SINKRONIZACIJA UPRAVLJANJA ELEKTROMOTORNIH POGONA ZA OSTVARIVANJE KRUŽNE PUTANJE KRETANJA

Pešut, Ivan

Master's thesis / Diplomski rad

2023

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

Permanent link / Trajna poveznica: https://urn.nsk.hr/urn:nbn:hr:190:416598

Rights / Prava: Attribution 4.0 International/Imenovanje 4.0 međunarodna

Download date / Datum preuzimanja: **2024-05-09**

Repository / Repozitorij:

Repository of the University of Rijeka, Faculty of Engineering

SVEUČILIŠTE U RIJECI

TEHNIČKI FAKULTET

Diplomski sveučilišni studij elektrotehnike

Diplomski rad

SINKRONIZACIJA UPRAVLJANJA ELEKTROMOTORNIH POGONA ZA OSTVARIVANJE KRUŽNE PUTANJE KRETANJA

Rijeka, rujan 2023.

Ivan Pešut

0069083650

SVEUČILIŠTE U RIJECI

TEHNIČKI FAKULTET

Diplomski sveučilišni studij elektrotehnike

Diplomski rad

SINKRONIZACIJA UPRAVLJANJA ELEKTROMOTORNIH POGONA ZA OSTVARIVANJE KRUŽNE PUTANJE KRETANJA

Mentor: Prof. dr. sc. Dario Matika

Rijeka, rujan 2023.

Ivan Pešut

0069083650

SVEUČILIŠTE U RIJECI
TEHNIČKI FAKULTET
POVJERENSTVO ZA DIPLOMSKE ISPITE

Rijeka, 18. travnja 2023.

Zavod: **Zavod za automatiku i elektroniku**
Predmet: **Automatizacija postrojenja i procesa**
Grana: **2.03.06 automatizacija i robotika**

ZADATAK ZA DIPLOMSKI RAD

Pristupnik: **Ivan Pešut (0069083650)**
Studij: Sveučilišni diplomski studij elektrotehnike
Modul: Automatika

Zadatak: **Sinkronizacija upravljanja elektromotornih pogona za ostvarivanje kružne putanje kretanja/ Synchronizing the control of electric motor drives to realize a circular path of movement**

Opis zadatka:

Potrebno je projektirati i izraditi upravljački program za kontrolu dva mehanički neovisna elektromotorna pogona kako bi se ostvarila kružna putanja kretanja izvršnog elementa. Upravljački program treba sadržavati sekvencu podizanja tereta tijekom koje se treba izvršiti kružno gibanje izvedeno preko dva elektromotrona pogona. Projketiranje uključuje detaljan opis sustava, razvoj programske aplikacije upravljanja i vizualizacije u programskom okruženju Siemens Tia portal, te funkcionalno ispitivanje sustava korištenjem aplikacije za simulaciju na programabilnom logičkom kontroleru Siemens Tia Portal S7-1500.

Rad mora biti napisan prema Uputama za pisanje diplomskih / završnih radova koje su objavljene na mrežnim stranicama studija.

Zadatak uručen pristupniku: 20. ožujka 2023.

Mentor:

Prof. dr. sc. Dario Matika

Dr. sc. Nikola Turk (komentor)

Predsjednik povjerenstva za diplomski ispit:

Prof. dr. sc. Dubravko Franković

IZJAVA

Izjavljujem da sam samostalno izradio diplomski rad prema zadatku preuzetom dana 20. ožujka 2023.

Rijeka, 14. rujan 2023.

_____

Ivan Pešut

ZAHVALA

Zahvaljujem se svome mentoru prof. dr. sc. Dariu Matiki i komentoru dr. sc. Nikoli Turku na svim konstruktivnim savjetima i izdvojenom vremenu pri pisanju ovoga diplomskog rada.

Također želio bih se zahvaliti tvrtki Danieli Systec d.o.o. na danoj mogućnosti za izradu diplomskog rada. Isto tako zahvaljujem se g. Toniju Tumpiću na mentorstvu prilikom izrade rada.

Sadržaj

1. Uvod 1

2. Kran i komponente 2

2.1. Mosni kran 2

2.2. Dijelovi mosnog krana 3

2.2.1. Kranska kolica 3

2.2.2. Vitlo 5

2.3. Opis pokreta dizanja lonca 7

3. Hardver i softverska podrška sustava 9

3.1. Programibilni logički kontroler 9

3.2. Senzori i aktuatori u sustavu 11

3.3. Komunikacija između PLC-a i ostalih uređaja 14

3.4. Komunikacija frekvencijski pretvarač ⇔ PLC 16

3.5. Organizacija programskih blokova i upravljačka logika sustava 29

3.5.1. Sekvenca i automatsko upravljanje kranom 48

3.5.2. Simulacija sekvence i podizanja nosača lonca 62

4. Vizualizacija procesa 66

4.1. Općenito o HMI-u 66

4.2. Izgled HMI stranica 67

5. Zaključak 80

Sažetak i ključne riječi 82

Abstract and keywords 82

Dodatak A 83

1. Uvod

Proizvodnja čelika je poznata po svojoj složenosti procesa, gdje besprijekorna koordinacija opreme i strojeva postaje ključna za osiguranje kvalitete, produktivnosti i sigurnosti unutar same proizvodnje. Unutar ovog raznolikog spektra komponenti i strojeva, kranovi igraju izuzetno važnu ulogu u manipulaciji materijalima te prijevozu sirovina, poluproizvoda i gotovih čeličnih proizvoda tijekom cjelokupnog proizvodnog procesa. Premda je glavna funkcija kranova naravno linearno kretanje, neosporno je da u nekim specifičnim proizvodnim operacijama postoji potreba za izvođenjem kružnog kretanja kako bi se postigla optimalna obrada i podizanje nekih tereta.

U uvodnom dijelu rada pružen je opis kranova i njihovih komponenti, te je istaknuto u kojim situacijama se najčešće zahtijeva kružno gibanje. Drugi dio fokusira se na komunikaciju između različitih uređaja i senzora, s posebnim naglaskom na interakciju između frekvencijskog pretvarača i PLC-a.

Glavni dio je ujedno i najveći izazov ovog rada gdje je objašnjeno postizanje sinkroniziranog upravljanja elektromotora kako bi se omogućilo precizno i kontinuirano kretanje duž kružne putanje. U većini slučajeva, kranovi su upravljani ručnim komandama, gdje operater usmjerava rad samo jednog dijela krana u isto vrijeme. Za postizanje optimalnih performansi, ključno je osigurati sinkronizaciju između različitih komponenti krana, kao što su glavno vitlo, kranska kolica i most krana. Ova sinkronizacija omogućava njihovo koordinirano kretanje, posebno važno kada se zahtijeva izvođenje preciznih kružnih putanja.

2. Kran i komponente

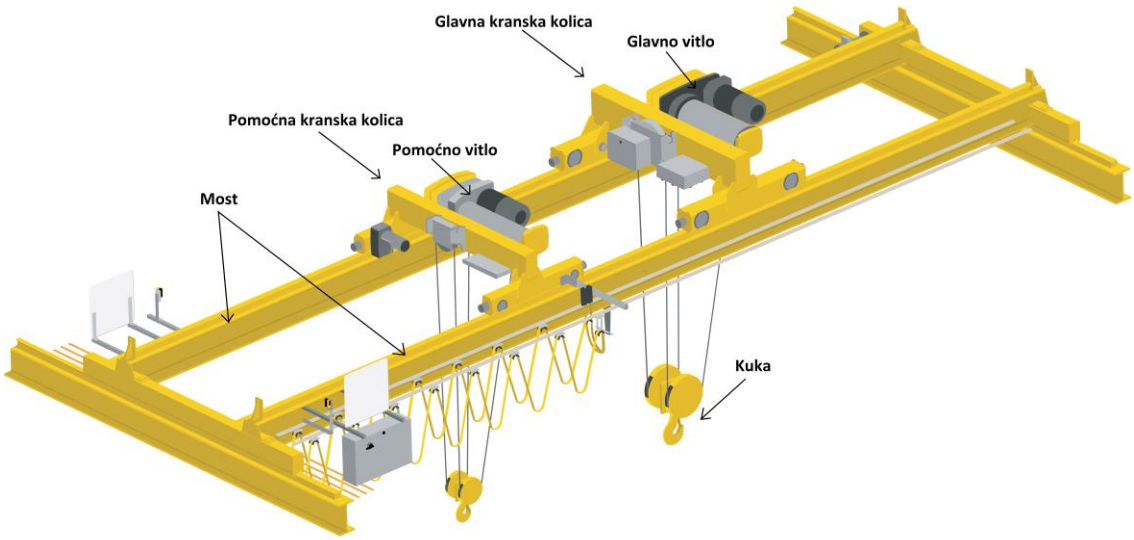
2.1. Mosni kran

Mosni kran, vrsta je fiksne dizalice poduprte nosačem s pomičnim mostom koji se pomiče na nosačima ili šinama ovješenim na visini. To omogućuje pomicanje tereta u tri osi u bilo kojoj točki radnog prostora dizalice, što znači da se može pomicati lijevo i desno, gore i dolje te naprijed i nazad po šinama.

Mosne dizalice su teška oprema za rukovanje materijalom s velikim nosivostima. Operater upravlja mosnom dizalicom ručno iz kontrolne sobe, pomoću žičane viseće stanice, bežične kontrole ili automatski u kompliciranim kretnjama koje zahtijevaju preciznost koja se ne može postići ručnim upravljanjem. Mosne dizalice se obično koriste za višesmjerni transport materijala koji pomaže u proizvodnji, skladištenju, utovaru i istovaru u industrijskom objektu, a u ovom slučaju služi za prenošenje velikih lonaca rastaljenog čelika na razne lokacije u tvornici.

Mosne dizalice omogućuju širi raspon kretanja i veću fleksibilnost od većine ostalih dizalica zbog svoje robusnosti i dizajna. Stoga su idealne za premještanje velikih ili teških tereta u više smjerova, što može biti osobito korisno u proizvodnim pogonima gdje se proizvodi često transportiraju između različitih radnih stanica. Na slici 2.1. prikazana je mosna dizalica sa svojim dijelovima.

Konkretno u ovom radu opisati će se rad samo kranskih kolica(*eng. The Trolley*) i glavnog vitla(*eng. Main Hoist*) jer se pomoću njih kran radi kružnu putanju pri dizanju i spuštanju lonca s čelikom.



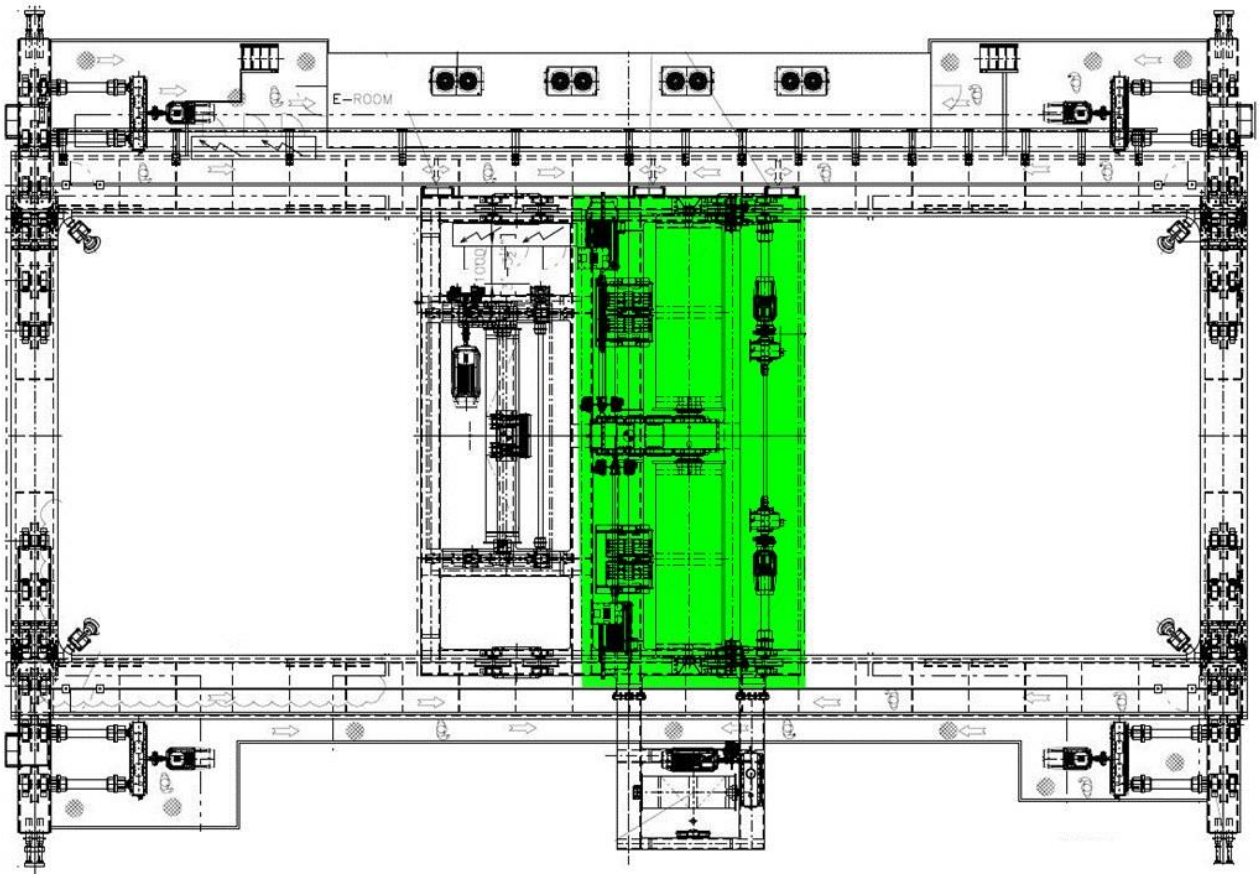
Slika 2.1. Prikaz dijelova krana [1]

2.2. Dijelovi mosnog krana

2.2.1. Kranska kolica

Kranska kolica (slika 2.2.) na sebi imaju kotače te motor koji pomiče kolica lijevo desno po mostu, također kolica na sebi sadrže i glavno vitlo koje diže i spušta teret. Kolica imaju jedan asinkroni motora 600VAC 60 Hz nazivne snage 74kW i nazivne brzine 1187 okretaja u minuti.

Motor sadrži i PT100 temperaturni senzor te je opremljen jednim ventilatorom za hlađenje od 1,1kW 600VAC 60Hz montiranim na motor kako bi se smanjila temperatura motora tijekom rada(uključuje se svaki put kada se kolica koriste). Motor je spojen na zupčasti prijenos, redukcije omjera 1:56.75. Također kolica sadrže i kočnicu kojom upravlja frekvencijski pretvarač korištenjem digitalnih izlaza. Na motor montiran apsolutni enkoder kako bi se znala točna pozicija kranskih kolica u prostoru. Korišteni pretvarač jest Siemens S120, 200kW, 500-690V, 50/60Hz te nazivne struje 215A.[2]



Slika 2.2. Prikaz glavnih kranskih kolica [2]

U tablici 2.1. se vide svi navedeni nazivni podaci motora kranskih kolica kojim upravlja frekvencijski pretvarač.

Tablica 2.1. Nazivni podaci motora [2]

Nazivni podaci motora	
Snaga	74 kW
Brzina vrtnje	1187 o/min
Napon	600 V – 60 Hz
Struja	95 A
Moment	601 Nm

Tablica 2.2. prikazuje postavke brzina pri pomicanju kranskih kolica. Koraci 10, 20, 75 i 100% se odnose na brzinu kranskih kolica u ručnom načinu rada, te ih je moguće postići pomicanjem upravljačke palice koja se pomiče u četiri koraka. Također prikazano je vrijeme akceleracije i deceleracije kranskih kolica od 0 do 100% i obratno.

Tablica 2.2. Postavke brzina pomicanja kranskih kolica [2]

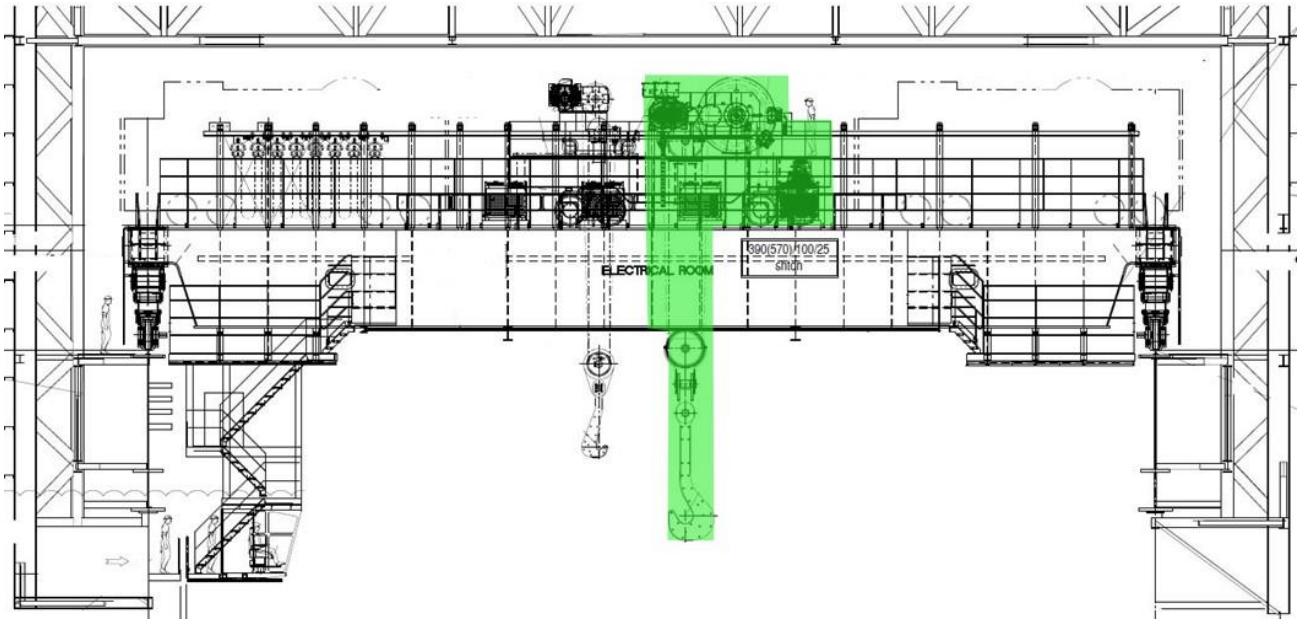
Postavke brzine			**Opterećenje**
Maksimalna brzina	660.4 mm/s		Maks. opt.
Korak 1	81.28 mm/s	10%	Maks. opt.
Korak 2	208.28 mm/s	25%	Maks. opt.
Korak 3	416.56 mm/s	75%	Maks. opt.
Korak 4	660.4 mm/s	100%	Maks. opt.
Akceleracija 0…100%		6 s	
Deceleracija 100…0%		4 s	

2.2.2. Vitlo

Vitlo (slika 2.3.) je dizajnirano za podizanje i spuštanje željenog tereta i pričvršćeno je na okvir kranskih kolica. Vitlo korišteno u projektu koristi čelične sajle te ima nosivost od 354 tone.

Glavno vitlo opremljeno je s dva identična asinkrona motora 600VAC 60Hz nazivne snage 448kW i nazivne brzine 1193o/min. Dva PT100 termistora mjere temperaturu svakog motora. Motor je opremljen jednim ventilatorom za hlađenje od 3kW 600VAC 60Hz koji je montiran na motor kako bi se smanjila temperatura motora tijekom rada. Svaki motor je upravljan jednim Siemens S120 pretvaračem, nazivne snage 710kW, napona napajanja 500 do 690V i frekvencije 50/60Hz. Maksimalna izlazna struja pretvarača iznosi 794A.

Vitlo također kao i kranska kolica ima jedan planetarni zupčasti prijenos s redukcijom 1:76,6. Prijenos je povezan s bubnjem koji spušta i podiže kuku, isto tako bubanj sadrži i senzor za preklapanje sajle, kako bi prepoznali grešku pri slaganju sajle kod podizanja tereta. Na bubnju se također nalaze i dvije kočnice koje su upravljane impulsom sa svakog od pretvarača.[2]



Slika 2.3. Prikaz glavnog vitla [2]

U tablici 2.3. se vide svi navedeni nazivni podaci motora glavnog vitla kojim upravlja frekvencijski pretvarač.

Tablica 2.3. Nazivni podaci motora [2]

Nazivni podaci motora	
Broj motora	2
Snaga	448 kW
Brzina vrtnje	1193 o/min
Napon	600 V – 60 Hz
Struja	557 A
Moment	4177 Nm

Tablica 2.4. ilustrira konfiguraciju brzina za pokretanje glavnog vitla. Brzine od 10, 20, 75 i 100% odnose se na brzinu vitla u ručnom režimu, koju je moguće postići pomicanjem upravljačke palice kroz četiri različita koraka. Također, prikazano je vrijeme potrebno za akceleraciju i deceleraciju glavnog vitla od 0 do 100% i obratno.

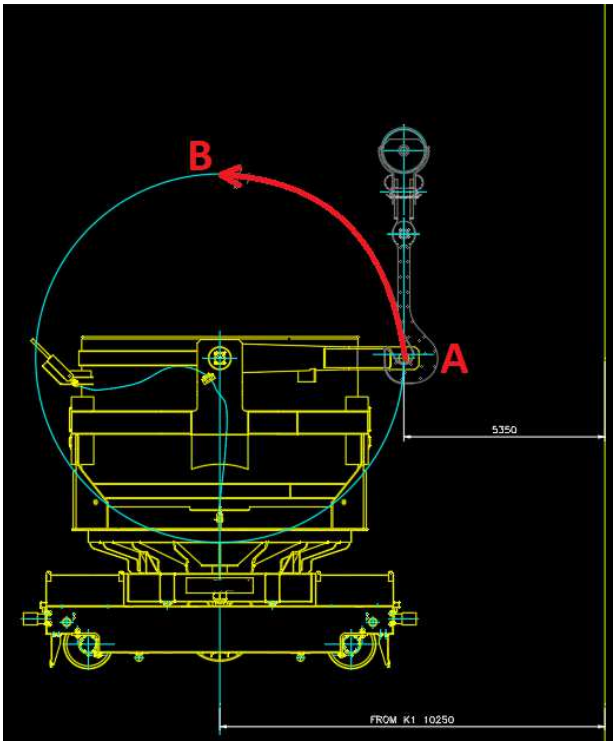
Tablica 2.4. Postavke brzina pomicanja glavnog vitla [2]

Postavke brzine			Opterećenje
Maksimalna brzina	132.08 mm/s		Maks. opt.
Korak 1	20.32 mm/s	10%	Maks. opt.
Korak 2	50.8 mm/s	25%	Maks. opt.
Korak 3	101.6 mm/s	75%	Maks. opt.
Korak 4	132.08 mm/s	100%	Maks. opt.
Akceleracija 0…100%		3 s	
Deceleracija 100…0%		3 s	

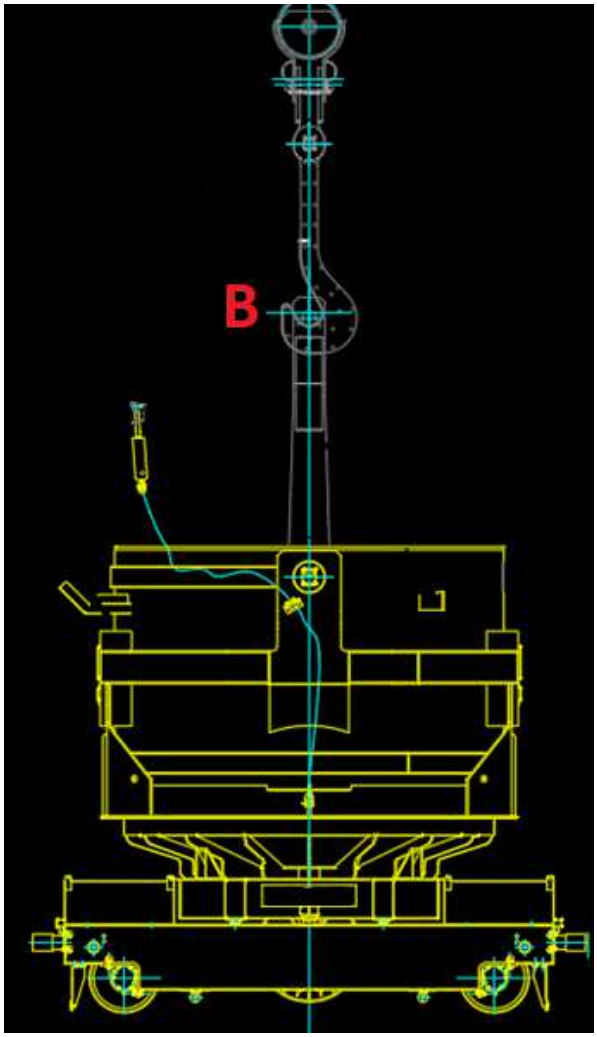
2.3. Opis pokreta dizanja lonca

Obično kod mosnih kranova i svih dizalica općenito operater kod ručnog načina rada može upravljati i pomicati samo jednu od osi krana istovremeno. Ograničavanje kontrole operatera se izvodi kako bi se zaštitila oprema i osoblje, zbog kompleksnosti pokreta te same mase tereta kojeg se pomiče. Također zbog mogućnosti njihanja tereta ukoliko se prebrzo počne pomicati pretvarači nove generacije poput ovog korištenog u projektu, Siemens S120, sadrže i neki tip softverske zaštite od njihanja tereta kako bi operaterima olakšali upravljanje. Kod Siemensa je to DBSC(*eng. Drive base Sway Control*) koji ne koristi kamere za razliku od nekih drugih proizvođača već radi na principu matematičkog modela njihala, gdje se kontinuirano računa pozicija vitla u prostoru na temelju podataka tereta. Kao rezultat DBSC kontrolira pretvarač, a time i motor osi koja se pomiče kako bi se eliminiralo njihanje. [3]

Nekad je potrebno izvršiti kompleksan pokret gdje je potrebno pomicati teret u više osi istovremeno, u tom slučaju prethodno se programiraju funkcije koje izračunavaju potrebne brzine i pozicije osi koje se pomiču kako bi se obavio taj pokret i time olakšalo operateru te povećala sigurnost. Jedan od takvih pokreta jest dizanje lonca s čelikom te premještanje na neku od lokacija u tvornici. Potrebno je kuku okačiti na nosače lonca te pomaknuti nosač iz pozicije A(slika 2.4.) u poziciju B(slika 2.5.).



Slika 2.4. Kuka zakačena na poziciji A [2]



Slika 2.5. Kuka zakačena na poziciji B [2]

Opisan kružni pomak (slika 2.4.) ostvaruje se sinkroniziranim upravljanjem dvjema osima, konkretno glavnim vitlom kojim upravljamo vertikalnim kretnjama i glavnim kranskim kolicima kojim upravljamo horizontalnim kretnjama. Pokreti krana moraju biti vrlo precizni i pratiti točnu kružnu putanju kako se čelik u loncu ne bi istresao te uništio opremu i ugrozio sigurnost radnika. Sama sekvenca, izračuni brzina i pozicija svakog od motora bit će objašnjeni u nastavku ovog rada.

3. Hardver i softverska podrška sustava

3.1. Programibilni logički kontroler

PLC je kratica za programibilni logički kontroler. PLC je programibilni računalni uređaj koji se koristi za upravljanje procesima, obično u industriji. PLC se ponekad naziva i industrijskim računalom. PLC nadzire stanja ulaznih uređaja, kao što su signali s senzora, i donosi odluke o sljedećem stanju izlaznih uređaja, kao što je uključivanje ili isključivanje npr. motora.

PLC-ovi se također koriste za prijenos informacija s uređaja u tvornicama ili s lokacija izvan tvornice do centraliziranih upravljačkih aplikacija koje se često izvode na osobnim računalima kako bi se nadzirao i upravljao cjelokupni sustav. PLC-ovi se obično koriste za nadzor i izvješćivanje o stanjima uređaja, za dijagnosticiranje grešaka u hardverskim uređajima kao što su industrijski strojevi i alati te za automatizaciju postrojenja. Za potrebe ovog rada korišten je Siemens S7-1500 PLC koji je prikazan na slici 3.1., uz napajanje i 4 I/O modula.[4]

Konkretno, korišten je Siemens PLC model 1518-4 PN/DP. Brojka 4 označava da ovaj PLC ima četiri komunikacijska porta. U slučaju ovog CPU-a, tri su PROFINET porta, dok je četvrti PROFIBUS konekcijski port. Detaljni opisi ovih komunikacijskih protokola bit će pruženi u poglavlju 3.3.

Ovaj PLC ima radnu memoriju od 24MB i impresivno vrijeme procesiranja za bit operacije od samo 1ns. Ovo je izuzetno brza i pouzdana karakteristika koja se pokazala vrlo korisnom u preciznim kretnjama u kranskom sustavu.

Slika 3.1. Siemens S7-1500 PLC s napajanjem i I/O karticama [Autor]

PLC-ovi (*eng. Programmable Logic Controllers*) su nasljednici relejnih logičkih sustava koji su služili kao kontrolni sustavi za nadzor i upravljanje uređajima, kao što su sklopke, releji, mjerači vremena, aktuatori i motori. U logičkim sustavima releja, releji izvode logičke operacije aktiviranjem i deaktiviranjem zavojnica u električnim krugovima. Releji su bili idealni za upravljanje i koordinaciju raznih industrijskih automatiziranih uređaja jer su mogli spajati strujne krugove različitih strujnih i naponskih razina. Međutim, nedostatak relejnog logičkog sustava je bio u tome što je bio teško ožičen i zahtijevao mnogo žica, što ga je činilo teškim za održavanje. Razvoj mikrokontrolera riješio je ovaj problem omogućujući programiranje relejnih logičkih procesa i njihovo pohranjivanje u računalo. Takav tip programiranja naziva se ljestvičast dijagram(*eng. Ladder diagram*), te radi na principu releja, gdje imamo NO(*eng. Normally open*) i NC(*eng. Normally closed*) kontakte. Te kontakte postavljamo logičkim slijedom kako bi uključivali ili isključivali neke izlaze tj. "špule". Još postoje SCL(*eng. Structured Control Language*) programski jezik koji je nalik C-u te STL(*eng. Statement List*) jezik koji je jednak asembler programskom jeziku.[4]

PLC se sastoji od sljedećih elemenata:

- CPU(*eng. Central Processing Unit*) – centralna procesorska jedinica
- RAM(*eng. Random Access Memory*) memorije
- ROM(*eng. Read Only Memory*) memorije
- I/O modula
- Kartice za napajanje

I/O moduli ili kartice mogu biti digitalni ili analogni. Digitalni I/O moduli služe za komunikaciju sa jednostavnim senzorima, aktuatorima, ventilima i sl., odnosno s uređajima koji rade sa diskretnim signalom. Diskretni signal sa digitalnog I/O modula može poprimiti vrijednosti 0 ili 24 V.

Analogni I/O moduli služe za rad s kontinuiranim signalima tj. analognim signalima, konvertirajući različite naponske i strujne razine u informacije pogodne za PLC. Analogni moduli omogućavaju preciznu kontrolu opreme kao što su motori, proporcionalni ventili, regulatori temperature i sl. Također analogni moduli rade na različitim naponskim i strujnim razinama kao što su 0 do 10V, -10V do 10V i 4 do 20mA.

Kod Siemens S7-1500 PLC-a potrebno je imati karticu za napajanje, koja će konvertirati izmjenični mrežni napon na DC napon 24V kako bi se napajao CPU i dodatne modularne kartice.

3.2. Senzori i aktuatori u sustavu

Prekidač za prepoznavanje preklapanja sajle(*eng. Rope guiding limit switch*) instalira se na glavno vitlo krana. Svrha tog prekidača je prepoznati eventualno preklapanje sajle i obavijestiti PLC o tome kako bi se odmah prekinulo gibanje, ukoliko je to moguće. U slučaju preklapanja sajle dozvoljava se samo spuštanje tereta kako bi se sajla na vitlu odmotala. Korišteni prekidač se vidi na slici 3.2..

Slika 3.2. Prikaz prekidača za prepoznavanje preklapanja sajle [2]

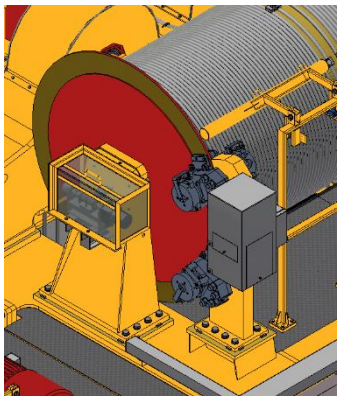
Pozicijski krajnji prekidači postavljeni su na krajevima kranskih kolica te na vitlu kako bi znali kada se nalazimo blizu maksimalnog dosega krana. Postoje dvije vrste prekidača, krajnji prekidači(*eng. Limit switch*) i prekidači koji označavaju kada kran treba usporiti(*eng. Slowdown switch*). Postavljeni su tako da kran prvo pređe preko slowdown prekidača, pri čemu se kran uspori na neku zadanu brzinu te staje tek kad dođe do krajnjeg prekidača. Oba prekidača su jednakog dizajna kao na slici 3.3..

Slika 3.3. Prikaz krajnjeg prekidača [2]

Glavne disk kočnice su postavljene na bržu stranu zupčastog prijenosa vitla i kranskih kolica. Kočnice su upravljane pomoću signala sa pretvarača, te se otvaraju kada se motor magnetizira i zatvaraju kada je brzina vrtnje motora oko nule. Također kočnica se zatvara ukoliko je došlo do greške u pretvaraču ili je pritisnuta sigurnosna tipka. Kočnice su dizajnirane tako da izdrže puni moment motora plus sigurnosni faktor. Na ovoj kočnici također je ugrađen prekidač koji PLC-u šalje povratnu informaciju je li kočnica otvorena. Kočnica krana je prikazana na slici 3.4..

Slika 3.4. Prikaz kočnica krana [2]

Kran također na vitlu ima sigurnosne kočnice koje rade nezavisno od PLC-a. Sigurnosne kočnice aktiviraju se ukoliko dođe do prebrzog spuštanja vitla, a zbog velike mase tereta i krana aktiviraju se oko 15cm od prvotne pozicije kada je počelo propadanje. Također one se zatvaraju ako nestane napajanja na motoru i pretvaraču te pritiskom na sigurnosnu tipku. Sigurnosne kočnice montirane na vitlo se vide na slici 3.5..[2]



Slika 3.5. Prikaz sigurnosnih kočnica na vitlu [2]

Za precizno određivanje pozicija X, Y i Z osi krana koriste se apsolutni enkoderi koji su direktno povezani s sekundarnim dijelom zupčastog prijenosa, osiguravajući omjer prijenosa od 1:1. Sva tri smjera oslanjaju se na upotrebu enkodera AMF60 (slika 3.6.), koji omogućuje povezivanje putem PROFINET-a, odnosno Etherneta. Svaka pozicija krana odgovara jedinstvenom kodu. Čak i pokreti koji se dogode dok je kran isključen precizno se transformiraju u odgovarajuće položajne vrijednosti čim se enkoder ponovno aktivira.

Slika 3.6. Prikaz AMF60 enkodera [2]

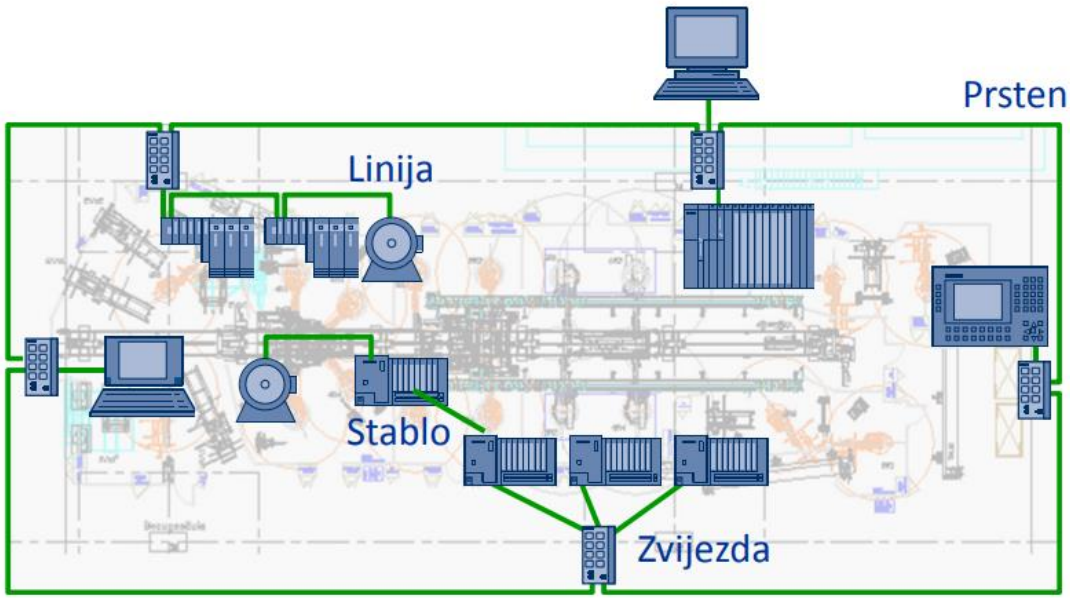
Jedna od važnijih sigurnosnih komponenti je sigurnosna tipka (slika 3.7.), koja se ponekad naziva i gljivom. Ova komponenta ima ključnu ulogu u prekidu rada cijelog postrojenja ili njegovog određenog dijela kako bi se zaštitili operator ili stroj od mogućeg oštećenja. Nakon što se sigurnosna tipka pritisne, ostaje u tom položaju sve dok se ne resetira, obično rotacijom same tipke. U svim scenarijima, tipka je tipa NC (*eng. Normally closed*), što znači da se kontakt otvara kada se pritisne. Ovo dovodi do logičke nule na PLC-u na koji je tipka spojena, što pokreće određene akcije isključivanja stroja ili njegovog dijela gdje je sigurnosna tipka pritisnuta.

Slika 3.7. Prikaz sigurnosne tipke[2]

3.3. Komunikacija između PLC-a i ostalih uređaja

Siemens PLC-i podržavaju različite komunikacijske protokole koji omogućavaju jednostavnu komunikaciju između PLC-ova, HMI-a (*eng. Human Machine Interface*), pretvarača i ostalih uređaja koji zahtijevaju složeniju komunikaciju s PLC-om. Dva najkorištenija takva protokola su PROFINET koji je zapravo industrijski Ethernet te PROFIBUS, što je zapravo standardni serijski komunikacijski protokol.[5]

PROFINET omogućuje cikličku i acikličku komunikaciju između komponenti, uključujući dijagnostiku, sigurnost, alarme i informacije. Za povezivanje svih ovih komponenti PROFINET koristi standardni Ethernet kao komunikacijski medij. Ethernet kabeli povezuju PROFINET komponente u mrežu, što omogućuje koegzistenciju drugih Ethernet protokola u istoj infrastrukturi. Okruženja industrijske automatizacije često zahtijevaju velike brzine i deterministčku komunikaciju. Deterministička komunikacija znači isporuku poruka točno onda kada se očekuju. PROFINET mora osigurati da se poruke isporučuju odgovarajućom brzinom i determinizmom za zadatak. Ne zahtijevaju sve aplikacije iste performanse. Na primjer, učitavanje konfiguracijskih podataka za procesni instrument može potrajati nekoliko minuta bez utjecaja na proizvodnju. S druge strane, kašnjenje od samo nekoliko milisekundi u komunikaciji između PLC kontrolera i frekvencijskih pretvarača koji upravlja brzinom motora može značajno utjecati na proces. Različite vrste PROFINET konfiguracije se mogu vidjeti na slici 3.8..[5]



Slika 3.8. Prikaz različite PROFINET komunikacije među uređajima u nekom postrojenju [6]

PROFIBUS(*eng. Process Field Bus*) je široko korišten serijski komunikacijski standard korišten i kod Siemens-ovih PLC-a. PROFIBUS podržava dva komunikacijska tipa, PROFIBUS PA(*eng. Process Automation*) i PROFIBUS DP(*eng. Decentralised Peripherials*).

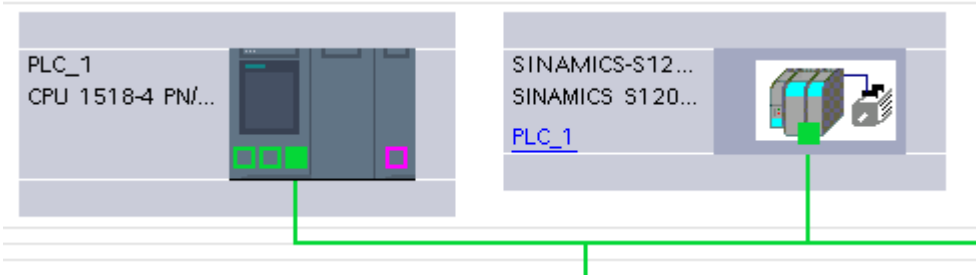
PROFIBUS DP je često korišten kod procesa koji zahtijevaju bržu komunikaciju između PLC-a i nekih decentraliziranih uređaja, senzora ili aktuatora. PROFIBUS DP jest većinom ciklički, brz i deterministički komunikacijski protokol. Njegova brzina prijenosa iznosi od 9,6 Kbit/s do 12 Mbit/s što naravno ovisi o tipu kabela i njegovoj duljini. Koristi se RS 485 standard za serijsku komunikaciju. U PROFINET DP komunikacijskoj mreži su definirana dva tipa uređaja, aktivni/nadređeni uređaj(*eng. master*) i pasivni/podređeni uređaj(*eng. slave*). Nadređeni uređaj klase 1 jest PLC i on upravlja podređenim uređajima te ima najviši prioritet u komunikaciji na mreži. Nadređeni uređaji klase 2 su većinom HMI ili neki drugi upravljački, nadzorni sustav koji služi za puštanje u pogon ili namještanje određenih parametara u pogonu. Klasa 2 ima niski prioritet na mreži. Podređeni uređaji su periferni uređaji koji procesiraju signale i prenose ih nadređenom uređaju ili samo djeluju kako im je određeno sa nadređenog uređaja. Oni ne mogu inicirati komunikaciju već samo odgovarati na zahtjev nadređenog uređaja. U ovom radu i projektu koristi se samo PROFINET komunikacija zbog svoje superiorne brzine i jednostavnosti nad PROFIBUS mrežom.[5]

3.4. Komunikacija frekvencijski pretvarač ⇔ PLC

U ovom radu koriti se SINAMICS S120 frekvencijski pretvarač za kontrolu svih motora krana te komunicira sa PLC-om preko PROFINET-a, to znači da pretvarač i PLC izmjenjuju telegrame preko industrijskog Etherneta.

Telegram jest komunikacijska poruka koja se šalje unutar sustava kako bi se razmijenile naredbe i podaci među različitim uređajima. Svaki telegram sadrži kontrolne riječi(*eng. Control word*), njihov broj u telegramu ovisi o vrsti telegrama. Kontrolna riječ je podatak unutar telegrama koji sadrži niz bitova koji upravljaju različitim funkcijama frekvencijskog pretvarača ili uređaja s kojim se komunicira. Kontrolna riječ također omogućava postavljanje parametara, promjenu brzine motora, smjer vrtnje te mnogo drugih funkcija.

SINAMICS pretvarači podržavaju različite standardne telegrame za razmjenu podataka s PLC-om. U ovom slučaju za razmjenu podataka između PLC-a i pretvarača koristi se telegram 1. Konkretno korišten je Standard telegram 1, PZD-2/2 i razmjenjuju se dvije riječi podataka između PLC-a i pretvarača. PLC pogonu šalje telegram koji sadrži kontrolnu riječ pretvarača i postavljenu brzinu, a pretvarač šalje telegram PLC-u koji sadrži riječ statusa pretvarača i stvarnu brzinu. [7]



Slika 3.9. Prikaz PROFINET mreže sa PLC-om i SINAMICS-S120 pretvaračem [Autor]

Siemens nudi nekoliko telegrama za komunikaciju između procesora i pogona. Najčešći su: standardni telegram 1, standardni telegram 20, Siemensov telegram 350, Siemensov Telegram 352, Siemensov Telegram 353 i Siemensov Telegram 354. Izbor vrste telegrama koji se koristi ovisi o namjeni kontrole. Na primjer, ako je potrebna kontrola brzine, koristi se standardni telegram 1 ili Siemensov telegram 352. S druge strane, ako je potrebna kontrola položaja, koristi se standardni telegram 111. Slika 3.9. prikazuje povezanost PLC-a i frekvencijskog pretvarača pomoću PROFINET mreže.[7]

Standard telegram 1, PZD-2/2 s njegovim kontrolnim riječima prikazani su na tablicama 3.1. i 3.2 za komunikaciju u smjeru frekvencijski pretvarač ⇨PLC.

Tablica 3.1. Kontrolna riječ 0 u komunikaciji pretvarač ⇨ PLC [7]

BIT	**OPIS**	**PARAMETAR**
0	Spremno za uključivanje	r899.0
1	Spremo za rad	r899.1
2	Rad omogućen	r899.2
3	Greška prisutna	r2139.3
4	Nema slobodnog klizanja	r899.4
5	Brzo zaustavljanje neaktivno	r899.5
6	Zabrana rada	r899.6
7	Alarm prisutan	r2139.7
8	Aktivna promjena brzine	r2197.7
9	Zahtjev za kontrolu aktivan	r899.9
10	f ili n vrijednost dosegnuta ili premašena	r2199.1
11	Maksimalni moment nije dosegnut	r1407.7
12	Otvorena kočnica	r899.12
13	Nema alarma prevelike temperature	r2135.14
14	Smjer vrtnje motora	r2197.3
15	Nema alarma toplinskog preopterećenja jedinice napajanja	r2135.15

Tablica 3.2. Kontrolna riječ 1 u komunikaciji pretvarač ⇨ PLC [7]

INT	**OPIS**	**PARAMETAR**
1	Stvarna brzina motora	r63

U tablicama 3.3. i 3.4 prikazane su kontrolne riječi za komunikacijski smjer PLC ⇨ pretvarač, tj. riječi za upravljanje pretvaračem.

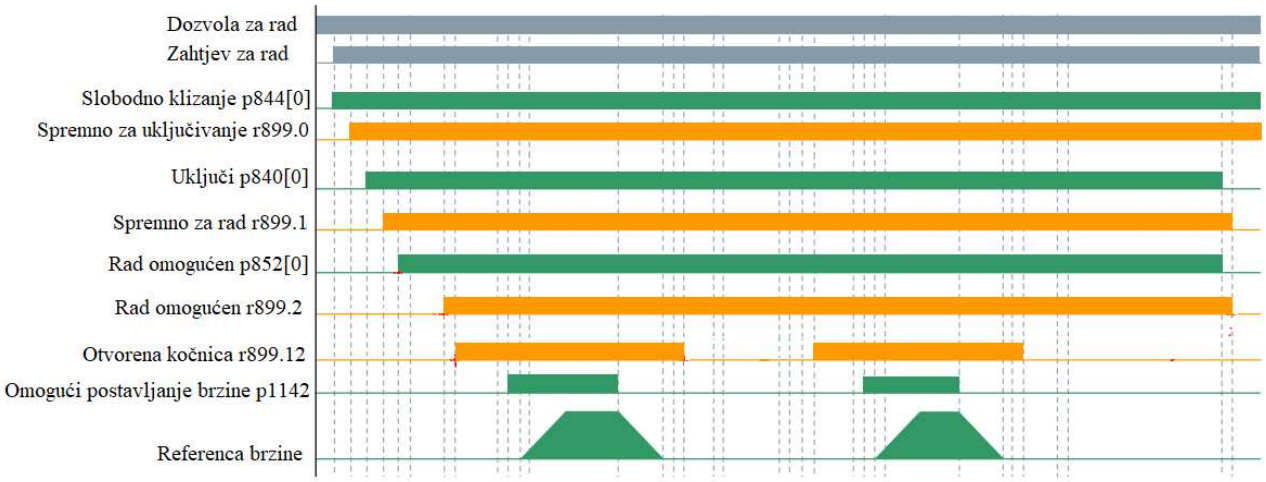
Tablica 3.3. Kontrolna riječ 0 u komunikaciji PLC ⇨ pretvarač [7]

BIT	**OPIS**	**PARAMETAR**
0	Uključi/Isključi	p840[0]
1	Slobodnog klizanja	p844[0]
2	Brzo zaustavljanje	p848[0]
3	Rad omogućen	p852[0]
4	Omogući rampu	p1140[0]
5	Nastavi kontrolu po rampi	p1141[0]
6	Omogući postavljanje brzine	p1142[0]
7	Potvrda greške	p2103[0]
8	Rezervirano	
9	Rezervirano	
10	Kontrola sa PLC-a	p854[0]
11	Inverzija postavljene brzine	p1113[0]
12	Rezervirano	
13	Povećaj brzinu pomoću potenciometra	p1035[0]
14	Smanji brzinu pomoću potenciometra	p1036[0]
15	Rezervirano	

Tablica 3.4. Kontrolna riječ 1 u komunikaciji PLC ⇨ pretvarač [7]

INT	**OPIS**	**PARAMETAR**
1	Željena brzina	p1070[0]

Svaki pretvarač ima svoju specifičnu startnu sekvencu koja zahtijeva precizno vrijeme i ispravan redoslijed slanja komandi prema pretvaraču. U slučaju SINAMICS S120 pretvarača namijenjenih za upravljanje kranovima, ključna startna sekvenca prikazana je na slici 3.10.



Slika 3.10. Prikaz startne sekvence SINAMICS-S120 pretvarača u radu s motorima krana [Autor, 2]

Kako bi se realizirala ovakva komunikacija bilo je potrebno izraditi blok za komunikaciju i upravljanje pretvarača. Na ulazu u funkciju prikazanu na slici 3.12. dolaze sve vrijednosti potrebne za upravljanje pretvaračem.

Ulazi su redom:

- EMERGENCY_OK – Signal koji dolazi iz sigurnosnog dijela logike te onemogućava rad ukoliko je neka od sigurnosnih tipki pritisnuta.
- RUN_PERMISSIVE_OK – Provjerava jesu li svi uvjeti za start pretvarača i motora zadovoljeni.
- RUN_REQUEST – Ukoliko je dana bilo koja komanda za start, u ovom slučaju za gore ili dolje, pošalji zahtjev za startom pretvarača.
- BRAKE_OPN – Povrat iz polja da je kočnica otvorena.
- FW – Komanda za smjer vrtnje motora naprijed.
- RW – Komanda za smjer vrtnje motora unazad.
- FAULT_RESET – Komanda za resetiranje alarma koja dolazi s HMI-a.
- SPEED_TARGET_RPM – Željena brzina vrtnje motora u o/min.
- MOTOR_MAX_RPM – Maksimalna brzina vrtnje motora u o/min.

- MOTOR_MAX_MMPS – Maksimalna brzina linearne kretnje motora u mm/s.
- DP_ADRESS – Adresa pretvarača.
- SIM_ON – Provjerava je li simulacija krana uključena.
- RampFunction_Instance – Podatkovni blok za kontrolu rampe u internoj simulaciji pretvarača.

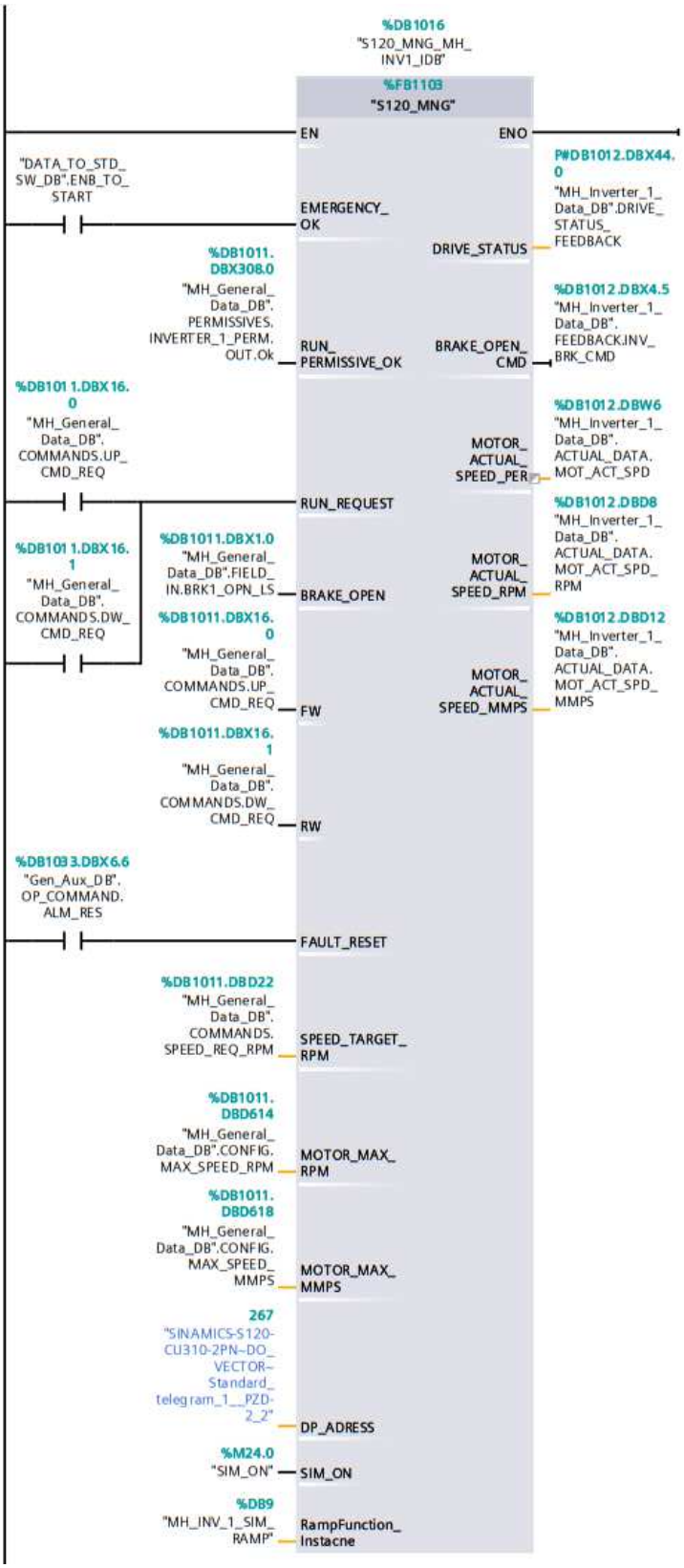
Izlazi iz bloka:

- DRIVE_STATUS – Povrat informacija pročitanih s pretvarača.
- BRAKE_OPEN_CMD – Komanda za otvaranje kočnice koja dolazi iz pretvarača.
- MOTOR_ACTUAL_SPEED_PER – Stvarna brzina vrtnje u postotku.
- MOTOR_ACTUAL_SPEED_RPM - Stvarna brzina vrtnje u o/min.
- MOTOR_ACTUAL_SPEED_MMPS - Stvarna linearna brzina u mm/s.

Kako bi se lakše komuniciralo s pretvaračem i kasnije koristilo podatke sa pretvarača izrađen je poseban tip podatka „SINAMICS_S120_STATUS“ koji sadrži sve podatke koji su potrebni za rad sustava. Slika 3.11. prikazuje sve podatke sadržane u izrađenom tipu podatka.

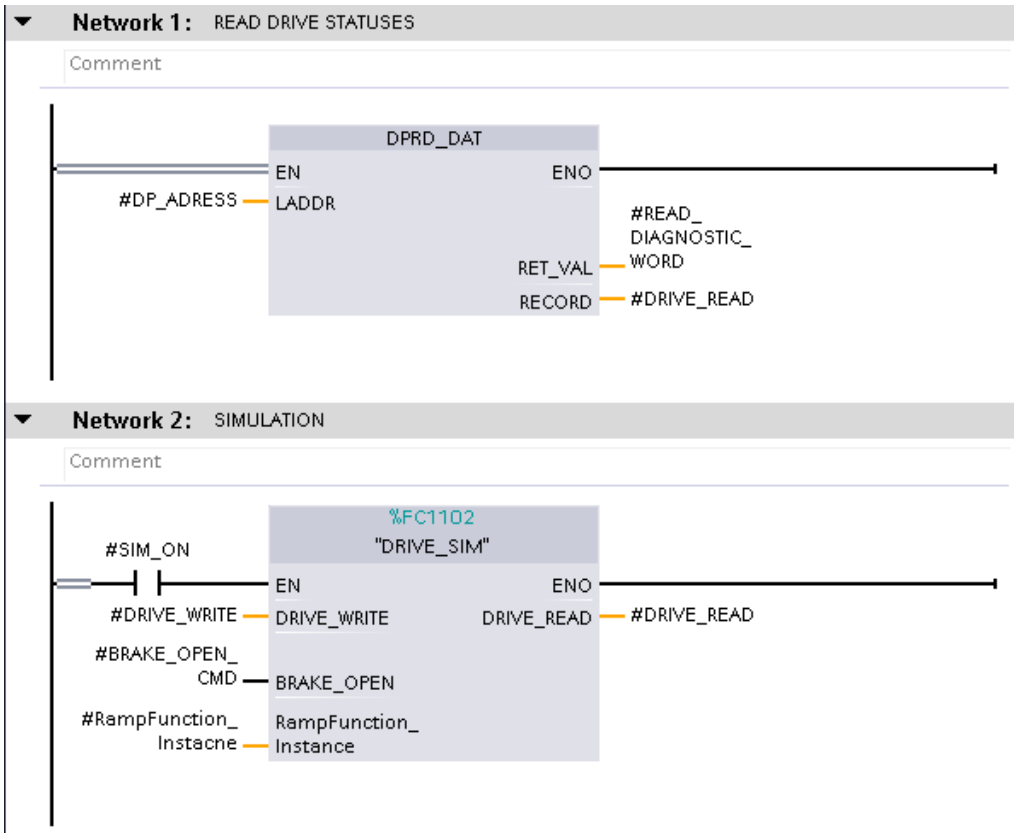
6	DRIVE_STATUS_FEEDBACK	"SINAMICS_S120_STATUS"
7	RDY_FOR_SWCH_ON	Bool
8	READY	Bool
9	OP_ENABLED	Bool
10	FLT_PRESENT	Bool
11	PWR_ON_INHIB_ACT	Bool
12	CTRL_REQ	Bool
13	BRAKE_HOLD_OPN	Bool
14	NO_MOTOR_TEMP_ALM	Bool
15	NO_ALM_PRESENCE	Bool
16	ZERO_SPEED	Bool
17	FORWARD_DIR	Bool
18	BACKWARD_DIR	Bool
19	SPARE1	Bool
20	SPARE2	Bool
21	SPARE3	Bool
22	SPARE4	Bool

Slika 3.11. Prikaz SINAMICS_S120_STATUS tipa podatka korištenog za čitanje povratnih informacija s pretvarača [Autor]



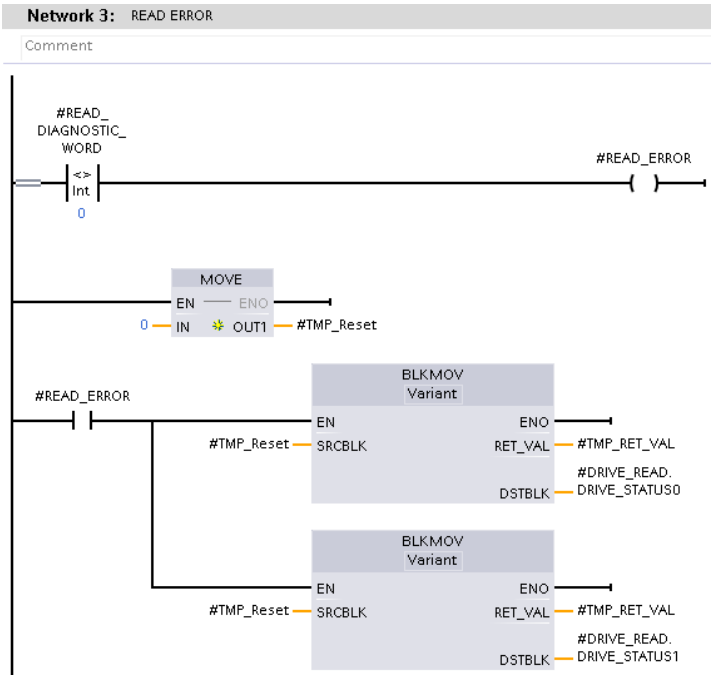
Slika 3.12. Prikaz S120_MNG funkcijskog bloka za kontrolu SINAMICS S120 pretvarača [Autor]

U nastavku je opis funkcijskog bloka koji upravlja komunikacijom s pretvaračem. Funkcijski blok obuhvaća ključne mehanizme i postupke koji omogućuju učinkovitu interakciju i razmjenu podataka između kontrolnog sustava i pretvarača.



Slika 3.13. Prikaz mreže 1 i 2 S120_MNG bloka [Autor]

Mreža jedan(slika 3.13.) vrši čitanje podataka koji stižu s pretvarača pomoću Siemens-ovog bloka DPRD_DAT i sprema ih u lokalnu pomoćnu varijablu kako bi mogli te podatke dalje koristiti u funkciji, također sprema se i povratna riječ nakon čitanja podataka kako bi se utvrdilo ukoliko je komunikacija uredu. Ukoliko je čitanje podataka izvršeno bez greške povratna riječ će biti jednaka nuli. Mreža dva provjerava je li uključena simulacija, ukoliko je DRIVE_SIM blok simulira podatke koje šalje pretvarač na temelju onih koje PLC šalje prema njemu.



Slika 3.14. Prikaz mreže 3 S120_MNG bloka [Autor]

Treća mreža(slika 3.14.) provjerava je li komunikacija s pretvaračem uspješna, ukoliko nije, povratna riječ će biti različita od nule i u tom slučaju potrebno je resetirati sve bitove koji čitaju podatke na nulu, to se radi pomoću bloka BLKMOV koji odjednom prebriše sve bitove na nulu umjesto pojedinačno.

Sljedeće dvije mreže sadržavaju komande koje su uvijek uključene ili isključene.

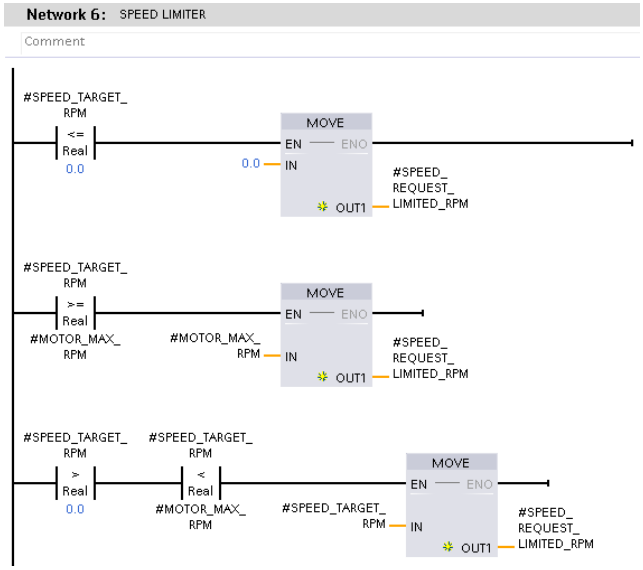
Uvijek uključene:

- Slobodno klizanje
- Kontrola sa PLC-a
- Omogući rampu
- Nastavi kontrolu po rampi

Uvijek isključene:

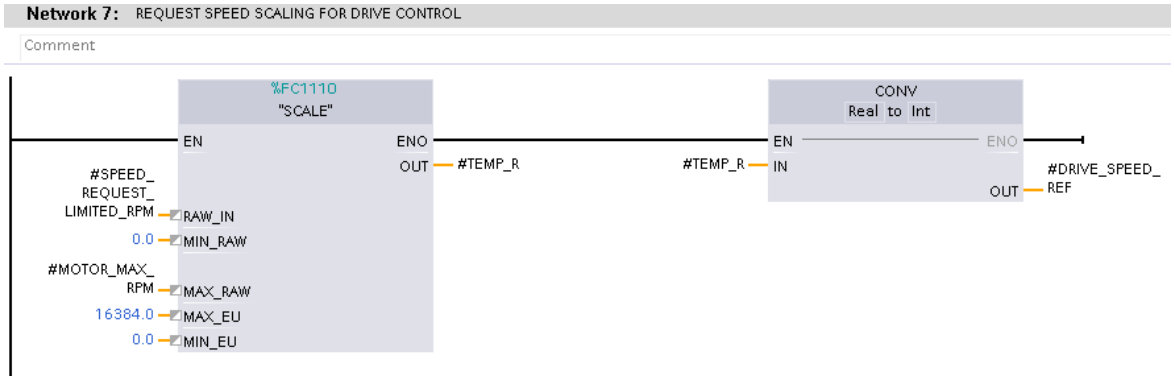
- Povećaj brzinu pomoću potenciometra
- Smanji brzinu pomoću potenciometra

Pri upravljanju brzinom, nužno je uvesti limitatore brzine kako bismo izbjegli premašivanje nazivnih vrijednosti motora i ostale opreme. Ograničenje brzine se ostvaruje u okviru mreže 6 (slika 3.15.). Kada je željena ulazna brzina manja od nule, postavljamo brzinu na nulu. U slučaju kada je brzina veća od maksimalno dopuštene, postavljamo je na tu maksimalnu vrijednost. Ako se nalazi između ovih granica, postavljamo je na odgovarajuću vrijednost unutar tog raspona.



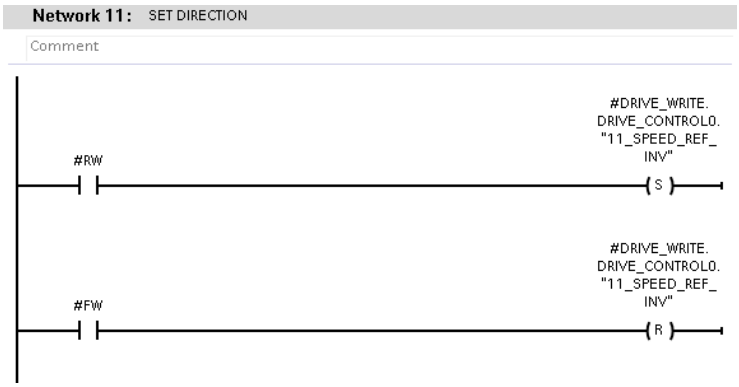
Slika 3.15. Prikaz mreže 6 S120_MNG bloka [Autor]

Nakon što se brzina ograniči, potrebno ju je skalirati kako bi se dobila izlazna vrijednost unutar raspona od 0 do 16384. Ovaj postupak se provodi pomoću izrađenog SCALE bloka unutar mreže 7 (slika 3.16.). Slično kao što je bila skalirana ulazna referentna brzina, isti postupak primjenjuje se na stvarnu trenutnu brzinu vrtnje motora, koja se čita s pretvarača. U mrežama 8, 9 i 10 ta brzina se pretvara u postotak, okretaje po minuti i linearnu brzinu osi kojom pretvarač upravlja.



Slika 3.16. Prikaz mreže 7 S120_MNG bloka [Autor]

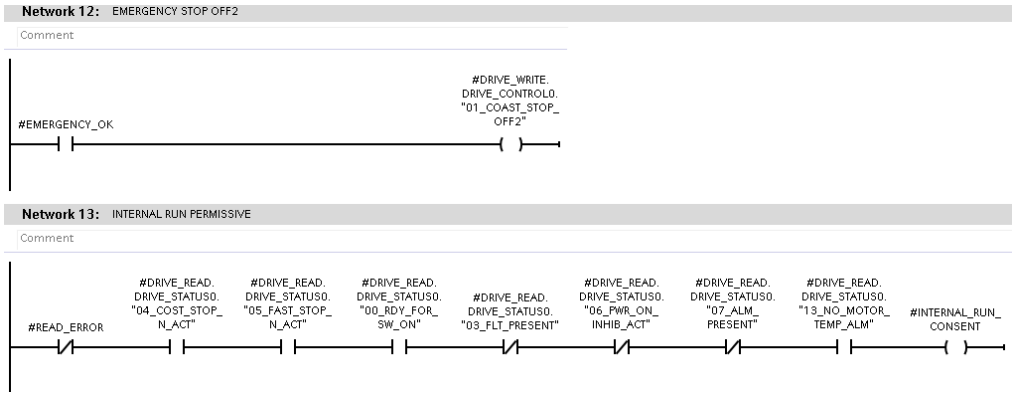
U mreži 11 (slika 3.17.) određuje se smjer vrtnje motora prema ulazima u funkcijski blok. U slučaju da je zadan smjer vrtnje unazad, tag koji se piše u pretvarač te upravlja smjerom vrtnje postavlja se u jedinicu te se time mijenja i smjer vrtnje motora, u suprotnom tag ostaje nula i smjer ostaje prema naprijed.



Slika 3.17. Prikaz mreže 11 S120_MNG bloka [Autor]

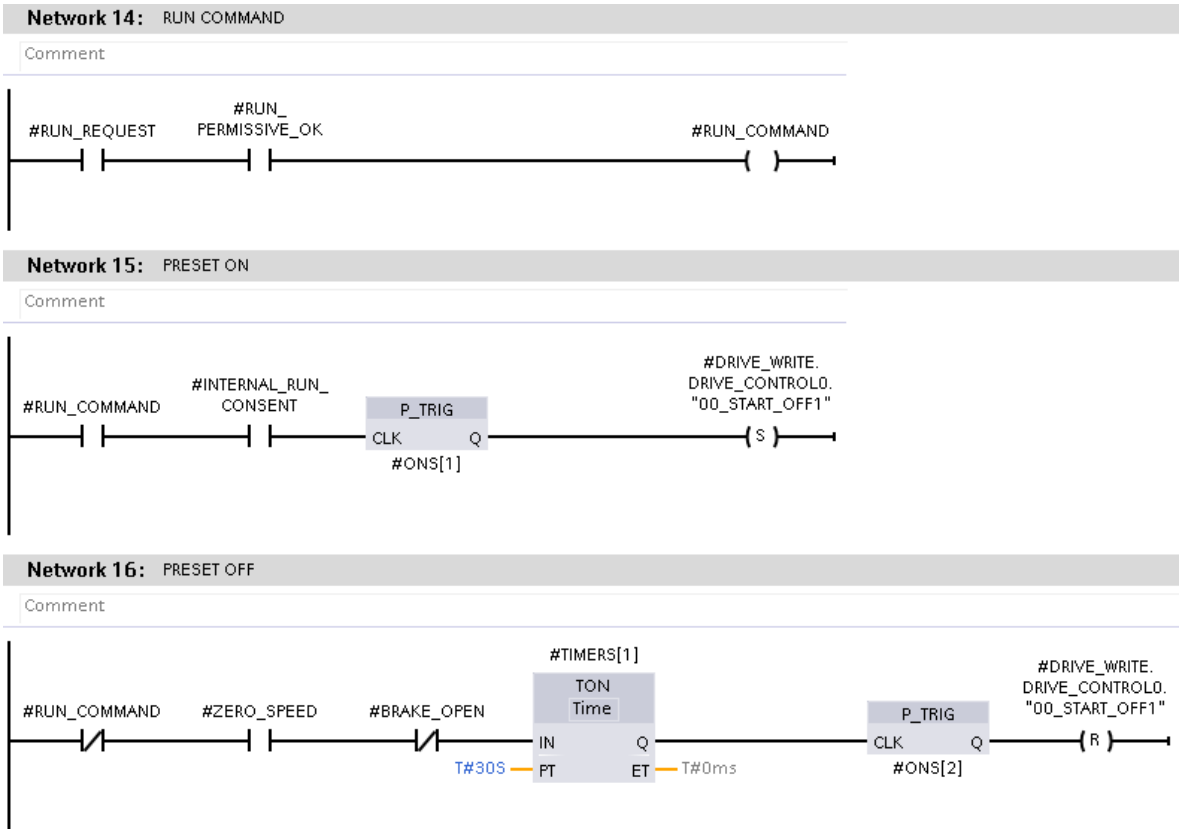
Nakon što se odredi smjer vrtnje, provjerava se jesu li ispunjeni svi uvjeti za pokretanje pretvarača. Ovi uvjeti su podijeljeni na vanjske i unutarnje. Vanjski uvjeti za rad dolaze kao jedan tag i ulaz u blok, dok unutarnje uvjete provjeravamo unutar samog bloka, to jest prema stanjima koja se čitaju s pretvarača. Također, potrebno je provjeriti je li sigurnosna tipka aktivna; u slučaju da je, pretvarač prestaje s radom. Sigurnosna komponenta se nalazi u mreži 12, dok mreža 13 (slika 3.18.) provjerava unutarnju dozvolu za pokretanje.

Unutarnji uvjeti koji moraju biti zadovoljeni za normalan početak rada su: nema pogrešaka u čitanju podataka, slobodno klizanje nije aktivno, brzo zaustavljanje nije aktivno, pretvarač je spreman za uključivanje, nema aktivnih grešaka, nema zabrane uključivanja, nema aktivnih alarma i temperatura motora je unutar normalnog raspona.



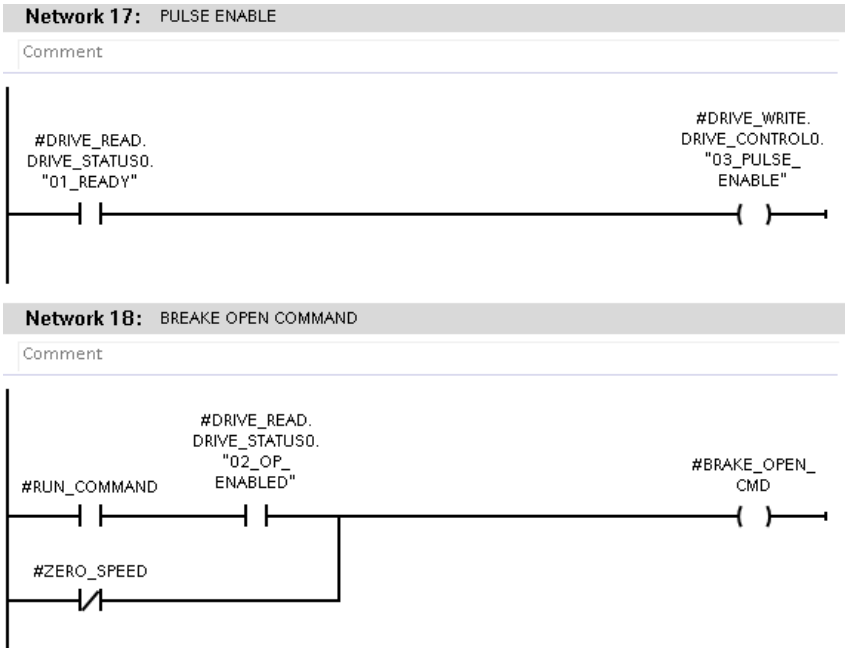
Slika 3.18. Prikaz mreže 12 i 13 S120_MNG bloka [Autor]

Nakon provjere unutarnjih startnih uvjeta, potrebno je provjeriti je li uopće dana vanjska komanda za start motora, ukoliko je, te ako su vanjski startni uvjeti također zadovoljeni pretvarač se postavlja u startno, spremno stanje(*eng. Preset on*). Kod kranova je bitno da „preset“ stanje ostane uključeno neko vrijeme iako nema vanjske start komande, upravo zato jer se kran ne miče konstantno, već operater ima pauze između micanja osi krana. Konkretno „preset“ ovdje ostaje uključen 30 sekundi nakon zadnje aktivnosti, to se vidi na slici 3.19. pod mrežom 16.



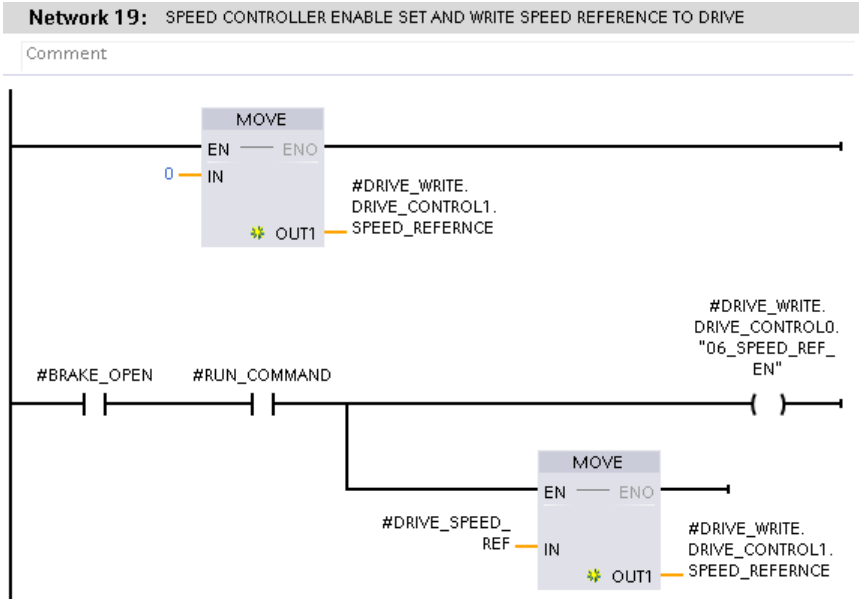
Slika 3.19. Prikaz mreže 14, 15 i 16 S120_MNG bloka [Autor]

Kada s pretvarača stigne informacija da je motor spreman za rad, motor dolazi pod napon(*eng Motor energized*) te se potom daje komanda za otvaranje kočnice koja se drži otvorenom sve dok brzina motora ne padne na nulu ili se i dalje daje komanda za start. To prikazuje slika 3.20..



Slika 3.20. Prikaz mreže 17 i 18 S120_MNG bloka [Autor]

Nakon potvrde da se kočnica otvorila zapisuje se referentna brzina vrtnje motora u pretvarač, što je vidljivo u mreži 19 na slici 3.21.. Sljedećih nekoliko mreža služi za resetiranje grešaka pretvarača te prepisivanje statusa na izlaz bloka, kao i pisanje komandi u pretvarač pomoću DPWR_DAT bloka što se vidi na slici 3.22..



Slika 3.21. Prikaz mreže 19 S120_MNG bloka [Autor]

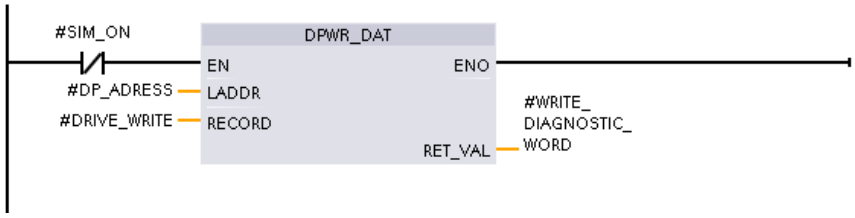
Network 23: OUTPUT STATUS WRITE

Comment

```
#DRIVE_STATUS.RDY_FOR_SWCH_ON := #DRIVE_READ.DRIVE_STATUS0."00_RDY_FOR_SW_ON";
#DRIVE_STATUS.READY := #DRIVE_READ.DRIVE_STATUS0."01_READY";
#DRIVE_STATUS.OP_ENABLED := #DRIVE_READ.DRIVE_STATUS0."02_OP_ENABLED";
#DRIVE_STATUS.FLT_PRESENT := #DRIVE_READ.DRIVE_STATUS0."03_FLT_PRESENT";
#DRIVE_STATUS.PWR_ON_INHIB_ACT := #DRIVE_READ.DRIVE_STATUS0."04_COST_STOP_N_ACT";
#DRIVE_STATUS.CTRL_REQ := #DRIVE_READ.DRIVE_STATUS0."09_CTRL_REQ";
#DRIVE_STATUS.BRAKE_HOLD_OPN := #DRIVE_READ.DRIVE_STATUS0."12_BRAKE_HOLD_OPN";
#DRIVE_STATUS.NO_MOTOR_TEMP_ALM := #DRIVE_READ.DRIVE_STATUS0."13_NO_MOTOR_TEMP_ALM";
#DRIVE_STATUS.NO_ALM_PRESENCE := #DRIVE_READ.DRIVE_STATUS0."15_NO_ALM_PRESENCE";
#DRIVE_STATUS.ZERO_SPEED := #ZERO_SPEED;
```

Network 24: WRITE TO DRIVE

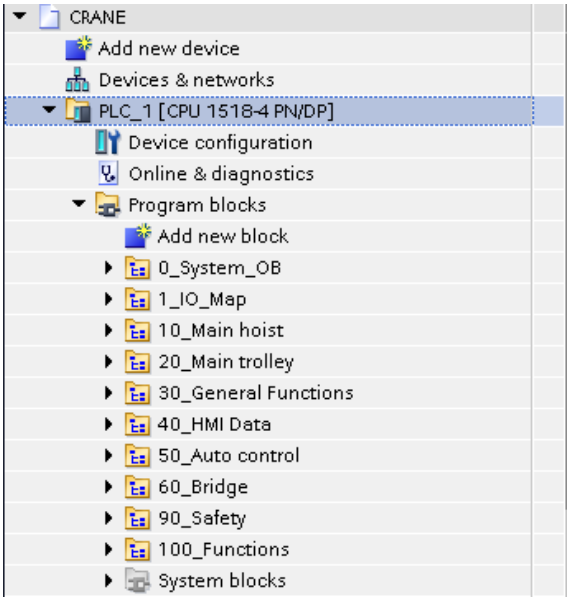
Comment



Slika 3.22. Prikaz mreže 23 i 24 S120_MNG bloka [Autor]

3.5. Organizacija programskih blokova i upravljačka logika sustava

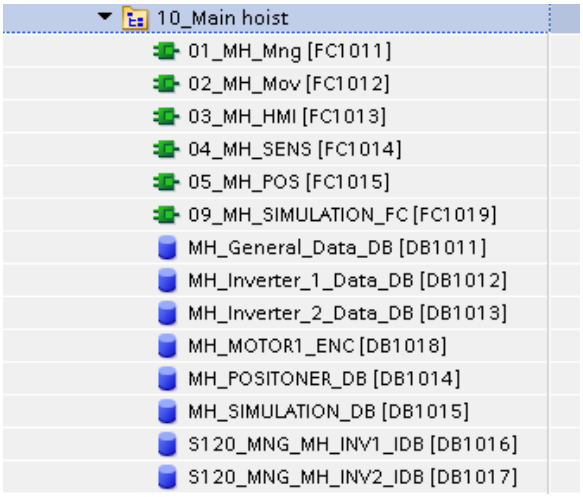
Organizacija logike sustava je strukturirana putem programskih blokova, što je prikazano na slici 3.23. Unutar mape 0_System_OB smještena je osnovna funkcija "Main" ili OB1 koja uključuje sve potrebne funkcije za izvođenje i automatizaciju stroja. "Main" funkcija predstavlja centralnu funkciju PLC-a i sve što se nalazi unutar nje izvršava se svakog ciklusa. Trajanje ciklusa varira ovisno o performansama PLC-a i količini logike, odnosno programskog koda koji se izvršava. U okviru ovog projekta, minimalno trajanje ciklusa postavljeno je na 5ms, dok je maksimalno trajanje 150ms.



Slika 3.23. Organizacija projekta po mapama [Autor]

Mapa 1_IO_Map sadrži funkcije koje mapiraju fizičke ulaze i izlaze PLC-a na podatkovne blokove, organizirajući ih prema specifičnim dijelovima stroja. To omogućava lakšu navigaciju i bolju organizaciju. Unutar mape 30_General Functions nalaze se funkcije koje povezuju različite dijelove stroja. Mapa 40_HMI Data Block sadrži podatkovne blokove koji služe kao veza između HMI-a i PLC-a. U mapi 50_Auto Control smještena je automatska sekvenca i funkcija za automatsko upravljanje. Mape 90_Safety i 100_Functions su općenite mape u koje su pohranjene sve funkcije i podatkovni blokovi koji se koriste u ostatku logike.

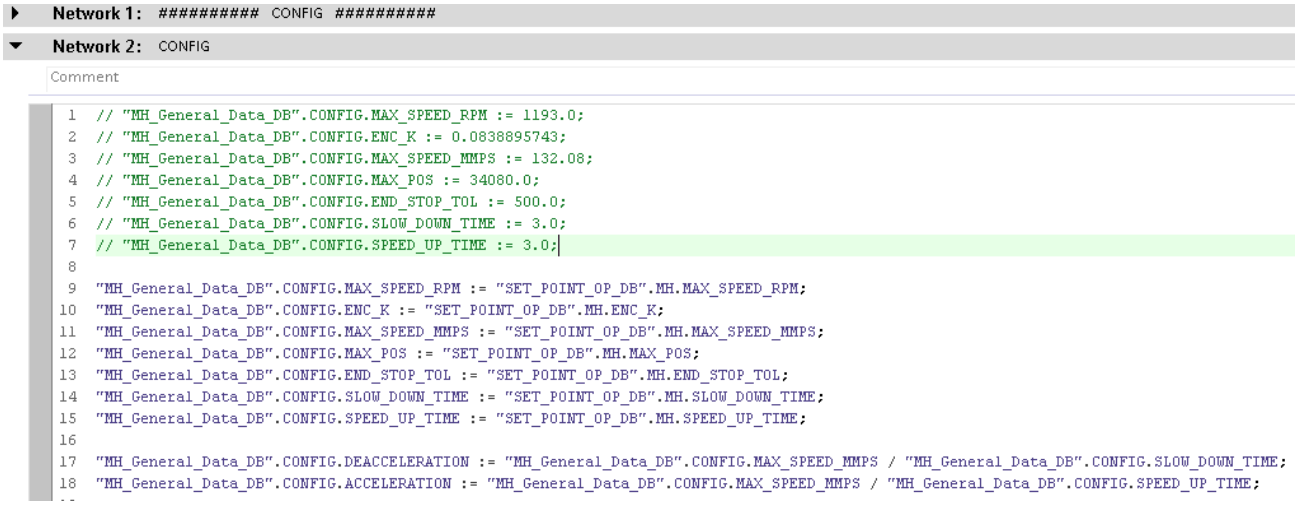
Mape dijelova stroja su sve organizirane na isti način, kao primjer je dano glavno vitlo, 10_Main hoist, te se na slici 3.24. vidi njegova organizacija.



Slika 3.24. Organizacija mape glavnog vitla [Autor]

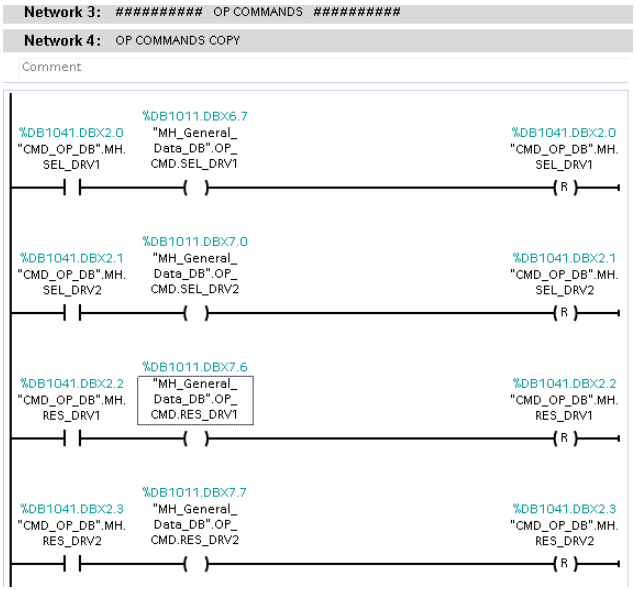
Kao što je već spomenuto sustav ima dva načina rada, ručni i automatski. Od toga se ručni koristi 90% vremena u upravljanju kranovima dok se automatski koristi samo za određene situacije koje zahtijevaju veliku preciznost te se ponavljaju. U nastavku će biti objašnjeno ručno pokretanje kompletnog glavnog vitla, no na istom principu te uz vrlo sličnu logiku rade i ostali dijelovi stroja kao što su kranska kolica i most.

Funkcija 01_MH_Mng nam služi kao poveznica među različitim načinima upravljanja i svim pojedinim dijelovima vitla. Prvo što je potrebno kako bi sustav ispravno funkcionirao jest napraviti konfiguraciju te unesti sve potrebne informacije u upravljačku logiku. Taj dio se učitava sa HMI-a te će izgled stranice i unos biti prikazan kasnije u radu. U mreži 2 na slici 3.25. vidimo prepisivanje tih podataka sa HMI-a te iznad zakomentirane linije koda koje prikazuju njihove zadane vrijednosti.



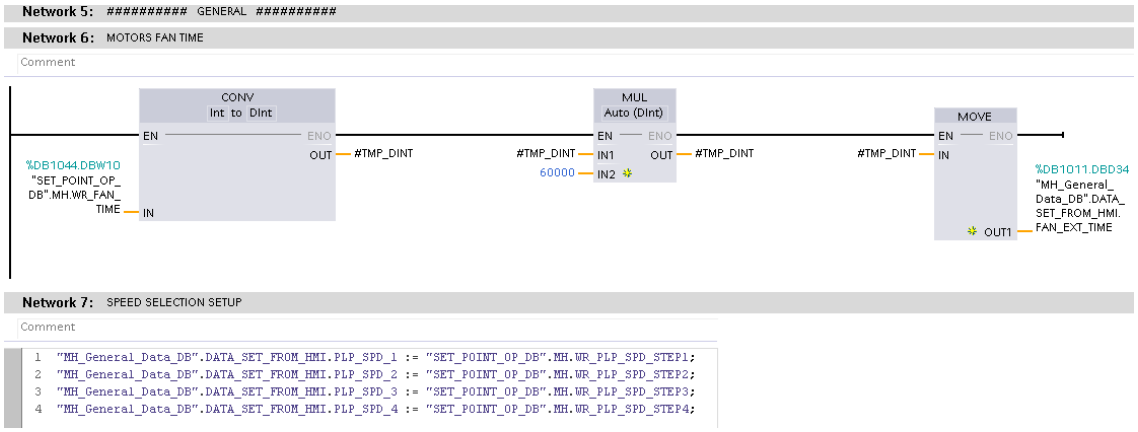
Slika 3.25. Konfiguracija glavnog vitla [Autor]

Nakon konfiguracije je potrebno očitati komande koje dolaze sa HMI-a i koje se koriste u ovoj funkciji. To su komande za odabir pojedinačnog pretvarača te komande za resetiranje radnih sati svakoga od njih. Komande rade na principu da su aktivne samo jedan ciklus rada PLC-a te su prikazane na slici 3.26..

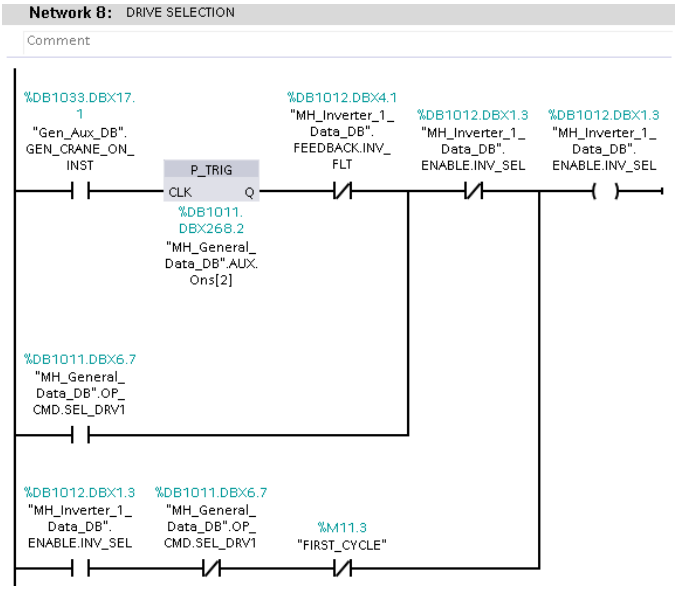


Slika 3.26. Komande sa HMI-a [Autor]

Kada su se sve komande očitale potrebno je postaviti opće postavke stroja, kao što su vrijeme rada ventilatora motora nakon isključivanja motora te četiri razine brzine (slika 3.27.). Postoje četiri različite razine brzine jer ručna upravljačka palica(*eng. joystick*) za upravljanje svakom osi ima upravo četiri prekidača tj. četiri moguće brzine od mirovanja do gibanja punom brzinom. Također kao i ostale postavke brzinu je potrebno namjestiti na HMI sučelju. U općem dijelu logike se odvija i odabir pretvarača na slici 3.28..

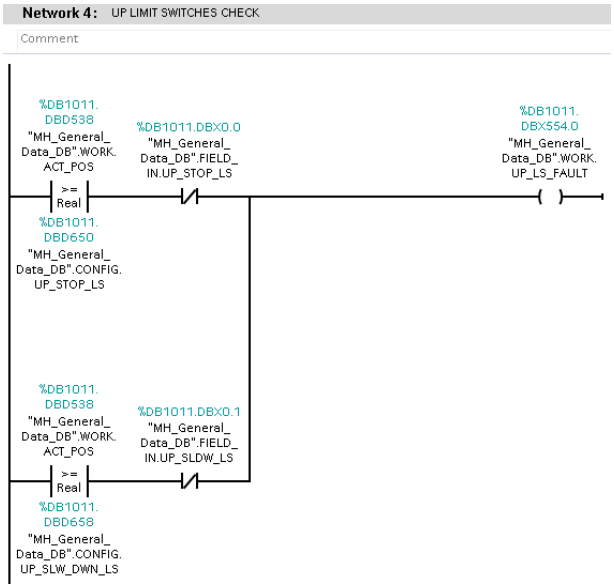


Slika 3.27. Postavke ventilatora i ručnih brzina[Autor]



Slika 3.28. Odabir pretvarača 1[Autor]

Nakon što su osnovne postavke vitla konfigurirane, potrebno je definirati pokrete i njihove brzine pri kretanju. Ovaj proces se obavlja u posebnoj funkciji 02_MH_Mov, koju pozivamo unutar glavne funkcije 01_MH_Mng. Prvi korak je provjera graničnih prekidača (*eng. Limit switch*). Ova provjera se obavlja na dva načina: prvo, provjerava se je li krajnji prekidač pritisnut, a drugo, provodi se provjera pozicije putem enkodera. U slučaju da je pozicija prešla određenu vrijednost na kojoj bi trebao biti granični prekidač, stroj signalizira grešku te prekida kretanje u tom smjeru jer granični prekidač nije reagirao te se javlja alarm. Ova situacija je prikazana na slici 3.29..

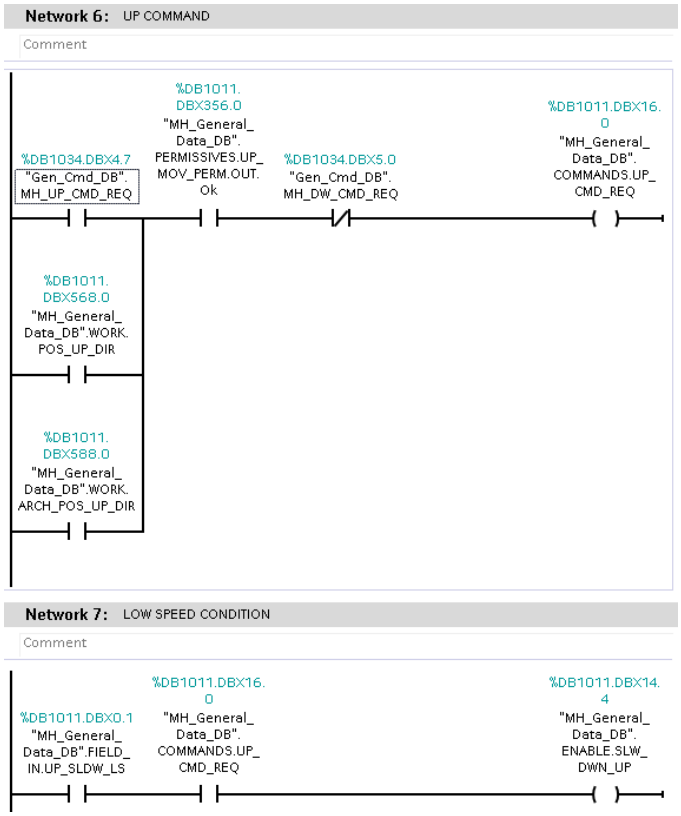


Slika 3.29. Provjera graničnih prekidača[Autor]

Kao primjer uzeta je kretnja vitla prema gore, no i ostale kretnje su vrlo slične. Za kretnju prema gore potrebno je zadovoljiti nekoliko uvjeta, a to su:

- Nije pritisnut krajnji prekidač za zaustavljanje
- Eksterna vaga vitla je u funkciji
- Vitlo nije u preopterećenju
- Prekidači za preklapanje sajle nisu aktivni
- Pretvarač 1 ili 2 su odabrani na HMI sučelju
- Svi krajnji prekidači su u funkciji

Ako su svi uvjeti ispunjeni, provjerava se dolazi li naredba za pokretanje dizanja vitla prema gore, bilo putem ručne upravljačke palice ili putem jedne od automatskih sekvenci. Naravno, ako je aktivno spuštanje vitla prema dolje, naredba za podizanje će biti onemogućena. Ova provjera se izvodi u mreži 6, što je prikazano na slici 3.30. Paralelno s tim, u mreži 7 se jednostavno provjerava status krajnjeg prekidača za usporavanje brzine vitla.

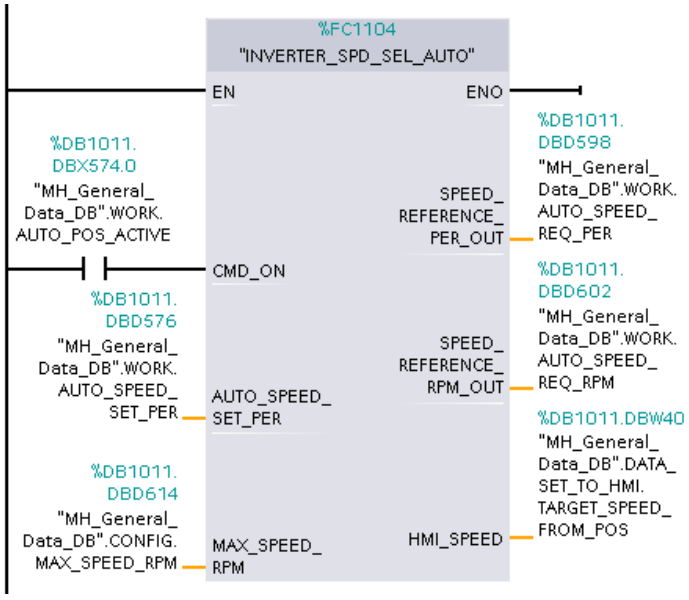


Slika 3.30. Komanda za start gibanja vitla[Autor]

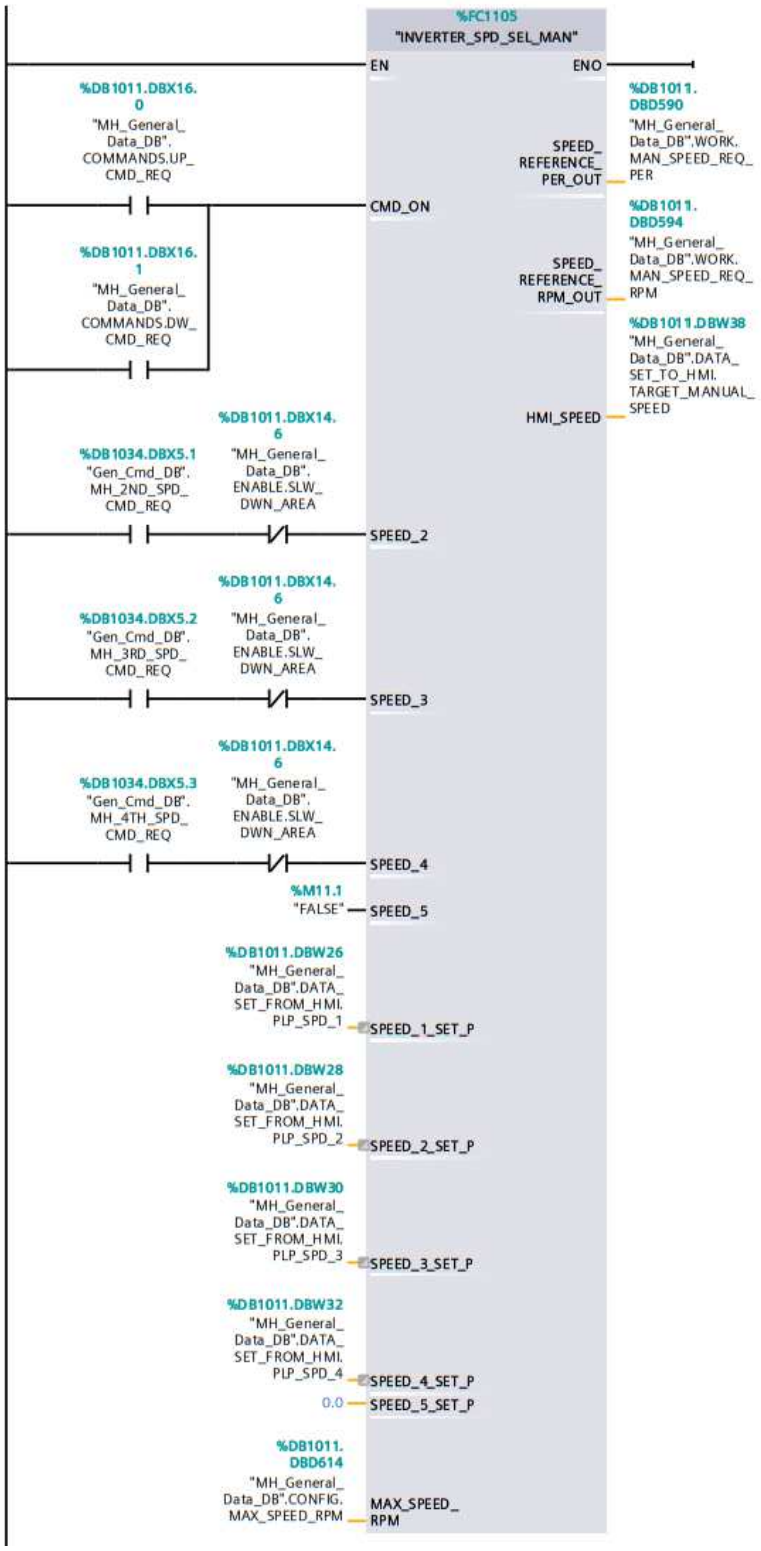
Nakon pozivanja funkcije 02_MH_Mov kojom se upravlja kretanjem vitla, potrebno je definirati dodatne postavke i uvjete prije slanja informacija pretvaraču za upravljanjem vitlom. Jedna od tih ključnih informacija je brzina vrtnje motora, koja se dobiva iz različitih izvora, ovisno o trenutnom načinu rada. Konačna brzina vrtnje određuje se kroz dva bloka: „INVERTER_SPD_SEL_MAN“ i „INVERTER_SPD_SEL_AUTO“.

Konkretno, ručni selektor brzine, kao što je prikazano na slici 3.32., kada prima smjernu naredbu s ručne upravljačke palice, postavlja brzinu na izlazu bloka na istu vrijednost kao što je unesena na ulazu za SPEED_1_SET_P. Svaka daljnja razina upravljačke palice povećava brzinu prema predefiniranim vrijednostima na ulazu. Ukoliko je aktiviran krajnji prekidač za usporavanje, brzina na izlazu će biti postavljena na najsporiju predefiniranu brzinu.

Automatski selektor brzine, slično kao ručni selektor, prima ulaznu naredbu samo ako su aktivni automatski načini rada, te preslikava ulaznu brzinu iz postotaka na izlaz u okretajima po minuti. Ova brzina je bitna za komunikaciju s pretvaračem. Zadane brzine unesene u pretvarač bit će one iz ručnog načina rada ako automatski način nije aktivan; u suprotnom, bit će preuzete iz „INVERTER_SPD_SEL_AUTO“ bloka sa slike 3.31..



Slika 3.31. Blok INVERTER_SPD_SEL_AUTO [Autor]



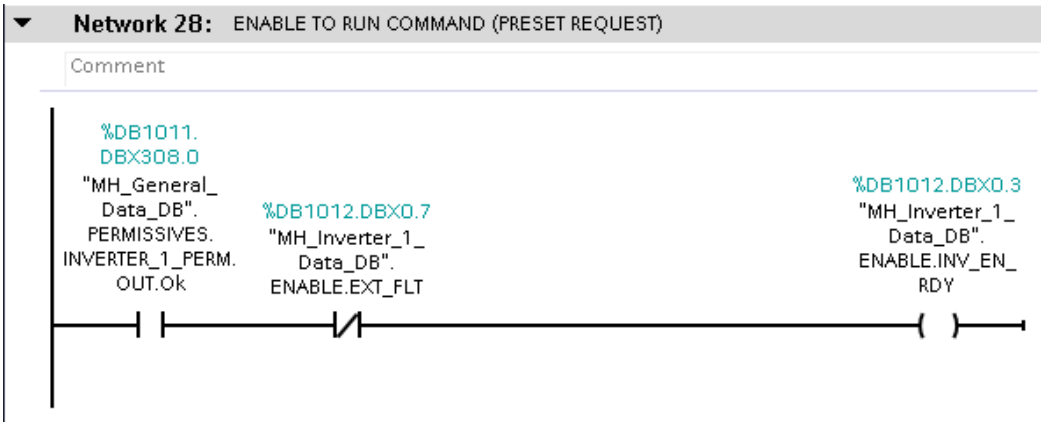
Slika 3.32. Blok INVERTER_SPD_SEL_MAN [Autor]

Nakon određivanja referentne brzine dolazi se do dijela logike koji upravlja pretvaračem. Kako bi pretvarač uopće mogao biti pokrenut potrebno je zadovoljiti nekoliko uvjeta za pokretanje.

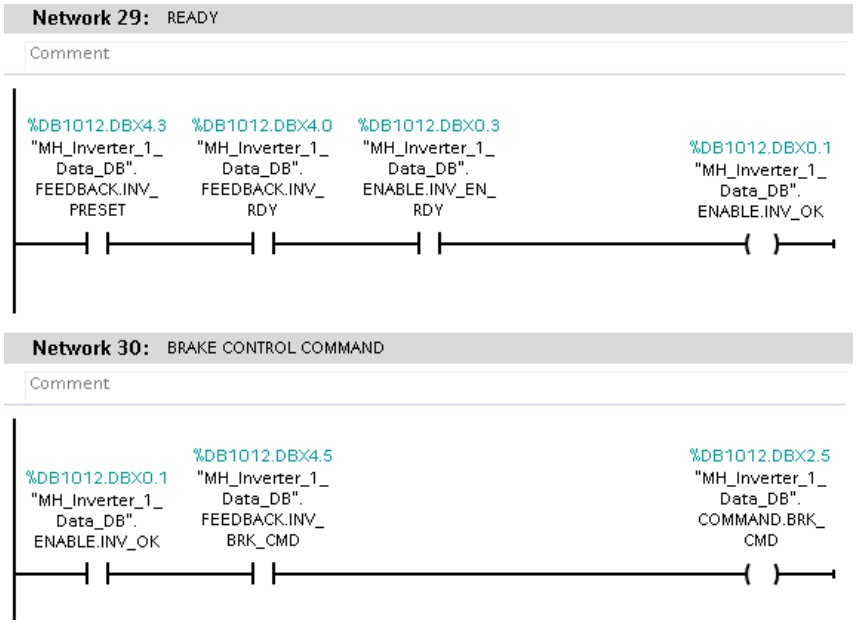
Uvjeti za pokretanje pretvarača 1 prvog motora vitla:

- Odabran je pretvarač 1
- Temperatura motora 1 je u normalnim granicama
- Kočnica 1 je spremna za rad
- Kočnica 2 je spremna za rad
- Alarm za istek vremena pri otvaranju kočnice 1 nije aktiviran
- Alarm za istek vremena pri otvaranju kočnice 2 nije aktiviran
- Povratni prekidač o otvorenosti kočnice 1 u funkciji
- Povratni prekidač o otvorenosti kočnice 2 u funkciji
- Povratne informacije s pretvarača dostupne
- Alarm za istek vremena za magnetiziranje motora neaktivan
- Alarm za istek vremena za start motora je aktivan

Ukoliko su svi uvjeti zadovoljeni te nema vanjskih grešaka koje onemogućuju rad vitla i pretvarača dobiva se OK status koji omogućuje otvaranje kočnica i pokretanje motora. To se vidi na slikama 3.33. i 3.34..



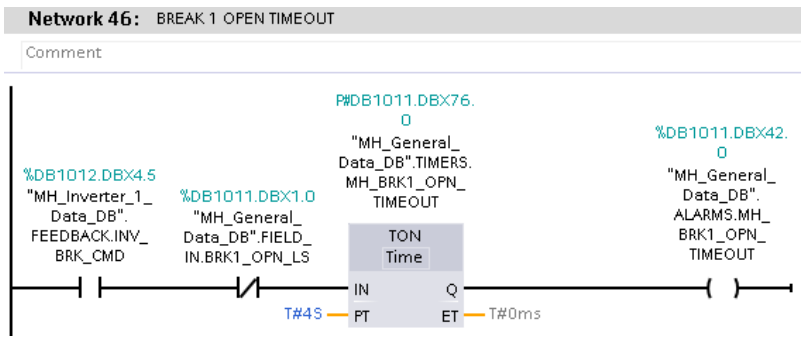
Slika 3.33. Vanjska dozvola za rad pretvarača[Autor]



Slika 3.34. Uvjeti za otvaranje kočnica [Autor]

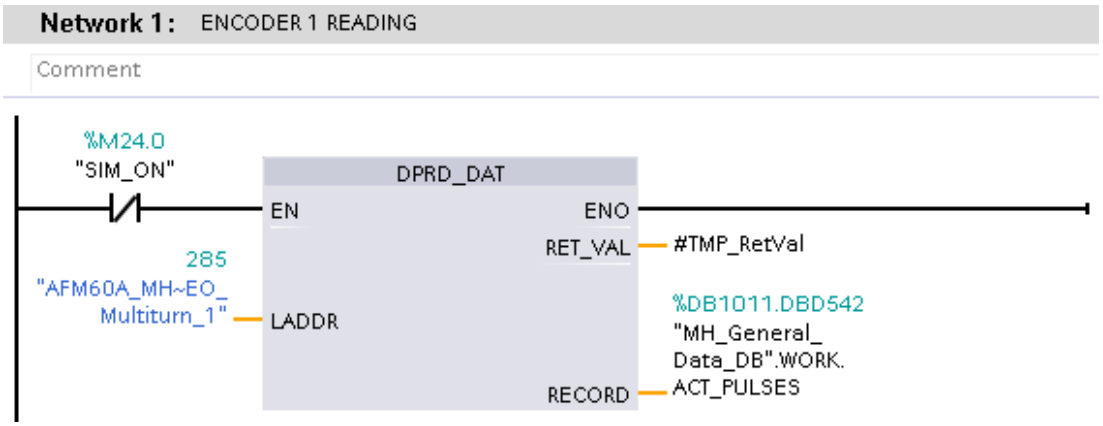
Nakon što su svi potrebni parametri i uvjeti za pokretanje motora i pretvarača ispunjeni, slijedi komunikacija s pretvaračem. Ova faza ostvaruje se putem posebno izrađenog bloka S120_MNG, čija je funkcionalnost detaljno objašnjena u poglavlju 3.4. U slučaju glavnog vitla, postoje dva motora, svaki s vlastitim pretvaračem koji upravlja njegovim radom. Stoga je bilo nužno ponoviti cijelu gornju logiku pokretanja i komunikacije za svaki od pretvarača. Osim toga, budući da postoje dva motora, također je bitno pratiti radne sate oba motora kako bi se njihova upotreba izjednačila, osobito kada ne rade istovremeno. Time se sprječava prekomjerno opterećenje jednog motora dok je drugi neaktivan.

Zadnji korak u logici funkcije 01_MH_Mng uključuje poziv alarma za istek vremena(*eng. Timeout alarm*)(slika 3.35.). Primjerice, tu je postavljen alarm isteka vremena za otvaranje kočnice koji će se aktivirati nakon 4 sekunde ukoliko povratna informacija o otvorenoj kočnici ne stigne u predviđenom roku.



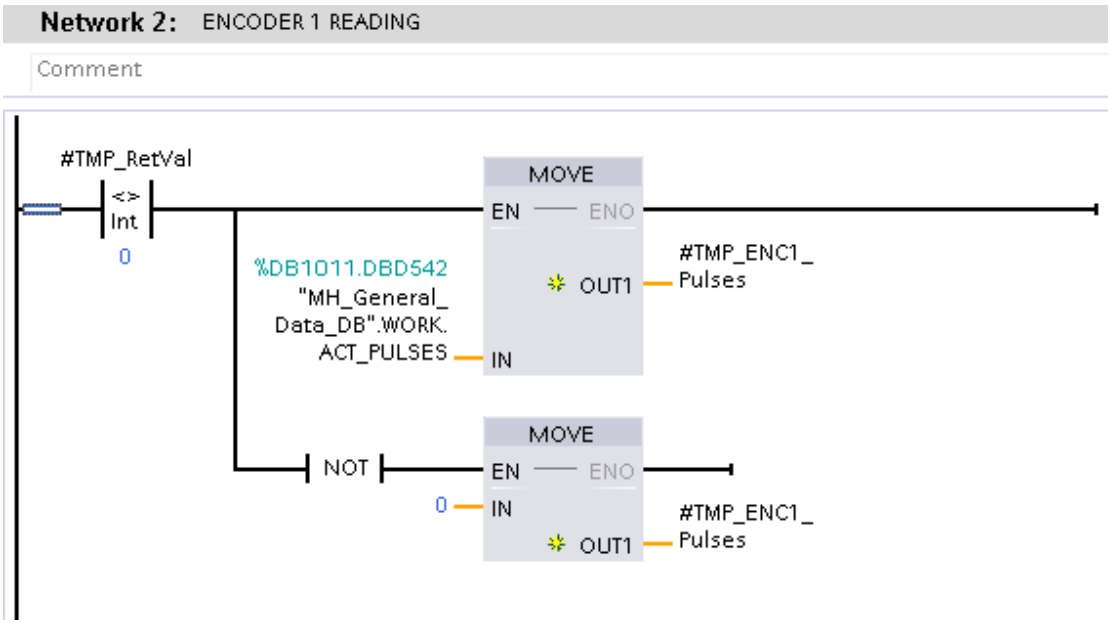
Slika 3.35. Alarm isteka vremena za otvaranje kočnice [Autor]

Još jedna značajna funkcija je 04_MH_SENS, koja provodi očitavanje pozicije s enkodera. Slično kao i kod frekvencijskog pretvarača, i ovdje koristi se DPRD_DAT blok za čitanje informacija koje stižu s enkodera preko PROFINET mreže. Kao što je prikazano na slici 3.36., povratna informacija s enkodera daje trenutni broj impulsa koji označava trenutnu poziciju.



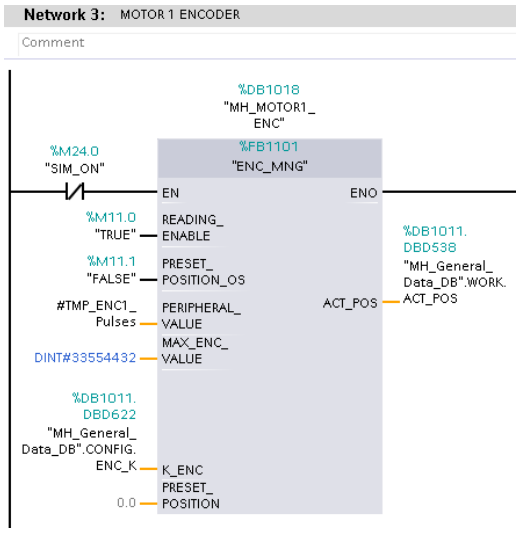
Slika 3.36. Očitavanje informacija s enkodera [Autor]

Također se provodi provjera dobivenih informacija na način da se, ako je povratni kod iz DPRD_DAT bloka različit od nule, upisuje nula u broj pulseva. To se čini kako bi se spriječilo krivo očitavanje pozicije u slučaju problema s komunikacijom. Ova provjera je implementirana u mreži 2, kako je prikazano na slici 3.37.



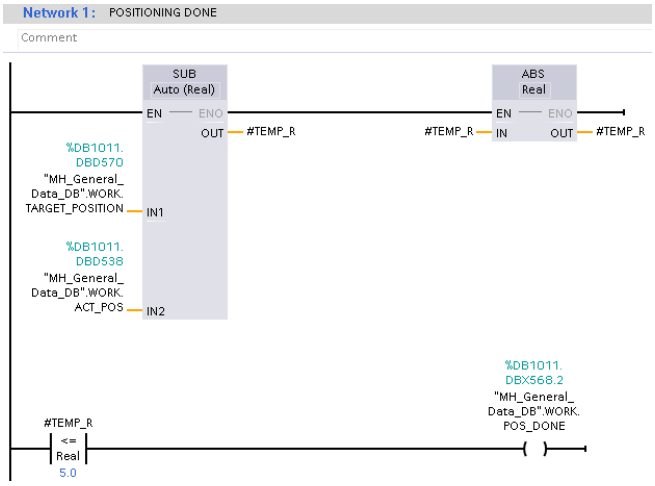
Slika 3.37. Provjera komunikacije s enkoderom [Autor]

Blok za rad s enkoderom funkcionira na vrlo jednostavnom principu te prema promjeni pulseva u sekundi izračuna trenutnu poziciju. Kao ulaz u blok je potrebno unesti samo broj trenutnih enkoderskih pulseva i K vrijednost enkodera koja se izražava u pulsevima po milimetru. Taj blok je vidljiv na slici 3.38.



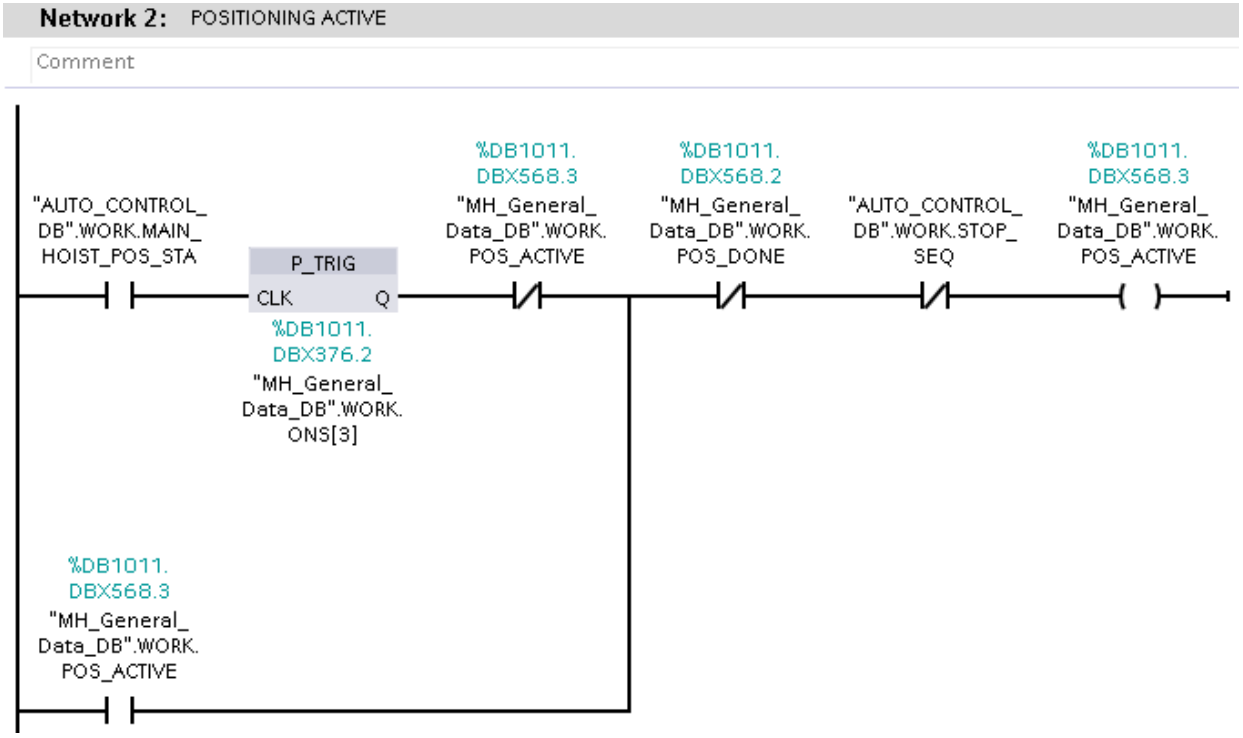
Slika 3.38. Blok za izračun pozicije enkodera [Autor]

Sljedeća važna funkcija u upravljanju glavnim vitlom je funkcija 05_MH_POS. Ova funkcija obuhvaća logiku za automatsko pozicioniranje i djeluje kao dodatni segment unutar automatske upravljačke logike koja upravlja pravocrtnim pokretima samo jedne osi. U prvom dijelu funkcije provjerava se je li proces pozicioniranja završen. To se postiže usporedbom razlike između zadane i trenutne pozicije, te se ukoliko ta razlika iznosi manje od 5 mm prekida komanda za aktivaciju bloka za pozicioniranje. Ovaj proces je ilustriran na slici 3.39.



Slika 3.39. Provjera kraja pozicioniranja [Autor]

Aktivacija bloka za pozicioniranje provodi se putem druge funkcije koja je odgovorna za automatsko upravljanje i koja će biti detaljnije objašnjena kasnije. Ukoliko je primljena naredba za pokretanje te pozicioniranje nije trenutno aktivno ili nije zaustavljeno, tada se pokreće blok za pozicioniranje prema sljedećoj logici prikazanoj na slici 3.40.



Slika 3.40. Komanda za aktiviranje bloka za pozicioniranje [Autor]

Blok za pozicioniranje „POSITIONER“ na slici 3.41.. Njegova logika je napisana u SCL programskom jeziku radi jednostavnosti i potrebnih proračuna. Funkcija mu je da na temelju ulaznih parametara kao što su trenutna brzina i pozicija dovede vitlo u točno zadanu poziciju upravljanjem brzine motora.

Ulazni parametri „POSITIONER“ bloka:

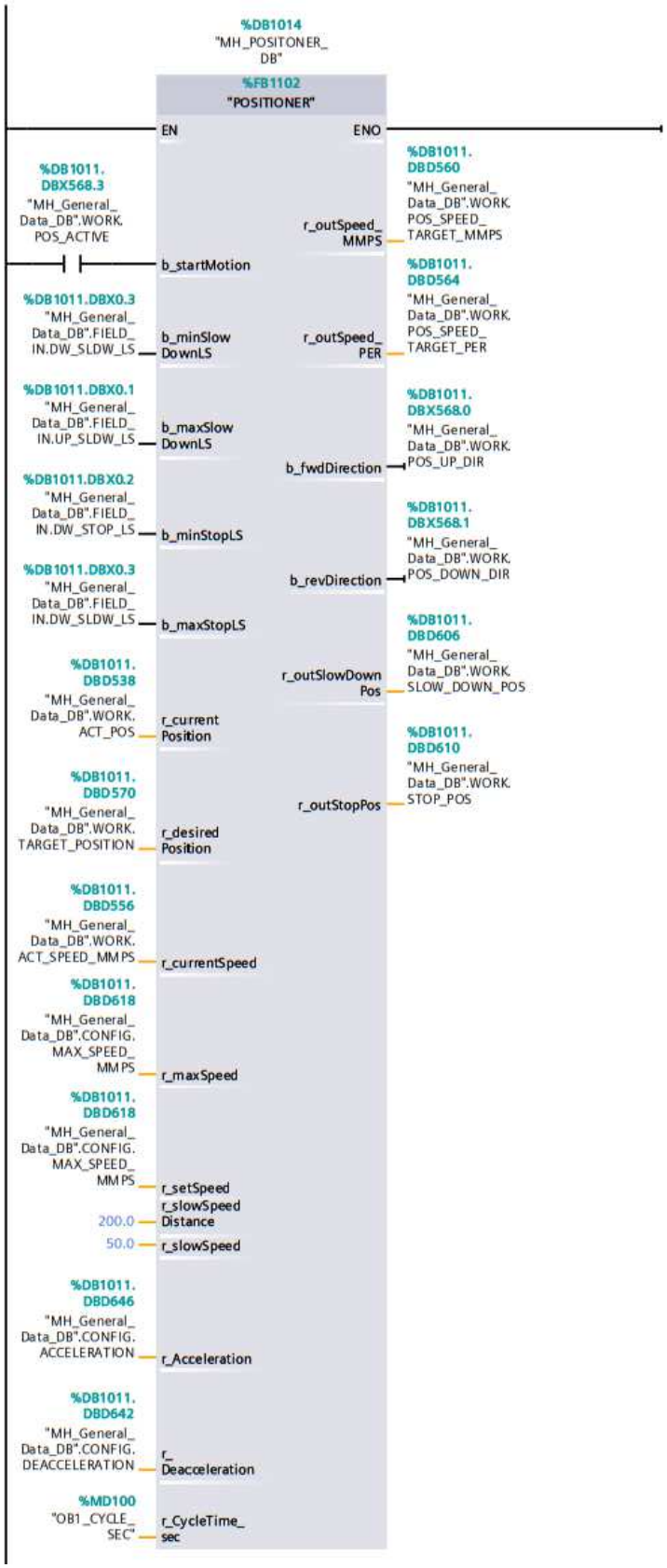
- Aktivacijska komanda za start rada bloka
- Krajnji prekidači za usporavanje

- Krajnji prekidači za prekidanje rada
- Trenutna pozicija u mm
- Željena pozicija u mm
- Maksimalna brzina kretanja u mm/s
- Zadana brzina kretanja u mm/s
- Udaljenost na kojoj će motor preći u sporiju brzinu prije zadane pozicije u mm
- Sporija brzina kretanja u mm/s
- Akceleracija motora u mm/s^2
- Deceleracija motora u mm/s^2
- Vrijeme PLC ciklusa u sekundama

Izlazne vrijednosti „POSITIONER“ bloka:

- Referentna brzina motora u mm/s
- Referentna brzina motora u %
- Smjer vrtnje motora
- Pozicija za početak usporavanja brzine motora
- Krajnja pozicija zaustavljanja

Funkcijski blok „POSITIONER“ (slika 3.41.) se nalazi u trećoj mreži 05_MH_POS funkcije nakon komande za aktiviranje bloka za pozicioniranje (slika 3.40.).



Slika 3.41. POSITIONER blok [Autor]

Logika bloka POSITIONER započinje određivanjem maksimalne brzine kretanja, tj. ograničavanjem izlazne brzine na zadanu vrijednost ako je ona manja od maksimalne, ili postavljanjem na maksimalnu vrijednost ako je veća od te maksimalne brzine. Ovaj korak se obavlja na početku koda, kako je prikazano na slici 3.42.

```
3  IF #r_setSpeed >= #r_maxSpeed THEN
4      #r_speed := #r_maxSpeed;
5  ELSE
6      #r_speed := #r_setSpeed;
7  END_IF;
```

Slika 3.42. Ograničavanje brzine [Autor]

Kako bi se odredila pozicija u sljedećem ciklusu PLC-a potrebno je izračunati promjenu pozicije kao umnožak vremena ciklusa PLC-a i trenutačne brzine, kao u liniji 10 na slici 3.43..

Sljedeća sekcija koda odnosi se na izračun pozicije kada motor treba krenuti u usporavanje kako bi postigao zadanu sporu brzinu točno na zadanoj udaljenosti od krajnje pozicije zaustavljanja. Osnovni uvjet za provjeru i usporavanje jest da su ta udaljenost i spora brzina različite od nule. Put koji kran prijeđe prilikom usporavanja izračunava se prema izrazu (3.1), no samo ukoliko je trenutna brzina različita od nule, u suprotnom taj put je jednak nuli.

$$slowDownArea = \frac{currentSpeed^2 - slowSpeed^2}{2 \cdot Deceleration} \tag{3.1}$$

Gdje je:

- slowDownArea – prijeđeni put tijekom usporavanja
- currentSpeed – trenutna brzina kretanja
- slowSpeed – zadana spora brzina kretanja
- Deceleration – deceleracija

Nakon izračuna potrebnog puta za usporavanje, potrebno je izvršiti provjeru je li potrebno započeti usporavanje. Prvo je potrebno izračunati na kojoj poziciji je potrebno započeti usporavanje prema izrazu (3.2).

$$slowDownPos = desiredPosition - slowDownArea - slowSpeedDistance \quad (3.2)$$

Gdje je:

- slowDownPos – točna pozicija na kojoj je potrebno započeti usporavanje
- desiredPosition – željena pozicija spore brzine
- slowDownArea – prijeđeni put tijekom usporavanja
- slowSpeedDistance – udaljenost na kojoj će motor preći u sporiju brzinu prije zadane pozicije

Nakon što je izračunata točna pozicija za usporavanje, potrebno je provjeriti je li kran dosegao tu poziciju. U tu svrhu, trenutnoj poziciji se dodaje delta pozicija koju kran prijeđe u jednom ciklusu PLC-a. Ovo omogućuje da se već u sljedećem ciklusu može preći u usporavanje. Također, provjerava se aktivnost krajnjeg prekidača za usporavanje, koji također pokreće proces usporavanja krana. Gornji opis s formulama precizno opisuje kod prikazan na slici 3.43., pri čemu se primarno odnosi na pozitivan smjer brzine kretanja. Za negativni smjer, pristup je isti, s razlikom što se promijeni predznak nekih članova u formulama.

```
 9  // Delta position - to check how much we travel in actual PLC cycle with current speed
10  #r_deltaPosition := #r_CycleTime_sec * #r_currentSpeed;
11
12  // Calculation for slow down position in which actual speed must decelerate to slow speed
13  // Calculation is done only if slow speed distance is bigger then 0.0 mm and if slow speed is different then 0 mm/s.
14 IF #r_slowSpeedDistance > 0.0 AND #r_slowSpeed > 0.0 THEN
15      // Slow down space neccessary to go from fast speed to slow speed
16      // It will be checked only if actual speed is different than 0
17      IF #r_currentSpeed <> 0.0 THEN
18          #r_slowDownArea := (SQR(#r_currentSpeed) - SQR(#r_slowSpeed)) / (2.0 * #r_Deacceleration);
19      ELSE
20          #r_slowDownArea := 0.0;
21      END_IF;
22
23      IF #r_currentSpeed > 0.0 THEN
24          #r_slowDownPos := #r_desiredPosition - #r_slowDownArea - #r_slowSpeedDistance;
25
26          // Condition is checked for next PLC cycle.
27          // If (actual position + delta position) is bigger then slowdown position then target speed must be slow speed
28          IF (#r_currentPosition + #r_deltaPosition >= #r_slowDownPos) OR (#b_maxSlowDownLS) THEN
29              #b_slowSpeedSet := TRUE;
30          END_IF;
31      END_IF;
32
33      // Calculation for slow down position in which actual speed must decelerate to slow speed
34      IF #r_currentSpeed < 0.0 THEN ... END_IF;
43  END_IF;
```

Slika 3.43. Usporavanje krana na zadanu brzinu [Autor]

Kod za izračunavanje trenutka kada kran treba započeti zaustavljanje jednak je onome za usporavanje, s promjenom što je sada zadana brzina postavljena na nulu, kao što se vidi u izrazu (3.3). Također, izračun za zaustavnu poziciju je prilagođen, kako je prikazano u formuli (3.4). Potpuni kod za pozitivan smjer vrtnje je prikazan na slici 3.44.

$$stopArea = \frac{currentSpeed^2}{2 \cdot Deceleration} \tag{3.3}$$

Gdje je:

- stopArea – prijeđeni put tijekom zaustavljanja
- currentSpeed – trenutna brzina kretanja
- Deceleration – deceleracija

$$stopPos = desiredPosition - stopArea \tag{3.4}$$

Gdje je:

- stopPos – točna pozicija na kojoj je potrebno započeti zaustavljanje
- desiredPosition – željena pozicija
- stopArea – prijeđeni put tijekom zaustavljanja

```
107  // Calculation for stop position in which actual speed must decelerate to 0 speed
108  #r_StopArea := (SQR(#r_currentSpeed)) / (2.0 * #r_Deacceleration);
109
110 IF #r_currentSpeed > 0.0 THEN
111      #r_stopPos := #r_desiredPosition - #r_StopArea;
112
113      // Condition is checked for next PLC cycle.
114      // If (actual position + delta position) is bigger then stop position then target speed must be 0 speed
115      IF (#r_currentPosition + #r_deltaPosition >= #r_stopPos) OR (#b_maxStopLS) THEN
116          #b_stopRequest := TRUE;
117      END_IF;
118  END_IF;
```

Slika 3.44. Zaustavljanje krana na određenu poziciju [Autor]

Do sada je provjeravana situacija za zaustavljanje i usporavanje brzine krana. Međutim, posljednji dio koda upravlja izlaznim parametrima funkcije. Ako se kran ne nalazi na zadanoj poziciji, pokreće se ako je primljena vanjska komanda, a smjer vrtnje motora postavlja se prema poziciji.

Također, postavlja se brzina na zadanu vrijednost ukoliko niti jedan od prethodno objašnjenih uvjeta za usporavanje i zaustavljanje nije ispunjen. Ukoliko je vanjska komanda za start prekinuta sve vrijednosti se resetiraju te brzina krana pada na nulu. Cijela posljednja sekcija koda se vidi na slici 3.45., dok se potpun kod POSITIONER funkcije nalazi u dodatku A.

```
IF #b_startMotion THEN
    // Direction
    IF #r_currentPosition < #r_desiredPosition THEN
        #b_fwdDirection := TRUE;
        #b_revDirection := FALSE;

    ELSIF #r_currentPosition > #r_desiredPosition THEN
        #b_fwdDirection := FALSE;
        #b_revDirection := TRUE;
    END_IF;

    IF #b_slowSpeedSet AND NOT #b_stopRequest THEN
        #r_outSpeed_MMPS := #r_slowSpeed;
        #r_outSpeed_PER := #r_slowSpeed / #r_maxSpeed * 100.0;
    ELSIF #b_stopRequest THEN
        #r_outSpeed_MMPS := 0;
        #r_outSpeed_PER := 0;
    ELSIF NOT #b_slowSpeedSet AND NOT #b_stopRequest THEN
        #r_outSpeed_MMPS := #r_speed;
        #r_outSpeed_PER := #r_speed / #r_maxSpeed * 100.0;
    END_IF;

    IF NOT #b_slowSpeedSet THEN
        #r_outSlowDownPos := #r_slowDownPos;
    END_IF;

    IF NOT #b_stopRequest THEN
        #r_outStopPos := #r_stopPos;
    END_IF;

ELSE
    #b_slowSpeedSet := FALSE;
    #b_stopRequest := FALSE;
    #b_fwdDirection := FALSE;
    #b_revDirection := FALSE;
    #b_firstCycleDone := FALSE;
    #b_slowDownStarted := FALSE;
    #b_slowSpeedSet := FALSE;
    #r_outSpeed_MMPS := 0;
END_IF;
```

Slika 3.45. Upravljanje izlaznim vrijednostima funkcije [Autor]

Kao test i simulacija funkcije POSITIONER korištena su kranska kolica, zadano je da se pomaknu na poziciju 10000mm sa pozicije 5000mm. Na slici 3.46. prikazana je snimka pozicije u milimetrima i brzine motora u okretajima po minuti. Kao što je vidljivo na slici kolica ubrzaju na maksimalnu brzinu za 6 sekundi, kao što je i navedeno u tablici 2.2, neko vrijeme se gibaju tom brzinom te potom kreću u usporavanje kako bi dostigla brzinu od 50mm/s tj. 90o/min na točno 200mm od krajnje točke zaustavljanja, kako je i postavljeno na ulazu u funkciju.

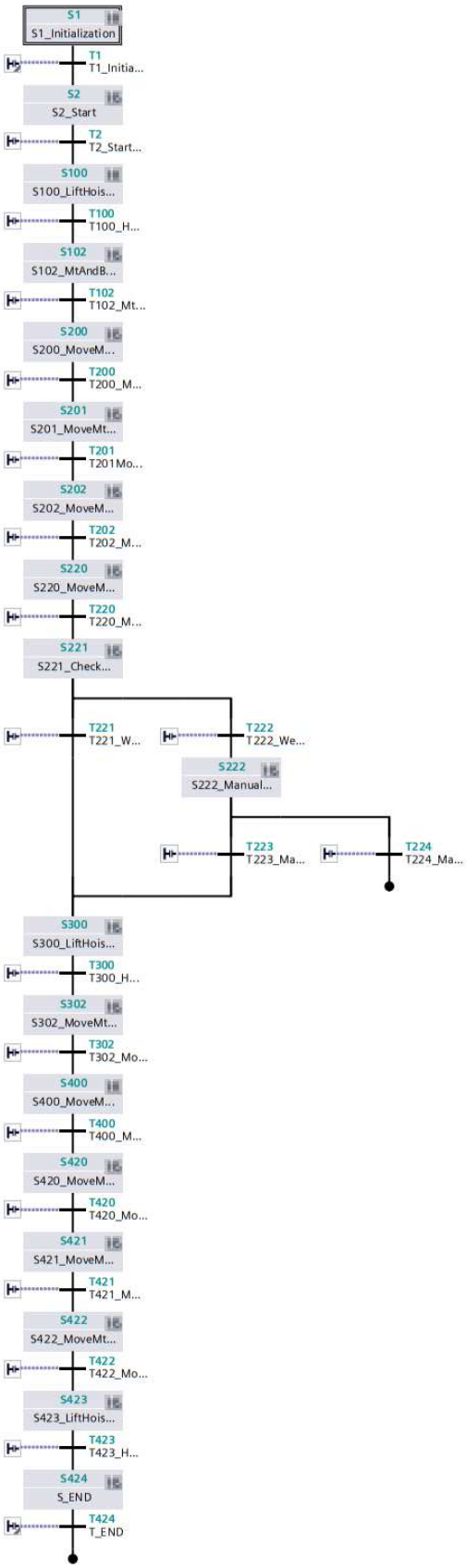
Slika 3.46. Snimka brzine i pozicije kranskih kolica upravljanim POSITIONER funkcijom [Autor]

3.5.1. Sekvenca i automatsko upravljanje kranom

Automatsko upravljanje kranom je ključno prilikom podizanja tereta, jer je potrebno usklađeno upravljati s kranskim kolicima i glavnim vitlom kako bi se ostvarila precizna i sinkronizirana putanja podizanja tereta, kao što je prikazano na slici 2.3. Također, za premještanje tereta s jedne na drugu poziciju potrebna je pažljivo osmišljena sekvenca koja osigurava visoku preciznost operacije. Ova sekvenca je programirana koristeći programski jezik GRAPH. Cijela sekvenca se vidi na slici 3.47..

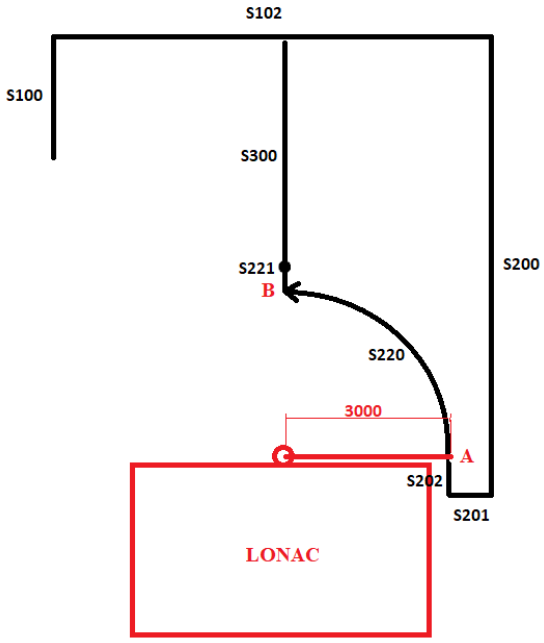
Koraci sekvence su sljedeći:

- S1 – inicijalizacija sekvence
- S2 – start sekvence
- S100 – podizanje glavnog vitla na sigurnu visinu za pomicanje
- S102 – pomicanje mosta i kranskih kolica na poziciju
- S200 – spuštanje glavnog vitla ispod položaja zaključavanja nosača
- S201 – pomicanje kranskih kolica na poziciju zaključavanja
- S202 – podizanje glavnog vitla u poziciju zaključavanja nosača lonca
- S220 – kružno dizanje nosača lonca sinkroniziranim upravljanjem vitla i kranskih kolica
- S221 – provjera mase tereta
- S222 – ručna konformacija mase
- S300 – podizanje glavnog vitla na sigurnu visinu za pomicanje
- S302 – pomicanje mosta i kranskih kolica na novu poziciju
- S400 – spuštanje glavnog vitla na novu poziciju
- S420 – kružno spuštanje nosača lonca sinkroniziranim upravljanjem vitla i kranskih kolica
- S421 – spuštanje glavnog vitla ispod položaja zaključavanja nosača
- S422 – pomicanje kranskih kolica van pozicije zaključavanja
- S423 – podizanje glavnog vitla na sigurnu visinu za pomicanje

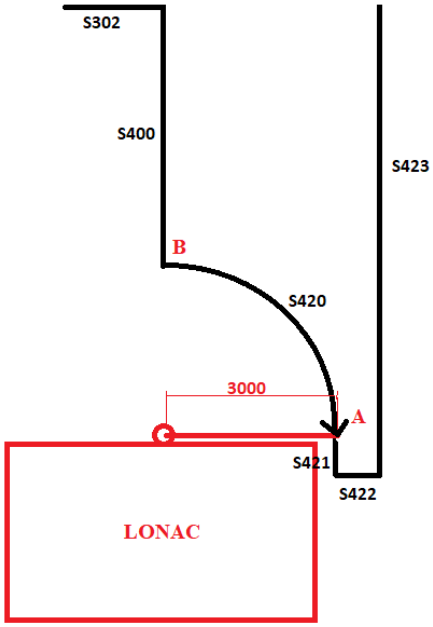


Slika 3.47. Automatska sekvenca premještanja lonca [Autor]

Na slikama 3.48. i 3.49. se vide koraci sekvence za dizanje i spuštanje lonca.



Slika 3.48. Koraci sekvence dizanja lonca[Autor]



Slika 3.49. Koraci sekvence spuštanja lonca [Autor]

Za pozicioniranje krana na određene pozicije koristi se blok POSITIONER, koji je opisan nešto ranije u ovom radu. Međutim, za postupak kružnog podizanja tereta, koji zahtijeva sinkronizirano upravljanje glavnim vitlom i kranskim kolicima, koristi se blok ARCH_POSITIONER. Ova blok funkcija je implementirana pomoću SCL programskog jezika kako bi se olakšala implementacija matematičkih izraza. U narednim dijelovima će biti objašnjeni svi ulazni i izlazni parametri bloka, te će biti detaljno opisano kako blok funkcija radi. Vizualni prikaz ovog bloka je dostupan na slici 3.50.

Ulazni parametri ARCH_POSITIONER bloka:

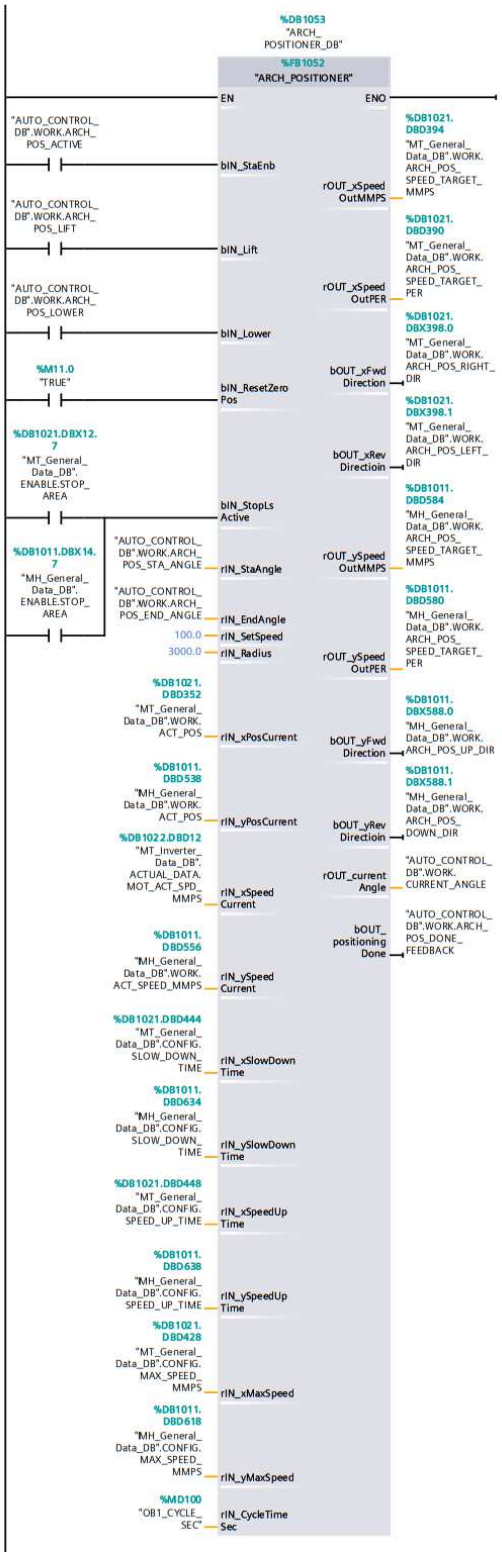
- Komanda za omogućavanje starta
- Komanda za podizanje nosača lonca
- Komanda za spuštanje nosača lonca
- Resetiranje početne pozicije
- Aktivni krajnji prekidači
- Početni kut nosača lonca

- Krajnji kut nosača lonca
- Zadana maksimalna brzina
- Dužina nosača lonca
- Trenutna pozicija kranskih kolica
- Trenutna pozicija vitla
- Aktualna brzina kranskih kolica u mm/s
- Aktualna brzina vitla u mm/s
- Vrijeme deceleracije kranskih kolica s maksimalne brzine na nulu u sekundama
- Vrijeme deceleracije vitla s maksimalne brzine na nulu u sekundama
- Vrijeme akceleracije kranskih kolica iz mirovanja na maksimalnu brzinu u sekundama
- Vrijeme akceleracije vitla iz mirovanja na maksimalnu brzinu u sekundama
- Maksimalna brzina kranskih kolica u mm/s
- Maksimalna brzina vitla u mm/s
- Vrijeme trajanja posljednjeg ciklusa PLC-a u sekundama

Izlazni parametri ARCH_POSITIONER bloka:

- Referentna brzina kranskih kolica u mm/s
- Referentna brzina kranskih kolica u postotku
- Komanda za pozitivan smjer gibanja kranskih kolica
- Komanda za negativan smjer gibanja kranskih kolica
- Referentna brzina vitla u mm/s
- Referentna brzina vitla u postotku
- Komanda za pozitivan smjer gibanja vitla
- Komanda za negativan smjer gibanja vitla

- Trenutni kut nosača lonca u stupnjevima
- Povratna informacija o kraju pozicioniranja nosača lonca



Slika 3.50. Prikaz ARCH_POSITIONER bloka [Autor]

Prvi korak u ARCH_POSITIONER funkciji je izračunavanje deceleracije za vitlo i kranska kolica, kao i linearna brzina kretanja duž kružnice. Izračun deceleracije se postiže upotrebom formule (3.5), dok se linearna brzina određuje prema formuli (3.6). Ovaj izračun se inicijalno obavlja na početku koda, kako je prikazano na slici 3.51. te će se izračunavati svaki ciklus, bez obzira na ostatak koda.

$$Deceleration = \frac{MaxSpeed}{slowDownTime} \tag{3.5}$$

Gdje je:

- Deceleration – deceleracija
- MaxSpeed – maksimalna brzina kretanja
- slowDownTime – vrijeme usporavanja s maksimalne brzine

$$linearSpeed = \sqrt{|xSpeedCurrent|^2 + |ySpeedCurrent|^2} \tag{3.6}$$

Gdje je:

- linearSpeed – linearna brzina kretanja
- xSpeedCurrent – trenutna brzina kretanja kranskih kolica
- ySpeedCurrent – trenutna brzina kretanja vitla

```
1  //CALCULATING DEACELERATION ND ACCELERATION OF EACH AXIS
2  #r_xDeacceleration := #rIN_xMaxSpeed / #rIN_xSlowDownTime;
3  #r_yDeacceleration := #rIN_yMaxSpeed / #rIN_ySlowDownTime;
4
5  //CALCULATING LINEAR CIRCULAR SPEED
6  #r_linearSpeed := SQRT(SQR(ABS(#rIN_xSpeedCurrent)) + SQR(ABS(#rIN_ySpeedCurrent)));
7
```

Slika 3.51. Izračun deceleracije i linearne brzine [Autor]

Nakon izračuna deceleracije i linearne brzine, provjerava se jesu li na ulazu funkcije dane komande za omogućavanje rada te za podizanje ili spuštanje nosača lonca. Ako su oba uvjeta ispunjena, izvršava se dio koda koji se aktivira tijekom jednog PLC ciklusa. Taj dio koda služi za postavljanje brzina i osnovnih pozicija. Ovaj segment koda je prikazan na slici 3.52.

Kao što je već spomenuto na početku ove sekcije koda, maksimalna brzina postavlja se na nižu od dviju maksimalnih brzina vitla i kranskih kolica. Ako je unesena željena brzina manja od obiju, ta će se brzina koristiti kao maksimalna izlazna brzina kretanja za oba sustava.

Nulta pozicija je ona pozicija oko koje se vrti nosač lonca tj. osovina nosača. Pozicija te točke je izračunata pomoću dvije formule za svaku od osi posebno, no samo ako je na ulazu bloka aktivna komanda za resetiranje te pozicije. Izrazi za izračun te pozicije dani su u formulama (3.7) i (3.8):

$$xZeroPos = xPosCurrent - Radius \cdot \cos\left(StaAngle \cdot \frac{\pi}{180}\right) \tag{3.7}$$

Gdje je:

- xZeroPos – X koordinata osovine nosača
- xPosCurrent – trenutna pozicija kranskih kolica
- Radius – dužina nosača lonca
- StaAngle – početni kut od kojeg se podiže nosač lonca

$$yZeroPos = yPosCurrent - Radius \cdot \cos\left((90 - StaAngle) \cdot \frac{\pi}{180}\right) \tag{3.8}$$

Gdje je:

- yZeroPos – Y koordinata osovine nosača
- yPosCurrent – trenutna pozicija vitla
- Radius – dužina nosača lonca
- StaAngle – početni kut od kojeg se podiže nosač lonca

Također, važno je izračunati točku na kojoj se nalazi 0 stupnjeva, kao i točku na kružnom luku na kojoj moramo zaustaviti kretanje i koja pripada krajnjem kutu zadanom na ulazu u blok. Nadalje, trebamo izračunati i razliku između početnog i krajnjeg kuta, budući da će nam to biti potrebno za daljnje kalkulacije. U konačnici, u ovoj sekciji koda unutarnje komande za zaustavljanje svake od osi također su resetirane. Položaj nultog stupnja opisan je u izrazima (3.9) i (3.10), položaj krajnjeg kuta u izrazima (3.11) i (3.12), dok je radni kut opisan u izrazu (3.13).

$$xZeroAnglePoint = xZeroPos + Radius \tag{3.9}$$

Gdje je:

- xZeroAnglePoint – X koordinata 0 stupnjeva na kružnom liku nosača
- xZeroPos – X koordinata osovine nosača
- Radius – dužina nosača lonca

$$yZeroAnglePoint = yZeroPos \tag{3.10}$$

Gdje je:

- xZeroAnglePoint – X koordinata 0 stupnjeva na kružnom liku nosača
- xZeroPos – X koordinata osovine nosača

$$xEndPoint = xZeroPos + Radius \cdot \cos\left(EndAngle \cdot \frac{\pi}{180}\right) \tag{3.11}$$

Gdje je:

- xEndPoint – X koordinata krajnje pozicije nosača konca
- xZeroPos – X koordinata osovine nosača
- Radius – dužina nosača lonca
- EndAngle – krajnji kut do kojeg se podiže nosač lonca

$$yEndPoint = yZeroPos + Radius \cdot \sin\left(EndAngle \cdot \frac{\pi}{180}\right) \tag{3.12}$$

Gdje je:

- yEndPoint – Y koordinata krajnje pozicije nosača konca
- yZeroPos – Y koordinata osovine nosača
- Radius – dužina nosača lonca

- EndAngle – krajnji kut do kojeg se podiže nosač lonca

```
IF #bIN_StaEnb AND (#bIN_Lift OR #bIN_Lower) THEN
    //FIRST STEP, SETTING MAX SPEED
    IF NOT #b_FirstStepDone THEN
        IF #rIN_SetSpeed >= #rIN_yMaxSpeed OR #rIN_SetSpeed >= #rIN_xMaxSpeed THEN
            IF #rIN_xMaxSpeed > #rIN_yMaxSpeed THEN
                #r_MaxSpeed := #rIN_xMaxSpeed;
            ELSE
                #r_MaxSpeed := #rIN_yMaxSpeed;
            END_IF;
        ELSE
            #r_MaxSpeed := #rIN_SetSpeed;
        END_IF;

        //CALCULATING ROTATING POINT OF BUCKET
        IF #bIN_ResetZeroPos THEN
            #r_xZeroPos := #rIN_xPosCurrent - #rIN_Radius * COS(#rIN_StaAngle * 3.14159 / 180);
            #r_yZeroPos := #rIN_yPosCurrent - #rIN_Radius * COS((90 - #rIN_StaAngle) * 3.14159 / 180);
        END_IF;
        //ZERO ANGLE POINT IS 0° ON A CIRCLE, END POINT IS WHERE MOVEMENT IS FINISHED, WORK ANGLE IS DIFERENCE BETWEEN START AND END ANGLE
        #r_xZeroAnglePoint := #r_xZeroPos + #rIN_Radius;
        #r_yZeroAnglePoint := #r_yZeroPos;
        #r_WorkAngle := ABS(#rIN_StaAngle - #rIN_EndAngle);
        #r_xEndPoint := #r_xZeroPos + #rIN_Radius * COS(#rIN_EndAngle * 3.14159 / 180);
        #r_yEndPoint := #r_yZeroPos + #rIN_Radius * SIN(#rIN_EndAngle * 3.14159 / 180);

        //RESETING SPEEDS
        #b_xSlowDownStarted := FALSE;
        #b_ySlowDownStarted := FALSE;

        #b_FirstStepDone := TRUE;
```

Slika 3.52. Početna konfiguracija i izračuni [Autor]

Nakon izvršavanja prvog ciklusa funkcije i postavljanja početne konfiguracije kreće druga sekcija koda, koja se bazira na izračunu trenutnog kuta nosača lonca u prostoru te potrebne brzine za oba sustava, glavno vitlo i kranska kolica, kako bi se postiglo gibanje po kružnoj putanji od početne do krajnje točke. Ova sekcija koda je prikazana na slici 3.53.

Kako bi se izračunao trenutni kut u prostoru u kojem se nalazi nosač lonca potrebna je formula (3.17), koja je podijeljena na nekoliko izraza zbog svoje duljine(izrazi 3.13 do 3.16):

$$A = (xZeroAnglePoint - xZeroPos) \cdot (xPosCurrent - xZeroPos) \tag{3.13}$$

Gdje je:

- xZeroAnglePoint – X koordinata 0 stupnjeva na kružnom luku nosača
- xZeroPos – X koordinata osovine nosača
- xPosCurrent – trenutna pozicija kranskih kolica

$$B = (yZeroAnglePoint - yZeroPos) \cdot (yPosCurrent - yZeroPos) \quad (3.14)$$

Gdje je:

- yZeroAnglePoint – Y koordinata 0 stupnjeva na kružnom luku nosača
- yZeroPos – Y koordinata osovine nosača
- yPosCurrent – trenutna pozicija vitla

$$C = \sqrt{(xZeroAnglePoint - xZeroPos)^2 + (yZeroAnglePoint - yZeroPos)^2} \quad (3.15)$$

Gdje je:

- xZeroAnglePoint – X koordinata 0 stupnjeva na kružnom luku nosača
- xZeroPos – X koordinata osovine nosača
- yZeroAnglePoint – Y koordinata 0 stupnjeva na kružnom luku nosača
- yZeroPos – Y koordinata osovine nosača

$$D = \sqrt{(xPosCurrent - xZeroPos)^2 + (yPosCurrent - yZeroPos)^2} \quad (3.16)$$

Gdje je:

- xPosCurrent – trenutna pozicija kranskih kolica
- xZeroPos – X koordinata osovine nosača
- yPosCurrent – trenutna pozicija vitla
- yZeroPos – Y koordinata osovine nosača

$$CurrentAngle = acos\left(\frac{A + B}{C \cdot D}\right) \cdot \frac{180}{\pi} \quad (3.17)$$

Gdje je:

- CurrentAngle – trenutni kut nosača lonca u prostoru

Također izračunava se i pređeni kut nosača lonca koji će trebati kasnije u provjeri za početak zaustavljanja kretanja pomoću izraza (3.18):

$$ChangeAngle = |StaAngle - CurrentAngle| \tag{3.18}$$

Gdje je:

- ChangeAngle – prijeđeni kut nosača lonca
- StaAngle – početni kut od kojeg se podiže nosač lonca
- CurrentAngle – trenutni kut nosača lonca u prostoru

Nakon izračuna trenutnog i prijeđenog kuta vrši se izračun potrebnog područja zaustavljanja svakog od sustava prema njihovim trenutnim brzinama, to se računa svaki ciklus sve dok se ne aktivira komanda tj. uvjet koji pokreće zaustavljanje te osi gibanja. Spomenuti izrazi su (3.19) i (3.20)

$$xSlowDownPath = \frac{xSpeedCurrent^2}{2 \cdot xDeceleration} \tag{3.19}$$

Gdje je:

- xSlowDownPath – prijeđeni put kranskih kolica tijekom zaustavljanja
- xSpeedCurrent – trenutna brzina kretanja kranskih kolica
- xDeceleration – deceleracija kranskih kolica

$$ySlowDownPath = \frac{ySpeedCurrent^2}{2 \cdot yDeceleration} \tag{3.20}$$

Gdje je:

- ySlowDownPath – prijeđeni put vitla tijekom zaustavljanja
- ySpeedCurrent – trenutna brzina kretanja vitla
- yDeceleration – deceleracija vitla

```
39          //AFTER FIRST STEP DONE
40      ELSIF #b_FirstStepDone THEN
41
42          //CALCULATING CURRENT ANGLE TO DETERMINE SPEED FOR EACH AXIS
43          #r_CurrentAngle := ACOS(((#r_xZeroAnglePoint - #r_xZeroPos) * (#rIN_xPosCurrent - #r_xZeroPos) + (#r_yZeroAnglePoint - #r_yZeroPos) ...
44          #rOUT_currentAngle := #r_CurrentAngle;
45          #r_ChangeAngle := ABS(#rIN_StaAngle - #r_CurrentAngle);
46
47          //CALCULATING SLOW DOWN ANGLE, CHANGES BASED ON CURRENT SPEED
48          IF NOT #b_xSlowDownStarted THEN
49              #r_xSlowDownPath := #rIN_xSpeedCurrent / (2.0 * #r_xDeceleration);
50          END_IF;
51
52          IF NOT #b_ySlowDownStarted THEN
53              #r_ySlowDownPath := #rIN_ySpeedCurrent / (2.0 * #r_yDeceleration);
54          END_IF;
55
```

Slika 3.53. Izračun trenutnog kuta i puta usporavanja [Autor]

Slijedeći odlomak koda odnosi se na postavljanje brzina tijekom podizanja nosača lonca iz pozicije A u B, što je prikazano na slici 3.48. Cjelokupni kod za podizanje može se vidjeti na slici 3.54.

Prva provjera ima za cilj zaustavljanje ili sprječavanje daljnjeg gibanja ukoliko je pređeni kut nosača lonca veći ili jednak radnom kutu, ili ako je aktiviran bilo koji od krajnjih prekidača za zaustavljanje. Ako je prethodno navedeni izraz istinit, brzine vitla i kranskih kolica postavljaju se na nulu.

Sljedeća provjera se odnosi na provjeru početka usporavanja svake od osi zasebno kako bi se kretanje precizno zaustavilo na željenoj točki. Ovaj uvjet je izveden prema izrazu (3.21), koji sadrži uvjet za vitlo. Sličan pristup primijenjen je i na kranska kolica, nekoliko redova niže u kodu. Koncept je da se provjerava apsolutna razlika između trenutne pozicije i krajnje pozicije zaustavljanja i uspoređuje sa stvarnom udaljenošću koja je potrebna da se kretanje zaustavi. Ukoliko se uvjet zadovolji brzina te osi se postavlja u nulu i daje se komanda za početak zaustavljanja koja prekida daljnje računanje zaustavnog puta.

$$|yPosCurrent - yEndPoint| \leq ySlowDownPath \tag{3.21}$$

Gdje je:

- yPosCurrent – trenutna pozicija vitla
- yEndPoint – Y koordinata krajnje pozicije nosača konca
- ySlowDownPath – prijeđeni put vitla tijekom zaustavljanja

Postavljanje izlaznih brzina je takvo da jedna od osi služi kao referentna brzina tj. ona se postavlja na maksimalnu zadanu brzinu kretanja, dok je druga os „prati“ tj. prilagođava se ovisno o kojem kutu na kružnom luku se radi.

Kod dizanja nosača lonca ukoliko je trenutni kut nosača manji od 45 stupnjeva(od 0 do 45 stupnjeva), referentna brzina će biti brzina glavnog vitla te se ona postavlja na maksimalnu brzinu a brzina kolica se računa. Drugih 45 stupnjeva pokreta(od 45 do 90 stupnjeva) referentna brzina će biti brzina kranskih kolica dok će se brzina vitla računati. Nakon postavljanje referentne brzine računa se koeficijent pogreške pozicije koji računa grešku između pozicije kranskih kolica u kojoj se trenutno nalaze i u kojoj bi trebala biti s obzirom na poziciju glavnog vitla. Ta pogreška se množi sa koeficijentom 0.5 te se kasnije to pribraja izračunatoj brzini kranskih kolica kao neka vrsta regulatora. Izraz za izračun tog koeficijenta dan je u (3.22):

$$xErrCoef = \left(\left(\sqrt{Radius^2 - (yPosCurrent - yZeroPos)^2} + xZeroPos\right) - xPosCurrent\right) \cdot 0.5 \tag{3.22}$$

Gdje je:

- xErrCoef – koeficijent pogreške pozicije kranskih kolica
- Radius – dužina nosača lonca
- yPosCurrent – trenutna pozicija vitla
- yZeroPos – Y koordinata osovine nosača
- xZeroPos – X koordinata osovine nosača
- xPosCurrent – trenutna pozicija kranskih kolica

Izračun potrebne brzine kranskih kolica kako bi se postigla kružna putanja se računa pomoću linearne brzine gibanja po kružnici, tako što se izračuna projekcija brzine za os vitla na temelju trenutnog kuta nosača lonca u prostoru te pridoda koeficijent pogreške pozicije kranskih kolica. Izraz je dan u (3.23):

$$xSpeed = linearSpeed \cdot \cos\left((CurrentAngle + 90) \cdot \frac{\pi}{180}\right) + xErrCoef \tag{3.23}$$

Gdje je:

- xSpeed – izračunata potrebna brzina kranskih kolica
- linearSpeed – linearna brzina gibanja po kružnici
- CurrentAngle – trenutni kut nosača lonca

- xErrCoef – koeficijent pogreške pozicije kranskih kolica

Zadnja dio koda dizanja nosača lonca kada je brzina vitla referentna brzina(0 od 45 stupnjeva) ograničava izračunatu brzinu kranskih kolica na maksimalnu zadanu ukoliko je prekoračuje. Na slici 3.54. također se vidi i kod za kut veći od 45 stupnjeva gdje je referentna brzina kranskih kolica. Logika je identična i za spuštanje nosača, uz uvjet da su brzine onda suprotnog predznaka.

```
        //LIFTING
        //SET SPEED TO ZERO IF CURRENT ANGLE IS BIGGER THAN END ANGLE
        IF #bIN_Lift AND (NOT #bIN_Lower) THEN
            IF #r_ChangeAngle >= #r_WorkAngle OR #bIN_StopLsActive THEN
                #r_xSpeed := 0.0;
                #r_ySpeed := 0.0;
            END_IF;

            IF ABS(#rIN_yPosCurrent - #r_yEndPoint) <= #r_ySlowDownPath THEN
                #r_ySpeed := 0.0;
                #b_ySlowDownStarted := TRUE;
            END_IF;

            IF ABS(#rIN_xPosCurrent - #r_xEndPoint) <= #r_xSlowDownPath THEN
                #r_xSpeed := 0.0;
                #b_xSlowDownStarted := TRUE;
            END_IF;

            IF #r_CurrentAngle <= 45.0 AND NOT (#b_xSlowDownStarted OR #b_ySlowDownStarted) THEN
                #r_ySpeed := #r_MaxSpeed;
                #r_xErrCoef := ((SQRT(SQR(#rIN_Radius) - SQR(#rIN_yPosCurrent - #r_yZeroPos)) + #r_xZeroPos) - #rIN_xPosCurrent) * 0.5;
                #r_xSpeed := #r_linearSpeed * COS((#r_CurrentAngle + 90.0)*3.14159/180) + #r_xErrCoef;

                IF #r_xSpeed >= #r_MaxSpeed THEN
                    #r_xSpeed := #r_MaxSpeed;
                ELSIF #r_xSpeed <= - #r_MaxSpeed THEN
                    #r_xSpeed := - #r_MaxSpeed;
                END_IF;

            ELSIF #r_CurrentAngle > 45.0 AND NOT (#b_xSlowDownStarted OR #b_ySlowDownStarted) THEN
                #r_xSpeed := -#r_MaxSpeed;
                #r_yErrCoef := ((SQRT(SQR(#rIN_Radius) - SQR(#rIN_xPosCurrent - #r_xZeroPos)) + #r_yZeroPos) - #rIN_yPosCurrent) * 0.5;
                #r_ySpeed := #r_linearSpeed * SIN((#r_CurrentAngle + 90.0) * 3.14159 / 180) + #r_yErrCoef;

                IF #r_ySpeed >= #r_MaxSpeed THEN
                    #r_ySpeed := #r_MaxSpeed;
                ELSIF #r_ySpeed <= - #r_MaxSpeed THEN
                    #r_ySpeed := - #r_MaxSpeed;
                END_IF;
            END_IF;
```

Slika 3.54. Izračun potrebnih brzina za kružno gibanje [Autor]

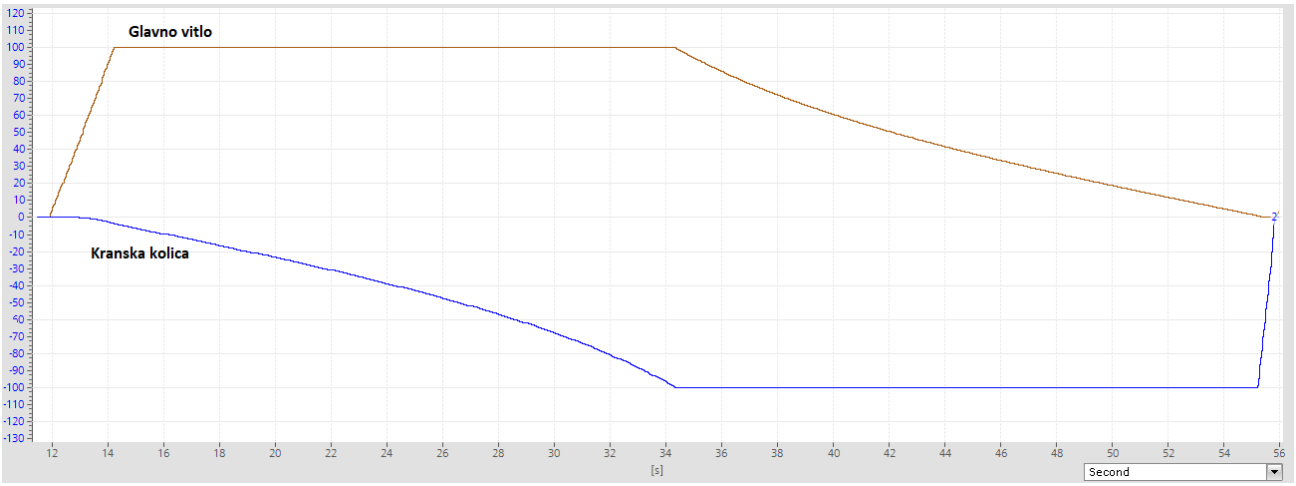
Posljednji korak jest postaviti izračunate parametre na izlaz bloka kao na slici 3.55.

```
    IF #r_xSpeed < 0.0 THEN
        #rOUT_xSpeedOutMMPS := ABS(#r_xSpeed);
        #rOUT_xSpeedOutPER := (#rOUT_xSpeedOutMMPS / #rIN_xMaxSpeed) * 100.0;
        #bOUT_xFwdDirection := FALSE;
        #bOUT_xRevDirectioin := TRUE;
    ELSIF #r_xSpeed > 0.0 THEN
        #rOUT_xSpeedOutMMPS := ABS(#r_xSpeed);
        #rOUT_xSpeedOutPER := (#rOUT_xSpeedOutMMPS / #rIN_xMaxSpeed) * 100.0;
        #bOUT_xFwdDirection := TRUE;
        #bOUT_xRevDirectioin := FALSE;
    ELSE
        #rOUT_xSpeedOutMMPS := 0.0;
        #rOUT_xSpeedOutPER := 0.0;
        #bOUT_xFwdDirection := FALSE;
        #bOUT_xRevDirectioin := FALSE;
    END_IF;
```

Slika 3.55. Postavljanje izlaza iz bloka[Autor]

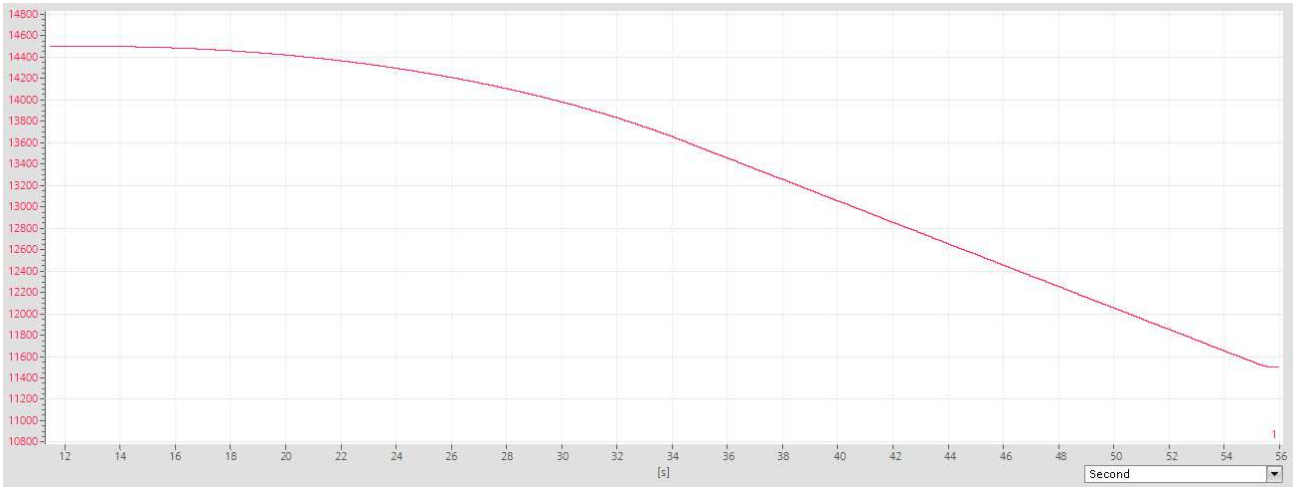
3.5.2. Simulacija sekvence i podizanja nosača lonca

Prvo je odrađena simulacija podizanja nosača lonca, tj. funkcije ARCH_POSITIONER. Na slici 3.56. se vide brzine glavnog vitla i kranskih kolica u milimetrima po sekundi. Kao što je vidljivo na slici glavno vitlo ubrza na maksimalnu zadanu brzinu te prvih 45 stupnjeva služi kao referentna pozicija i brzina. Pozicija i brzina vitla se potom koriste za izračun brzine kranskih kolica da se postigne kružna putanja. Drugih 45 stupnjeva je situacija obrnuta te kranska kolica služe kao referentna.

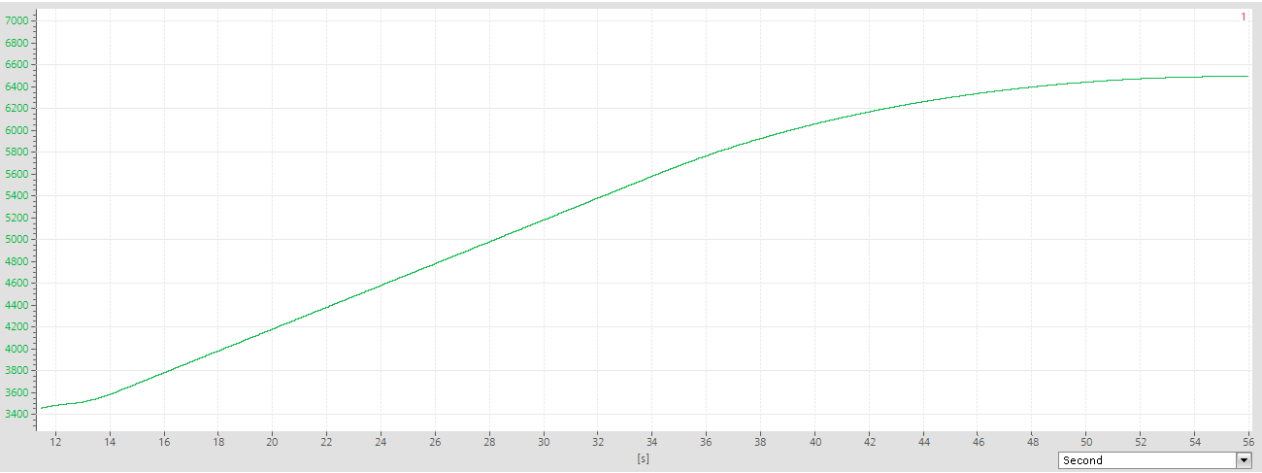


Slika 3.56. Brzine glavnog vitla i kranskih kolica prilikom dizanja nosača lonca [Autor]

Pozicije kroz pokret dizanja su prikazane na slikama 3.57. za kranska kolica te 3.58. za glavno vitlo.

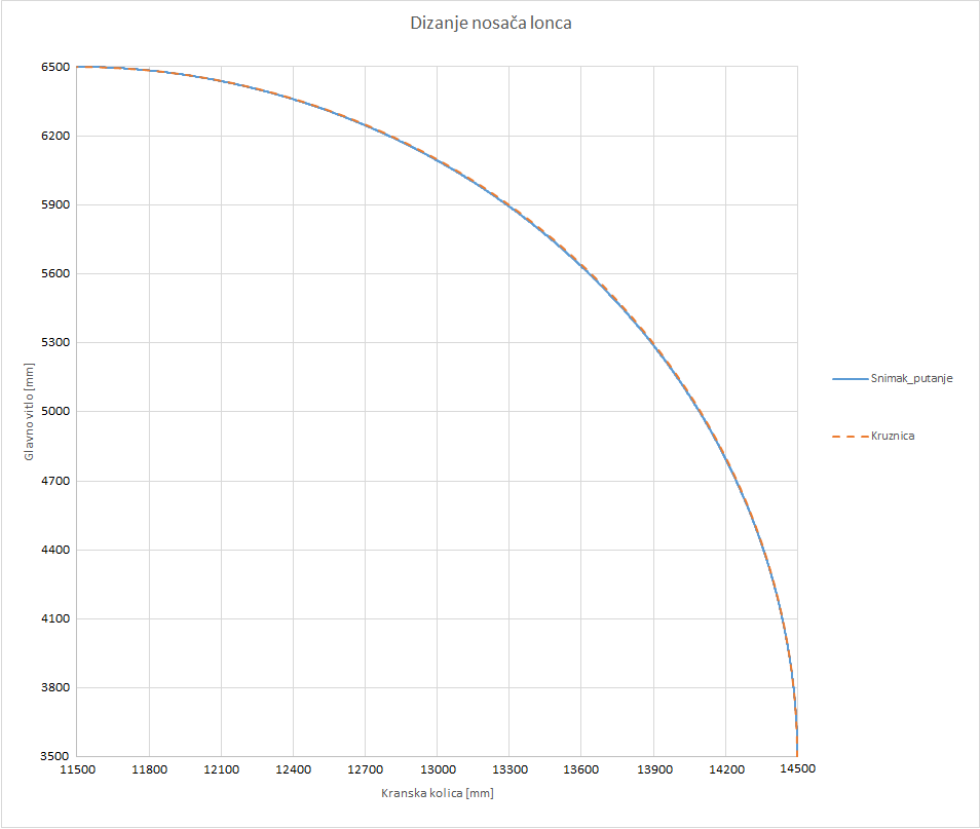


Slika 3.57. Pozicija kranskih kolica prilikom dizanja nosača lonca [Autor]



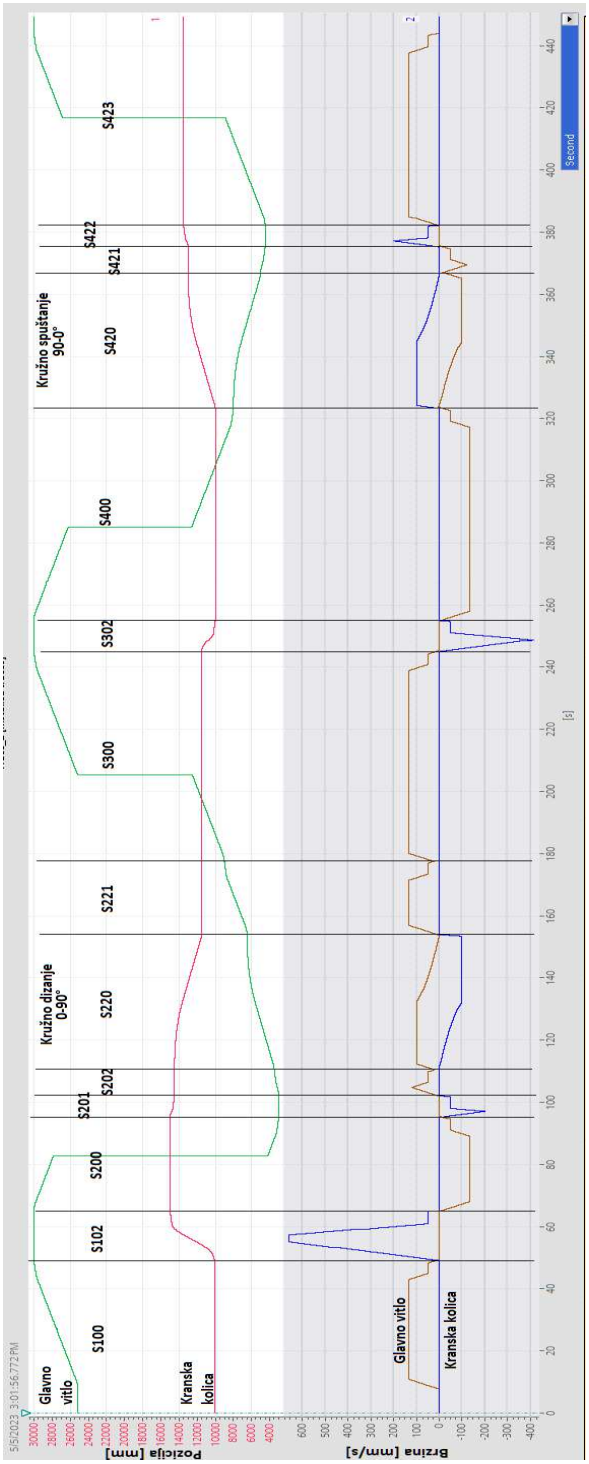
Slika 3.58. Pozicija glavnog vitla prilikom dizanja nosača lonca [Autor]

Kao provjera točnosti kretanja po kružnom luku kružnice eksportirane su vrijednosti pozicija te pomoću excela je nacrtana zajednička pozicija u prostoru. Također uspoređena je sa pravilnom kružnicom po kojoj bi se kran trebao gibati. Na slici 3.59. plavom linijom je označena snimljena putanja dobivena pomoću simulacije, dok je isprekidanom crvenom linijom točna putanja po kružnici. Može se vidjeti da su putanje u potpunosti jednake.



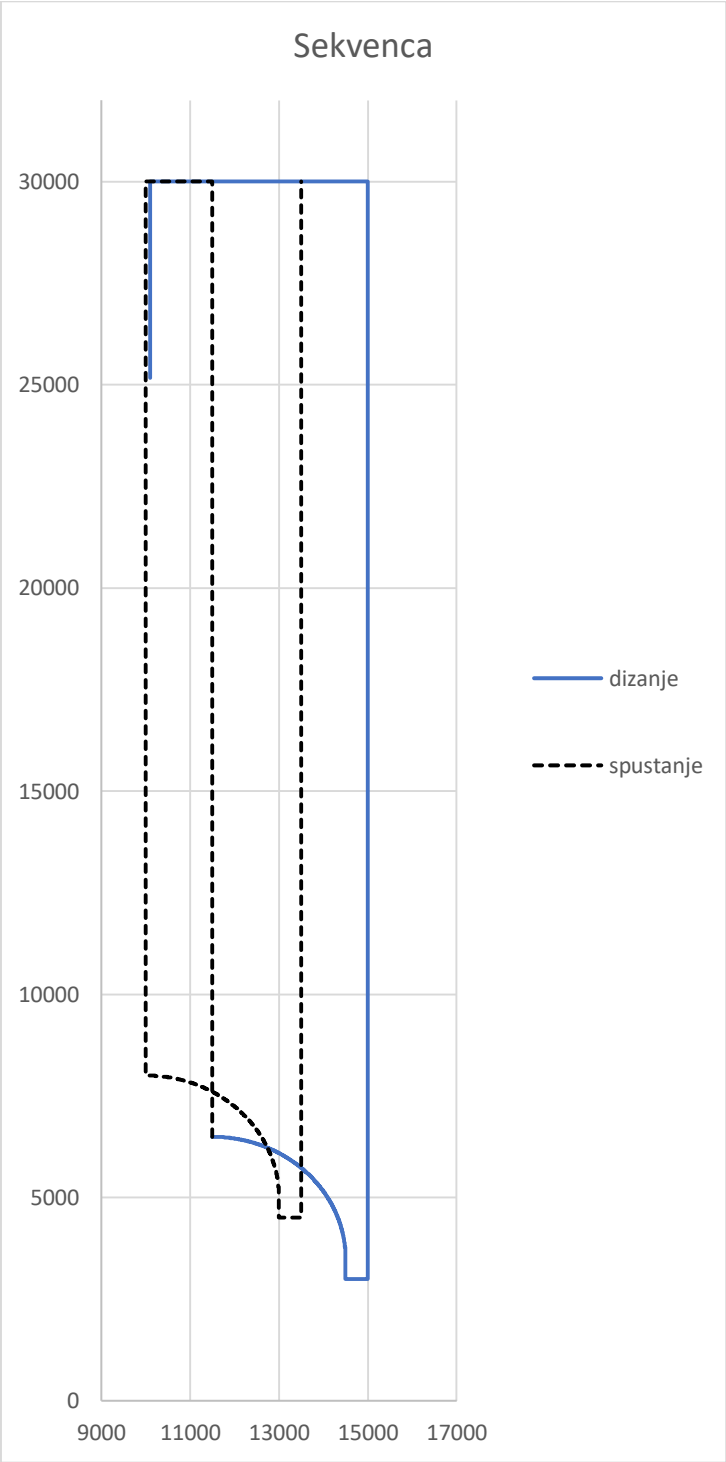
Slika 3.59. Usporedba prave i snimljene putanje po kružnici [Autor]

Nakon simulacije funkcije za podizanje nosača lonca simulirana je i testirana sekvenca za premještanje lonca s jedne lokacije tvornice na drugu. Cijela sekvenca sa brzinama, pozicijom i pripadnim segmentima sekvence se vidi na slici 3.60.. Neki dijelovi za dizanje i spuštanje vitla su ubrzani kako bi se uspijela snimiti cijela sekvenca, to su dijelovi označeni sa S200, S300, S400 i S423.



Slika 3.60. Snimka pozicija i brzina glavnog vitla i kranskih kolica tijekom sekvence [Autor]

Kao provjera su pozicije kao i prije eksportirane u excel gdje je napravljen graf koji pokazuje gibanje u 2D prostoru kao što je prikazano na slici 3.61.. Dizanje je označeno sa punom plavom linijom dok je spuštanje označeno crnom isprekidanom linijom. Dio za spuštanje traje sve od trenutka vaganja mase tereta lonca pa do konačne pozicije lonca.



Slika 3.61. Pozicija glavnog vitla i kranskih kolica u prostoru tijekom sekvence [Autor]

4. Vizualizacija procesa

4.1. Općenito o HMI-u

HMI(*eng. Human-Machine Interface*) tj. sučelje čovjek-stroj. Putem HMI se zapravo ostvaruje interakcija između čovjeka i stroja, tipično je to kroz neko grafičko sučelje. HMI je dizajniran tako da što jednostavnije prikaže i upravlja sustav koji se nadgleda.

U svrhu ovog diplomskog rada HMI je izrađen u Tia Portalu 17, konkretno na panelu TP900 Comfort (slika 4.1.).

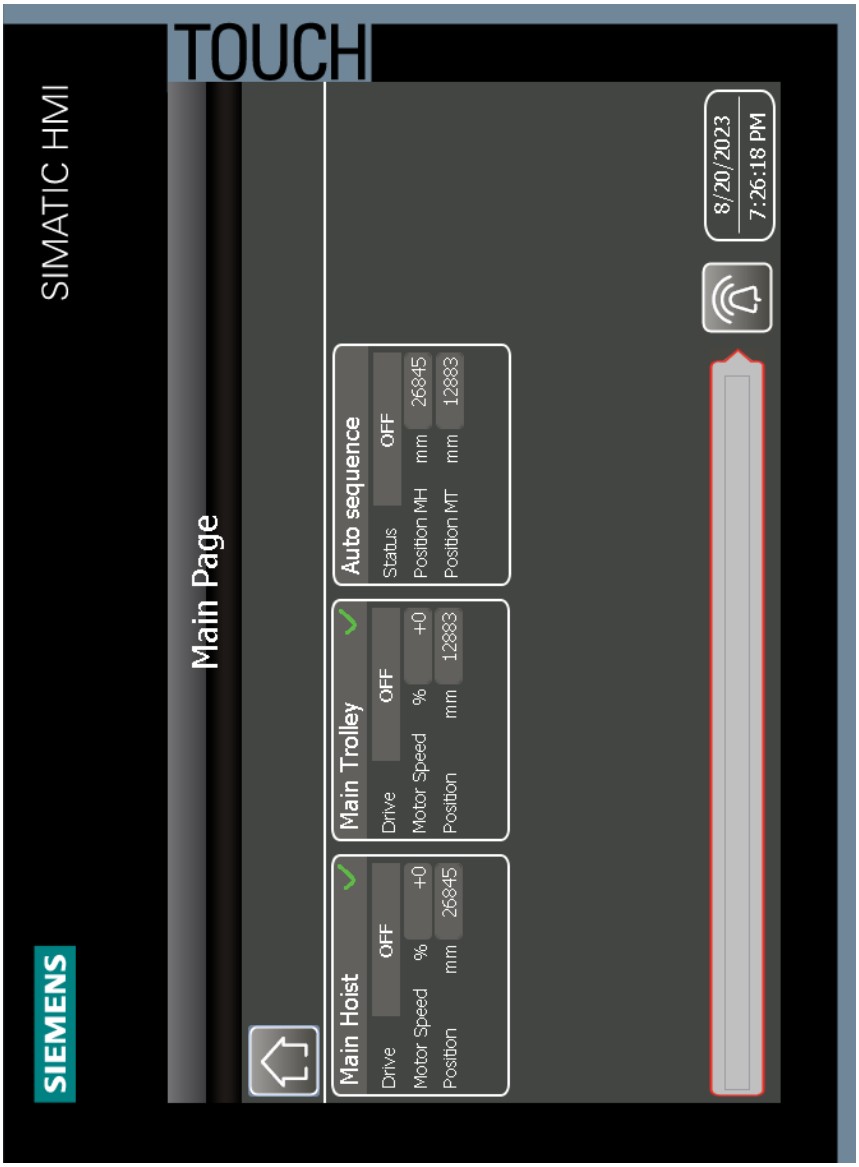
Slika 4.1. TP900 Comfort HMI panel [8]

TP900 Comfort specifikacije:

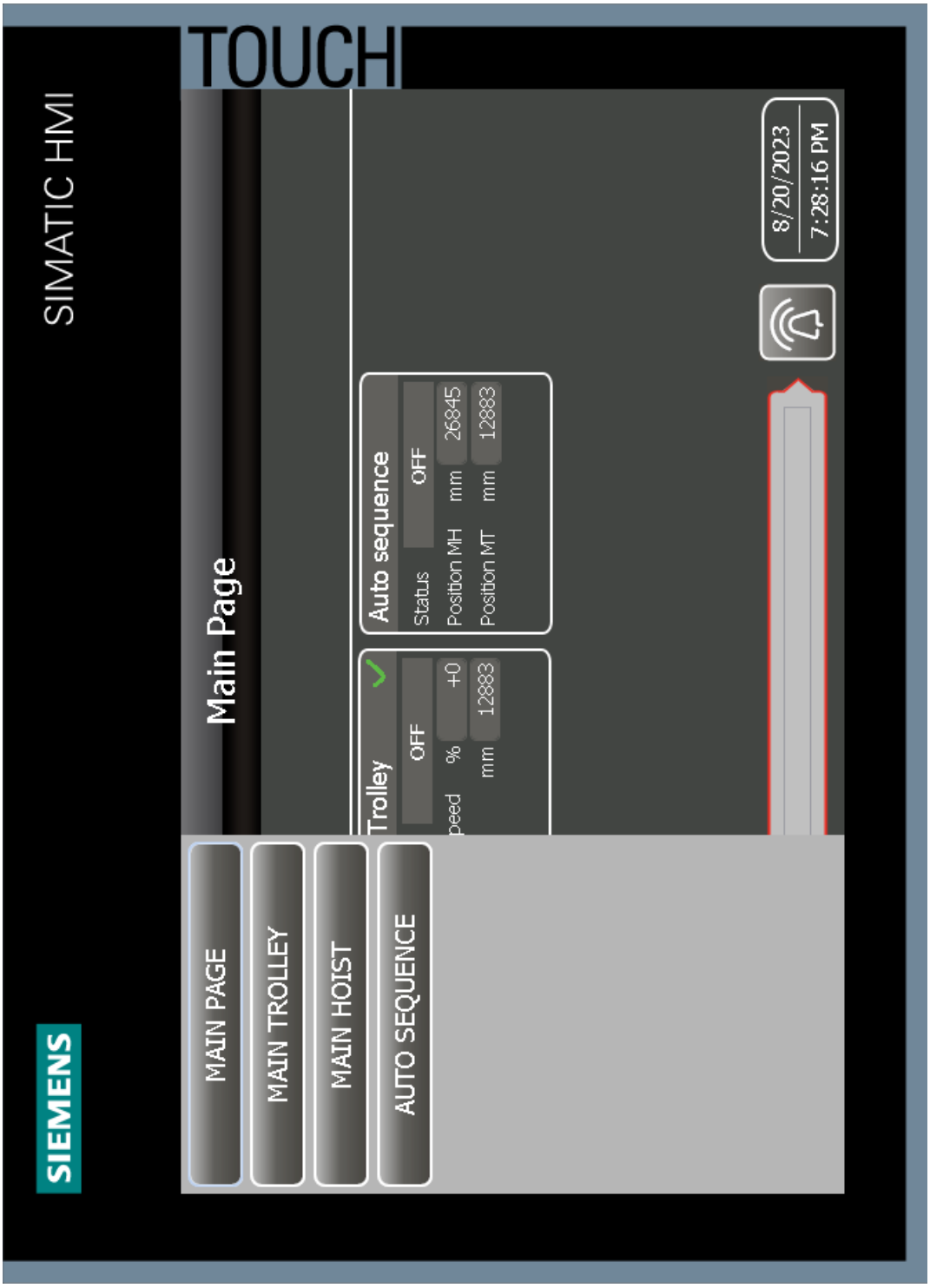
- Ulazno napajanje 24VDC 18W
- 9 inč displej
- Rezolucija 800x480 piksela
- 2 PROFINET porta za komunikaciju
- Konfiguracijski softver WinCC Comfort

4.2. Izgled HMI stranica

Na prvoj početnoj stranici sa slike 4.2. su prikazane tri kućice sa osnovnim informacijama o svakom od sustava, Glavnog vitla (*eng. Main Hoist*), Kranskih kolica (*eng. Main Trolley*) i automatske sekvence (*eng. Auto Sequence*). Za vitlo i kolica je prikazano stanje frekvencijskog pretvarača (*eng. Drive*), brzina vrtnje motora (*eng. Motor Speed*) i pozicija (*eng. Position*), dok je za auto sekvencu prikazano stanje sekvence koje može biti isključeno (*eng. Off*) ili aktivno (*eng Active*). Također na svakoj od stranica nalazi se ime stranice na vrhu, dugme sa kućicom koja otvara meni sa izborom stranica (slika 4.3.) koje se mogu otvoriti te na dnu informacije o zadnjem ili trenutno aktivnom alarmu, kao i dugme za otvaranje svih alarma.



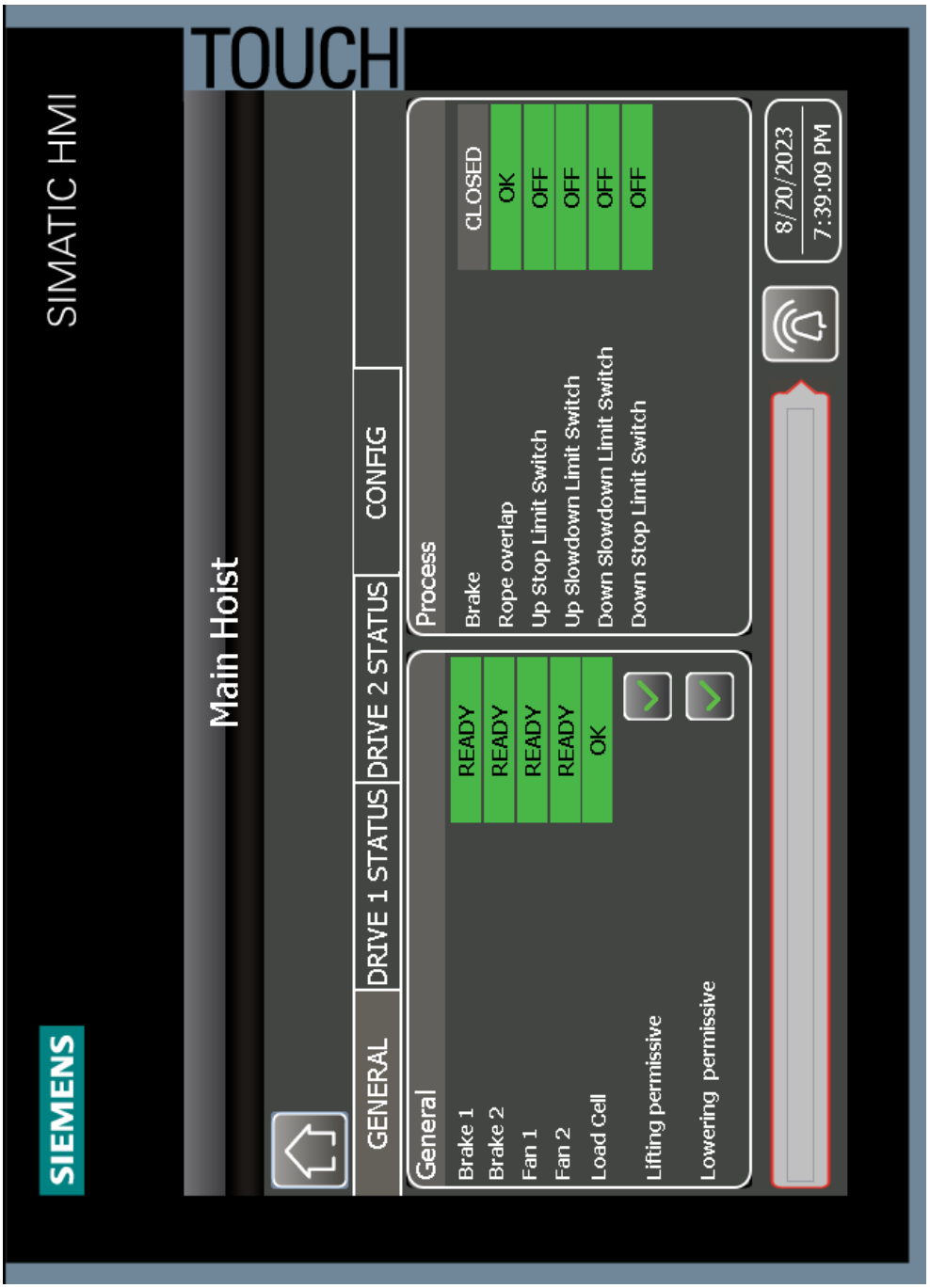
Slika 4.2. Početna stranica [Autor]



Slika 4.3. Meni za otvaranje stranica [Autor]

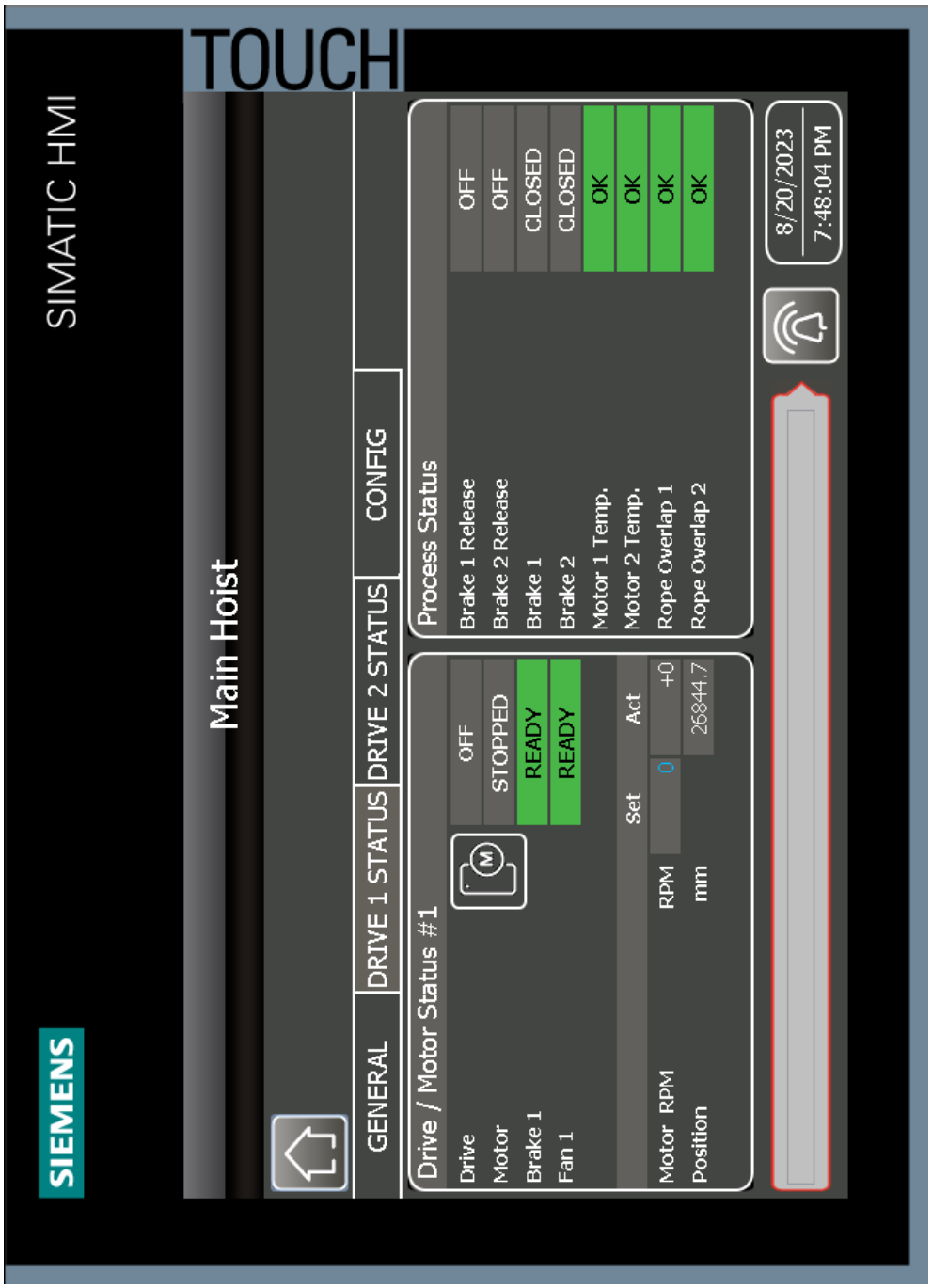
Kao što je moguće kliknuti na dugme iz menija za otvaranje stranica, tako je moguće kliknuti i na bilo koji od prozora na početnoj stranici kako bi se otvorila ta određena stranica.

Stranica glavnog vitla sadrži 4 kartice. Prva je generalne informacije (*eng. General*) druge dvije su rezervirane za informacije o pretvaračima (*eng Drive status*) te posljednja kartica je za konfiguraciju (*eng. Config*). Stranica naziva General prikazuje statuse kočnica, ventilatora te vage. Također prikazuju se neke informacije o procesu, kao što su otvorenost kočnice, preklapanje sajle te stanja krajnjih prekidača. Kao što se vidi na slici 4.4. postoje i prikaz dozvola za rad, u ovom slučaju to su dozvole za dizanje i spuštanje vitla.



Slika 4.4. General stranica glavnog vitla [Autor]

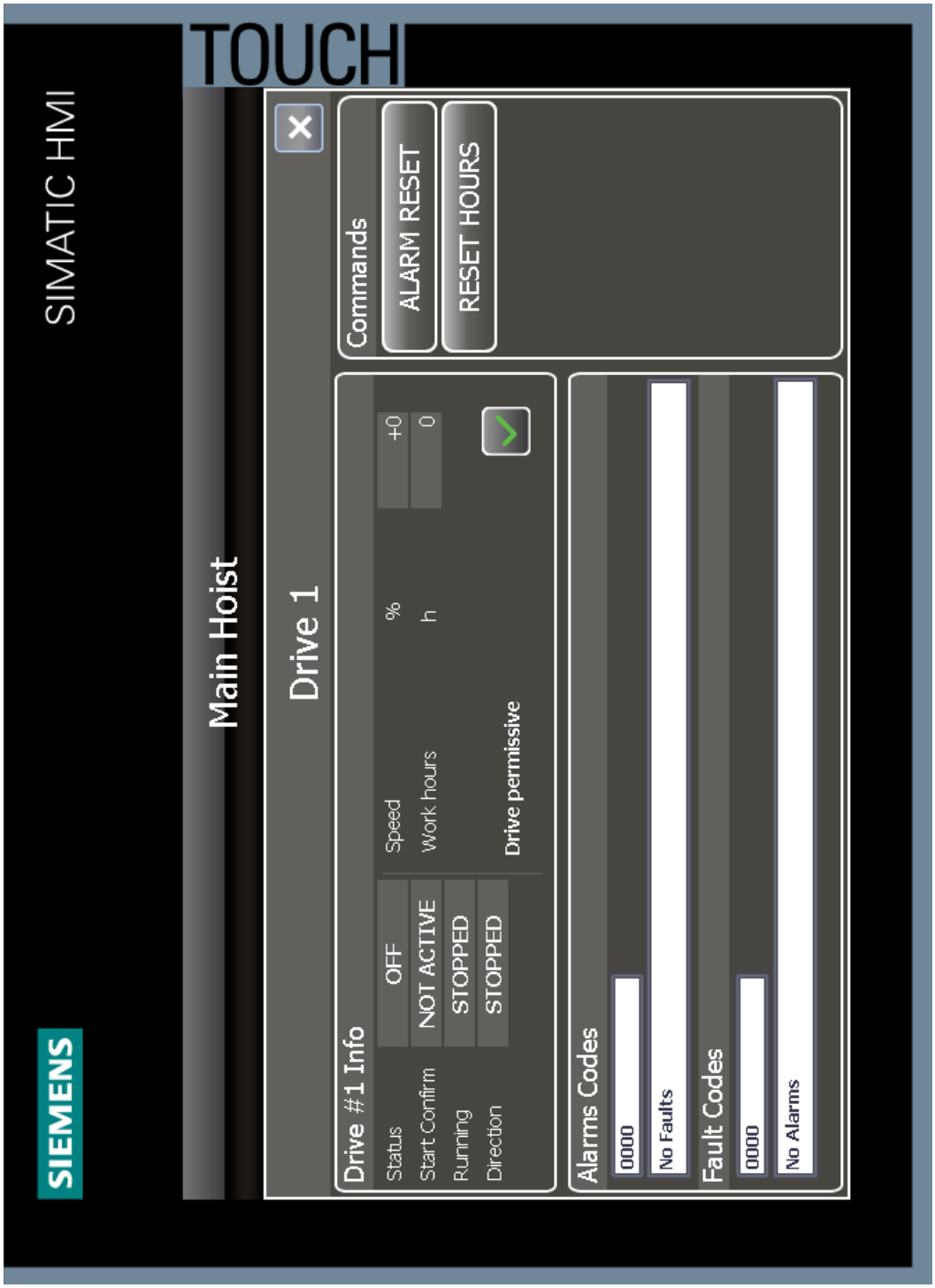
Na kartici za prikaz informacija o pretvaraču je prikazano stanje u kojem se nalazi pretvarač, kao i motor. Također vidljiva su stanja kočnica glavnog krana, kao i zaštita temperature motora te preklapanje sajle vitla. Na slici 4.5. se vidi i brzina vrtnje motora kao referentna ili željena vrijednost te aktualna u okretajima po minuti. Prikazuje se i pozicija vitla u prostoru. Slika 4.5. prikazuje stanje vitla u mirovanju. Klikom na dugme sa slikom pretvarača otvara se dodatna stranica sa informacijama o pretvaraču.



Slika 4.5. Drive status stranica glavnog vitla [Autor]

Dodatna pop-up pretvarač stranica(slika 4.6.) prikazuje sve statuse pretvarača zasebno kao i smjer vrtnje motora, brzinu u postotku te radne sate motora. Također prikazan je i uvjet za rad pretvarača te komande za resetiranje alarma na pretvaraču te komanda za resetiranje radnih sat.

Na dnu se prikazuju alarmi i greške pretvarača sa svojim kodom.



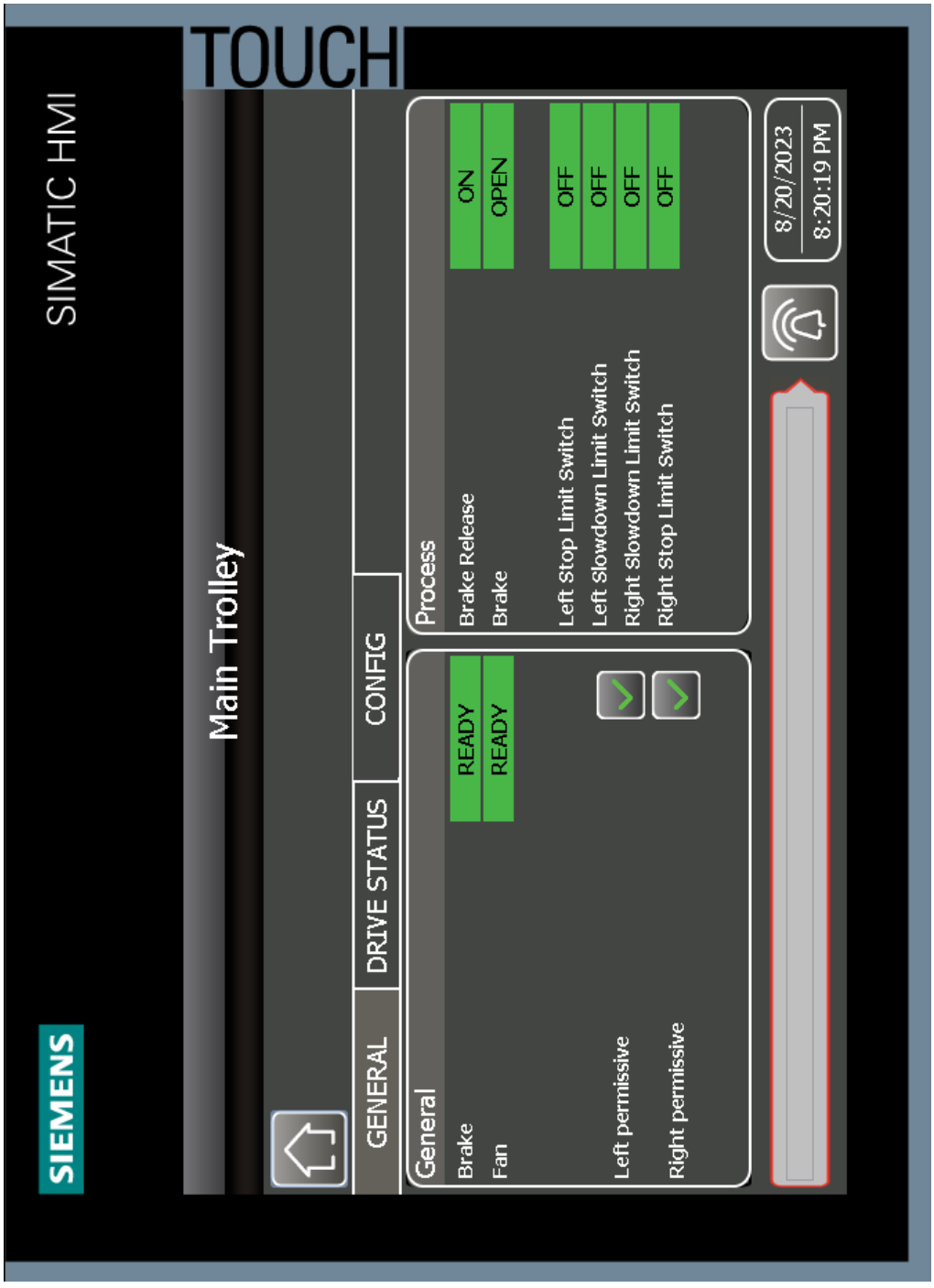
Slika 4.6. Drive pop-up stranica glavnog vitla [Autor]

Konfiguracijska stranica sadrži sve postavke glavnog vitla koje se mogu mijenjati, poput maksimalne brzine vrtnje motora, vremena akceleracije i deceleracije, koeficijent enkodera, ručne brzine za svaku razinu upravljačke palice itd. Config stranica je prikazana na slici 4.7.

Slika 4.7. Config stranica glavnog vitla [Autor]

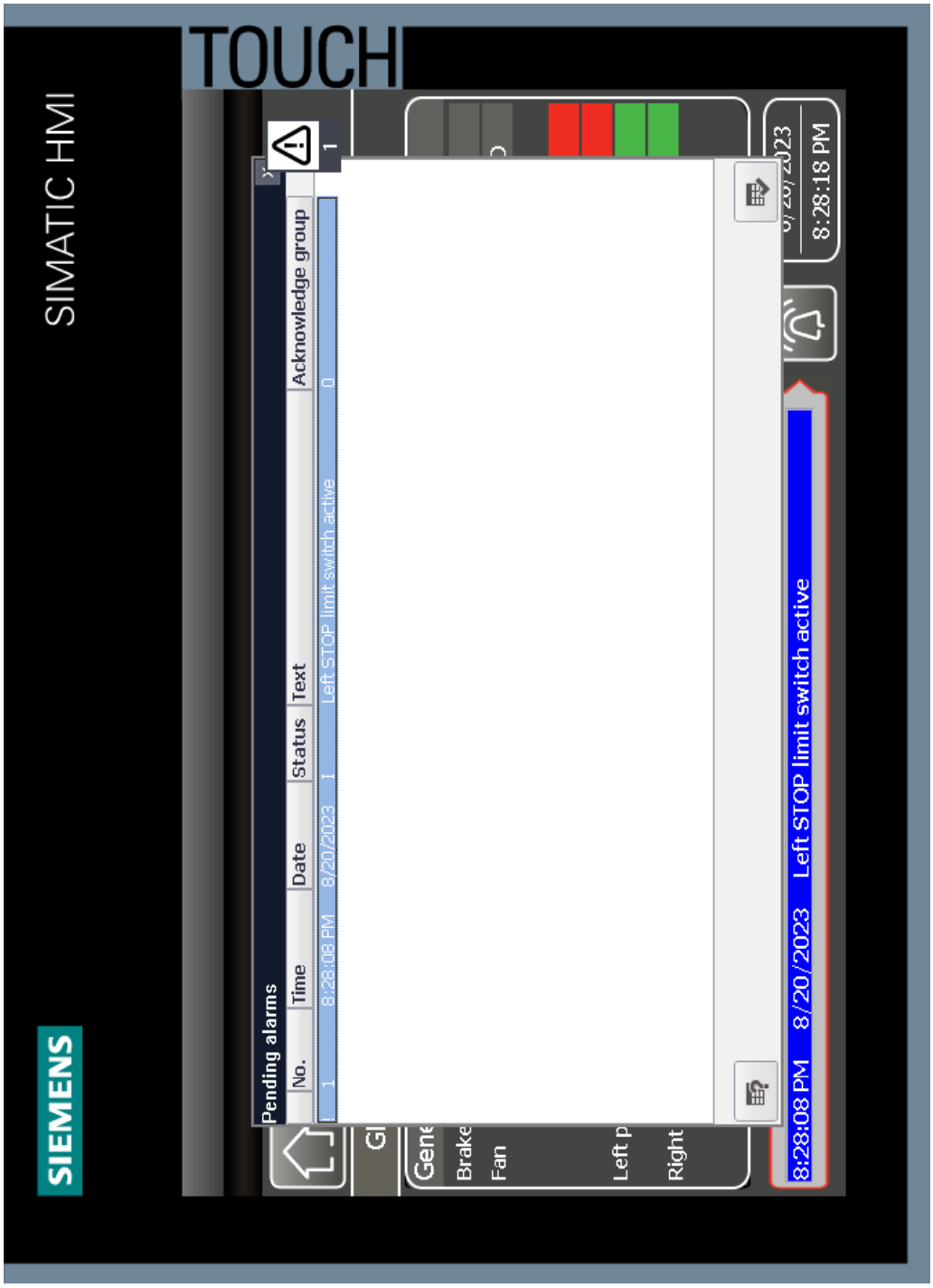
Glavno vitlo je bilo prikazano u stanju mirovanja, tako da će se kranska kolica prikazati u aktivnom stanju tj. s gibanjem ulijevo. Kao i kod vitla na početnoj stranici kranskih kolica imamo 3 kartice, opće informacije, pretvarač te konfiguracijsku karticu.

Kao što se vidi na slici 4.8. komanda za otvaranje kočnice je aktivna te je kočnica otvorena, svi krajnji prekidači su neaktivni te se gibanje nastavlja.

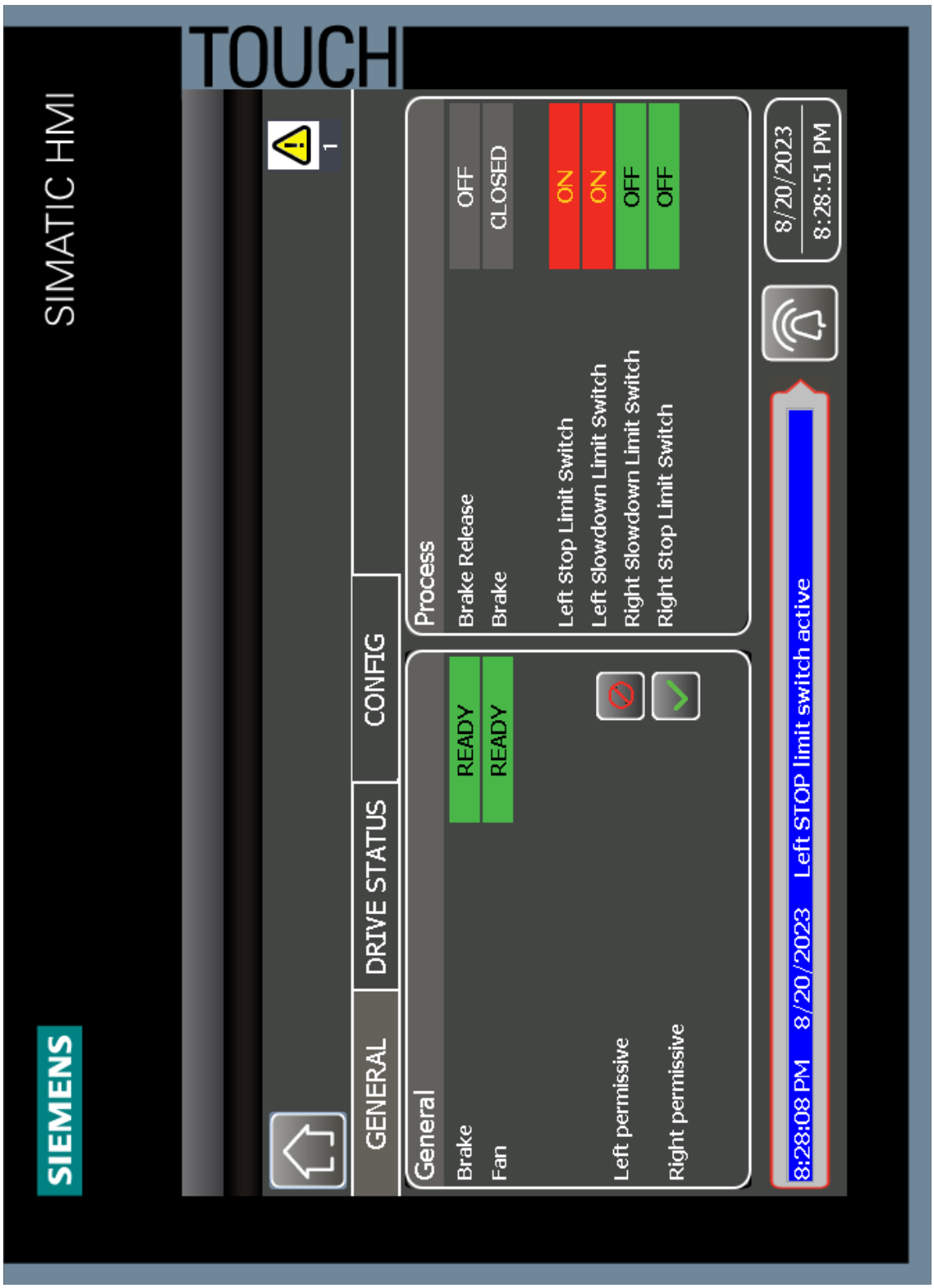


Slika 4.8. General stranica kranskih kolica u normalnom radu [Autor]

Nastavljanjem gibanja kranska kolica dolaze do kraja te aktiviraju lijevi krajnji stop prekidač, komanda za pokretanje kolica se prekida, kočnica zatvara te se javlja alarm. Također se može vidjeti da je dopuštenje za gibanje ulijevo poništeno dok je gibanje u desno i dalje omogućeno. Na slici 4.9. može se vidjeti kako se pojavi alarm za krajnji prekidač, Dok je na slici 4.10. stanje sustava kolica nakon aktivacije krajnjeg prekidača.

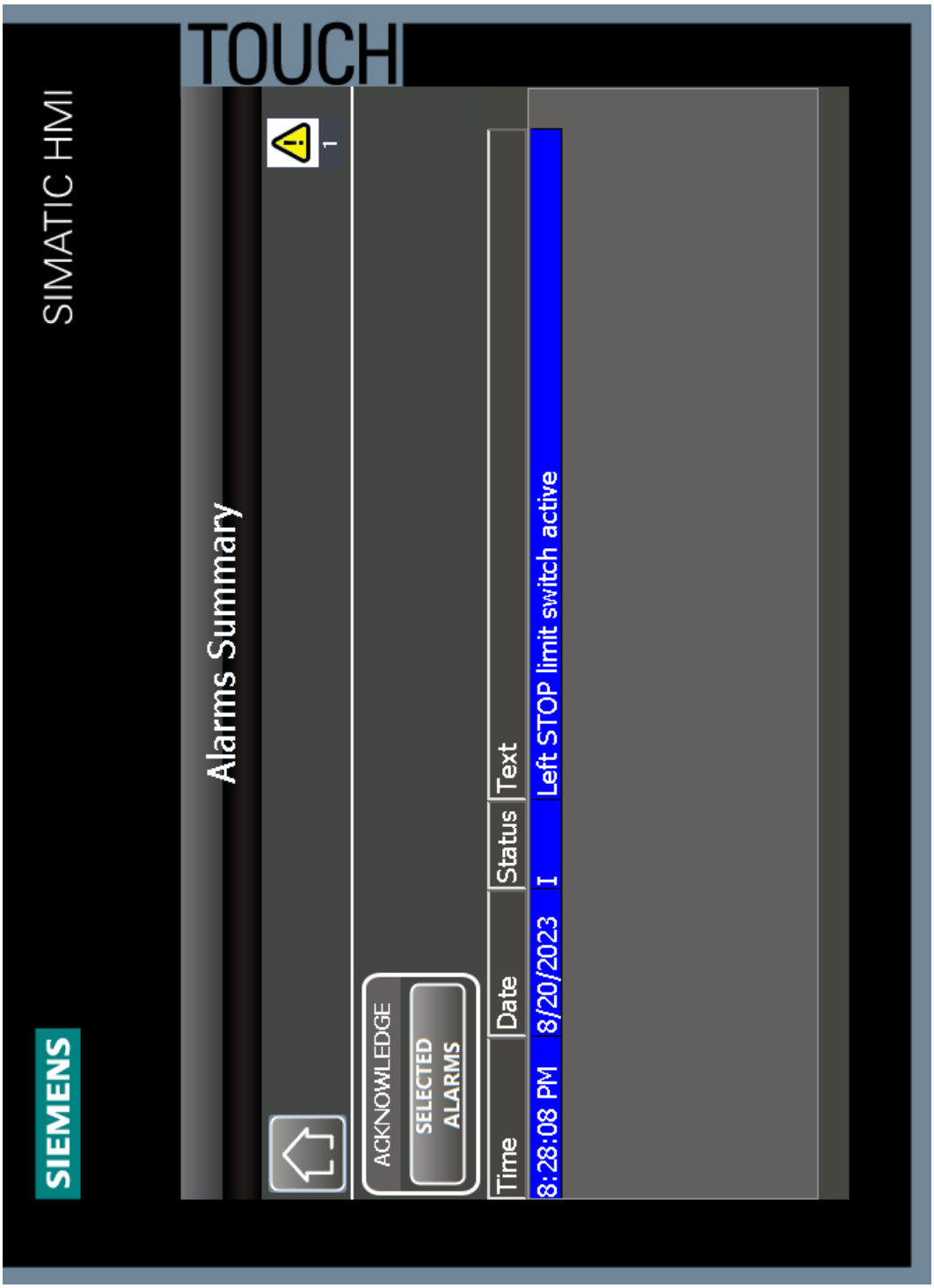


Slika 4.9. Aktivan alarm krajnjeg lijevog prekidača [Autor]



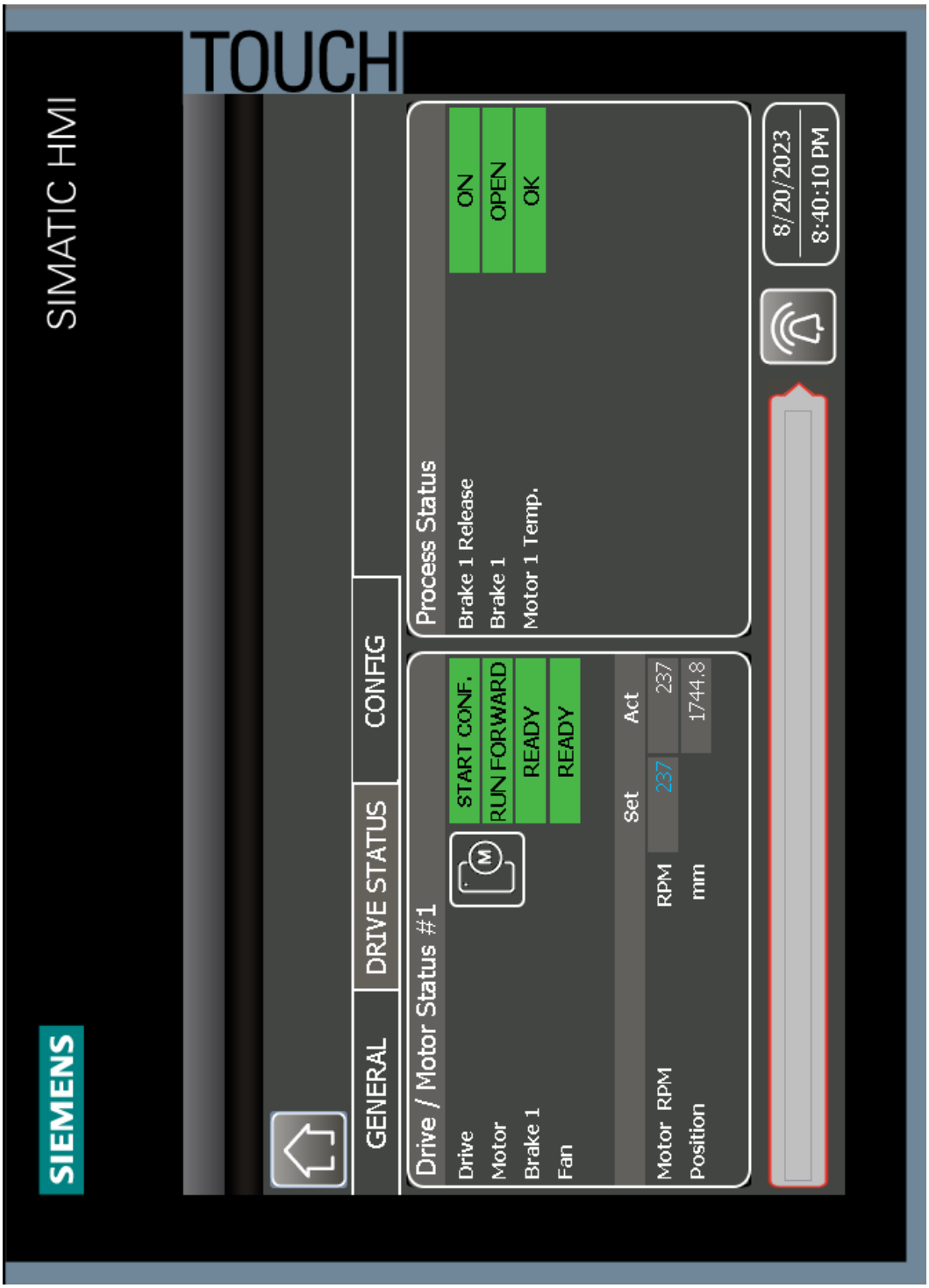
Slika 4.10. General stranica uz aktivan alarm krajnjeg lijevog prekidača [Autor]

Kako bi se alarm izbrisao potrebno je prvo pokrenuti kranska kolica udesno te prihvatiti alarm na stranici alarmi koja je dostupna klikom na dugme sa zvonom, kao na slici 4.11. na kojoj se nalazi lista svih alarma u sustavu do sada. Odabirom alarma koje se želi prihvatiti i klikom na dugme *selected alarms*, prihvate se svi označeni alarmi.



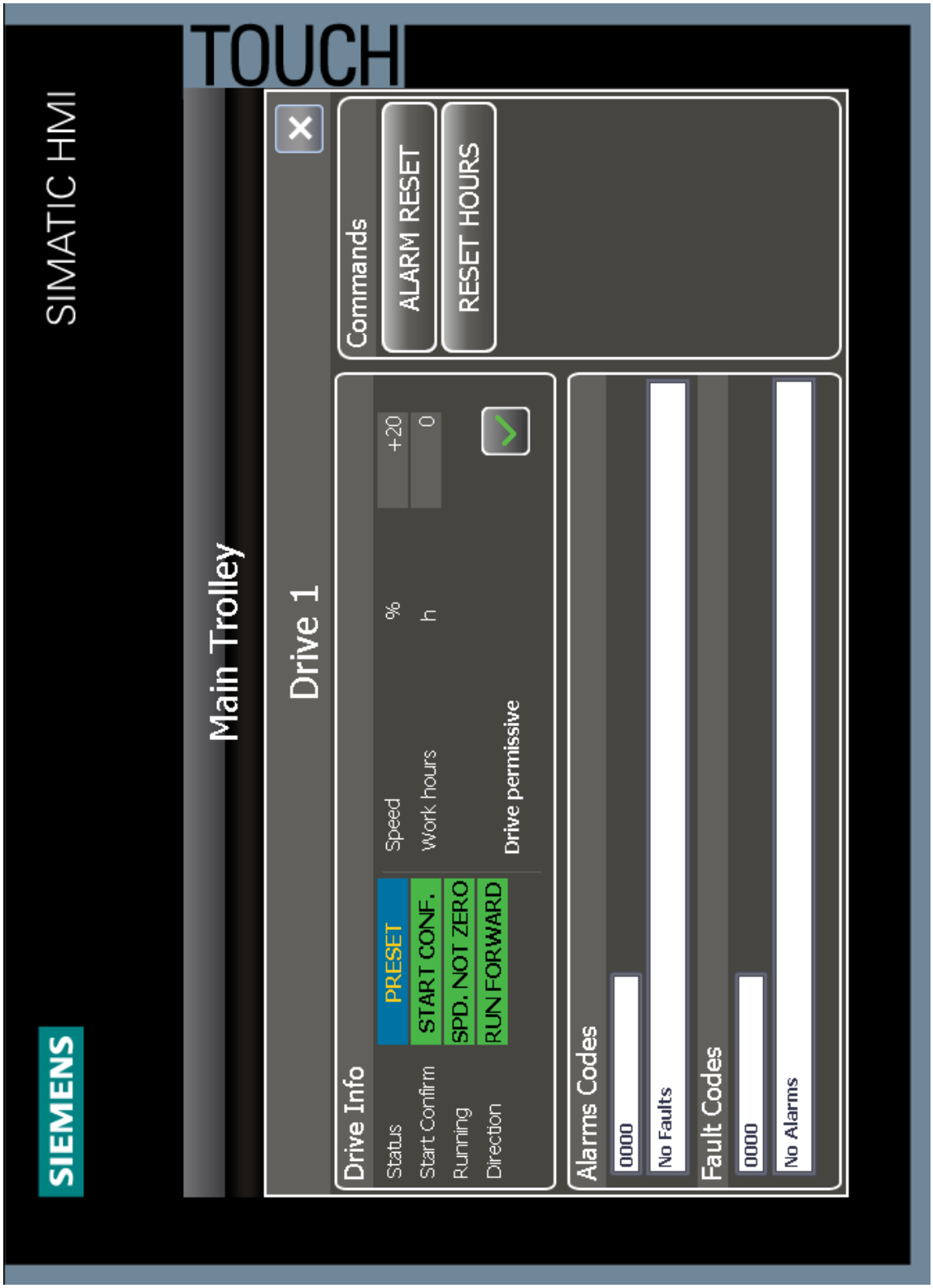
Slika 4.11. Stranica za prikaz svih alarma[Autor]

Sljedeća kartica kranskih kolica prikazuje status pretvarača, referentnu te trenutnu brzinu i poziciju pri kretanju u desno. Na slici 4.12. se vidi da je status pretvarača potvrđeni start te da se motor vrti prema naprijed.



Slika 4.12. Drive status stranica kranskih kolica [Autor]

Klikom na pretvarač dugme otvara se pop-up prozor sa dodatnim informacijama o pretvaraču(slika 4.13.), vidi se da je omogućen start pretvarača, te da se motor giba brzinom 20% maksimalne.



Slika 4.13. Drive pop-up stranica kranskih kolica [Autor]

Konfiguracijska stranica kranskih kolica je identična onoj glavnog vitla te se vidi na slici 4.14.

Slika 4.14. Config stranica kranskih kolica [Autor]

5. Zaključak

Izrada ovog diplomskog rada bila je izuzetno izazovna jer je zahtijevala detaljno proučavanje električnih shema mosnog krana sa svim njegovim aktuatorima i senzorima kako bi se razumjelo njegovo funkcioniranje te koje zadatke izvršava i na koji način.

Ključna komponenta za rješavanje problema kružnog kretanja bila je razrada upravljačke logike krana. Stoga je rad strukturiran tako da prvo opisuje osnovne karakteristike mosnog krana, komunikaciju sa specifičnim uređajima, te upravljačku logiku mosnog krana. Tek nakon što su ti temelji postavljeni, objašnjava se rješenje problema kružnog gibanja krana u prostoru.

Tijekom izrade programskog koda za kružno gibanje isprobana su mnoga moguća rješenja, no najbolje se pokazalo ono koje je opisano u ovom radu. Ukratko, ideja je postaviti brzinu jedne osi kao referentnu i na temelju pozicije i brzine te osi izračunati brzinu druge osi. Kada kut kružnog gibanja prijeđe 45 stupnjeva, referentna os i os kojoj se brzina računala zamijene do krajnjeg položaja, odnosno 90 stupnjeva.

Cijeli sustav je prošao kroz testiranje putem simulacije, a rezultati ukazuju na ispravno funkcioniranje upravljačke logike. Kružni pokret krana prati stvarni oblik kružnice u simuliranim uvjetima. Važno je napomenuti da treba pristupiti tim rezultatima s oprezom, s obzirom na to da simulacijski uvjeti pružaju idealne okolnosti. U stvarnom radnom okruženju postoje brojni vanjski faktori koji mogu utjecati na kretnju krana. Primarni među njima su njihanje vitla, elastičnost sajle, preciznost senzora te drugi potencijalni problemi zbog kojih je potrebno izvršiti i testiranje u radnom okruženju.

Literatura

[1] Intelligent motion: „Crane kits“, s Interneta, https://www.cmco.com/en-us/products/crane-systems/crane-kits/, 28. srpnja 2023.

[2] Dokumentacija dobivena u firmi Danieli Systec

[3] Simocrane: „Drive - Based Sway Control“, s interneta, https://assets.new.siemens.com/siemens/assets/api/uuid:e59ee3f4-f77f-4428-aded-5159e15d0450/simocrane-drive-based-sway-control.pdf, 31. srpnja 2023.

[4] Na, L.: „Application of Siemens Smart200 and S7-1200 Plc Complex Instructions in Industrial Automation“

[5] Michael, B.: „What is Profinet? – Profinet Explained“, s interneta, https://us.profinet.com/profinetexplained/#:~:text=PROFINET%20is%20an%20open%20Industrial,well-adopted%20Industrial%20Ethernet%20solution, 3. kolovoza 2023.

[6] Kutija, M.; Sumina, D.: „Industrijski komunikacijski protokol Profinet u elektromotornim pogonima“, s interneta, https://www.fer.unizg.hr/_download/repository/EMP_Profinet[6].pdf, 3. kolovoza 2023.

[7] Siemens, „Integration of Siemens S Drives into Studio 5000“, s interneta, https://cache.industry.siemens.com/dl/files/369/92045369/att_1086002/v1/Integration_of_Siemens_S_Drives_into_Studio_5000.pdf, 10. kolovoza 2023.

[8] Siemens, „HMI devices Comfort Panels INOX, ITC INOX“, s interneta, https://cache.industry.siemens.com/dl/files/115/109478115/att_904796/v1/hmi_comfort_itc_inox_compact_operating_instructions_en-US.pdf, 20. kolovoza 2023.

Sažetak i ključne riječi

U zahtjevnom okruženju kao što je proizvodnja čelika, postizanje preciznosti i učinkovitosti ima ključan značaj. Pomicanje teških materijala i proizvoda, osobito u industriji čelika, zahtijeva izuzetno složenu kontrolu i koordinaciju među različitim sustavima. Kranovi igraju ključnu ulogu u tim procesima, omogućujući rukovanje materijalima, sirovinama i transport proizvoda tijekom cjelokupnog proizvodnog lanca. Ovaj rad detaljno analizira sofisticiranu kontrolu krana, komunikaciju između PLC-a i frekvencijskog pretvarača, sveobuhvatan softver potreban za upravljanje krana te na kraju, funkciju koja usklađuje kretanje dvaju mehanički neovisnih sustava - glavnog vitla i kranskih kolica - kako bi zajedno ostvarili preciznu kružnu putanju. Konkretno, kružna putanja krana koristi se prilikom podizanja nosača lonca radi premještanja lonca s jedne na drugu lokaciju u tvornici. Posljednji dio rada usmjeren je na razvoj HMI korisničkog sučelja za prikaz i upravljanje cijelim sustavom.

Ključne riječi: kran, upravljanje, motor, softver, sinkronizirano upravljanje, HMI, PLC, Tia portal

Abstract and keywords

In a demanding environment like steel production, achieving precision and efficiency holds paramount significance. Moving heavy materials and products, particularly in the steel industry, necessitates intricate control and coordination among various systems. Cranes play a pivotal role in these processes, enabling the handling of materials, raw goods, and product transportation throughout the entire production chain. This study extensively delves into the intricate crane control, communication between the PLC and drive, comprehensive software required for crane management, and ultimately, the function that synchronizes the movement of two mechanically independent systems - the main hoist and the trolley - to achieve a precise circular trajectory. Specifically, the crane's circular path is employed when lifting a ladle carrier for relocating it from one position to another within the factory. The final section of the study is focused on developing an HMI user interface for displaying and managing the entire system.

Keywords: crane, control, motor, software, synchronized control, HMI, PLC, Tia Portal

Dodatak A

Totally Integrated Automation Portal

CRANE / PLC_1 [CPU 1518-4 PN/DP] / Program blocks / 100_Functions

POSITIONER [FB1102]

POSITIONER Properties							
General							
Name	POSITIONER	Number	1102	Type	FB	Language	SCL
Numbering	Manual						
Information							
Title		Author		Comment		Family	
Version	0.1	User-defined ID					

POSITIONER

Name	Data type	Default value	Retain	Accessible from HMI/OPC UA/Web API	Writable from HMI/OPC UA/Web API	Visible in HMI engineering	Setpoint	Supervision	Comment
▼ Input									
b_startMotion	Bool	false	Non-retain	False	False	False	False		
b_minSlowDownLS	Bool	false	Non-retain	False	False	False	False		
b_maxSlowDownLS	Bool	false	Non-retain	False	False	False	False		
b_minStopLS	Bool	false	Non-retain	False	False	False	False		
b_maxStopLS	Bool	false	Non-retain	False	False	False	False		
r_currentPosition	Real	0.0	Non-retain	False	False	False	False		[mm]
r_desiredPosition	Real	0.0	Non-retain	False	False	False	False		[mm]
r_currentSpeed	Real	0.0	Non-retain	False	False	False	False		[mm/s]
r_maxSpeed	Real	0.0	Non-retain	False	False	False	False		[mm/s]
r_setSpeed	Real	0.0	Non-retain	False	False	False	False		[mm/s]
r_slowSpeedDistance	Real	0.0	Non-retain	False	False	False	False		[mm]
r_slowSpeed	Real	0.0	Non-retain	False	False	False	False		[mm/s]
r_Acceleration	Real	0.0	Non-retain	False	False	False	False		[mm/s^2]
r_Deacceleration	Real	0.0	Non-retain	False	False	False	False		[mm/s^2]
r_CycleTime_sec	Real	0.0	Non-retain	False	False	False	False		[s]
▼ Output									
r_outSpeed_MMPS	Real	0.0	Non-retain	False	False	False	False		[mm/s]
r_outSpeed_PER	Real	0.0	Non-retain	False	False	False	False		[mm/s]
b_fwdDirection	Bool	false	Non-retain	False	False	False	False		
b_revDirection	Bool	false	Non-retain	False	False	False	False		
r_outSlowDownPos	Real	0.0	Non-retain	False	False	False	False		[mm]
r_outStopPos	Real	0.0	Non-retain	False	False	False	False		[mm]
InOut									
▼ Static									
r_slowDownArea	Real	0.0	Non-retain	False	False	False	False		
r_slowDownPos	Real	0.0	Non-retain	False	False	False	False		
b_slowDownStarted	Bool	false	Non-retain	False	False	False	False		
b_slowSpeedSet	Bool	false	Non-retain	False	False	False	False		
b_firstCycleDone	Bool	false	Non-retain	False	False	False	False		
b_stopRequest	Bool	false	Non-retain	False	False	False	False		
r_deltaPosition	Real	0.0	Non-retain	False	False	False	False		
r_StopArea	Real	0.0	Non-retain	False	False	False	False		
r_stopPos	Real	0.0	Non-retain	False	False	False	False		
▼ Temp									
r_speed	Real								
Constant									

```
0001 IF #r_setSpeed >= #r_maxSpeed THEN
0002   #r_speed := #r_maxSpeed;
0003 ELSE
0004   #r_speed := #r_setSpeed;
0005 END_IF;
0006
0007 // Delta position - to check how much we travel in actual PLC cycle with current speed
0008 #r_deltaPosition := #r_CycleTime_sec * #r_currentSpeed;
0009
0010 // Calculation for slow down position in which actual speed must decelerate to slow speed
0011 // Calculation is done only if slow speed distance is bigger then 0.0 mm and if slow speed is different then 0 mm/s.
0012 IF #r_slowSpeedDistance > 0.0 AND #r_slowSpeed > 0.0 THEN
0013
0014   // Slow down space neccessary to go from fast speed to slow speed
0015   // It will be checked only if actual speed is different than 0
0016   IF #r_currentSpeed <> 0.0 THEN
0017     #r_slowDownArea := (SQR(#r_currentSpeed) - SQR(#r_slowSpeed)) / (2.0 * #r_Deacceleration);
0018   ELSE
0019     #r_slowDownArea := 0.0;
0020   END_IF;
0021
0022   IF #r_currentSpeed > 0.0 THEN
0023     #r_slowDownPos := #r_desiredPosition - #r_slowDownArea - #r_slowSpeedDistance;
0024
```

```
0025      // Condition is checked for next PLC cycle.
0026      // If (actual position + delta position) is bigger then slowdown position then target speed must be slow speed
0027      IF (#r_currentPosition + #r_deltaPosition >= #r_slowDownPos) OR (#b_maxSlowDownLS) THEN
0028        #b_slowSpeedSet := TRUE;
0029      END_IF;
0030
0031    END_IF;
0032
0033    // Calculation for slow down position in which actual speed must decelerate to slow speed
0034    IF #r_currentSpeed < 0.0 THEN
0035      #r_slowDownPos := #r_desiredPosition + #r_slowDownArea + #r_slowSpeedDistance;
0036
0037      // Condition is checked for next PLC cycle.
0038      // If (actual position - delta position) is smaller then slowdown position then target speed must be slow speed
0039      IF (#r_currentPosition - #r_deltaPosition <= #r_slowDownPos) OR (#b_minSlowDownLS) THEN
0040        #b_slowSpeedSet := TRUE;
0041      END_IF;
0042    END_IF;
0043
0044  END_IF;
0045
0046  // Calculation for stop position in which actual speed must decelerate to 0 speed
0047  #r_StopArea := (SQR(#r_currentSpeed)) / (2.0 * #r_Deacceleration);
0048
0049  IF #r_currentSpeed > 0.0 THEN
0050    #r_stopPos := #r_desiredPosition - #r_StopArea;
0051
0052    // Condition is checked for next PLC cycle.
0053    // If (actual position + delta position) is bigger then stop position then target speed must be 0 speed
0054    IF (#r_currentPosition + #r_deltaPosition >= #r_stopPos) OR (#b_maxStopLS) THEN
0055      #b_stopRequest := TRUE;
0056    END_IF;
0057
0058  END_IF;
0059
0060  // Calculation for stop position in which actual speed must decelerate to 0 speed
0061  IF #r_currentSpeed < 0.0 THEN
0062    #r_stopPos := #r_desiredPosition + #r_StopArea;
0063
0064    // Condition is checked for next PLC cycle.
0065    // If (actual position + delta position) is bigger then stop position then target speed must be 0 speed
0066    IF (#r_currentPosition - #r_deltaPosition <= #r_stopPos) OR (#b_minStopLS) THEN
0067      #b_stopRequest := TRUE;
0068    END_IF;
0069  END_IF;
0070
0071
0072  IF #b_startMotion THEN
0073
0074    // Direction
0075    IF #r_currentPosition < #r_desiredPosition THEN
0076
0077      #b_fwdDirection := TRUE;
0078      #b_revDirection := FALSE;
0079
0080    ELSIF #r_currentPosition > #r_desiredPosition THEN
0081
0082      #b_fwdDirection := FALSE;
0083      #b_revDirection := TRUE;
0084
0085    END_IF;
0086
0087    IF #b_slowSpeedSet AND NOT #b_stopRequest THEN
0088      #r_outSpeed_MMPS := #r_slowSpeed;
0089      #r_outSpeed_PER := #r_slowSpeed / #r_maxSpeed * 100.0;
0090    ELSIF #b_stopRequest THEN
0091      #r_outSpeed_MMPS := 0;
0092      #r_outSpeed_PER := 0;
0093    ELSIF NOT #b_slowSpeedSet AND NOT #b_stopRequest THEN
0094      #r_outSpeed_MMPS := #r_speed;
0095      #r_outSpeed_PER := #r_speed / #r_maxSpeed * 100.0;
0096    END_IF;
0097
0098    IF NOT #b_slowSpeedSet THEN
0099      #r_outSlowDownPos := #r_slowDownPos;
0100    END_IF;
0101
0102    IF NOT #b_stopRequest THEN
0103      #r_outStopPos := #r_stopPos;
0104    END_IF;
0105
0106  ELSE
0107    #b_slowSpeedSet := FALSE;
0108    #b_stopRequest := FALSE;
0109    #b_fwdDirection := FALSE;
0110    #b_revDirection := FALSE;
0111    #b_firstCycleDone := FALSE;
0112    #b_slowDownStarted := FALSE;
```

```
0113    #b_slowSpeedSet := FALSE;
0114    #r_outSpeed_MMPS := 0;
0115 END_IF;
0116
0117
0118
0119
0120
0121
0122
0123
0124
0125
```

Symbol	Address	Type	Comment
#b_firstCycleDone		Bool	
#b_fwdDirection		Bool	
#b_maxSlowDownLS		Bool	
#b_maxStopLS		Bool	
#b_minSlowDownLS		Bool	
#b_minStopLS		Bool	
#b_revDirection		Bool	
#b_slowDownStarted		Bool	
#b_slowSpeedSet		Bool	
#b_startMotion		Bool	
#b_stopRequest		Bool	
#r_currentPosition		Real	[mm]
#r_currentSpeed		Real	[mm/s]
#r_CycleTime_sec		Real	[s]
#r_Deacceleration		Real	[mm/s^2]
#r_deltaPosition		Real	
#r_desiredPosition		Real	[mm]
#r_maxSpeed		Real	[mm/s]
#r_outSlowDownPos		Real	[mm]
#r_outSpeed_MMPS		Real	[mm/s]
#r_outSpeed_PER		Real	[mm/s]
#r_outStopPos		Real	[mm]
#r_setSpeed		Real	[mm/s]
#r_slowDownArea		Real	
#r_slowDownPos		Real	
#r_slowSpeed		Real	[mm/s]
#r_slowSpeedDistance		Real	[mm]
#r_speed		Real	
#r_StopArea		Real	
#r_stopPos		Real	