treba primijeniti za pomjeranje kroz sve kvadrante [1]

θ_S (mehanički) [1].

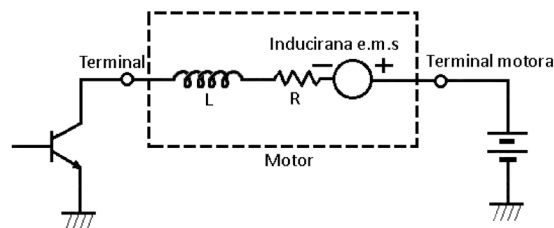
Međutim, kod microstepping-a cilj je izazvati rotaciju statičkog magnetnog polja za mali ugao $\theta \ll 90^\circ$ kako je prikazano relacijom 3.1 i 3.2.

$$I_2 b = I_R \cos \theta \quad (3.1)$$

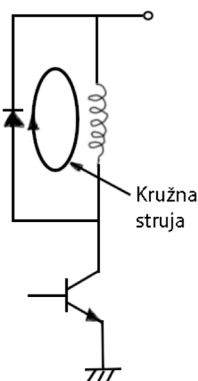
$$I_1 a = I_R \sin \theta \quad (3.2)$$

Sada se statorsko magnetno polje pomjerilo za ugao $\theta \ll 90^\circ$ (električki), što je praćeno rotacijom rotora za taj ugao, odnosno za $\Delta\theta \ll \theta_S$ (mehanički). U tabeli 3.7 su prikazane vrijednosti odnosa $I_2 b$ i $I_1 a$ koji izazivaju rotaciju za $\theta = (1/8)\theta_S$ po ulaznom impulsu.

Tabela 3.8 prikazuje struje koje statorski namotaji trebaju ostvariti radi omogućavanja upravljanja microsteppng-om kroz sve kvadrante.



Slika 3.4: Ekvivalentno kolo namotaja koračnog motora [4]



Slika 3.5: Prigušenje diodom [4]

3.2 Drajveri

Drajveri napajaju motor i utiču na njegovo ponašanje. Namotaj koračnog motora je induktivnog karaktera, a prilikom okretanja motora javlja se inducirano elektromagnetno polje. U skladu s tim namotaj se može predstaviti kao serijska veza induktiviteta, otpornosti, te izvora napona kao što je prikazano na slici 3.4. Prilikom isključenja tranzistora sa slike 3.4 veliki napon se inducira na induktivitetu, što može oštetiti tranzistor. Da bi se to spriječilo potrebno je prigušiti naponski skok, što je moguće izvesti na nekoliko načina, koji će biti opisani u nastavku [4].

Prigušenje sa diodom se vrši dodavanjem diode paralelno sa namotajem kao što je prikazano na slici 3.5. U ovome slučaju kada se tranzistor isključi struja će cirkulirati kroz diodu i vremenom iščeznuti. Potencijal na kolektoru će biti zbir napona napajanja i napona diode. Nedostatak ovoga metoda je što se kružna struja sporo disipira [4].

Prigušenje sa diodom i otpornikom se vrši dodavanjem otpornika u seriju sa diodom kao na slici 3.6. Ovo ubrzava disipaciju kružne struje, ali mijenja napon na kolektoru prema formuli 3.3:

$$V_{CE} = E + IR_S + V_{DF}, \quad (3.3)$$

gdje je:

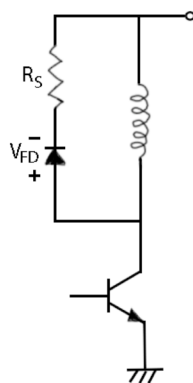
E - napon napajanja,

I - struja pred isključenje tranzistora,

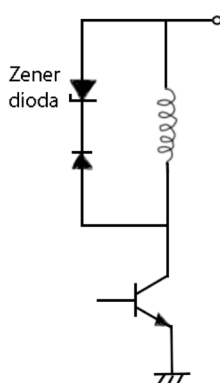
R_S - dodana otpornost,

V_{DF} - napon diode.

Što je veća otpornost R_S struja se brže disipira, ali je i veći napon na kolektoru što ugrožava tranzistor.



Slika 3.6: Prigušenje diodom i otpornikom[4]

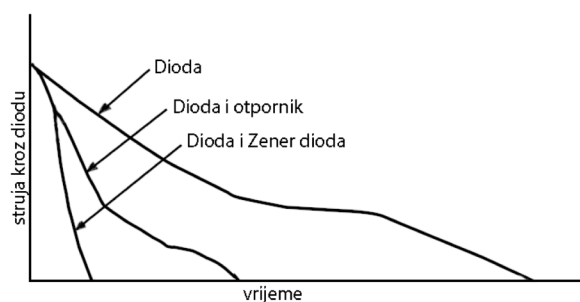


Slika 3.7: Prigušenje Zener diodom [4]

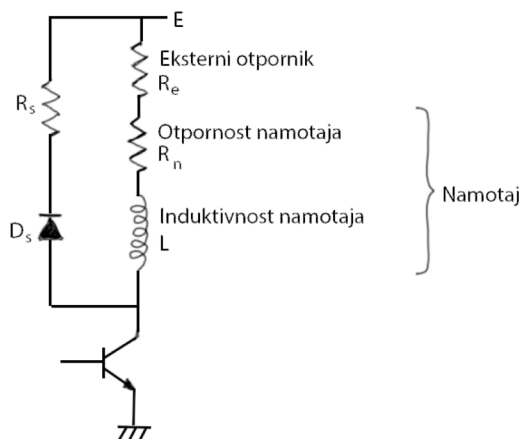
Prigušenje sa Zener diodom se vrši dodavanjem Zener diode u seriju sa diodom kao na slici 3.7. Ovim putem je postignuta najbrža disipaciju kružne struje, a napon na kolektoru predstavlja zbir napona napajanja i napona Zener diode. Na slici 3.8 je prikazano poređenje različitih vrsta prigušenja i njihov uticaj na smanjenje kružne struje.

3.2.1 Ubrzavanje porasta struje u koračnom motoru

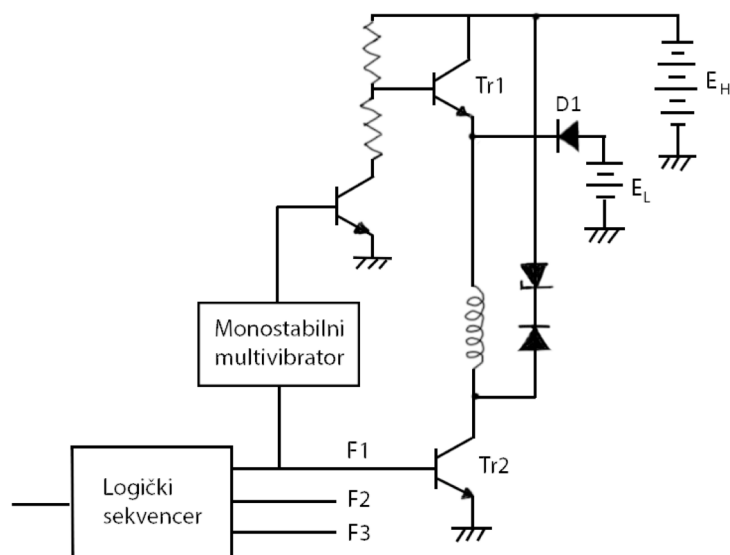
Uključenjem tranzistora s ciljem uzbuđenja faza napojna jedinica mora nadvladati efekat namotajnog induktiviteta prije rada sa nominalnom strujom, jer se induktivnost opire promjeni struje. Kako se frekvencija rada povećava, vrijeme porasta struje postaje veliko i rezultira smanjenim



Slika 3.8: Poređenje efekata različitih vrsta prigušenja [4]



Slika 3.9: Ubrzavanje porasta struje dodavanjem otpornika R_e u seriju s namotajem i povećanjem napona napajanja E [4]



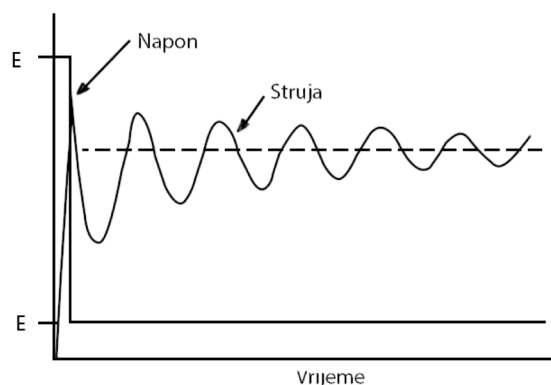
Slika 3.10: Ubrzavanje porasta struje uz korištenje drajvera dvostrukog napajanja [4]

momentom i sporom reakcijom. Postoji više metoda za skraćivanje vremena porasta i poboljšanja momentne karakteristike na velikim brzinama koje će biti navedene u nastavku [4].

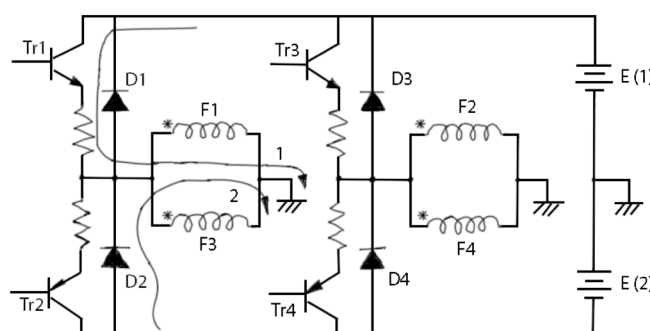
Dodavanje otpora u seriju s namotajem je najjeftiniji način, te je prikazan na slici 3.9. Napon napajanja E je odabran tako da sa dodanom otpornošću provede nominalnu struju u stacionarnom stanju sistema. Vremenska konstanta kruga je sada smanjena sa L/R_w na $L/(R_e + R_w)$. Iako je ova metoda najjednostavnija, nije i najpoželjnija usljed velike disipacije snage na serijskim otporima.

Drajver sa dvostrukim napajanjem se koristi kako bi se smanjila disipacija snage i poboljšao rad koračnog motora. Shema za jednu fazu je prikazana na slici 3.10.

Po dolasku komandnog pulsa za koračanje na ulaz logičkog sekvencera, signal visokog naponskog nivoa (logička jedinica) se šalje na jedan od izlaznih terminala, radi pobuđenja faznog namotaja. Nakon ovoga signal oba tranzistora $Tr1$ i $Tr2$ su uključeni, te se visoki napon E_H primjenjuje na namotaj faze. Dioda $D1$ je sada inverzno polarisana te izolira niže napajanje E_L od višeg E_H . Javlja se brz porast struje zbog visokog napona E_H . Vremenska konstanta



Slika 3.11: Karakteristika napona i struje prilikom korištenja drajvera dvostrukog napajanja [4]



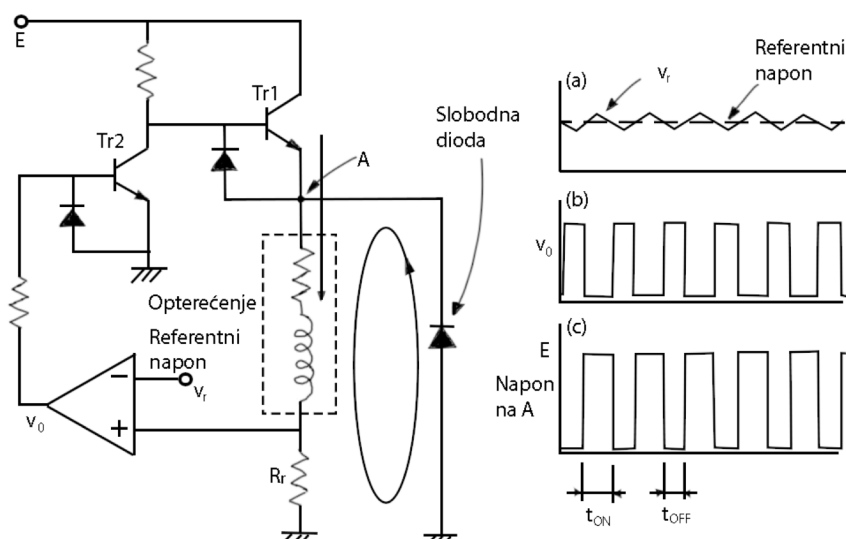
Slika 3.12: Mosni drajver za bipolarno upravljanje [4]

monostabilnog multivibratora je odabrana tako da se tranzistor Tr1 isključi kada struja namotaja malo pređe nominalnu struju. Time se visoko napajanje odvaja, te dioda D1 počne voditi i namotaj se napaja sa napajanja nižeg nivoa E_L . Tipičan izgled strujne karakteristike je prikazan na slici 3.11.

3.2.2 Mosni drajver

Efikasnost motora je veća kada su svi namotaji uvijek pobuđeni, te time proizvode efektivni moment, te je iz tog razloga je bipolarni drajver odličan. Kako su kod bipolarnog drajvera sva četiri namotaja uvijek uzbuđena on može postići poboljšanje momenta od 20% do 35% u odnosu na unipolarni drajver. Na slici 3.12 je prikazan jedan mosni krug za upravljanje bipolarnim motorima koji koristi dva izvora napajanja.

Ukoliko motor ima osam odvojenih terminala oni su spojeni kao na slici 3.12. Kolo funkcionira na poprilično jednostavan način. Samo jedan od tranzistora T1 i T2 može biti uključen u bilo kojem trenutku, isto vrijedi i za tranzistore T3 i T4. Kada je tranzistor T1 (T3) uključen struja teče kroz faze F1 i F3 (F2 i F4) u pozitivnom, dok ukoliko je tranzistor T2 (T4) uključen. Da bi se spriječilo oštećenje tranzistora u situaciji kada su oba tranzistora T1 i T2 (ili T3 i T4) uključeni u isto vrijeme dodani su otpornici između njih. Dioda služe za prigušivanje skokova napona koji se javljaju kada se bilo koji tranzistor isključi. Ako Tr1 vodi struja teče kroz kolo na način prikazan krivom 1 na slici 3.12, međutim nakon gašenja tranzistora T1 i uključenja T2 struja neće odmah promijeniti smjer, te će teći kroz diodu D2 kako je prikazano krivom 2.



Slika 3.13: PWM drajver [4]

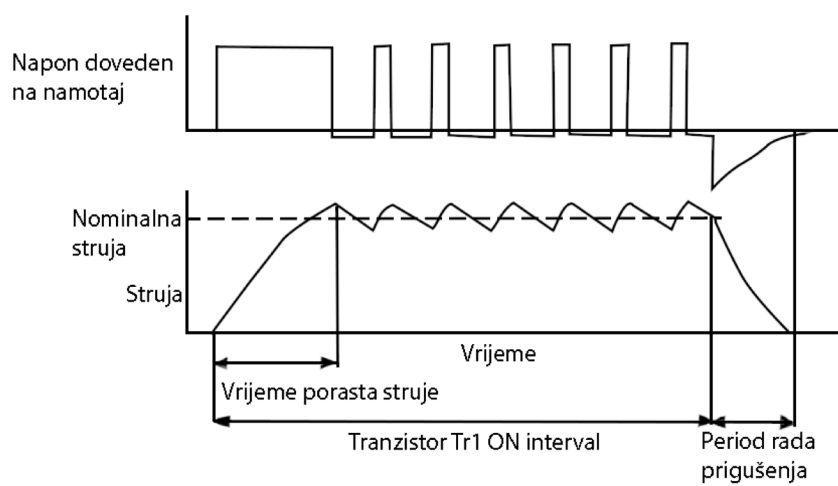
- (a) - Promjena referentnog napona v_r
- (b) - Promjena napona v_0 na izlazu iz operacionog pojačala
- (c) - Promjena napona u tački A

3.2.3 PWM drajver

PWM ("Pulse-width modulation" - širinsko-impulsno modulacioni) drajver je odličan drajver koji nudi brz porast struje sa malim gubitkom. Osnovna funkcija PWM drajvera je ilustrirana slikom 3.13. Ovdje induktivno opterećenje unutar pravougaonika sa isprekidanim linijama predstavlja uobičajni drajver. Napon na opterećenju R_r se poredi sa referentnim naponom v_r korištenjem amplitudnog pojačala sa visokim pojačanjem. Referentni napon predstavlja superpoziciju visokofrekventnog trokuta ili sinusoidne komponente i DC komponente kao što se vidi na slici 3.13 (a). Ako je DC komponenta referentnog signala i napona na opterećenju R_r skoro ista signal na izlazu operacionog pojačala će izgledati kao na slici 3.13 (b). Kako operaciono pojačalo ima visoko pojačanje izlazni napon oscilira između napona zasićenja i odsijecanja. Signal se invertuje tranzistorom Tr2 da bi se na bazu tranzistora Tr1. Kada je on uključen struja se vuče iz napajanja prema opterećenju, dok ukoliko je tranzistor Tr2 isključen struje teče preko slobodne diode prema opterećenju kako je prikazano konturom na slici 3.13. Kako se ovim drajverom napon odsijeca on se naziva i čoper (engl. odsijecajući) [4].

Prednosti PWM drajvera su korištenje jednog izvora napajanja, mali gubitak snage, te automatski prilagođen napon da upravlja sa zahtijevanom strujom. Na slici 3.14 je prikazana strujna i naponska karakteristika PWM drajvera. Prilikom uključivanja tranzistora Tr1 struja na namotaju se počne povećavati, ali je manja od referentne vrijednosti, te je tranzistor Tr2 uključen i visoki napon napajanja se dovodi na fazu motora. Kada struja poraste blizu opsega vrijednosti oko v_r/R_r , tranzistor Tr2 se počne uključivati/isključivati, te započinje chopper način rada. Nakon isključenja tranzistora Tr1 struja brzo opadne zbog prigušenja na paralelnoj slobodnoj diodi, te na dodanom otporu.

U PWM drajveru frekvencija choppera je određena vanjskim izvorom i njegova mana je postojanje električnog i akustičnog šuma izazvanog samim drajverom.



Slika 3.14: Strujna i naponska karakteristika PWM dražvera [4]

Poglavlje 4

Opis sistema za upravljanje dvoosnim manipulatorom

4.1 Korištena oprema

4.1.1 Popis (specifikacija) opreme

Korištene komponente i oprema za rad su sljedeće:

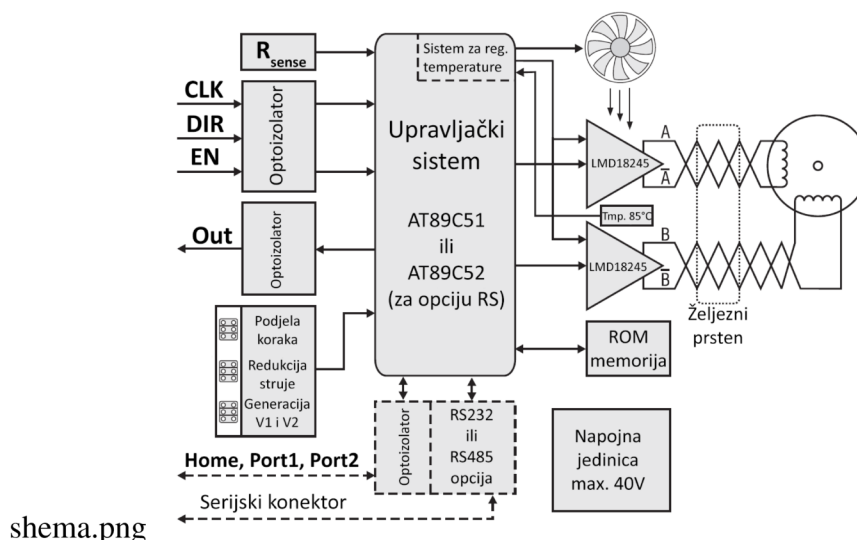
- PLC TM241CEC24R (Schneider Electric) kao kontrolni uređaj za motore (drajvere)
- Magelis HMISCU6A5 korisnički panel sa tasterima za upravljanje
- dva bipolarna dvofazna step motora sa korakom od 1.8° , KP56RM2-045 za Z osu i KP56RM2-027 za X osu
- dva drajvera za upravljanje koracima step motora, model SMC64
- napojne jedinice MDR-60-12 koje prouzvodí Mean Well
- dva fotoelektrična senzora OPB 981 koje proizvodi TT Electronics
- dva granična senzora (krajnje sklopke)
- osigurači 09056, 03386 koje proizvodi Legrand
- računar

4.1.2 Kratak opis najznačajnijih komponenti

Modicon M241 PLC - TM241CEC24R

TM241CEC24R je proizvodni broj Modicon M241 programabilnog logičkog kontrolera koga proizvodi Schneider Electric koji ima mogućnost ostvarivanja Ethernet, Modbus, Profibus DP i USB komunikacije.

Za svrhe ovoga rada naročito je značajno postojanje dva para brzih izlaza. Oni omogućuju slanje povorke impulsa, te signala koji označava smjer na oba motora sa frekvencijom koja doseže 200kHz. Pored toga mogućnost komunikacije putem Ethernet-a i MODBUS-a omogućuje povezivanje ovog podsistema sa ostatkom ETFCam sistema, te upravljanje motora sa višeg nivoa. Za programiranje ovog-a PLC-a koristi se softverski paket SoMachine koji omogućuje programiranje u ladder logici, putem funkcijskih blokova, te struktuiranog teksta.



Slika 4.1: Blok dijagram SMC64 drajvera[6]

Magelis HMISCU6A5

Magelis HMISCU6A5, koga proizvodi Schneider Electric, se sastoji od programabilnog logičkog kontrolera i korisničkog panela koji se može koristiti za davanje komandi bilo svom ili nekom drugom PLC-u. Sadrži ekran sa dijagonalom dužine 3.5 inča (8.89cm) i rezolucijom od 320 x 240 piksela, 2 brza i 14 digitalnih ulaza, 2 brza i 8 digitalnih izlaza, te mogućnost komunikacije putem Ethernet-a, Modbus-a, CANopen-a i USB-a.

Programski paketi SoMachine i Vijeo Designer omogućavaju jednostavno povezivanje ovog HMI-a sa glavnim kontrolerom M241 putem Ethernet-a. Usljed toga možemo ga jednostavno koristiti u kombinaciji sa M241 PLC-om kao panel za manualno upravljanje sistemom, te zadavanje komandi PLC-u.

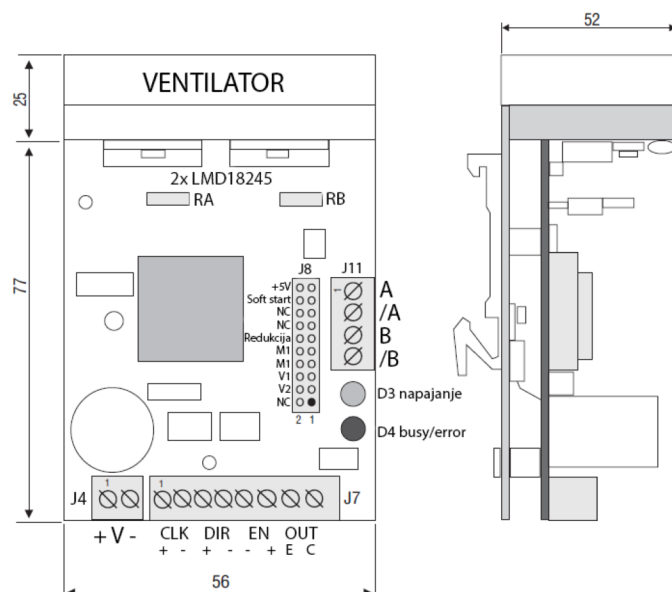
SMC64 drajver

SMC64 je minijturni kontroler dizajniran da radi sa dvofaznim koračnim motorom sa bipolarnim namotajem (sa 8 ili 4 žice) ili sa unipolarnim namotajem (sa 6 žica) korišten kao bipolarni. Omogućava kontrolu sa punim, polu-korakom, te 1/4 ili 1/8 koraka, te održava konstantnu struju kroz namotaj motora koja može biti podešena dodavanjem dodatnih otpora na predviđeno mjesto drajvera između vrijednosti od 1A do 3.5A.

Kao takav predstavlja pogodan drajver koji se ujedno ponaša i kao logički sekvencer, te ga možemo direktno spojiti sa motorom, kao što je prikazano na slici 3.1 (a). Na njegov ulaze dovodimo povorku impulsa koja daje signal za pomjeranje, digitalni signal koji predstavlja smjer, te signal za uključenje/isključenje motora. Oni su dovedeni sa PLC-a na CLK, DIR i EN ulaze drajvera. Međutim prije njihovog spajanja potrebno je proračunati otpore koje je potrebno dodati da bi drajver radio sa predviđenom strujom i naponom što je prikazano u narednom poglavlju.

Prije korištenja drajvera potrebno je obezbjediti odgovarajuće napajanje (max 40V) i strujnu efikasnost. Napojna mora također da upije energiju kočenja koju obezbjeđuju početni kondenzatori kapacitetom od bar 4700μF. Također za minimizaciju šuma na vezi između drajvera i motora preporučuje se uvrtnje kablova koji vode signal do faza motora u parovima (A sa /A fazom, B sa /B fazom). Na slici 4.1 prikazan je blok dijagram ovoga drajvera, dok je na slici 4.2 prikazan izgled konektora.

Oznake na slici 4.2 odgovaraju:



Slika 4.2: Izgled konektora drajvera[6]

- J1 napajanje
 1. plus napojne jedinice (+)
 2. uzmljenje napajanja (-)
- J2 kontrolni signali
 1. anoda upravljačkog signala (CLK)
 2. katoda
 3. anoda direkcionog signala (DIR)
 4. katoda
 5. anoda omogućavajućeg signala (EN)
 6. katoda
- J8 signali sa opcijama
 - 1-2 NC - ne spajati
 - 3-4 V2 - V2 odabir brzine
 - 5-6 V1 - V1 odabir brzine
 - 7-8 M2 podjela koraka
 - 9-10 M1 podjela koraka
 - 11-12 zaustavljanje redukcije struje
 - 13-14 NC - ne spajati
 - 15-16 An - Ain aktivacija ulaza
 - 17-18 Soft start
 - 19- izlaz od 5V

Nominalna struja po fazi	1.0	1.5	1.7	2.0	2.5	3.0	3.5
Dotatne otpornosti R_A i R_B	-	30	24	20	15	12	10

Tabela 4.1: Tabela za odabir nominalne struje izborom otpornika R_A i R_B [6]

- J11 Izlaz za motor

1. A faza
2. /A faza
3. B faza
4. /B faza

Odabir rezolucije se vrši kratkospajanjem M1 i/ili M2 i to na način da: oba budu kratkospajena za upravljanje sa punim korakom korakom, samo M2 za upravljanje sa pulukorakom, samo M1 za 1/4 koraka, te nijedan za 1/8 koraka, dok se izbor nominalne struje na izlazu vrši biranjem otpornika prema tabeli 4.1.

Drajverom je moguće upravljati na više načina. U osnovnom načinu rada eksterni koračni signal se dovodi na CLK+ i CLK- ulaze, te se na svaku uzlaznu ivicu signala motor pomjera za jedan korak. Prilikom rada sa odabranom brzinom drajver omogućava izbor jedne od ponuđenih frekvencija koji se biraju sa V1 i V2 izvodima. U ovome slučaju nije potrebno dovoditi eksterni CLK signal, drajver će ga sam reproducirati. Drajver možemo također direktno spojiti sa računarom putem serijske konekcije.

4.1.3 Proračun dodatnih korištenih komponenti

Drajver za upravljanje motorom je predviđen za rad sa naponom od 5V na ulazu. Kako je izlaz iz PLC-a podešen za davanje napona od 25V, potrebno je dodati otpornosti na CLK (clock), DIR (direction) i EN (enable) ulaze drajvera kako bi se spustio napon na ulazu (odnosno smanjila struja), te time spriječio kvar drajvera. U nastavku je njihov proračun.

Za ulaze drajvera EN iskorišteni su otpornici prema shemi sa slike 4.3 i sljedećem proračunu:

$$I = \frac{5V - 1.5V}{330\Omega} = 10.61mA \quad (4.1)$$

$$R = \frac{25V - 5V}{10.61mA} = 1.79096k\Omega \quad (4.2)$$

Uzeta je vrijednost otpornika od 2k Ω .

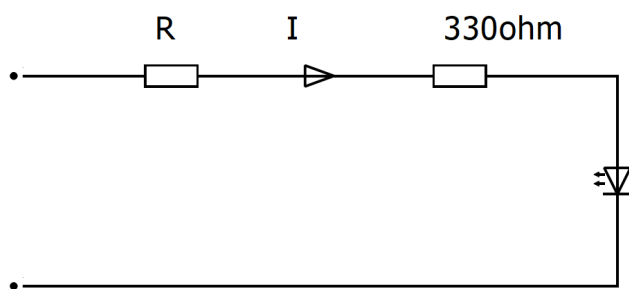
Za ulaze drajvera CLK i DIR iskorišteni su otpornici prema shemi sa slike 4.4 i sljedećem proračunu:

$$I = \frac{5V - 1.5V}{120\Omega + 120\Omega} = 14.6mA \quad (4.3)$$

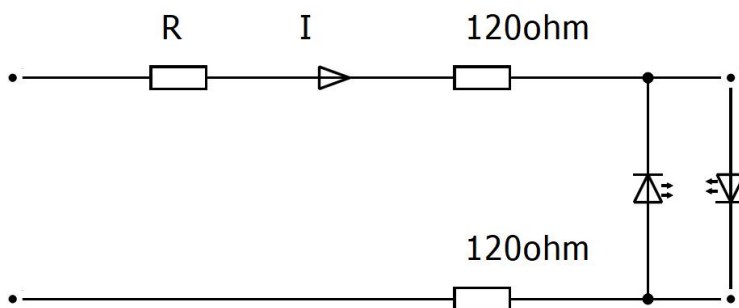
$$R = \frac{24V - 5V}{14.6mA} = 1.30136k\Omega \quad (4.4)$$

Uzete su vrijednosti otpornika 1.2k Ω .

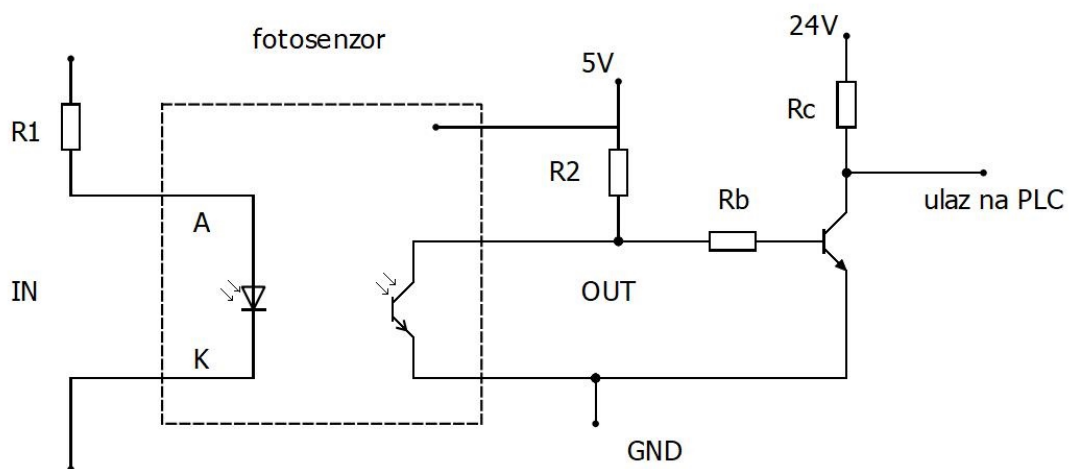
Također je bilo potrebno napraviti proračun za fotoelektrične senzore pri odabiru otpornika za ograničavanje struja diode i tranzistora, kao i za odabir otpornika pojačala izlaznog kruga



Slika 4.3: Shema za proračun otpornika na EN ulazima drajvera



Slika 4.4: Shema za proračun otpornika na ulazima drajvera CLK i DIR



Slika 4.5: Shema spajanja fotoelektričnog senzora OPB 981

senzora čija je uloga dizanje izlaznog napona senzora sa 5V na 24V za adaptaciju na digitalni ulaz PLC-a. Slikom 4.5 prikazana je shema kruga fotosenzora sa pojačalom.

Vrijednost otpornika R1 biramo na osnovu ograničenja struje diode od 18mA i napona diode 1.7V:

$$R1 = \frac{5V - 1.7V}{18mA} = 183.3\Omega \quad (4.5)$$

Odabrana je vrijednost $R1 = 180\Omega$.

Vrijednost otpornika R2 biramo prema ograničenju kolektorske struje tranzistora od 12mA i pada napona tranzistora 0.2V u zasićenju:

$$R2 = \frac{5V - 0.2V}{12mA} = 400\Omega \quad (4.6)$$

Odabrana je vrijednost $R2 = 400\Omega$.

Za vrijednost kolektorske struje dodatnog tranzistora usvojimo 10mA, te na osnovu toga možemo proračunati vrijednost kolektorskog otpornika Rc:

$$Ic * Rc = 24V - Uce \quad (4.7)$$

$$Rc = \frac{24V - 0.2V}{10mA} = 2.38k\Omega \quad (4.8)$$

Odabrana je vrijednost $Rc = 2.2k\Omega$.

Sada za $\beta = 150$, vrijednost baznog otpornika Rb možemo proračunati na sljedeći način:

$$Ib = \frac{Ic}{\beta} = 66.7mA \quad (4.9)$$

$$Ib * Rb = 5V - Ube \quad (4.10)$$

$$Rb = \frac{5V - 0.7V}{66.7mA} = 64.5k\Omega \quad (4.11)$$

Odabrana je vrijednost $Rb = 56k\Omega$.

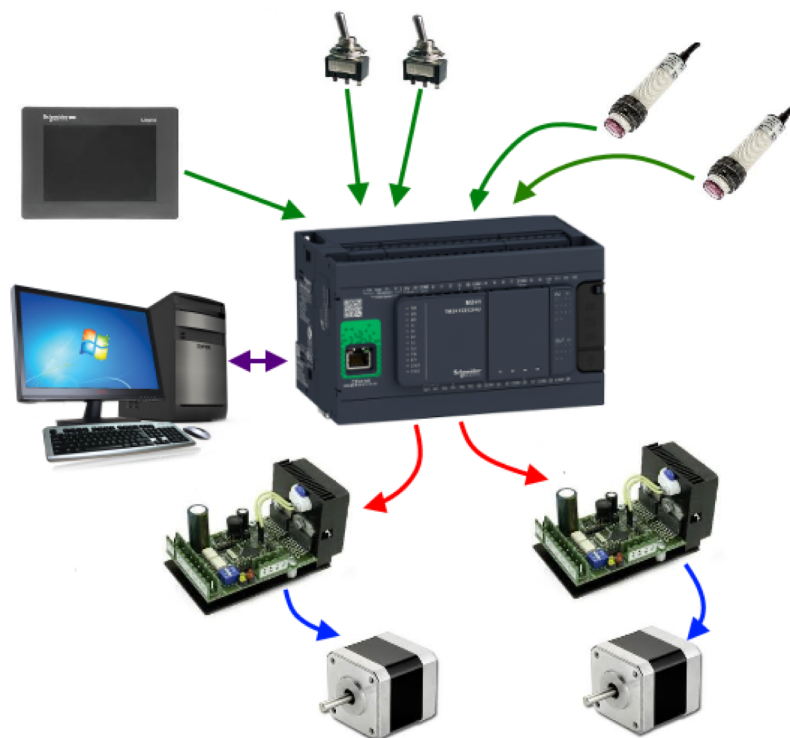
4.2 Principijelna shema

Na slici 4.6 je prikazana principijelna shema sistema koja nudi uvid u povezanost pojedinih elemenata sistema. Kompletan i detaljan shema je ponuđena u prilogu.

4.3 Ormar unutar kojega su smještene komponente

Radi realizacije projekta bilo je potrebno osmisliti ormar unutar kojega će se smjestiti sve komponente. Ovakav ormar treba omogućiti napajanje svih komponenti, njihovo postavljanje i povezivanje, te sačuvati komponente usljed neželjenog djelovanja (putem osigurača). Na vratima ormara koja su prikazana na slici 4.7 postavljeni su prekidači za uključenje pojedinih dijelova sistema, te određene indikacije. Oznake prikazane na slici 4.7 odgovaraju:

- H0 - Rezervna indikacija

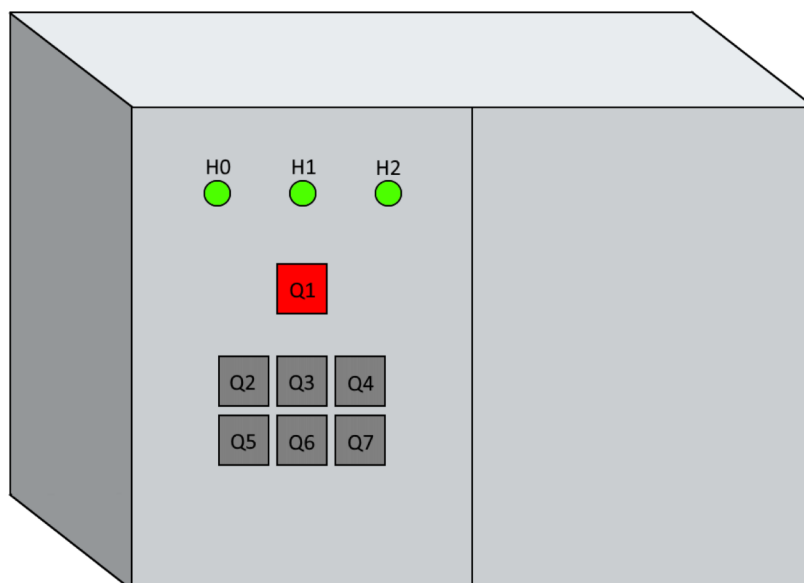


Slika 4.6: Principijelna shema sistema

- H1 - Indikacija eksternog napajanja
- H2 - Indikacija internog napajanja
- Q1 - Glavni razvod
- Q2 - Napojna 1 - DC 24V
- Q3 - Napojna 2 - DC 24V
- Q4 - Napojna 3 - DC 12V
- Q5 - Napojna 4 - DC 5V
- Q6 - Rezerva
- Q7 - Napajanje PLC M241

U unutrašnjosti ormara smještena su sve potrebne komponente za upravljanje dvoosnim manipulatorom kao što možemo vidjeti na slici 4.8. Oznake u unutrašnjosti ormara na slici 4.8 odgovaraju:

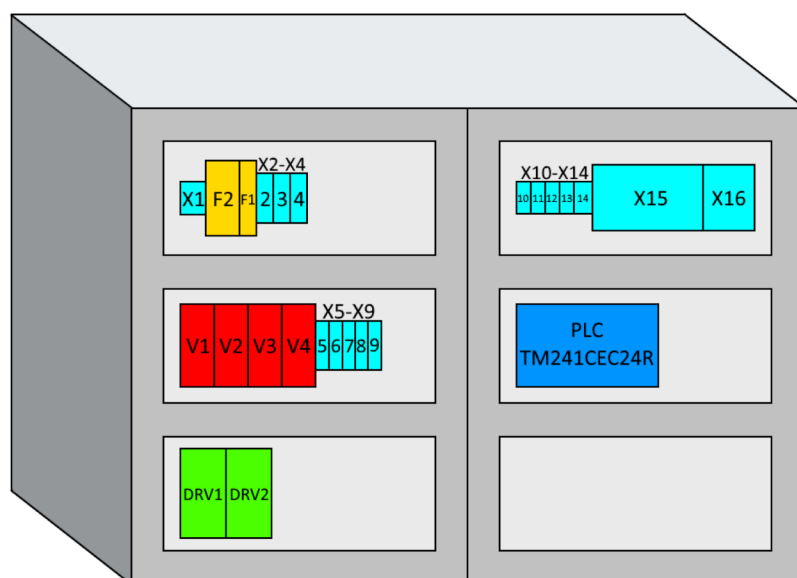
- X1-X16 - Ulazna stezaljka
- F1 - Osigurač
- F2 - FID sklopka
- V1 - Napojna 1 - DC 24V



Slika 4.7: Izgled vrata ormara

- V2 - Napojna 2 - DC 24V
- V3 - Napojna 3 - DC 12V
- V4 - Napojna 4 - DC 5V
- DRV1 - Step driver - X osa
- DRV2 - Step driver - Z osa

Svaka od navedenih komponenti spomenutih kroz oznake ormara, te način njenog spajanja sa ostatkom sistema može se naći u detaljnoj shemi ponuđenoj u prilogu rada.



Slika 4.8: Izgled unutrašnjosti ormara

Poglavlje 5

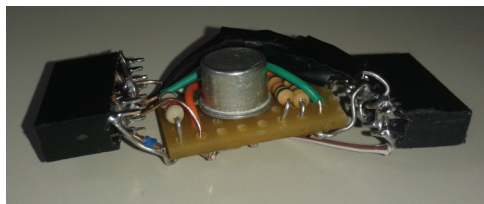
Realizacija sistema za upravljanje dvoosnim manipulatorom

Ovo poglavlje se bavi procesom realizacije sistema i upravljanjem dvoosnog manipulatora. Rad PLC-a je konfigurisan unutar SoMachine okruženja uz korištenje struktuiranog teksta (ST) i funkcijskih blokovskih dijagrama (FBD). Sam proces se razvoja sistema se odvijao u nekoliko faza:

- proračun shemi prilagođenja i pravljenje odgovarajućih kola,
- pravljenje namjenskog panela sa tasterima, te povezivanje sa PLC-om, zajedno sa ostalim potrebnim komponentama koje uključuju senzore i drajvere,
- analiza i testiranje rada pojedinih blokova u SoMachine-u i njihovog utjecaja na ponašanje motora,
- implementacija blokova i realizacija algoritma za upravljanje putem tastera u SoMachine-u,
- testiranje rada dijela sistema,
- uspostavljanje serijske komunikacije između računara i PLC-a koristeći MODBUS protokol,
- pravljenje aplikacije u C-u koja omogućuje slanje i primanje poruka na/sa PLC (-a) putem serijske konekcije,
- razvoj protokola za slanje koji omogućuje pokretanje komandi unutar SoMachine-a za upravljanje motorima i osigurava siguran rad motora i pravilno funkcionisanje upravljačkog algoritma,
- proširenje SoMachine koda da omogući upravljanjem motora putem serijske komunikacije.

Kroz ovo poglavlje biti će opisan čitav proces realizacije prije implementacije serijske komunikacije koji će se obraditi u zasebnom poglavlju.

Prije početka razvijanja samog programskog rješenja bilo je prilagoditi određene uređaje da bi se mogli povezati sa ulazima/izlazima PLC-a. Sam proračun komponenti je obrađen u ranijim poglavljima, te su otpornosti potrebne za povezivanje sa drajverima uklopljene unutar ožičenja, dok su za povezivanje sa fotoelektričnim senzorima napravljeni posebni moduli. Poseban izazov za njihovu realizaciju predstavljao je zahtjev za kompaktnost, te njihova konačna realizacija je prikazana slikama 5.1 i 5.2.



Slika 5.1: Modul za prilagođenje fotoelektričnog senzora - gornja strana

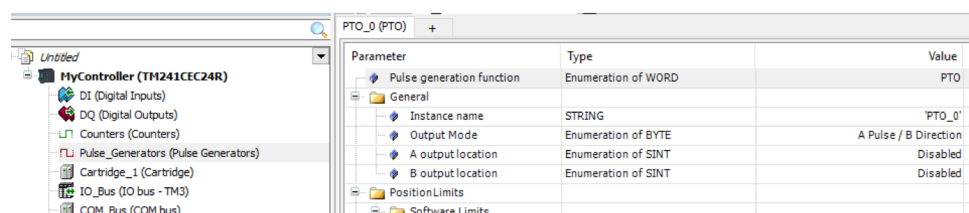


Slika 5.2: Modul za prilagođenje fotoelektričnog senzora - donja strana

Također je napravljen i fizički panel sa tasterima koji je prikazan na slici 5.3. Zatim su svi elementi sistema spojeni po shemi koja ja ponuđena u prilogu, te je započeta izrada programa za upravljanje. U nastavku će biti opisan taj proces počevši od blokova korištenih, te do implementacije kompletnog algoritma.



Slika 5.3: Panel sa tasterima za upravljanje



Slika 5.4: Podešavanje generatora pulsa

5.1 Generatori pulsa i SoMachine blokovi za njihovo upravljanje

Za generisanje potrebne povorke impulsa upravljačkog signala i smjera moguće je koristiti generatore pulsa koje SoMachine okruženje nudi. Za njih su razvijeni posebni PTO ('Pulse Train Output' - izlaz povorke impulsa) blokovi koji olakšavaju sam proces. Ti blokovi pružaju mogućnost traženja i postavljanja početne pozicije na osnovu očitavanja senzora, pomjeranja za relativnu udaljenost ili na tačnu poziciju, podešavanje brzine, te pamćenje i očitavanje trenutne pozicije motora u odnosu na početnu.

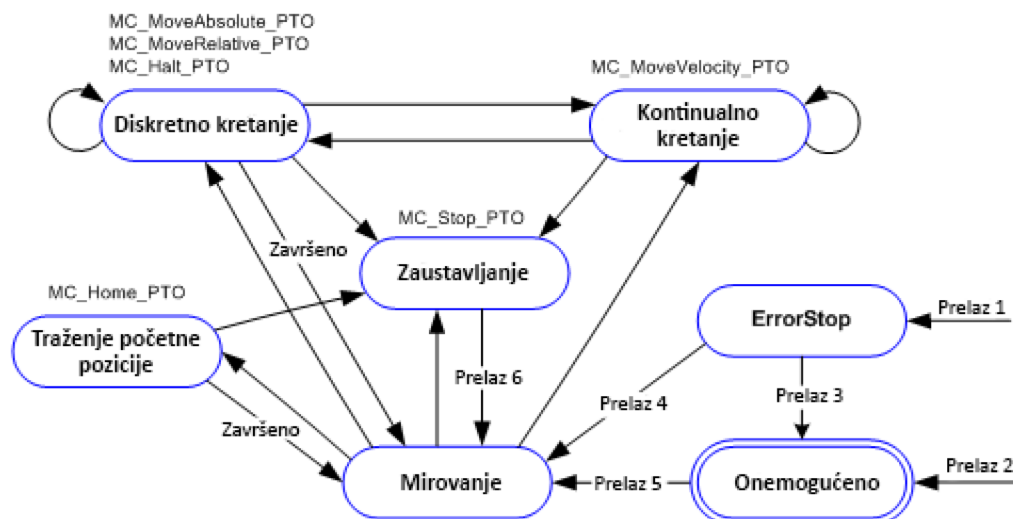
Prije korištenja ovih blokova potrebno je prvo dodati generator pulsa, odabrati da bude PTO tipa, te podesiti da izlaz bude tipa 'pulse/direction'. Sam generator pulsa se veže za par brzih izlaza, tako da je za potrebe ovoga projekta bio potreban PLC koji sadrži dva takva para. Na slici 5.4 je prikazan izgled podešenog generatora pulsa.

Nakon prethodne konfiguracije moguće je korištenje PTO blokova i u nastavku će ovako konfigurisan generator pulsa biti referiran kao osa. Najznačajniji blokovi koji služe za upravljanje kretanjem su:

- MC_Power_PTO blok - neophodan za funkcionisanje svih ostalih blokova za upravljanje kretanja ose,
- MC_Stop_PTO blok - zaustavlja kretanje ose, te vraća generator pulsa u stanje mirovanja,
- MC_Reset_PTO blok - resetovanje ose u slučaju greške (error-a),
- MC_Home_PTO blok - nalazi početnu poziciju ose na osnovu očitavanja senzora,
- MC_MoveAbsolute_PTO blok - pomjera osu na zadanu poziciju u odnosu na referentnu,
- MC_MoveRelative_PTO blok - pomjera osu za datu udaljenost sa trenutne pozicije,
- MC_MoveVelocity_PTO blok - vrši pomjeranje ose sa datom brzinom.

Na slici 5.5 su prikazana stanja ose, te kada se koje funkcije mogu koristiti u odnosu na stanje. U nastavku stanja će biti detaljnije objašnjenja zajedno sa pojedinim blokovima.

Svi blokovi koji vrše pomjeranje ose ostvaruju trapezni profil brzine prilikom kretanja do željene pozicije ili podešavanja brzine. Takav profil je prikazan na slici 5.6. Moguće je postavljanje ubrzanja ili usporavanja bloka u Hz/ms ili njihovog trajanja u ms. Također moguće je podesiti JerkRatio parametar koji rampu ubrzavanja/usporenja pretvara u profil S krive. Sam parametar predstavlja procenat vremena u kojem se mijenja ubrzanje/usporenje. Također, da bi ijedan od ovih blokova radio mora se EN ulaz drajvera postaviti na logičku jedinicu odnosno 'TRUE'.



Slika 5.5: Stanja generatora pulsa

Prelaz 1 - iz bilo kojeg stanja kada se detektuje greška (error)

Prelaz 2 - iz bilo kojeg stanja osim ErrorStop-a i kada je MC_Power_PTO.Status = FALSE

Prelaz 3 - MC_Reset_PTO.Done = TRUE i MC_Power_PTO.Status = FALSE

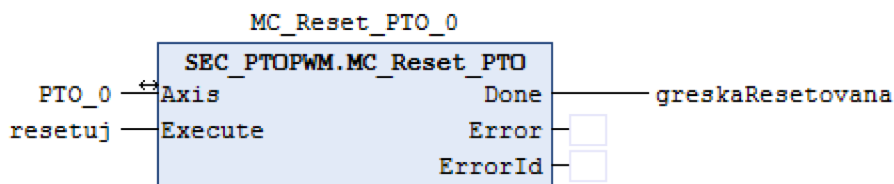
Prelaz 4 - MC_Reset_PTO.Done = TRUE i MC_Power_PTO.Status = TRUE

Prelaz 5 - MC_Power_PTO.Status = TRUE

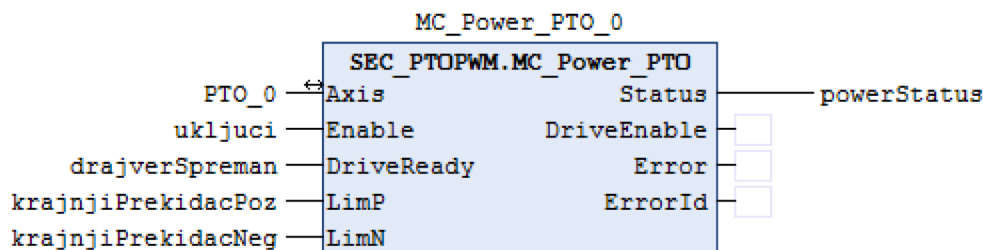
Prelaz 6 - MC_Stop_PTO.Done = TRUE i MC_Stop_PTO.Execute = FALSE



Slika 5.6: Trapezni profil brzine



Slika 5.7: Izgled MC_Reset_PTO bloka



Slika 5.8: Izgled MC_Power_PTO bloka

Pored ovih blokova značajan je i blok za čitanje trenutne pozicije. Jednostavan je za podesiti, te dokle god prima logičku jedinicu ('TRUE') na ulazu davati će trenutnu poziciju na izlazu. Iz tog razloga on neće biti detaljnije objašnjen.

5.1.1 ErrorStop i onemogućeno stanje, te prelazak u mirovanje

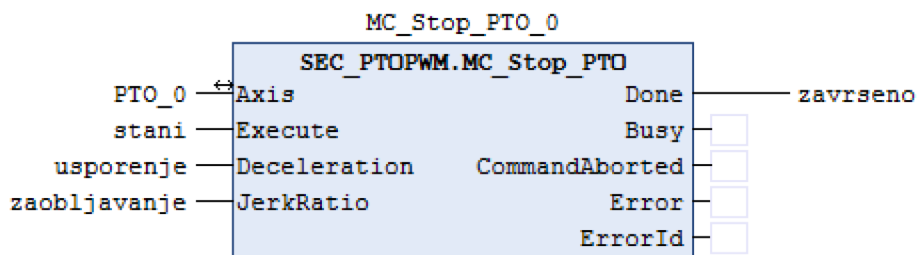
Kada se desi greška prilikom rada bilo kojeg PTO bloka osa se dovodi u ErrorStop stanje. Kada je osa u stanju greške nijedan drugi blok se ne može pozvati dok se greška ne riješi. Onemogućeno stanje se javlja kada god se isključi MC_Power_PTO blok, te se nakon njega također ne može pokrenuti nijedan blok za kretanje.

MC_Reset_PTO blok

Blok služi za izlazak iz ErrorStop stanja i njegov izgled je prikazan na slici 5.7. Potrebno je samo u trenutku detekcije greške postaviti 'Execute' ulaz u blok na 'TRUE'. Kada se greška ukloni izlaz 'Done' će biti postavljen na 'TRUE'.

MC_Power_PTO blok

Blok MC_Power_PTO je prikazan na slici 5.8 i neophodan je za rad svih drugih blokova kretanja. Kada se njegov status promjeni na 'FALSE' svako kretanje se obustavlja. Postavljanje ulaza 'Enable' na 'TRUE' omogućujemo kretanje, te on uvijek treba biti uključen. Na ulaz 'DriveReady' je moguće spojiti izlaz drajvera koji govori da je spreman, ovo omogućuje automatski prestanak slanja komandi u situaciji kada drajver nije u stanju izvršavanja pomjeranja. Također, na ulaze LimP i LimN se mogu povezati očitavanja krajnjih prekidača, te time automatski zaustaviti nedozvoljeno kretanje. Kada se 'Status' ovoga bloka tek promijeni na 'TRUE' osa prelazi u stajanje mirovanja ako ne postoji neka greška.



Slika 5.9: Izgled MC_Stop_PTO bloka

5.1.2 Zaustavljanje i stanje mirovanja

Stanje mirovanja je osnovno stanje i poželjno je, ukoliko se ne vrši kretanje, preći u njega, kako bi svaka funkcija za kretanje mogla biti pozvana. Stanje zaustavljanja vrši prelaz iz trenutnog stanja u stanje mirovanja i ono se aktivira sa MC_Stop_PTO blokom.

MC_Stop_PTO blok

Ovaj blok služi za zaustavljanje kretanja ose i prelazak u stanje mirovanja. Njegov izgled vidimo na slici 5.9, a kada ga želimo aktivirati samo je potrebno postaviti 'Enable' ulaz na 'TRUE'. Promjenom ulaza 'Deceleration' i 'JerkRatio' mijenjamo usporenje i profil brzine koji želimo postići na prethodno objašnjen način. Nakon zaustavljanja, 'Done' izlaz dobiva vrijednost 'TRUE' i osa prelazi u stanje mirovanja.

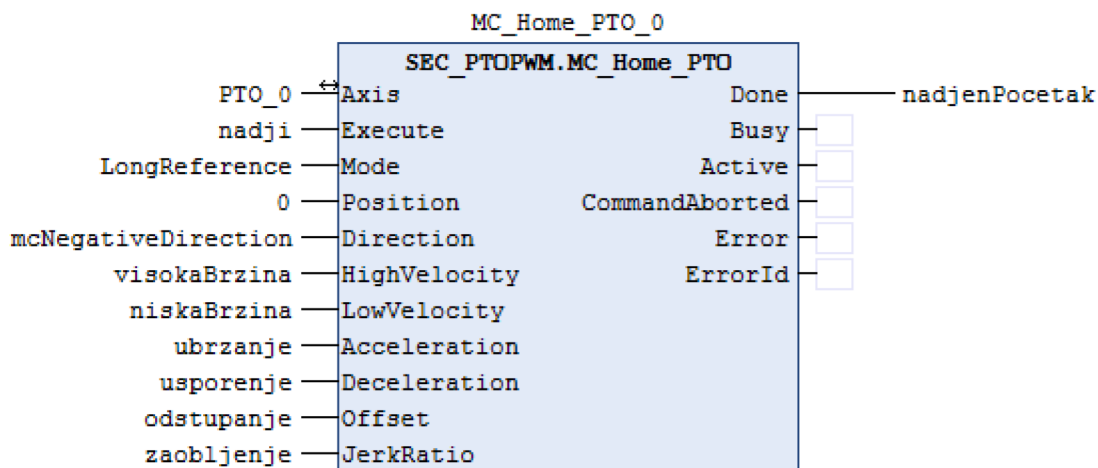
5.1.3 Traženje početne pozicije

Kako bi se ispravno izvršavale sve funkcije kretanja potrebno je pronalaženje početne pozicije za svaku osu. Također ukoliko smo prilikom konfiguracije generatora pulsa postavili gornje i donje software-sko ograničenje oni će se poštovati tek nakon pronalaženja početne pozicije. Prije korištenja bloka MC_Home_PTO koji ima tu funkciju potrebno je konfigurirati referentni i indeks ulaz koji je opcionalan u zavisnosti od načina traženja početne pozicije. Blok se može pozivati i nakon poočetnog postavljanja s ciljem uklanjanja greške nastale gubljenjem koraka prilikom rada motora.

MC_Home_PTO blok

Na slici 5.10 vidimo izgled bloka za nalaženje početne pozicije. Prije njegovog pokretanja potrebno je podesiti određene parametre. Ulaz 'Mode' označava način rada samog bloka na koji možemo dovesti slijedeće enumeracije:

- PositionSetting
- LongReference
- LongReferenceAndIndex
- ShortReference_Reversal
- ShortReference_NoReversal



Slika 5.10: Izgled MC_Home_PTO bloka

- ShortReferenceAndIndex_Outside
- ShortReferenceAndIndex_Inside

Prva tri načina rada su najvažnija, dok ostali predstavljaju blage varijacije u odnosu na njih, te će oni biti detaljnije objašnjeni u nastavku. Parametar 'Position' predstavlja vrijednost koja će biti postavljena prilikom zadavanja početne pozicije, dok je 'Direction' smjer u kome se traži početna pozicija. 'HighVelocity' i 'LowVelocity' parametri predstavljaju vrijednosti visoke i niske brzine, dok je 'Offset' udaljenost odstupanja za koju će se osa pomjeriti nakon pronalaska početne pozicije. Ostali parametri služe za postavku trapezoidnog profila brzine.

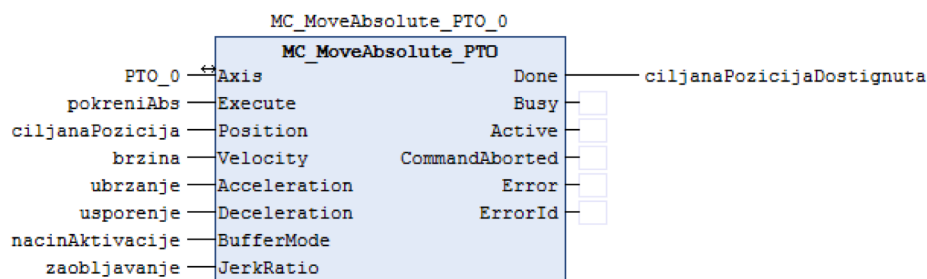
'PositionSetting' način rada postavlja trenutnu poziciju kao početnu i dodjeljuje joj vrijednost 'Position'. Ovaj način rada ne vrši pomjeranje ose.

'LongReference' način rada započinje traženje početne pozicije u podešenom smjeru sa odabranom visokom brzinom. Na uzlaznu ivicu referentnog ulaza on se zaustavlja, te se počne kretati u suprotnom smjeru nižom brzinom. Nakon dolaska silazne ivice postavlja tu poziciju kao početnu, te onda ili stane ili nastavi kretanje dok ne dosegne odstupanje.

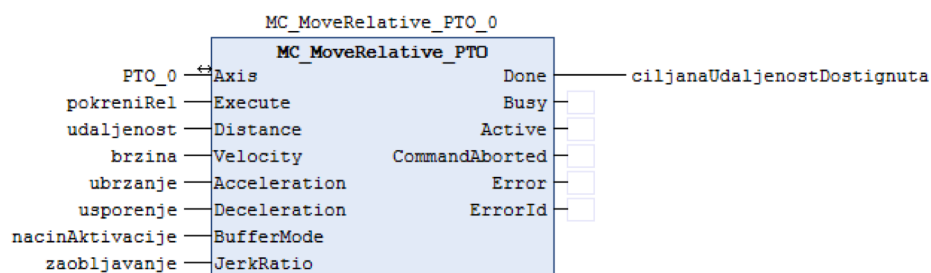
'LongReferenceAndIndex' način rada obavlja prvi dio traženja na isti način kao i prethodni 'LongReference' način. Međutim tokom kretanja niskom brzinom u suprotnom smjeru ne postavlja početnu poziciju odmah nakon dolaska silazne ivice referentnog signala, već nakon toga i dolaska uzlazne ivice indeks signala. Na indeks ulaz se obično vezuje senzor koji se više puta okida tokom pomjeranja ose u predviđenom opsegu i on obično predstavlja neku podjelu. I tokom ovoga načina rada nakon nalaženja početne pozicije kretanje se nastavlja dok se ne dostigne odstupanje. Blok se aktivira uključanjem 'Execute' ulaza, te na izlazu 'Done' daje vrijednost 'TRUE' kada završi. Nakon njegovog rada bitno je pokrenuti MC_Stop_PTO blok, kako bi se mogle vršiti druge funkcije pomjeranja nakon njega.

5.1.4 Diskretno i kontinualno kretanje

U stanju diskretnog kretanja pokreću se blokovi koji pomjeraju motore za zadanu udaljenost ili na željenu poziciju u odnosu na referentnu, dok u stanju kontinualnog kretanja motori se



Slika 5.11: Izgled MC_MoveAbsolute_PTO bloka



Slika 5.12: Izgled MC_MoveRelative_PTO bloka

pomjeraju željenom brzinom. Blokovi koji se aktiviraju unutar ova dva stanja mogu se pozivati bez povratka u stanje mirovanja. Kako su ulazi i izlazi blokova, te načini pokretanja dosta slični biti će obrađeni u istom dijelu.

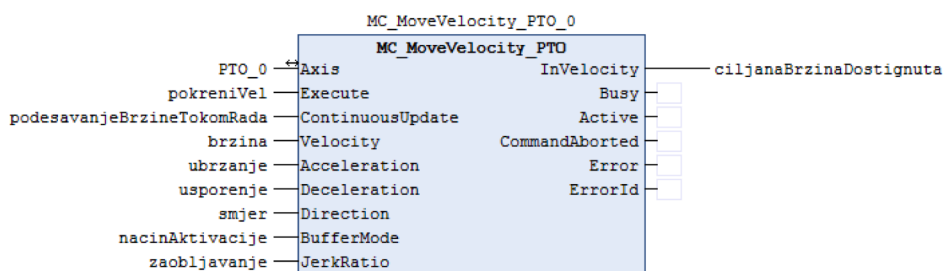
MC_MoveAbsolute_PTO, MC_MoveRelative_PTO i MC_MoveVelocity_PTO blokovi

MC_MoveAbsolute_PTO blok služi za pomjeranje ose na tačno zadanu poziciju i njegov izgled je prikazan slikom 5.11. Ciljana pozicija se postavlja na 'Position' ulazu i kada ona biva dostignuta 'Done' izlaz vrati logičku jedinicu ('TRUE').

MC_MoveRelative_PTO vrši za pomjeranje ose za datu udaljenost u odnosu na trenutnu i prikazan je slikom 5.12. Udaljenost se zadaje na 'Distance' ulazu, gdje predznak označava smjer, te kada se pomjeranje završi izlaz 'Done' također vrati logičku jedinicu ('TRUE').

MC_MoveVelocity_PTO pomjera osu zadanom brzinom, te izgled bloka vidimo na slici 5.13. Brzina se zadaje ulazom 'Velocity', dok se smjer dovodi na ulaz 'Direction'. Nakon dostizanja tražene brzine 'InVelocity' vrati logičku jedinicu ('TRUE').

Svaki od blokova sadrži parametre za podešavanje profila brzine, koji su prethodno objašnjeni.



Slika 5.13: Izgled MC_MoveVelocity_PTO bloka

Također svaki od njih se aktivira uzlaznom ivicom na 'Execute', ali se tokom njihovog rada dolaskom nove uzlazne ivice na 'Execute' ulaz bilo kojeg od ova tri bloka, uključujući i trenutno aktivni blok, može preći na rad novog bloka (ili istog bloka sa novom komandom). Način prelaska iz rada jednog bloka u drugi je definisan sa 'BufferMode' parametrom koji može imati naredne enumeracije:

- mcAborting - odmah pokreće rad novoga bloka, prethodni blok prestaje sa izvršavanjem,
- mcBuffered - novi blok se pokreće nakon što prethodni blok završi, ukoliko je prethodni blok bio diskretan novi blok počinje sa nultom brzinom,
- mcBlendingPrevious - novi blok počinje nakon što prethodni dosegne željenu poziciju ili brzinu, ukoliko je prethodni blok bio diskretan novi blok počinje sa brzinom prethodnog bloka (prethodni blok ne usporava prije dolaska do željene pozicije),
- seTrigger - rad novog bloka počinje nakon što se detektuje događaj na 'Probe' ulazu, a on se postavlja u konfiguraciji generatora pulsa,
- seBufferedDelay - novi blok se pokreće nakon što završi prethodni i postavljeno kašnjenje završi. Ono se može podesiti u MC_WriteParameter_PTO bloku sa 'ParameterNumber' postavljenim na 1000.

5.2 Implementacija PTO blokova i realizacija programa upravljanja

Realizacija programa upravljanja podrazumijevala je implementaciju i testiranje neophodnih blokova u funkcionalnim blokovskim dijagramima, da bi se unutar struktuiranog teksta mogli aktivirati za ostvarivanje određene funkcionalnosti. Sam SoMachine kod je konstruiran kao niz odvojenih slučajeva koji se aktiviraju kada se neki taster pritisne. Prvo se provjerava stanje hitnog stopa, te se tek ako on nije aktivan omogućava pozivanje nekih blokova za kretanje. Unutar slučajeva se razmatraju i stanja kontaktnih senzora, radi onemogućavanje rada motora u nedozvoljenim pozicijama. Za pamćenje pozicija korišten je niz, te su slučajevi koji su se bavili snimanjem i reprodukcijom sekvence pristupali i dodatnim varijablama za manipulaciju niza. U sklopu ovoga dijela rada uspješno je realizovano upravljanje pomoću tastera, a uputstvo za rad sa tasterima se nalazi u prilogu. Bitno je napomenuti da su vrijednosti horizontalne brzine podešene na 3000 Hz, dok vertikalne na 2000 Hz. Iako su brzine ovih motora testirane i do 10000 Hz, potrebno je napomenuti da se već za dvostruko povećanje brzine dešavaju značajni gubici koraka. Trajanje ubrzanja i usporenja je postavljeno na 500 ms. Testiranjem motora sa većim i manjim ubrzanjima nije pokazivalo dobre rezultate.

Poglavlje 6

Upravljanje sistema dvoosnog manipulatora korištenjem serijske komunikacije i MODBUS protokola

Ovo poglavlje opisuje proces uspostavljanja serijske komunikacije između računara i PLC-a, te korištenje MODBUS protokola za čitanje i pisanje podataka pohranjenih na PLC-u. Na računaru je razvijen program u C-u unutar Bloodshed Dev-C++ okruženja koji je imao mogućnosti uspostavljanja serijske komunikacije, te čitanja/pisanja podataka sa PLC-a. Podaci koje je moguće mijenjati označeni su kao markeri. Nadogradnjom SoMachine koda i interpretacijom podataka unutar markera omogućeno je upravljanje dvoosnog manipulatora pomoću računara. Kako je SoMachine kod nadograđen, program koji je pisan u C-u je mogao vršiti iste funkcionalnosti za koje je prethodno korišten panel kao i neke nove koje nisu prethodno bile dostupne, bez narušavanja prethodnih funkcionalnosti upravljanja s tasterima.

6.1 Uspostavljanje serijske komunikacije i korištenje MODBUS protokola za čitanje markera

Za omogućavanje serijske komunikacije na PLC-u dovoljno je podesiti 'Baud rate' i isključiti bit pariteta u opcijama za serijsku liniju. Zatim je potrebno varijablama dodijeliti adresu koja ima formu \$MWX, gdje je X broj koji predstavlja samu adresu. Računarski C-kod koji služi kao osnova za čitanje/pisanje podataka na serijskoj vezi dat je u prilogu, te je on podešen na isti 'Baud rate' i isključen je bit pariteta. Međutim da bi mogli pristupiti podacima na PLC-u potrebno je prvo poslati poruku koja primjenjuje MODBUS protokol. MODBUS serijski protokol posjeduje dva načina rada RTU i ASCII. U RTU načinu rada svaka poruka se sastoji od niza od 8 bajta, prva dva uvijek predstavljaju adresu slave-a, dok zadnja dva služe za provjeru greške i generišu se na osnovu niza prethodnih bajta. ASCII način rada koristi heksadecimalne ASCII karaktere i nešto drugačiju formu [8]. U nastavku je dat primjer izgleda osnovne poruke u C-u, prilikom RTU načina rada, gdje su promjenjive unutar poruke tipa 'WORD'.

Program 6.1: Osnovna poruka

```
1 char poruka[] = {slave, function, markerAdrH, markerAdrL, podatak1 , podatak2,
    Error1, Error2};
```

MarkerAdr varijable predstavljaju gornji i donji bajt adrese markera koje su prethodno unesene u SoMachine-u (označen sa X u formi %MWX), dok podaci mogu predstavljati gornji i

donji bit podatka koji želimo upisati na marker ili broj registara koji se želi pročitati. Varijable Error1 i Error2 nazivaju se CRC biti i kod za njihovo generisanje je prikazan u nastavku.

Program 6.2: Generisanje CRC koda [8]

```

1  WORD CRC16 (const BYTE *nData, WORD wLength)
2  {
3      static const WORD wCRCTable[] = {
4          0X0000, 0XC0C1, 0XC181, 0X0140, 0XC301, 0X03C0, 0X0280, 0XC241,
5          0XC601, 0X06C0, 0X0780, 0XC741, 0X0500, 0XC5C1, 0XC481, 0X0440,
6          0XCC01, 0X0CC0, 0X0D80, 0XCD41, 0X0F00, 0XCFC1, 0XCE81, 0X0E40,
7          0X0A00, 0XCAC1, 0XCB81, 0X0B40, 0XC901, 0X09C0, 0X0880, 0XC841,
8          0XD801, 0X18C0, 0X1980, 0XD941, 0X1B00, 0XDBC1, 0XDA81, 0X1A40,
9          0X1E00, 0XDEC1, 0XDF81, 0X1F40, 0XDD01, 0X1DC0, 0X1C80, 0XDC41,
10         0X1400, 0XD4C1, 0XD581, 0X1540, 0XD701, 0X17C0, 0X1680, 0XD641,
11         0XD201, 0X12C0, 0X1380, 0XD341, 0X1100, 0XD1C1, 0XD081, 0X1040,
12         0XF001, 0X30C0, 0X3180, 0XF141, 0X3300, 0XF3C1, 0XF281, 0X3240,
13         0X3600, 0XF6C1, 0XF781, 0X3740, 0XF501, 0X35C0, 0X3480, 0XF441,
14         0X3C00, 0XFCC1, 0XFD81, 0X3D40, 0XFF01, 0X3FC0, 0X3E80, 0XFE41,
15         0XFA01, 0X3AC0, 0X3B80, 0XFB41, 0X3900, 0XF9C1, 0XF881, 0X3840,
16         0X2800, 0XE8C1, 0XE981, 0X2940, 0XEB01, 0X2BC0, 0X2A80, 0XEA41,
17         0XEE01, 0X2EC0, 0X2F80, 0XEF41, 0X2D00, 0XEDC1, 0XEC81, 0X2C40,
18         0XE401, 0X24C0, 0X2580, 0XE541, 0X2700, 0XE7C1, 0XE681, 0X2640,
19         0X2200, 0XE2C1, 0XE381, 0X2340, 0XE101, 0X21C0, 0X2080, 0XE041,
20         0XA001, 0X60C0, 0X6180, 0XA141, 0X6300, 0XA3C1, 0XA281, 0X6240,
21         0X6600, 0XA6C1, 0XA781, 0X6740, 0XA501, 0X65C0, 0X6480, 0XA441,
22         0X6C00, 0XACC1, 0XAD81, 0X6D40, 0XAF01, 0X6FC0, 0X6E80, 0XAE41,
23         0XAA01, 0X6AC0, 0X6B80, 0XAB41, 0X6900, 0XA9C1, 0XA881, 0X6840,
24         0X7800, 0XB8C1, 0XB981, 0X7940, 0XBB01, 0X7BC0, 0X7A80, 0XBA41,
25         0XBE01, 0X7EC0, 0X7F80, 0XBF41, 0X7D00, 0XBDC1, 0XBC81, 0X7C40,
26         0XB401, 0X74C0, 0X7580, 0XB541, 0X7700, 0XB7C1, 0XB681, 0X7640,
27         0X7200, 0XB2C1, 0XB381, 0X7340, 0XB101, 0X71C0, 0X7080, 0XB041,
28         0X5000, 0X90C1, 0X9181, 0X5140, 0X9301, 0X53C0, 0X5280, 0X9241,
29         0X9601, 0X56C0, 0X5780, 0X9741, 0X5500, 0X95C1, 0X9481, 0X5440,
30         0X9C01, 0X5CC0, 0X5D80, 0X9D41, 0X5F00, 0X9FC1, 0X9E81, 0X5E40,
31         0X5A00, 0X9AC1, 0X9B81, 0X5B40, 0X9901, 0X59C0, 0X5880, 0X9841,
32         0X8801, 0X48C0, 0X4980, 0X8941, 0X4B00, 0X8BC1, 0X8A81, 0X4A40,
33         0X4E00, 0X8EC1, 0X8F81, 0X4F40, 0X8D01, 0X4DC0, 0X4C80, 0X8C41,
34         0X4400, 0X84C1, 0X8581, 0X4540, 0X8701, 0X47C0, 0X4680, 0X8641,
35         0X8201, 0X42C0, 0X4380, 0X8341, 0X4100, 0X81C1, 0X8081, 0X4040 };
36
37     BYTE nTemp;
38     WORD wCRCWord = 0xFFFF;
39
40     while (wLength--)
41     {
42         nTemp = *nData++ ^ wCRCWord;
43         wCRCWord >>= 8;
44         wCRCWord ^= wCRCTable[nTemp];
45     }
46     return wCRCWord;
47 }

```

Početak funkcije, gdje se poruka priprema za slanje naveden je u nastavku. Bitno je napomenuti da svaki MODBUS zahtjev vraća odgovor, tako da i u situacijama pisanja treba pročitati

dobivenu poruku.

Program 6.3: Kreiranje MODBUS poruke

```

1 WORD sendSerial(HANDLE hComm, WORD markerAdr, WORD data1, WORD data2, BOOL Status,
  WORD function) {
2
3     char    lpBuffer[] = {0x01, function, 0x00, markerAdr, data1 , data2, 0x00, 0
      x00};
4     DWORD   dNoOFBytestoWrite;           // No of bytes to write into the port
5     DWORD   dNoOfBytesWritten = 0;       // No of bytes written to the port
6
7     dNoOFBytestoWrite = sizeof(lpBuffer); // Calculating the no of bytes to write
      into the port
8     WORD crc = CRC16(lpBuffer, dNoOFBytestoWrite - 2);
9     lpBuffer[6] = crc;
10    crc = CRC16(lpBuffer, dNoOFBytestoWrite - 1);
11    lpBuffer[7] = crc;

```

Za čitanje poruke se na mjesto rezervisano za funkciju upisuje vrijednost 3. Tako da kada koristimo prethodno navedenu funkciju kao zadnji parametar unosimo vrijednost 3, kao što je prikazano u nastavku,

Program 6.4: Čitanje podatka sa registra PLC-a

```

1 WORD readMarker(HANDLE hComm, WORD markerAdr, BOOL Status) {
2     WORD data = sendSerial(hComm, markerAdr, 0, 2, Status, 3);
3     return data;
4 }

```

Za pisanje poruke koristi se na mjesto rezervisano za funkciju upisuje vrijednost 6, te je u nastavku prikazan kod koji koristi 'sendSerial' funkciju za unošenje podatka na željeni registar.

Program 6.5: Pisanje na registar PLC-a

```

1 void writeToMarkerFull(HANDLE hComm, WORD markerAdr, WORD data1, WORD data2, BOOL
  Status) {
2     WORD info = sendSerial(hComm, markerAdr, data1, data2, Status, 6);
3 }

```

Radi načina definisanja, te osiguravanja ispravnog rada PLC-a razvijen je protokol za slanje komandi na PLC, kao i za čitanje/pisanje podataka.

6.2 Protokol slanja komandi na PLC

U ovom dijelu će biti opisan niz koraka, odnosno poruka koje je potrebno razmijeniti između računara i PLC-a za ostvarivanje određene funkcionalnosti, kao i mogućnost njihovog proširivanja. Na PLC-u je rezervisano šest markera na osnovu kojih se ovaj protokol zasniva i oni su:

- marker 0 - predstavlja komandu koja se želi pozvati, njegova vrijednost se postavlja putem računara
- marker 1 - čuva podatak koji korisnik želi upisati/pročitati, vrijednost se postavlja ili na računar ili na PLC-u zavisno od komande

Rezervisane komande	Funkcija
10	komanda za hitan stop koja obustavlja rad motora
20	komanda za pronalaženje početne pozicije
50-54	skup komandi za pomjeranje manipulatora u željenom smjeru za dato vrijeme
100-102	skup komandi za pomjeranje manipulatora na željenu poziciju
200-204	skup komandi za pomjeranje manipulatora kroz skup zadanih pozicija

Tabela 6.1: Tabela sa rezervisanim komandama za PLC

- marker 2 - zahtjev za izvršavanje komande (ima vrijednost 1 ili 0) postavlja ga računar, dok se resetuje na kraju izvršavanja komande
- marker 3 - označava kraj odnosno uspješno izvršavanje komande, njegovu vrijednost isključivo mijenja PLC (također ima vrijednost 1 ili 0)
- marker 4 - PLC mu dodjeljuje vrijednost trenutne horizontalne pozicije, služi isključivo za čitanje
- marker 5 - PLC mu dodjeljuje vrijednost trenutne vertikalne pozicije, te se također koristi isključivo za čitanje

6.2.1 Podjela komandi i njihov odabir

U tabeli 6.1 je prikazan skup komandi koje su implementirane, te kratak opis njihove funkcionalnosti. Korisnik može definisati vlastite komande, koristeći vrijednosti koje ne spadaju definisanim komandama, a procedura pravljenja SoMachine, kao i C-koda koji će ih podržati će biti objašnjena u nastavku. Za odabir komande napisana je i funkcija u C-u koja podržava 255 komandi.

Program 6.6: Odabir komande u C-u.

```

1 void odaberiKomandu(HANDLE hComm, WORD komanda, BOOL Status) {
2     writeToMarker(hComm, 0, komanda, Status);
3 }

```

Prethodni kod se lahko može proširiti, te u nastavku ja prikazana funkciju koja može pozvati 16 bitnu komandu.

Program 6.7: Modifikovani kod za 16 bitnu komandu.

```

1 void odaberiKomandu(HANDLE hComm, int komanda, BOOL Status) {
2     WORD komanda1 = komanda/255;
3     WORD komanda2 = komanda;
4     writeToMarkerFull(hComm, 0, komanda1, komanda2, Status);
5 }

```

6.2.2 Procedure pozivanja pojedinih komandi

Kako bi se olakšao proces primjene koda ponuđenog ovim radom za razvijanje novih funkcionalnosti u ovom dijelu će biti opisani procesi pozivanja određenih komandi. Sam proces razvoja SoMachine koda koji može podržavati nove komande će biti opisan u narednom dijelu. Proces čitanja neke varijable sastoji se iz slijedeće procedure:

1. odabir komande koja služi za čitanje željene varijable, odnosno upis vrijednosti te komande u marker 0
2. pokretanje procesa čitanja upisujući vrijednost 1 u marker 2
3. čekanje da se podatak pripremi, odnosno čitanje markera 3 dok njegova vrijednost ne postane 1
4. čitanje traženog podatka na markeru 1

Proces za upis u neku varijablu se sastoji iz procedure:

1. odabir komande koja služi za pisanje podatka u željenu varijablu
2. upis željene vrijednosti u marker 1
3. pokretanje procesa pisanja preko markera 2
4. čekanje da se podatak upiše, čekajući potvrdu markera 3

Svaka komanda koja se treba poslati se može svesti na jednu od ovih procedura, te je pomoću njih moguće objasniti kompleksnije procedure. Jedini izuzetak je komanda hitnog stopa koja se odmah po odabiru aktivira. Marker 2 se obavezno postavi na nulu nakon završetka svake komande.

Tako se za pomjeranje u određenom smjeru prvo odabere komanda za željeni smjer (41 - lijevo, 42 - desno, 43 - gore, 44 - dolje). Zatim se provede prethodna procedura upisa vrijednosti vremena čekanja u ms, zatim počinje kretanje i dobivanjem potvrde na markeru 3 kretanje se završava. Za pomjeranje na poziciju prvo se pokrene procedura slanja horizontalne pozicije komandom 100, zatim vertikalne komandom 101, te se daje signal za pokretanjem procesom čitanja komandom 102 (ignoriše se pročitani podatak ili se preskoči korak čitanja). Procedura za pomjeranje kroz grupu tačaka je slična prethodnom samo sa dodatnim komandama za započinjanje nove sekvence, i prelaskom na novu tačku, tako da će za nju biti samo navedene komande. Komande za zadavanje sekvence kretanja su: 200 - započinjanje nove sekvence, 201 - postavljanje horizontalne brzine, 202 - postavljanje vertikalne brzine, 203 - prelazak na novu tačku, 204 - pokretanje sekvence.

Za pristupanje markerima 4 i 5 nije potrebna sekvenca, te se njima može pristupiti u bilo kojem trenutku.

6.2.3 Kreiranje proizvoljnih funkcionalnosti u SoMachine-u korištenjem prethodnog protokola

Sam proces pravljenja pravljenja novih komandi u SoMachine-u se svodi na upis markera na željene varijable ili upis željene varijable na marker, odnosno na 'get' i 'set' funkcionalnosti, tako da ćemo u nastavku navesti primjere za generičku 'get' i 'set' funkciju ('get' šalje vrijednost neke interne varijable računaru, dok 'set' funkcija postavlja neku internu varijablu na vrijednost dobivenu od strane računara).

Program 6.8: Generička komanda 'get'

```

1 IF marker0 = 31 THEN
2   //Komanda za nalazenje pocetne pozicije
3     IF marker2 THEN
4       marker1 := internaVarijabla;
```

```

5         marker3 := TRUE;
6         marker2 := FALSE;
7     END_IF
8 END_IF

```

Program 6.9: Generička komanda 'set'

```

1 IF marker0 = 31 THEN
2     IF marker2 THEN
3         internaVarijabla := marker1;
4         marker3 := TRUE;
5         marker2 := FALSE;
6     END_IF
7 END_IF

```

Ukoliko interna varijabla koju mijenjamo služi za aktiviranje nekog bloka poželjno je njeno aktiviranje korištenjem triggera. Također u tom slučaju marker 3 treba postaviti na 1, a marker 2 na nula tek nakon što se na 'Done' izlazu pojavi vrijednost 'TRUE'.

6.2.4 Korištenje imlementiranih funkcija u C kodu

U samom programu na računar su napravljene funkcije radi lakšeg pozivanja već imlementiranih komandi, te se one mogu koristiti umjesto korištenja procedura pozivanja komandi. Implementirane funkcije su :

- void hitanStop(HANDLE hComm, BOOL Status)
- void nadjiPocetak(HANDLE hComm, BOOL Status)
- void idiLijevo(HANDLE hComm, int time_ms, BOOL Status)
- void idiDesno(HANDLE hComm, int time_ms, BOOL Status)
- void idiGoreHANDLE hComm, int time_ms, BOOL Status)
- void idiDole(HANDLE hComm, int time_ms, BOOL Status)
- void moveToPos(HANDLE hComm, int x, int y, BOOL Status)
- void idiKrozTacke(HANDLE hComm, int x[], int y[], BOOL Status, int brojKoordinata)

Zaključak

Kroz ovaj rad analiziran je način rada koračnih motora, njihove prednosti i mane, te mogućnosti njihovog upravljanja. Ponuđen je detaljan opis, način korištenja, te mogućnosti sistema za upravljanje dvoosnog manipulatora. Također su date upute za moguća proširenja pojedinih dijelova sistema.

Sistem je uspješno realiziran, te je omogućeno manuelno upravljanje sistema pomoću panela sa tasterima. Također je ostvarena mogućnost pamćenja i ponavljanja sekvence koju korisnik pokreće uz pomoć tastera. Kako bi se postigla komunikacija s računarom uspostavljena je serijska veza uz korištenje MODBUS protokola za prenos informacija, te je ujedno napisan program u C-u koji omogućava upravljanje sistemom.

Potencijalna nadogradnja sistema je moguća, te su funkcionalnosti sistema razdvajane radi olakšavanja toga procesa. Također je razvijen protokol slanja i pisanja komandi odnosno poruka sa računara na PLC, da bi se olakšao proces dodavanja novih funkcionalnosti i njihovim upravljanjem putem računara. Jedna od preporučenih nadogradnji je dodavanje kontaktnih senzora na MC_Power_PTO blokove. Time se omogućava automatsko zaustavljanje motora u slučaju dolaska u nedozvoljenu poziciju, te se kod za upravljanje brzinom motora može znatno pojednostaviti. Međutim potrebno je imati na umu da će to onemogućiti daljnje kretanje motora. Bilo kakva promjena brzine i ubrzanja nije preporučljiva, naročito u situaciji kada su na traci prisutne prepreke, bez prethodnog testiranja ponašanja motora. Pri većim brzinama rizik gubljenja koraka postaje znatan, te je moguće sudaranje sa potencijalnim preprekama na traci i oštećenje sistema. Također, pomjeranje prema graničnim senzorima velikim brzinama je izuzetno opasno. Motori imaju inerciju i ne mogu se zaustaviti odmah, tako da je moguć sudar sa konstrukcijom koji može oštetiti motore ili druge dijelove sistema. Također za potencijalnu nadogradnju preporučuje dodavanje varijable koja se vezuje za 'BufferredMode' blokova koji služe za diskretno i kontinualno pomjeranje. Time se otvara veći broj opcija za način upravljanja. Recimo, sa odgovarajućom postavkom načina rada 'BufferredMode'-a moguće je slati komande sa više blokova za pomjeranje, koji će se vršiti jedan za drugim, te time postići sekvencijalno upravljanje (moguća ograničenja na dužinu ovakve sekvence nisu poznata). Također, sa odgovarajućom promjenom spomenute varijable moguće je odmah zaustaviti sekvencu i pokrenuti novu komandu pomjeranja.

Poželjno je i povećanje brzine slanja i izvršavanja komandi prilikom upravljanja pomoću računara uz korištenje serijske komunikacije. Ubrzavanjem procesa izvršavanja komandi se povećaju mogućnosti rada sistema u realnom vremenu.

Prilozi

Prilog A

Uputstvo za rukovanje panelom sa tasterima

Za upravljanje osa CNC mašine koristi se panel na slici A.1. Da bi se omogućilo kretanje mašine potrebno je pritisnuti i držati 'Enable' dugme sve vrijeme tokom kretanja. Dugmad sa desne strane služe za pomijeranje.

Sistem je osmišljen tako da prilikom pokretanja zahtijeva podešavanje početne pozicije. Iz tog razloga pomijeranje sa dugmadima za kretanje je usporeno. Za pronalaženje početne pozicije sistema potrebno je pritisnuti dugme 'Configure'. Nakon ove akcije sistem zapičinje kretanje prema početnoj poziciji, te nije dozvoljeno pritiskati druga dugmad dok se podešavanje ne završi jer to može imati nepredviđene posljedice. Naravno tokom cijele procedure podešavanja potrebno je držati 'Enable' dugme pritisnuto. Zaustavljanje sistema na obje ose označava kraj procedure podešavanja početne pozicije.

Nakon završetka procedure podešavanja omogućeno je pomijeranje mašine sa većim brzinama u radnoj oblasti. Međutim ukoliko se približimo njenim granicama brzina biva smanjena, te kada dođemo do granice radne oblasti definisane softverom, motor i/ili motori se zaustavljaju. Ukoliko iz nekog razloga softverska ograničenja ne uspiju zaustaviti kretanje motora, na sistem su spojeni i krajnji prekidači. Kada se oni aktiviraju zaustavlja se kretanje u tom smjeru.

Nakon podešavanja početne pozicije možemo koristiti i mogućnosti snimanja i puštanja određene sekvence kretanja. Za pokretanje procedure snimanja sekvence potrebno je pritisnuti i čuvati 'Record' dugme. Njegovim pritiskom pamti se početna pozicija mašine, te možemo vršiti njeno pomijeranje korištenjem odgovarajućih dugmadi. To pomijeranje možemo vršiti po X i po Z osi simultano. Kada pustimo svako dugme za kretanje pamti se dosegnuta pozicija i dodaje u sekvencu. Tokom čitave procedure snimanja potrebno je držati pritisnutim 'Enable' i 'Record' dugme. Nakon pomijeranja mašine željenom sekvencom puštanjem 'Record' dugmeta istu dodajemo u memoriju PLC-a, te je time završen proces snimanja.

Za pomijeranje motora prethodno snimljenom sekvencom koristi se dugme 'Play'. Kada ga pritisnemo sistem se pomjera u početnu poziciju sekvence, te se vrši sekvencijalno pomijeranje kroz tačke zapamćene prilikom snimanja. Ovo dugme nije potrebno držati pritisnutim prilikom ovog postupka. Dolaskom u krajnju tačku sekvence kretanje se zaustavlja. Prilikom puštanja sekvence pritiskanje drugih dugmadi može prouzrokovati nepredviđene posljedice, te ono nije dozvoljeno izuzev, naravno, 'Enable' dugmeta koje mora biti pritisnuto tokom čitave procedure.

Ukoliko primjetimo nepredviđeno ponašanje sistema, u svakom trenutku možemo pustiti tipku 'Enable' čime se zaustavlja rad sistema.



Slika A.1: Panel za upravljanje CNC mašinom

Literatura

- [1] Athani, V., Stepper motors: fundamentals, applications and design. New Age International, 1997.
- [2] Acarnley, P. P., Stepping motors: a guide to theory and practice. Iet, 2002, No. 63.
- [3] Condit, R., Jones, D. W., “Stepping motors fundamentals”, Microchip Inc. Publication AN907, 2004, str. 1–22.
- [4] Kenjo, T., Sugawara, A., Stepping motors and their microprocessor controls. Clarendon Press Oxford, 1994.
- [5] Sax, H., “Stepper motor driving”, AN235, 1995.
- [6] “Manual smc64”, PPH WObit, dostupno na: http://www.silniki.pl/download/smc64_instr.pdf
- [7] Douglas W Jones, U. o. I. D. o. C. S., Control of stepping motors : a tutorial. Douglas W. Jones, 1998.
- [8] “Modbus protocol description”, Witte Software, dostupno na: www.modbustools.com/modbus.html
- [9] S., R., “Serial port programming using win32 api in c”, Xanathium Enterprises, dostupno na: <https://github.com/xanathium-enterprises/Serial-Programming-Win32API-C>