Upravljanje sustavom dvaju kolica

Brkić, Davor

Undergraduate thesis / Završni rad

2018

Degree Grantor / Ustanova koja je dodijelila akademski / stručni stupanj: **Josip Juraj Strossmayer University of Osijek, Faculty of Electrical Engineering, Computer Science and Information Technology Osijek / Sveučilište Josipa Jurja Strossmayera u Osijeku, Fakultet elektrotehnike, računarstva i informacijskih tehnologija Osijek**

Permanent link / Trajna poveznica: https://urn.nsk.hr/urn:nbn:hr:200:170459

Rights / Prava: In copyright / Zaštićeno autorskim pravom.

Download date / Datum preuzimanja: **2024-03-29**

Repository / Repozitorij:

Faculty of Electrical Engineering, Computer Science and Information Technology Osijek

zir.nsk.hr

Image not found or type unknown

SVEUČILIŠTE JOSIPA JURJA STROSSMAYERA U OSIJEKU

FAKULTET ELEKTROTEHNIKE, RAČUNARSTVA I INFORMACIJSKIH TEHNOLOGIJA

Sveučilišni studiji

UPRAVLJANJE SUSTAVOM DVAJU KOLICA

Završni rad

Davor Brkić

Osijek, 2018.

FERIT
FAKULTET ELEKTROTEHNIKE, RAČUNARSTVA I INFORMACIJSKIH TEHNOLOGIJA OSIJEK

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

Osijek,14.09.2018.

Odboru za završne i diplomske ispite

Prijedlog ocjene završnog rada

Ime i prezime studenta:	Davor Brkić
Studij, smjer:	Preddiplomski sveučilišni studij Računarstvo
Mat. br. studenta, godina upisa:	R3755, 25.09.2017.
OIB studenta:	16305458229
Mentor:	Prof.dr.sc. Robert Cupec
Sumentor:	
Sumentor iz tvrtke:	Dario Brkić
Naslov završnog rada:	Upravljanje sustavom dvaju kolica
Znanstvena grana rada:	**Automatika (zn. polje temeljne tehničke znanosti)**
Predložena ocjena završnog rada:	Izvrstan (5)
Kratko obrazloženje ocjene prema Kriterijima za ocjenjivanje završnih i diplomskih radova:	Primjena znanja stečenih na fakultetu: 3 bod/boda Postignuti rezultati u odnosu na složenost zadatka: 3 bod/boda Jasnoća pismenog izražavanja: 3 bod/boda Razina samostalnosti: 3 razina
Datum prijedloga ocjene mentora:	14.09.2018.
Datum potvrde ocjene Odbora:	26.09.2018.

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

FERIT
FAKULTET ELEKTROTEHNIKE, RAČUNARSTVA
I INFORMACIJSKIH TEHNOLOGIJA **OSIJEK**

IZJAVA O ORIGINALNOSTIRADA

Osijek, 26.09.2018.

Ime i prezime studenta:	Davor Brkić
Studij:	Preddiplomski sveučilišni studij Računarstvo
Mat. br. studenta, godina upisa:	R3755, 25.09.2017.
Ephorus podudaranje [%]:	1 %

Ovom izjavom izjavljujem da je rad pod nazivom**: Upravljanje sustavom dvaju kolica**

izrađen pod vodstvom mentora Prof.dr.sc. Robert Cupec

i sumentora

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

Potpis studenta:

Sadržaj

1. UVOD 1
1.1 Zadatak završnog rada 1
2. RAZVOJNA OKRUŽENJA KORIŠTENA U RADU 2
2.1. Step 7 2
2.2. CoDeSys 2
3. KOLICA S LONCEM 3
3.1. Upravljanje brzinom vrtnje motora kolica s loncima 4
3.2. Mehanički prekidači za praćenje pozicije kolica sa loncima 13
3.3. Automatske sekvence kolica s loncima 15
3.4. Izbjegavanje kolizije kolica 19
3.5. Simulator pozicije - praćenje pozicije pomoću virtualnog enkodera 25
4. DIZALO 27
4.1. Podizanje i spuštanje dizala, izmjenjivanje kolica ispod dizala 27
5. VIZUALIZACIJA U CoDeSys-U 32
6. ZAKLJUČAK 35
LITERATURA 36
SAŽETAK 37
ABSTRACT 38
ŽIVOTOPIS 39
PRILOZI 40

1. UVOD

Talionice željeza za transport vrućeg željeza između peći koriste specijalno napravljene lonce koji se transportiraju pomoću kolica (engl. *ladle car*). Tema ovog rada je sustav koji se sastoji od dvaju kolica s loncima vrućeg željeza. Upravljački program sastoji se od programskih blokova za automatsko upravljanje dvojih kolica, mehaničkih prekidača, izmjenjivanja kolica, brzine vrtnje motora i dizala. Sustav upravljanja treba omogućiti automatsko pozicioniranje i pozicioniranje pomoću ručnih kontrola, izbjegavanje kolizije obaju kolica i upravljanje brzinom vrtnje motora. Poznavanje pozicije pokretnih objekata važno je zbog planiranja budućih kretanja. Praćenje dvaju kolica je omogućeno pomoću izrađenih virtualnih enkodera te digitalnih davača pozicije. Potrebno je voditi računa da ne dođe do kolizije između kolica, što nam omogućava enkoder i digitalni davač pozicije. Izmjena lonaca između kolica vrši se pomoću dizala. Lonac se može podignuti ili spustiti na kolica pomoću dizala koje omogućava vertikalno kretanje lonca. U programskom alatu Step7 [1] izrađeno je upravljanje sustavom kolica.

U drugom poglavlju navedena su razvojna okruženja potrebna za realizaciju završnog rada. U trećem poglavlju obrađuju se teme vezane za kolica s loncem, kao što je upravljanje brzinom vrtnje motora, mehanički prekidači za praćenje pozicije, automatske sekvence, simulator pozicije. Četvrto poglavlje posvećeno je dizalu koje služi za podizanje i spuštanje lonaca kolica. Peto poglavlje sadrži vizualizaciju izrađenu u CoDeSys razvojnom okruženju.

1.1 Zadatak završnog rada

Potrebno je izraditi programski blok za automatsko slijeđenje dvaju kolica u talionici željeza. Potrebno je slijeđenje obojih kolica uz pomoć virtualnih enkodera te digitalnih davača pozicija. Završni rad uključuje razvoj programske podrške (engl. *software*) i simulaciju u alatima Step7 [3] i CoDeSys te vizualizaciju u alatu CoDeSys.

2. RAZVOJNA OKRUŽENJA KORIŠTENA U RADU

U ovom poglavlju opisana su razvojna o kruženja i funkcije koje su korištene za izradu sustava upravljana dvaju kolica.

2.1. Step 7

U svim granama proizvodnje potrebni su upravljački elementi koji će omogućiti rad strojeva, uređaja i procesa. U ovom radu upravljanje sustavom dvaju kolica izvodi se pomoću programibilnog logičkog kontrolera (engl. *PLC - Programmable Logic Controller*). Programibilna logička jedinica programirati će se u alatu Step7. Step7 služi za pisanje programske podrške za PLC-e porodice SIMATIC. Razvijen je od Njemačke tvrtke Siemens. Programski jezik koji je korišten naziva se Ladder [2]. Linija programskog koda u Ladder programskom jeziku sastoji se od niza grafičkih simbola koji su zapravo naredbe i predstavljaju različite logičke operacije i komponente poput brojača, logičke operacije I, ILI, NE. Za izradu ovog rada koristila se verzija Step7 5.5.

2.2. CoDeSys

CoDeSys je programski jezik otvorenog koda koji se koristi u industrijskom programiranju. Sadrži integrirani sustav za vizualizaciju te alat koji omogućava razvoj i ispravljanje pogrešaka unutar koda. Za potrebe završnog rada koristi se CFC (engl. *Continous Function Chart*) pomoću kojeg je razvijena programska podrška. Vizualizacija je izrađena za potrebe sučelja između čovjeka i stroja (engl. *HMI- Human Machine Interfaces*).

3. KOLICA S LONCEM

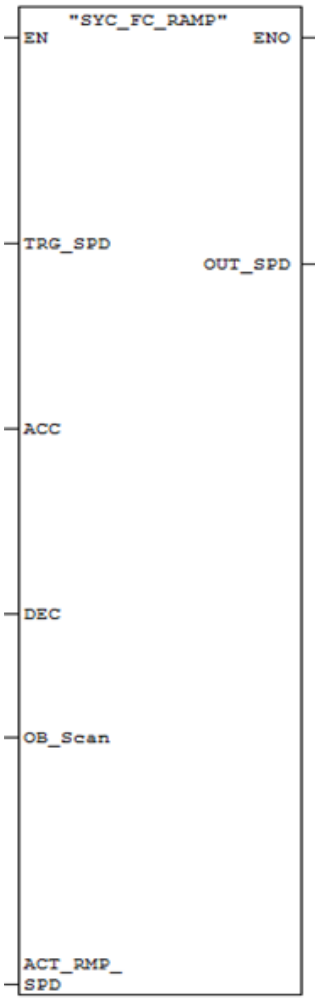
Svrha kolica je prijevoz vrućeg željeza koje se nalazi u loncu. Kolica se nalaze na tračnicama unutar talionice na predviđenom mjestu. Jedna kolica su smještena u području elektrolučne peći *(* engl. *Electric arc furnace*) poznatije kao EAF te se nazivaju kolica elektrolučne peći. Dok se druga nalaze u području *Ladle Refining Furnace* poznatije kao LRF i nazivaju se *Ladle Refining Furnace* kolica . Jedna kolica pokreće jedan asinkroni motor. Kolica mogu biti upravljana iz kontrolne sobe, kako u automatskim ciklusima tako i u ručnom pomicanju na željenu poziciju. Kolica s loncem smještena na tračnice prikazane su na slici 3.1.



Sl.3.1. Kolica s loncem smještena na tračnice

3.1. Upravljanje brzinom vrtnje motora kolica s loncima

Pomoću RAMP funkcije [4] se upravlja brzinom izmjeničnog elektromotora koji pokreće kolica. RAMP funkcija trenutnu vrijednost brzine linearnim povećanjem ili smanjenjem postavlja na željenu vrijednost brzine. Trenutna vrijednost brzine se povećava za iznos ubrzanja koje se uzima proizvoljno. Ukoliko je potrebno smanjiti trenutnu vrijednost brzine da bi se došlo do željene koristit će se usporenje. Brzinom se upravlja od 0 do 100% reference. Referenca predstavlja željenu brzinu koju šaljemo na motor. Kolica imaju pozitivan i negativan smjer kretanja. Ukoliko se EAF kolica kreću u smjeru od LRF-a do EAF-a imat će negativan smjer kretanja. U obrnutom slučaju imat će pozitivan smjer kretanja. Kolica LRF-a kada se kreću od parking pozicije do LRF-a imaju negativan smjer kretanja, u obrnutom slučaju imaju pozitivan smjer kretanja. Zbog pozitivnog i negativnog smjera kretanja postoji pozitivna i negativna referenca koju šaljemo na motor. RAMP funkcija prikazana je na slici 3.2.



Sl.3.2. Programsko rješenje RAMP funkcije

Prilikom izrade programa unutar Step7 alata koriste se različiti programski i podatkovni blokovi. Za izradu programske podrške korišten je organizacijski blok OB1, funkcije FC i podatkovni blokovi DB. OB1 tvori sučelje između operacijskog sustava i korisničkog programa. Operacijski sustav ciklički poziva OB1. Programski kod u OB1 sadrži samo pozive drugih blokova koji sadrže preostali kod. OB1 se ciklički izvršava. FC sadrži funkcionalnu cjelinu programa, te se poziva unutar OB1 zbog cikličkog izvršavanja. Unutar DB-a definiraju se varijable za pohranu vrijednosti korisničkog programa.

Na slici 3.2. prikazan je blok RAMP funkcije koja je pozvana unutar FC-ova za pozicioniranje EAF i LRF kolica. U poglavlju 3.1. prikazani su pojedini dijelovi RAMP funkcije koji su napravljeni unutar FC-a RAMP funkcije. RAMP funkcija ima pridružive parametre, tj., ulaze. Na ulazu u RAMP-u su svi parametri koji su potrebni da bi se izvršavao kod unutar FC-a u kojem se nalazi funkcionalna cjelina RAMP-e. Ulazne varijable definirane su unutar DB-a, te ih uz pomoć njihove memorijske adrese postavljamo na ulaz unutar RAMP-e. Na slici 3.2. vidljivi su ulazi koji služe da bi se pomoću njih upravljalo brzinom vrtnje motora EAF i LRF kolica. Za EAF i LRF kolica definirane su zasebne varijable.

Ulazi RAMP funkcije:

- ACC- predstavlja zadano ubrzanje, vrijednost je definirana unutar DB-a, također operater može mijenjati vrijednost
- DEC- zadano usporenje, vrijednost je definirana unutar DB-a, operater može mijenjati vrijednost prema potrebi
- OB_Scan - vrijeme cikličkog izvršavanja programibilnog kontrolera, dobiva se unutar zasebnog FC-a gdje je definirano kao izlaz iz funkcije, a u RAMP funkciji je ulaz
- ACT_RMP_SPD - trenutna vrijednost brzine, izračunava se unutar RAMP funkcije, definirana je kao ulazno-izlazna varijabla

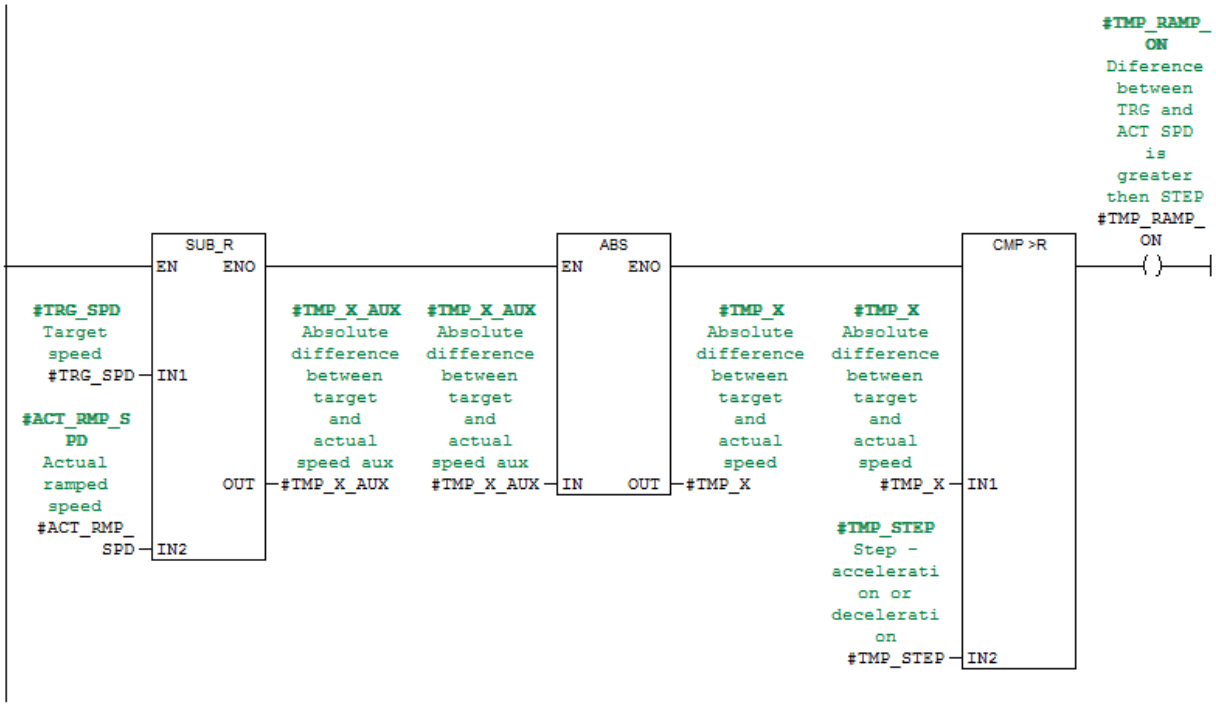
Izlaz RAMP funkcije:

- OUT_SPD - brzina koja se dobiva unutar RAMP funkcije njenim izvršavanjem

RAMP funkcija će se izvršavati za kretanje kolica u pozitivnom smjeru ako su zadovoljeni uvjeti koji su opisani u daljnjem tekstu

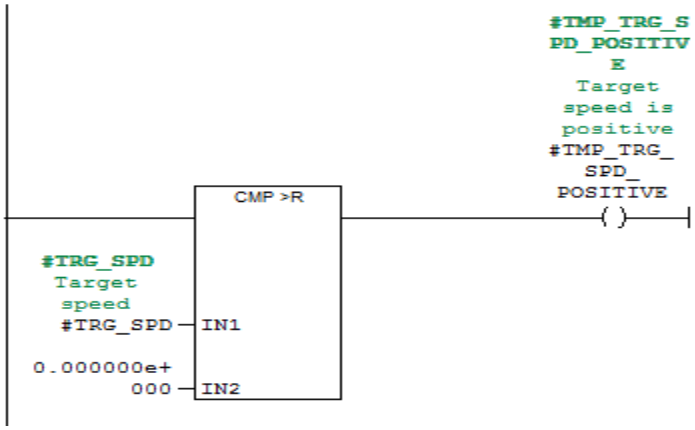
Varijable koje u imenu imaju oznaku "TMP" su privremene i dostupne samo unutar FC u kojem su definirane.

"TMP_RAMP_ON" je uvjet koji mora biti zadovoljen da bi se program izvodio. Kada je uvjet zadovoljen RAMP funkcija će biti aktivna. Označava da je razlika između željene brzine i trenutne brzine motora veća od vrijednosti "TMP_STEP" u koju se sprema apsolutna vrijednost iznosa ubrzanja ili usporenja. Ukoliko nije veća, promjena brzine se ne može ostvariti.



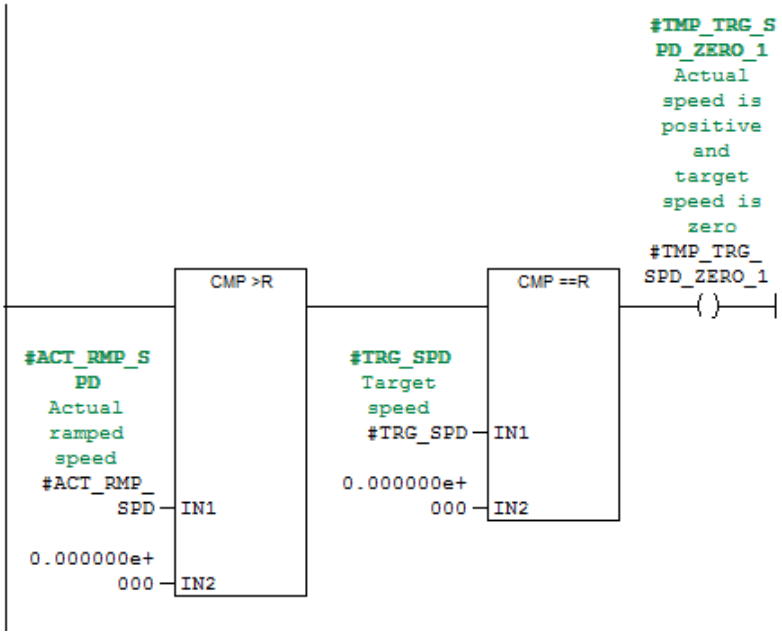
Sl.3.3. Programsko rješenje za izračunavanje uvjeta TMP_RAMP_ON

Na slici 3.3. je prikazan način pomoću kojeg se određuje hoće li uvjet "TMP_RAMP_ON" biti aktivan. Prvo se pomoću funkcije SUB kojoj su ulazi "TRG_SPD" na ulazu IN1 i "ACT_RMP_SPD" na ulazu IN2 izračunava razlika "TMP_X_AUX". "TRG_SPD" je ulazna varijabla RAMP funkcije i predstavlja željenu vrijednost brzine, zadaje ju korisnik, definira se unutar DB-a. Stvarna vrijednost brzine "ACT_RMP_SPD" je ulazno-izlazna varijabla čija se vrijednost izračunava unutar RAMP-e i ta vrijednost se dodjeljuje varijabli u DB-u. U ABS funkciji ulaz je razlika "TMP_X_AUX" a na izlazu iz funkcije je apsolutna vrijednost razlike "TMP_X". Funkcija CMP ima dva ulaza, apsolutna vrijednost razlike "TMP_X" na ulazu IN1 koja mora biti veća od iznosa spremljenog unutar drugog ulaza IN2,"TMP_STEP". Kada uvjet bude zadovoljen "TMP_RAMP_ON" će biti postavljen u logičku jedinicu.



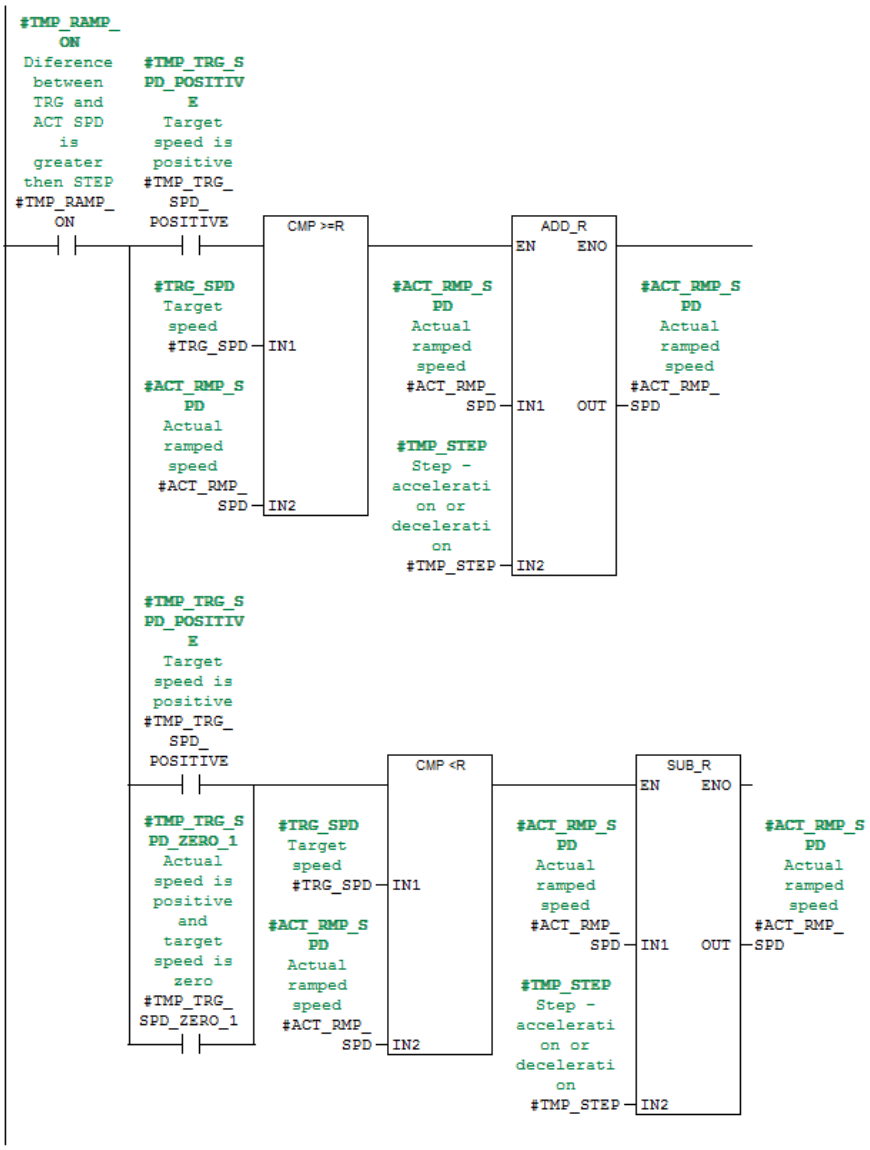
Sl.3.4.Provjera je li željena brzina pozitivna

"TMP_TRG_SPD_POSITIVE" je uvjet koji govori da se kolica gibaju u pozitivnom smjeru te da imaju pozitivnu referencu koju šaljemo na motor. Na slici 3.4. prikazano je kako se unutar programa provjerava je li željena brzina pozitivna. Pomoću funkcije CMP koja na ulazu IN1 ima željenu vrijednost brzine "TRG_SPD" usporedimo s nulom koja se nalazi na drugom ulazu IN2. Ukoliko je željena vrijednost brzine veća od nule uvjet je zadovoljen.



Sl.3.5. Provjera je li željena brzina jednaka nuli

Kada se kolica žele zaustaviti referenca koju šaljemo na motor bit će jednaka nuli. Potrebno je pomoću funkcije CMP provjeriti je li trenutna vrijednost brzine "ACT_RMP_SPD" na ulazu IN1 veća od vrijednosti nule na ulazu IN2. Nakon toga provjerava se je li željena vrijednost brzine "TRG_SPD" na ulazu IN1 u funkciju CMP jednaka nuli na ulazu IN2. Kada su oba uvjeta zadovoljena bit će zadovoljen uvjet da je stvarna brzina pozitivna i željena vrijednost brzine nula "TMP_TRG_SPD_ZERO_1". Na slici 3.5. je prikazano na koji način se unutar programa provjerava je li uvjet "TMP_TRG_SPD_ZERO_1" zadovoljen.



Sl.3.6. Programsko rješenje za upravljanje brzinom u pozitivnom smjeru kretanja kolica

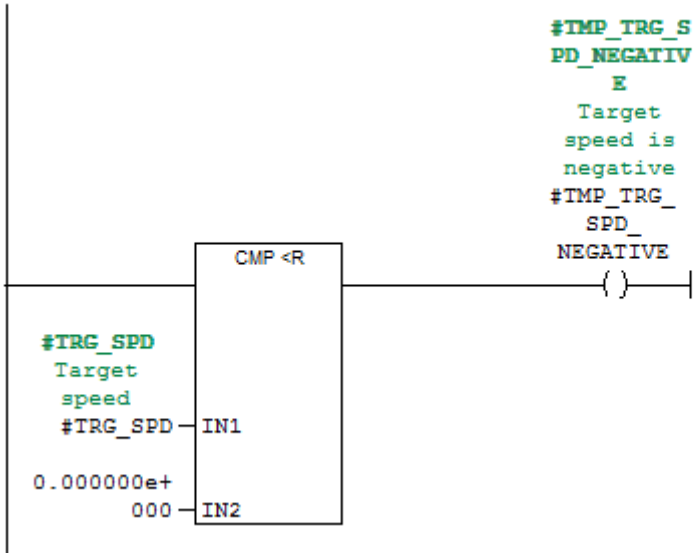
Nakon što je zadovoljen glavni uvjet "TMP_RAMP_ON" u programu se provjerava koji je uvjet od prethodno opisanih još zadovoljen te ovisno o tome se izvršava RAMP funkcija. Imamo tri moguća slučaja izvršavanja RAMP funkcije.

Prvi slučaj je ako je uz "TMP_RAMP_ON" zadovoljen i uvjet "TMP_TRG_SPD_POSITIVE" doći će do usporedbe pomoću funkcije CMP željene vrijednosti brzine "TRG_SPD" na ulazu IN1 sa trenutnom brzinom "ACT_RMP_SPD" na ulazu IN2. Ako je željena vrijednost brzine veća ili jednaka od trenutne, trenutnoj vrijednosti "ACT_RMP_SPD" na ulazu IN1 se pomoću funkcije ADD pridodaje vrijednost spremljena u "TMP_STEP" na ulazu IN2 koja označava iznos ubrzanja. Taj dio koda omogućava povećanje brzine motora. I nakon pridodavanja vrijednosti spremljene u "TMP_STEP" željena vrijednost brzine nam postaje stvarna vrijednost brzine nakon određenog broja zbrajanja.

Drugi slučaj je ukoliko se prilikom uspoređivanja "TRG_SPD" i "ACT_RMP_SPD" pomoću funkcije CMP utvrdi da je "TARGET_SPEED" na ulazu IN1 manji od "ACT_RMP_SPD" na ulazu IN2, doći će do smanjenja brzine motora. Smanjenje brzine motora je ostvareno pomoću funkcije SUB. Pomoću funkcije SUB od stvarne vrijednosti brzine "ACT_RMP_SPD" na ulazu IN1 oduzimamo iznos usporenja spremljen u "TMP_STEP" na ulazu IN2, te na taj način usporavamo motor.

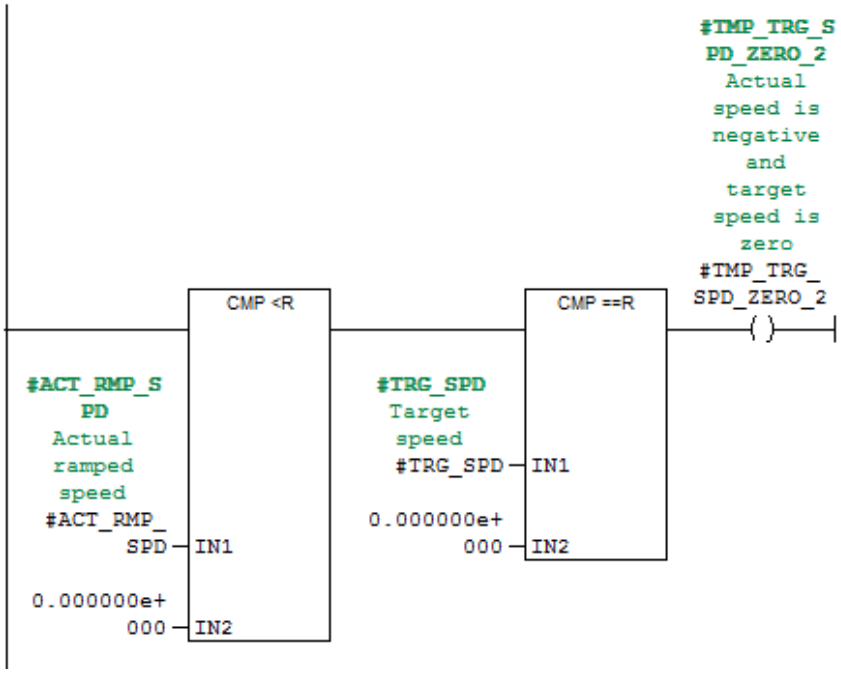
U trećem slučaju ako je uz "TMP_RAMP_ON" zadovoljen i uvjet "TMP_TRG_SPD_ZERO_1" i prilikom usporedbe željena vrijednost brzine "TRG_SPD" na ulazu IN1 funkcije CMP bude manja od trenutne "ACT_RMP_SPD" na ulazu IN2 motor će prestati s radom nakon što se stvarna vrijednost brzine u određenom broju koraka smanji do nule.

RAMP funkcija će se izvršavati za kretanje kolica u negativnom smjeru ako su zadovoljeni uvjeti koji su opisani u daljnjem tekstu. Važno je za naglasiti da u ovome slučaju šaljemo negativan iznos reference na motor.



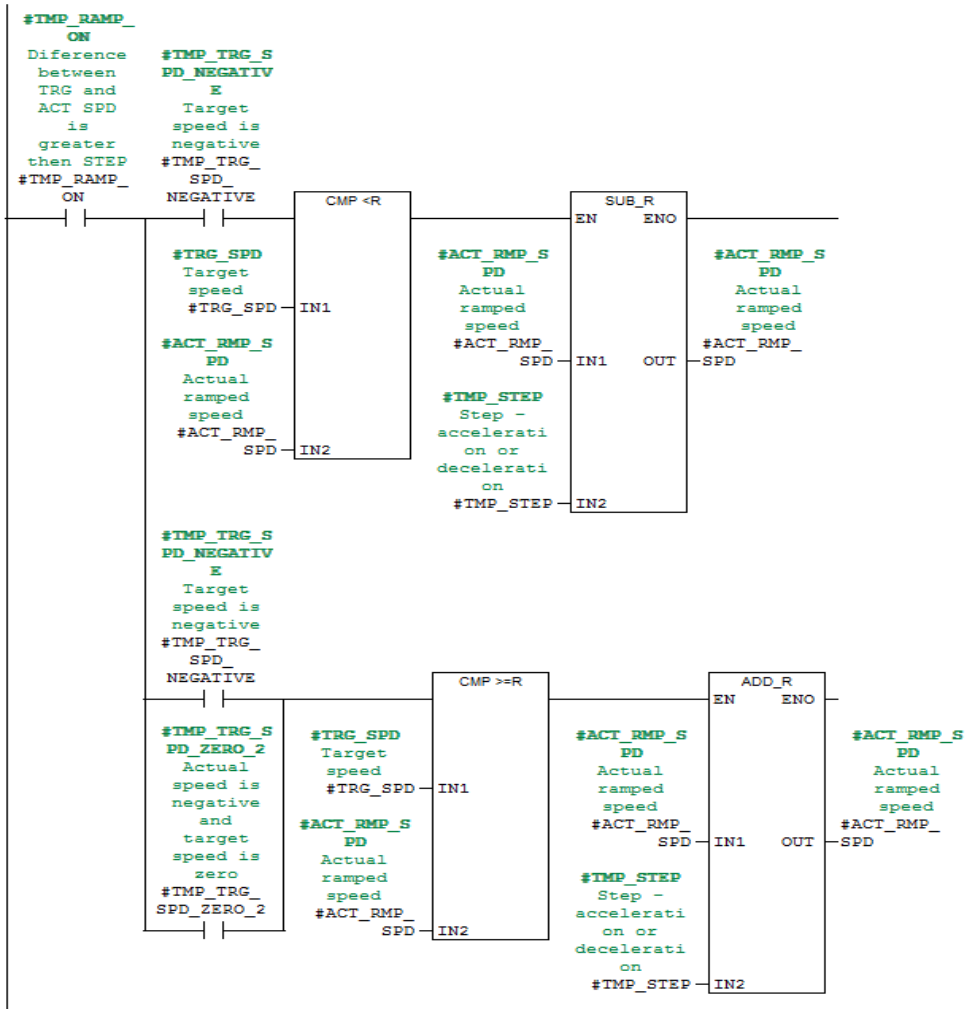
Sl.3.7.Provjera je li željena brzina negativna

"TMP_TRG_SPD_NEGATIVE" je uvjet koji govori da se kolica gibaju u negativnom smjeru te da imaju negativnu referencu koju šaljemo na motor. Na slici 3.7. prikazano je kako se unutar programa provjerava je li željena brzina negativna. Pomoću funkcije CMP koja na ulazu IN1 ima željenu vrijednost brzine "TRG_SPD" usporedi se s nulom na ulazu IN2. Ukoliko je željena vrijednost manja od nule uvjet je zadovoljen.



Sl.3.8. Provjera je li željena brzina jednaka nuli

Kada se kolica žele zaustaviti referenca koju šaljemo na motor bit će jednaka nuli. Potrebno je pomoću funkcije CMP provjeriti je li trenutna vrijednost brzine "ACT_RMP_SPD" na ulazu IN1 manja od nule na ulazu IN2. Nakon toga provjerava se je li željena vrijednost brzine "TRG_SPD" na ulazu IN1 jednaka nuli na ulazu IN2. Kada su oba uvjeta zadovoljena bit će zadovoljen uvjet da je stvarna brzina negativna i željena vrijednost brzine jednaka nula "TMP_TRG_SPD_ZERO_2". Na slici 3.8. je prikazano na koji način se unutar programa provjerava je li uvjet "TMP_TRG_SPD_ZERO_2" zadovoljen.



Sl.3.9. Programsko rješenje za upravljanje brzinom u negativnom smjeru kretanja kolica

Kod negativnog upravljanja brzinom također treba biti zadovoljen glavni uvjet "TMP_RAMP_ON". Postoje tri moguća slučaja izvršavanja RAMP funkcije kod upravljana brzinom s negativnom referencom.

Prvi slučaj je ako je uz "TMP_RAMP_ON" zadovoljen i uvjet "TMP_TRG_SPD_NEGATIVE" doći će do usporedbe pomoću funkcije CMP željene vrijednosti brzine "TRG_SPD" na ulazu IN1 sa trenutnom brzinom "ACT_RMP_SPD" na ulazu IN1. Ako je željena vrijednost brzine manja od trenutne, trenutnoj vrijednosti "ACT_RMP_SPD" na ulazu IN1 se pomoću funkcije SUB smanjuje iznos za iznos vrijednosti spremljen u "TMP_STEP" na ulazu IN2 koja označava iznos ubrzanja. Taj dio koda omogućava povećanje brzine motora. I nakon oduzimanja vrijednosti spremljene u "TMP_STEP" željena vrijednost brzine nam postaje stvarna vrijednost brzine nakon određenog broja oduzimanja.

Drugi slučaj je ukoliko se prilikom uspoređivanja "TRG_SPD" i "ACT_RMP_SPD" pomoću funkcije CMP utvrdi da je "TRG_SPD" na ulazu IN1 veći ili jednak od "ACT_RMP_SPD" na ulazu IN2 doći će do smanjenja brzine motora. Smanjenje brzine motora je ostvareno pomoću funkcije ADD. Pomoću funkcije ADD se stvarnoj vrijednosti brzine "ACT_RMP_SPD" na ulazu IN1 dodaje iznos usporenja spremljen u "TMP_STEP" na ulazu IN2 te na taj način usporavamo motor.

Treći slučaj uz "TMP_RAMP_ON" ima zadovoljen i uvjet "TMP_TRG_SPD_ZERO_2". Ako je željena vrijednost brzine na ulazu IN1 veća ili jednaka od trenutne na ulazu IN2 motor će prestati s radom nakon što se stvarna vrijednost brzine u određenom broju koraka smanji do nule.

3.2. Mehanički prekidači za praćenje pozicije kolica sa loncima

Mehanički prekidači služe da bi smo znali na kojoj se poziciji nalaze kolica. Nakon što kolica dodirnu prekidač, na PLC se šalje digitalni signal te pomoću njega znamo točnu poziciju kolica. Mehanički prekidači nalaze se kod EAF pozicije, pozicije zagrijavanja, LRF pozicije i pozicije parkiranja. Kretanje EAF kolica može ići od EAF pozicije do LRF pozicije i u suprotnom smjeru. LRF kolica mogu se kretati od pozicije za parkiranje do LRF pozicije i obratno. Pozicije mehaničkih prekidača prikazane su na slici 3.10. Mehanički prekidači su simulirani pomoću simulatora pozicije, svaki od njih se nalazi na određenoj udaljenosti. Prijeđena udaljenost kolica se uspoređuje s vrijednosti na koju je postavljen prekidač, kada prijeđena udaljenost bude jednaka udaljenosti na kojoj se nalazi prekidač simulira se da su kolica dodirnula neki od prekidača.

Mehanički prekidači EAF kolica :

- Prvi se nalazi na glavnoj EAF poziciji. Korišten je u svrhu zaustavljanja kolica na toj poziciji. Pored njega je smješten prekidač za usporavanje kolica kada se gibaju prema EAF poziciji.
- Na poziciji zagrijavanja mehanički prekidač koristi se u svrhu zaustavljanja kolica na toj poziciji.
- S obje strane pozicije zagrijavanja nalazi se po jedan prekidač. Ako kolica idu od EAF pozicije prema poziciji zagrijavanja, prekidač će samo u tome slučaju usporiti kretanje kolica kada je potrebno stati na poziciji zagrijavanja. Drugi

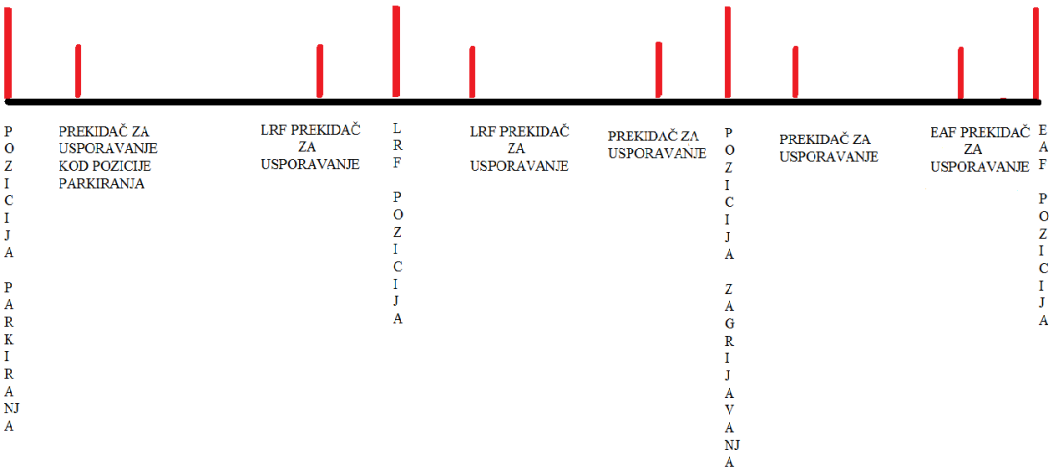
prekidač će usporiti kolica samo ako se kreću iz smjera LRF-a prema EAF poziciji i potrebno je stati na poziciji zagrijavanja.

Na LRF poziciju mogu doći EAF i LRF kolica. S obje strane LRF pozicije nalazi se po jedan mehanički prekidač. Jedan je za EAF kolica, a drugi za LRF kolica. Prekidač koji je korišten samo od stane EAF kolica koristi se za usporavanje kolica kada idu od EAF ili pozicije zagrijavanja prema LRF poziciji. Mehanički prekidač koji koriste LRF kolica služi za usporavanje kolica kada se kreću od pozicije parkiranja prema LRF poziciji. Kod LRF pozicije ne mogu se koristiti isti prekidači za oboja kolica jer onda nije moguće znati koja su kolica trenutno na LRF poziciji. Svaka kolica na LRF poziciji imaju svoj prekidač. Na LRF poziciji nalazi se i dizalo.

LRF kolica mogu se kretati od parking pozicije do LRF pozicije i u tome kretanju naići će na ukupno četiri mehanička prekidača.

Mehanički prekidači LRF kolica :

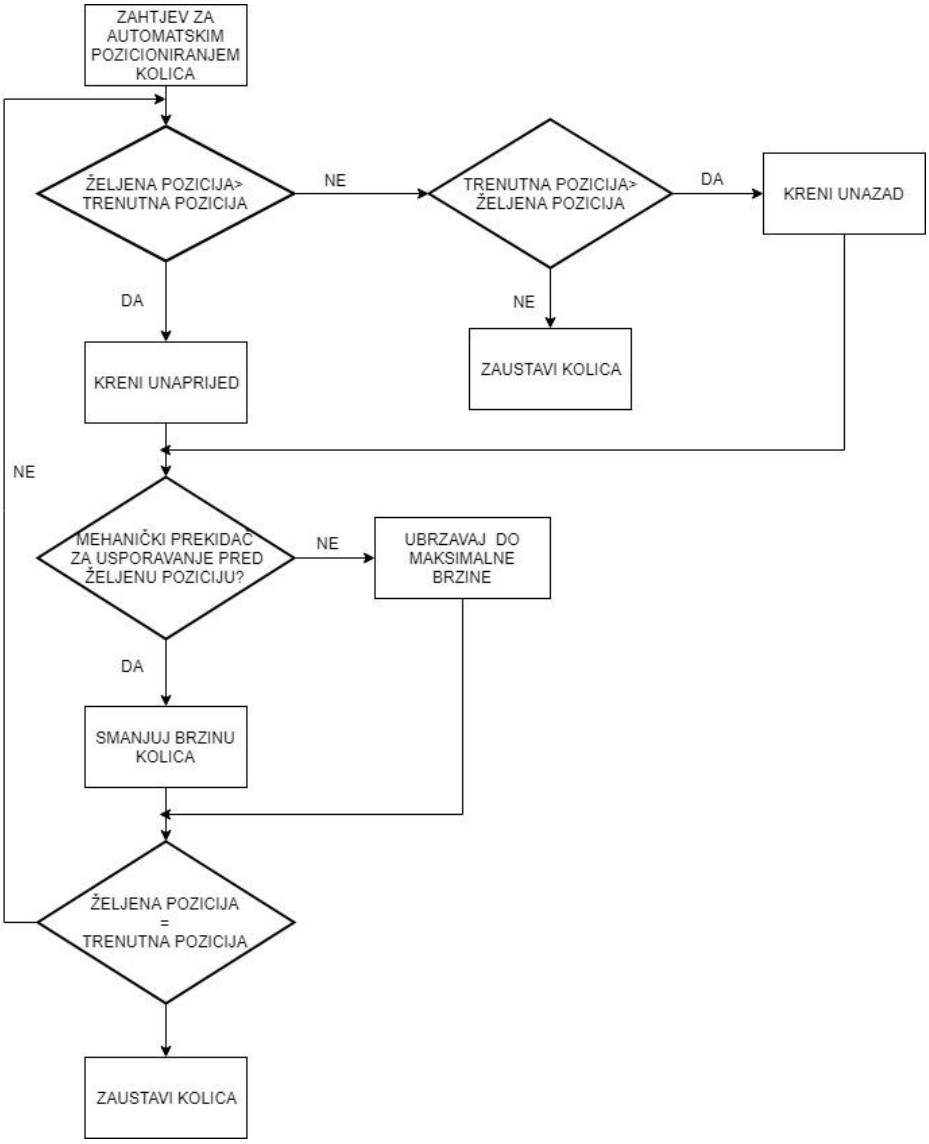
- Dva prekidača nalaze se na poziciji za parkiranje. Jedan služi za usporavanje kolica prije nego što dođu na poziciju parkiranja a drugi se nalazi na poziciji za parkiranje.
- Druga dva su objašnjena u predhodnom tekstu u primjeru kada se kolica gibaju prema LRF poziciji.



Sl.3.10. Prikaz pozicija mehaničkih prekidača

3.3. Automatske sekvence kolica s loncima

Automatske sekvence omogućavaju pozicioniranje kolica na zahtijevanu poziciju jednog od mehaničkih prekidača (EAF pozicija, pozicija zagrijavanja, LRF pozicija i parking pozicija). Pozicioniranje se izvodi iz kontrolne sobe u industrijskom postrojenju. Kolica će biti automatski poslana na željenu poziciju. Ukoliko dođe do pogreške, kolica se moraju pomoću ručnih kontrola postaviti na najbliži mehanički prekidač, kako bi ponovno uspostavili virtualno praćenje pomoću enkodera. Upravljane pozicijom pomoću automatskih sekvenci prikazano je dijagramom toka na slici 3.6.

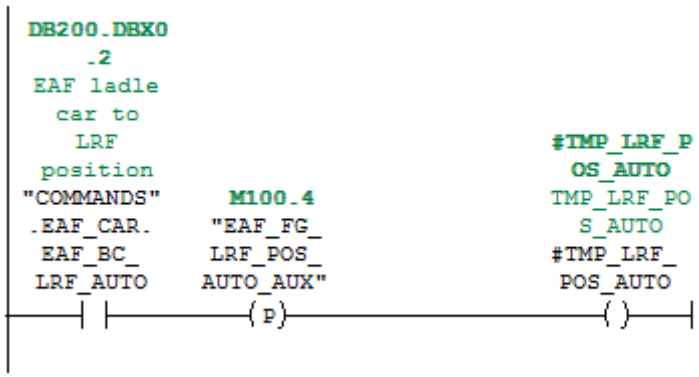


Sl.3.11. Dijagram toka upravljanja pozicijom pomoću automatskih sekvenci

Nakon što se dobije zahtjev za pozicioniranjem moguća su tri slučaja. U prvom slučaju potrebno je odrediti je li vrijednost željene pozicije veća od trenutne pozicije. Ukoliko je tvrdnja istinita pokrećemo kolica prema naprijed. Ako kolica naiđu na mehanički prekidač za usporavanje pred željenu poziciju, potrebno je usporiti brzinu kretanja kolica. Dok kolica ne naiđu na mehanički prekidač za usporavanje, kolica se ubrzavaju do maksimalne brzine koja je zadana za automatsko upravljanje kolicima. Zadnji je uvjet da je željena pozicija zapravo postala jednaka trenutnoj kada dolazi do zaustavljanja kolica. Ukoliko uvjet nije ispunjen potrebno je krenuti s ponovnim pozicioniranjem. Za drugi slučaj potrebno je da trenutna pozicija bude veća od željene, no ukoliko uvjet nije zadovoljeno doći će do zaustavljanja kolica. Za treći slučaj trenutna pozicija mora biti veća od željene pozicije. Kolica se pokreću unazad, a ostatak izvršavanja jednak je kao i kod prvog slučaja.

Automatsko pozicioniranje EAF i LRF kolica napravljeno je unutar zasebnih FC-ova koji se pozivaju u OB1. Varijable koje započinju automatsko pozicioniranje nakon što ih operater postavi u logičku jedinicu spremljene su unutar DB-ova.

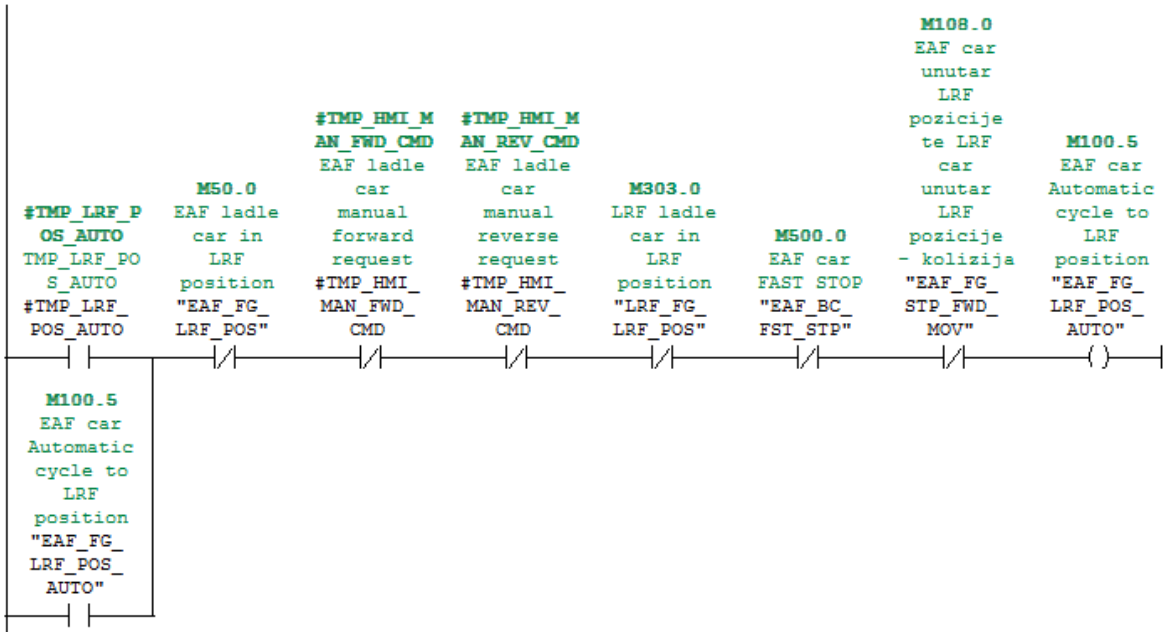
U daljnjem tekstu opisat će se automatsko pozicioniranje EAF kolica od EAF pozicije do LRF pozicije.



Sl.3.12. Prikaz zahtjeva za automatskim pozicioniranjem

Automatsko pozicioniranje EAF kolica u LRF poziciju započet će kada vrijednost varijable "EAF_BC_LRF_AUTO" bude u logičkoj jedinici. Varijablu "EAF_BC_LRF_AUTO" operater postavlja u logičku jedinicu. Kada se varijabla postavi u logičku jedinicu, varijabla "TMP_LRF_POS_AUTO" će biti aktivna samo za vrijeme jednog cikličkog izvršavanja programibilnog logičkog kontrolera. Varijabla će biti aktivna samo za vrijeme jednog cikličkog izvršavanja zbog funkcije $\cdots(p)\cdots$ koja će propustiti signal s rastućim bridom samo jednom (

eng. *one shot*). Varijabla "TMP_LRF_POS_AUTO" je dostupna samo unutar FC-a u kojem je definirana u svrhu što jednostavnijeg pisanja programa.



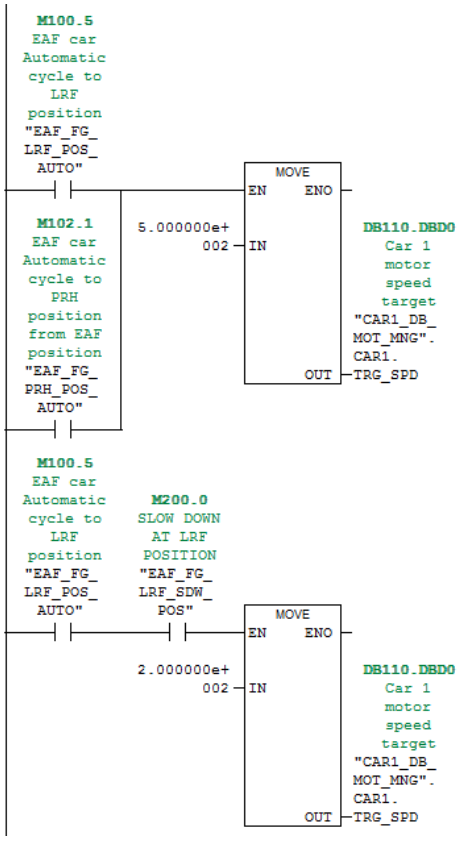
Sl.3.13. Održavanje automatskog ciklusa

Slika 3.13. prikazuje održavanje automatskog ciklusa. Varijabla "TMP_LRF_POS_AUTO" će propustiti signal samo u jednom ciklusu u marker "EAF_FG_LRF_POS_AUTO" koji se nalazi na adresi M100.5. Bit koji se nalazi na adresi M100.5 će biti u logičkoj jedinici u narednim ciklusima PLC-a i zbog toga je važno napraviti održavanje. Održavanje omogućuje automatsko pozicioniranje sve dok se pozicioniranje ne izvrši ili dok neki od uvjeta ne prekine automatsko pozicioniranje.

Uvjeti koji mogu prekinuti automatski ciklus su:

- "EAF_FG_LRF_POS" - EAF kolica su došla u LRF poziciju, pozicioniranje je uspješno obavljeno, željena pozicija je jednaka trenutnoj, ovaj signal dolazi od EAF simulatora pozicije.
- "TMP_HMI_MAN_FWD_CMD" - Ukoliko dođe do zahtjeva za ručnim pozicioniranjem kolica u pozitivnom smjeru od strane operatera.
- "TMP_HMI_MAN_REV_CMD" - Ukoliko dođe do zahtjeva za ručnim pozicioniranjem kolica u negativnom smjeru od strane operatera.

- "LRF_FG_LRF_POS" - Ako se LRF kolica nalaze unutar LRF pozicije, tada nije ni moguće započeti automatski ciklus. Ovaj signal dolazi od LRF simulatora pozicije.
- "EAF_BC_FST_STP" - Zahtjev za žurnim zaustavljanjem kolica od strane operatera.
- "LIF_LOW_POS" - Dizalo u donjoj poziciji. Signal dolazi od simulatora pozicije kada je dizalo u donjoj poziciji.
- "EAF_FG_STP_FWD_MOV" - EAF kolica unutar LRF pozicije i LRF kolica unutar LRF pozicije. Signal dolazi iz FC-a upravljanja pozicijom EAF kolica.



Sl.3.14. Raspodjela brzine u automatskom ciklusu

Kada dođe do zahtjeva za pozicioniranjem "EAF_FG_LRF_POS_AUTO", EAF kolica će ubrzavati do 50% reference. Pomoću funkcije MOVE vrijednost s ulaza šaljemo u varijablu željene brzine "TRG_SPD" koja poprima vrijednost od 50% reference. Prilikom gibanja kolica prema LRF poziciji, trenutna pozicija kolica se konstantno uspoređuje s pozicijom LRF mehaničkog prekidača za usporavanje i udaljenosti na koju je postavljena LRF pozicija. Kada je, uz "EAF_FG_LRF_POS_AUTO", zadovoljen i uvjet da se kolica nalaze kod "EAF_FG_LRF_SDW_POS" mehaničkog prekidača za usporavanje, kolica će početi usporavati

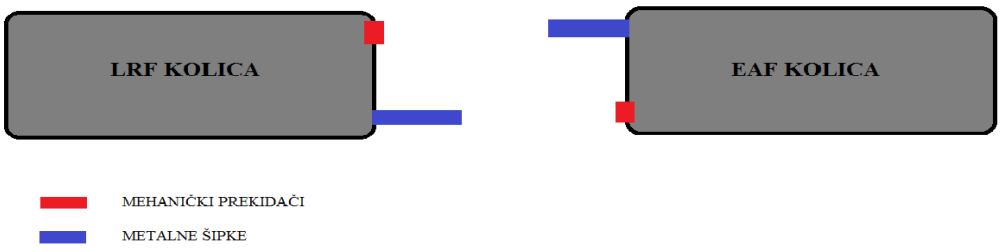
do 20% reference. S funkcijom MOVE vrijednost na ulazu šaljemo u varijablu željene brzine "TRG_SPD", zatim željena brzina "TRG_SPD" dobiva vrijednost u iznosu od 20% reference. Nakon što kolica dođu do LRF pozicije prekida se automatski ciklus i kolica se zaustavljaju. Raspodjela brzine u automatskom ciklusu EAF kolica od EAF pozicije do LRF pozicije prikazana je na slici 3.14. Varijabla "EAF_FG_PRH_POS_AUTO" nema nikakav utjecaj prilikom automatskog pozicioniranja EAF kolica u LRF poziciju. Varijabla "EAF_FG_PRH_POS_AUTO" se koristi prilikom automatskog pozicioniranja EAF kolica u poziciju zagrijavanja.

3.4. Izbjegavanje kolizije kolica

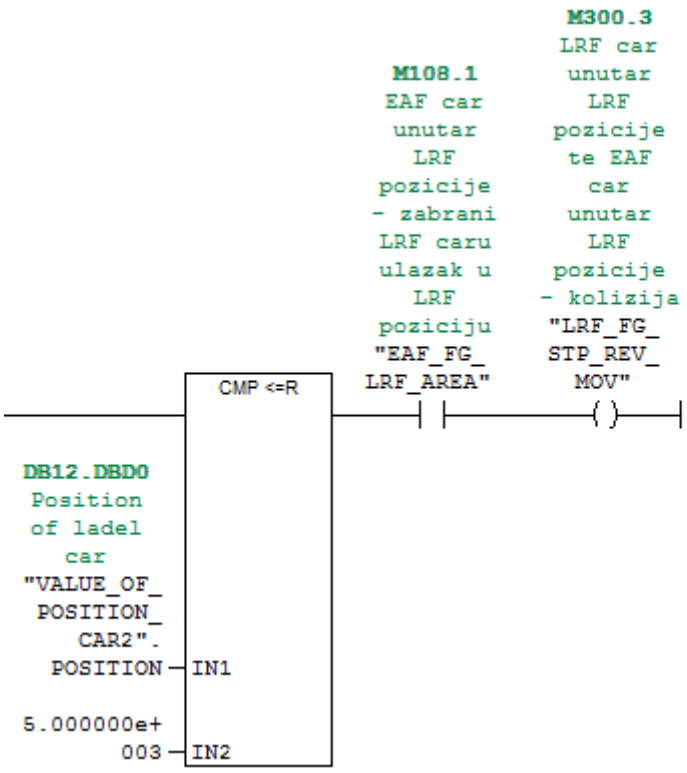
Prilikom pozicioniranja kolica potrebno je izbjeći njihovu koliziju. Koliziju je moguće izazvati ukoliko neki od mehaničkih prekidača ne radi ispravno ili zbog pogrešne kontrole prilikom upravljanja. Radi se o kompleksnim sustavima upravljanja kod kojih je potrebno izbjeći svaki mogući prekid rada te se postavljaju alarmi.

Na kolicima su postavljene metalne šipke koje su okrenute prema drugim kolicima. Primjena metalne šipke je ukoliko se LRF kolica krenu gibati prema EAF kolicima i uđu u njihovo područje, metalna šipka će dotaknuti mehanički prekidač za prepoznavanje kolizije koji se nalazi na EAF kolicima. Kolica će se zaustaviti. Isto vrijedi i za EAF kolica. Na slici 3.7. prikazana su kolica sa šipkama i mehaničkim prekidačima.

Unutar programske podrške napravljeno je izbjegavanje kolizije u slučaju automatskog i ručnog pozicioniranja unutar FC-ova za pozicioniranje EAF i LRF kolica. Pozicija kolica se dobiva pomoću simulatora pozicije.



Sl.3.15. Slikovni prikaz principa izbjegavanja kolizije

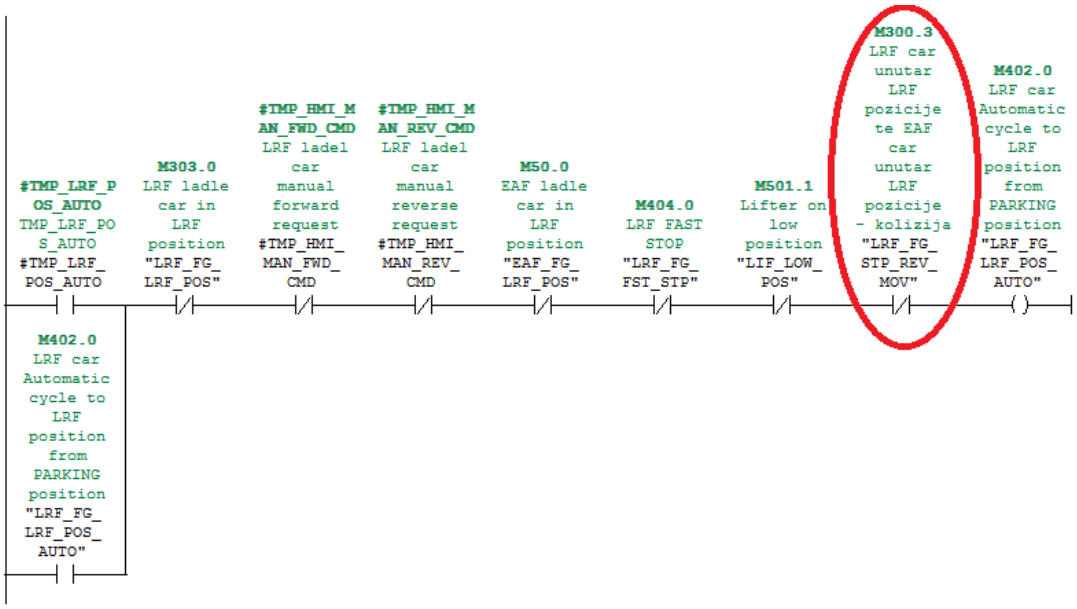


Sl.3.16. Programsko rješenje izbjegavanja kolizije u automatskom ciklusu LRF kolica

Uspoređivanjem pozicije LRF kolica "POSITION" na ulazu IN1 s 5000 mm na ulazu IN2 funkcije CMP u slučaju da je pokrenut automatski ciklus prikazano je na slici 3.16. Ako je pozicija LRF kolica manja ili jednaka od 5000 mm te ako su EAF kolica unutar LRF područja "EAF_FG_LRF_AREA" zaustavit će se gibanje LRF kolica. Pomoću "EAF_FG_LRF_AREA" znamo da se EAF kolica nalaze unutar LRF područja ako je zadovoljeno da je pozicija EAF kolica veća ili jednaka od 15 000 mm marker "EAF_FG_LRF_AREA" bit će postavljen u logičku jedinicu. Ako je marker "EAF_FG_LRF_AREA" u logičkoj jedinici također će onda biti i marker "LRF_FG_STP_REV_MOV" pomoću kojega će se prekinuti automatski ciklus. Pozicija EAF kolica dobiva se pomoću simulatora pozicije EAF kolica pozvanog unutar FC-a EAF kolica.

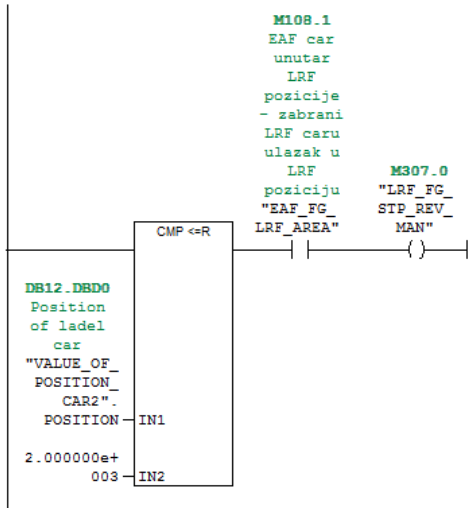
Signal "EAF_FG_LRF_AREA" dolazi iz FC-a EAF kolica.

Vrijednosti od 5000 mm i 15 000 mm predstavljaju vrijednosti udaljenosti EAF i LRF kolica na kojima ih je potrebno zaustaviti da ne dođe do kolizije uz prethodno navedene uvjete.



Sl.3.17. Prikaz markera "LRF_FG_STP_REV_MOV" unutar održavanja

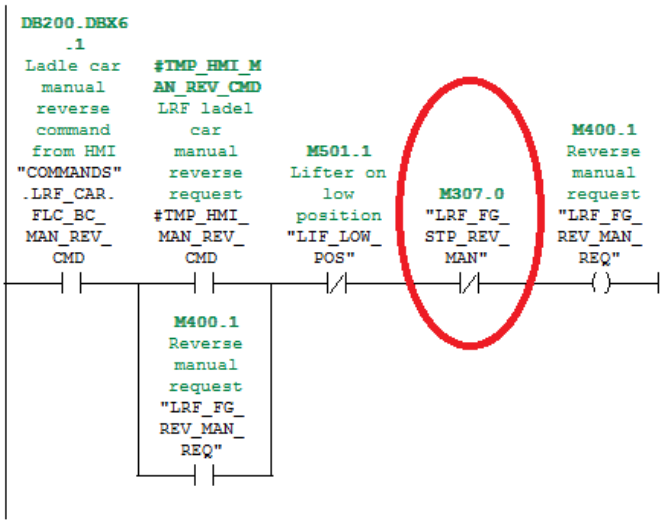
Na slici 3.17. prikazan je marker "LRF_FG_STP_REV_MOV" unutar održavanja automatskog ciklusa LRF kolica do LRF pozicije. Marker će prekinuti automatski ciklus kada bude u logičkoj jedinici.



Sl.3.18. Programsko rješenje izbjegavanja kolizije prilikom ručnog pozicioniranja LRF kolica

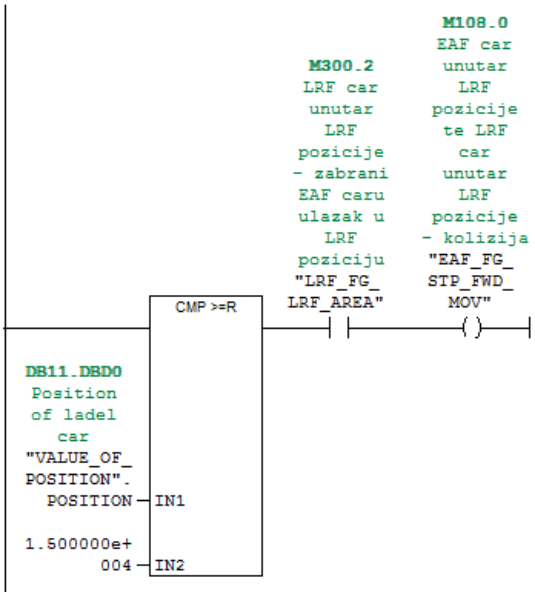
Prilikom ručnog pozicioniranja LRF kolica uspoređujemo udaljenost koju su prešla LRF kolica s 2000 mm. Ako je udaljenost manja ili jednaka od 2000 mm i ako su EAF kolica "EAF_FG_LRF_AREA" u LRF području onda će marker "LRF_FG_STP_REV_MAN", koji će poslužiti za zaustavljanje ručnog pozicioniranja, biti postavljen u logičku jedinicu. Kod ručnog

pozicioniranja dopušteno je LRF kolicima da dođu bliže LRF pozicije u odnosu na automatsko pozicioniranje.



Sl.3.19. Prikaz markera "LRF_FG_STP_REV_MAN"

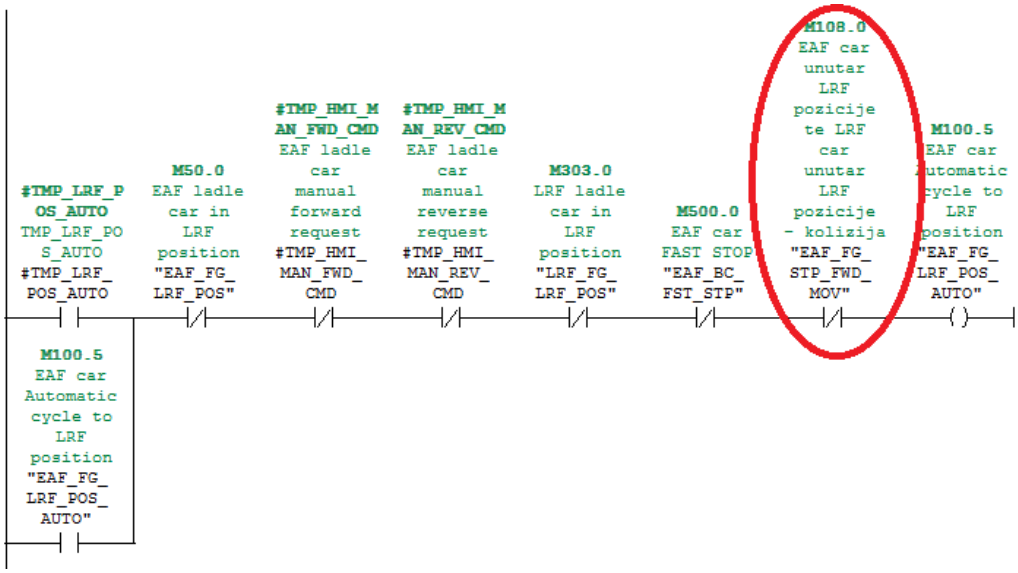
Na slici 3.19. prikazan je marker "LRF_FG_STP_REV_MAN" pomoću kojeg će se prekinuti ručno pozicioniranje LRF kolica u smjeru prema LRF poziciji.



Sl.3.20. Programsko rješenje izbjegavanja kolizije u automatskom ciklusu EAF kolica

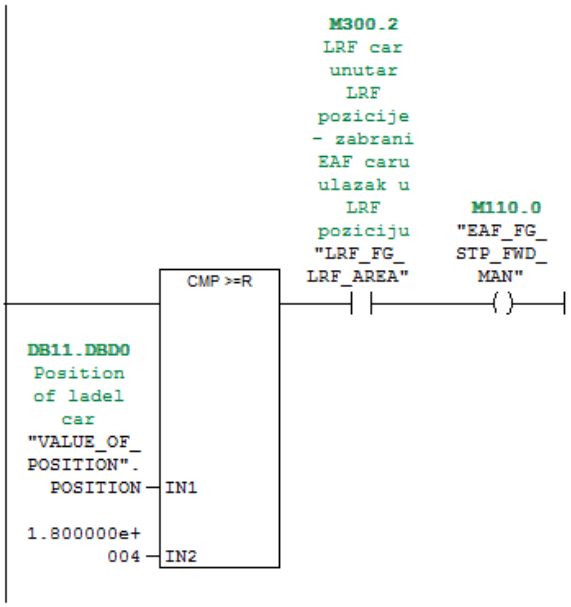
Uspoređivanjem pozicije EAF kolica "POSITION" na ulazu IN1 s 15 000 mm na ulazu IN2 funkcije CMP u slučaju da je pokrenut automatski ciklus prikazano je na slici 3.20. Ako je pozicija EAF kolica veća ili jednaka od 15 000 mm te ako su LRF kolica unutar LRF područja

"LRF_FG_LRF_AREA" zaustavit će se gibanje EAF kolica tj., bit će onemogućeno pozicioniranje EAF kolica u LRF poziciju. Pomoću "LRF_FG_LRF_AREA" znamo da se LRF kolica nalaze unutar LRF područja. Ako je zadovoljeno da je pozicija LRF kolica manja ili jednaka od 2000 mm marker "LRF_FG_LRF_AREA" bit će postavljen u logičku jedinicu. Ako je marker "LRF_FG_LRF_AREA" u logičkoj jedinici također će onda biti i marker "EAF_FG_STP_FWD_MOV" pomoću kojega će se prekinuti automatski ciklus. Pozicija LRF kolica dobiva se pomoću simulatora pozicije LRF kolica pozvanog unutar FC-a LRF kolica.



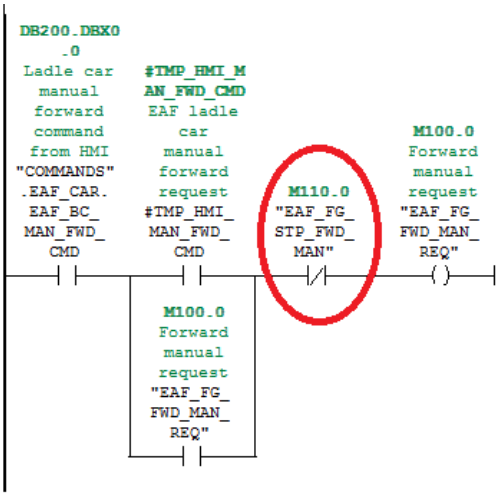
Sl.3.21. Prikaz markera "EAF_FG_STP_FWD_MOV" unutar održavanja

Na slici 3.21. prikazan je marker "EAF_FG_STP_FWD_MOV" unutar održavanja automatskog ciklusa EAF kolica do LRF pozicije. Marker će prekinuti automatski ciklus kada bude u logičkoj jedinici.



Sl.3.22. Programsko rješenje izbjegavanja kolizije prilikom ručnog pozicioniranja EAF kolica

Prilikom ručnog pozicioniranja EAF kolica uspoređujemo udaljenost koju su prešla EAF kolica s 18 000 mm. Ako je udaljenost veća ili jednaka od 18 000 mm i ako su LRF kolica "LRF_FG_LRF_AREA" u području LRF pozicije, "EAF_FG_STP_FWD_MAN" postavit će se u logičku jedinicu. Kod ručnog pozicioniranja dopušteno je EAF kolicima da dođu bliže LRF pozicije u odnosu na automatsko pozicioniranje.



Sl.3.23. Prikaz markera "EAF_FG_STP_FWD _MAN"

Na slici 3.23. prikazan je marker "EAF_FG_STP_FWD_MAN" pomoću kojeg će se prekinuti ručno pozicioniranje EAF kolica u smjeru prema LRF poziciji.

3.5. Simulator pozicije - praćenje pozicije pomoću virtualnog enkodera

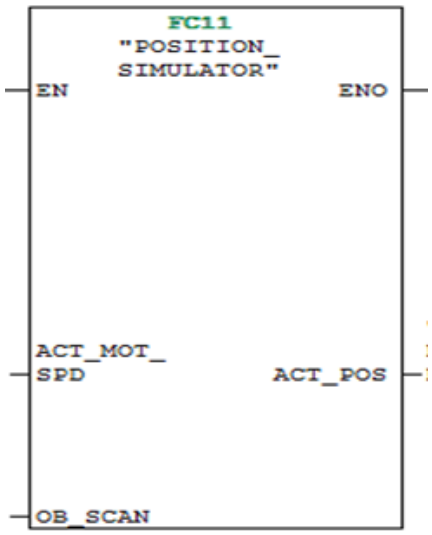
Simulator pozicije omogućava izračun prijeđene udaljenosti kolica. Koristi se apsolutni enkoder [5] koji je postavljen na osovinu motora, te uz pomoć njega mjerimo broj okretaja kotača kolica. Da bi se mogla izračunati prijeđena udaljenost potrebno je znati brzinu motora koju ćemo dobiti iz RAMP funkcije i vrijeme jednog ciklusa [6] cikličkog izvršavanja programibilnog logičkog kontrolera koje je izraženo u milisekundama.

$$v = s/OBscan \qquad (3\text{-}1)$$

$$s = v \times OBscan \qquad (3\text{-}2)$$

gdje je :

- s - prijeđeni put
- v - brzina motora
- $OBscan$ - vrijeme cikličkog izvršavanja programibilnog logičkog kontrolera



Sl.3.16. Programsko rješenje simulatora pozicije

Na slici 3.16. prikazan je simulator pozicije kojemu su na ulazu brzina motora "ACT_MOT_SPD" i vrijeme cikličkog izvršavanja programibilnog logičkog kontrolera OB_SCAN. Na izlazu se ispisuje pozicija koja se dobije množenje brzine motora i OBscan-a.

Simulator pozicije napravljen je unutar zasebnog FC-a koji se poziva unutar FC-ova za pozicioniranje EAF i LRF kolica.

Address	Symbol	Display format	Status value	Modify value
DB200.DBX 6.0	"COMMANDS".LRF_CAR.FLC_BC_MAN_FWD_CMD	BOOL	true	
DB200.DBX 6.1	"COMMANDS".LRF_CAR.FLC_BC_MAN_REV_CMD	BOOL	false	
DB115.DBD 12	"CAR2_DB_MOT_MNG".CAR2.ACT_SPD	FLOATING_POINT	300.0	
DB12.DBD 0	"VALUE_OF_POSITION_CAR2".POSITION	FLOATING_POINT	643.0	
DB115.DBD 0	"CAR2_DB_MOT_MNG".CAR2.TRG_SPD	FLOATING_POINT	300.0	

Sl.3.17. Prikaz prijeđene udaljenosti unutar simulacije

Na slici 3.17. prikazana je udaljenost koju su LRF kolica prešla tijekom pozicioniranja ručnim kontrolama. Udaljenost je izražena u milimetrima, a brzina motora je 30% reference.

4. DIZALO

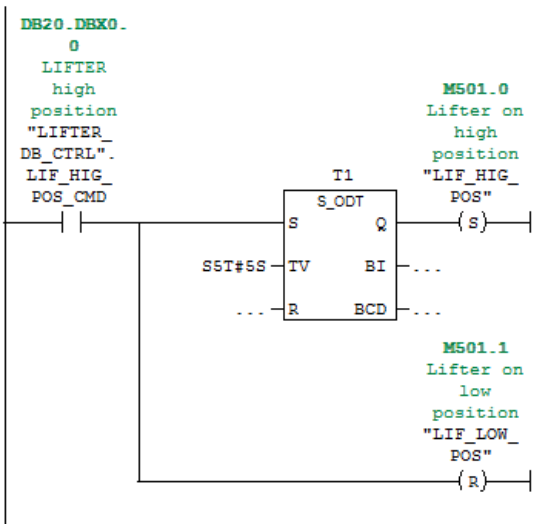
Dizalo omogućava podizanje i spuštanje lonaca na kolica tj. omogućava izmjenu lonaca između kolica. Dizalo je smješteno kod LRF pozicije.

4.1. Podizanje i spuštanje dizala, izmjenjivanje kolica ispod dizala

Podizanje i spuštanje dizala omogućeno je pomoću hidrauličke jedinice. Praćenje pozicije dizala je omogućeno pomoću mehaničkih prekidača. Mehanički prekidači će omogućiti prepoznavanje donje i gornje pozicije dizala. Poznavanje pozicije dizala je od ključne važnosti za podizanje ili spuštanje lonca.

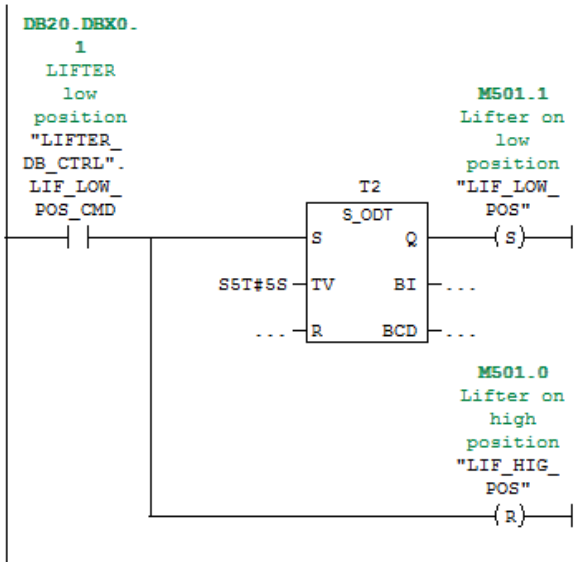
Kolica se kreću prema dizalu koje je spuštena tj. nalazi se u donjoj poziciji. Nakon što EAF kolica dođu na poziciju gdje je smješteno dizalo, dolazi do podizanja lonca. Nakon toga EAF kolica se izmiču ispod dizala dok je lonac podignut. LRF kolica dolaze na poziciju ispod dizala, dizalo spušta lonac na LRF kolica, te se istaljeni materijal prevozi na poziciju za parkiranje. Također je moguć i drugi scenariji kada prvo LRF kolica dovoze lonac te se lonac spušta na EAF kolica.

Programsko rješenje za upravljanje dizalom je napravljeno unutar zasebnog FC-a. Simulira se postavljanje dizala u gornju i donju poziciju, spuštanje lonca na EAF ili LRF kolica, te da se lonac nalazi na dizalu u gornjoj poziciji.



Sl.4.1. Programsko rješenje postavljanja dizala u gornju poziciju

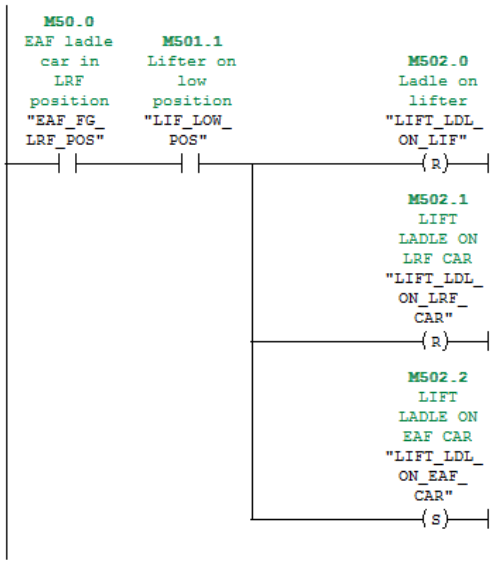
Dizalo će se postaviti u gornju poziciju, nakon što od strane operatera varijabla "LIF_HIG_POS_CMD" bude u logičkoj jedinici. Varijabla "LIF_HIG_POS_CMD" je definirana unutar DB-a i pomoću nje operater zahtjeva pozicioniranje dizala u gornju poziciju. Nakon što dođe do zahtjeva za pozicioniranjem u gornju poziciju nakon pet sekundi će se simulirati da se dizalo nalazi u gornjoj poziciji. Pomoću vremenskog brojača (*eng. Timer*) koji je nazvan T1 će se nakon 5 sekundi propustiti signal u marker "LIF_HIG_POS" koji će se pomoću funkcije SET (*s*) postaviti u logičku jedinicu. Marker "LIF_HIG_POS" kada je u logičkoj jedinici označava da se dizalo nalazi u gornjoj poziciji. Kada dođe do zahtjeva za pozicioniranjem u gornju poziciju pomoću funkcije RESET (*R*) marker "LIF_LOW_POS" postavit će se u logičku nulu ako je prethodno bio u logičkoj jedinici. Pomoću markera "LIF_LOW_POS" kada je u logičkoj jedinici znamo da se dizalo nalazi u donjoj poziciji. Na slici 4.1. prikazano je programsko rješenje za postavljanje dizala u gornju poziciju.



Sl.4.2. Programsko rješenje postavljanja dizala u donju poziciju

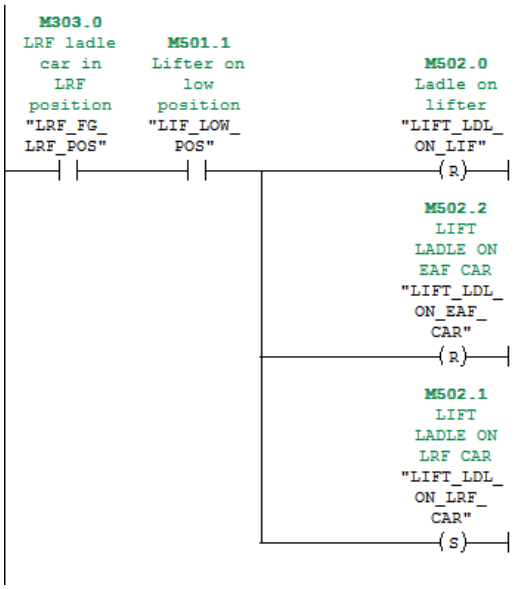
Dizalo će se postaviti u donju poziciju, nakon što od strane operatera varijabla "LIF_LOW_POS_CMD" bude u logičkoj jedinici. Varijabla "LIF_LOW_POS_CMD" je definirana unutar DB-a i pomoću nje operater zahtjeva pozicioniranje dizala u donju poziciju. Nakon što dođe do zahtjeva za pozicioniranjem u donju poziciju nakon pet sekundi će se simulirati da se dizalo nalazi u donjoj poziciji. Pomoću vremenskog brojača koji je nazvan T2 će se nakon 5 sekundi propustiti signal u marker "LIF_LOW_POS" koji će se pomoću funkcije SET (*s*) postaviti u logičku jedinicu. Marker "LIF_LOW_POS" kada je u logičkoj jedinici označava

da se dizalo nalazi u donjoj poziciji. Kada dođe do zahtjeva za pozicioniranjem u donju poziciju pomoću funkcije RESET (*R*) marker "LIF_HIG_POS" postavit će se u logičku nulu ako je prethodno bio u logičkoj jedinici. Pomoću markera "LIF_HIG_POS" kada je u logičkoj jedinici znamo da se dizalo nalazi u gornjoj poziciji. Na slici 4.2. prikazano je programsko rješenje za postavljanje dizala u donju poziciju.



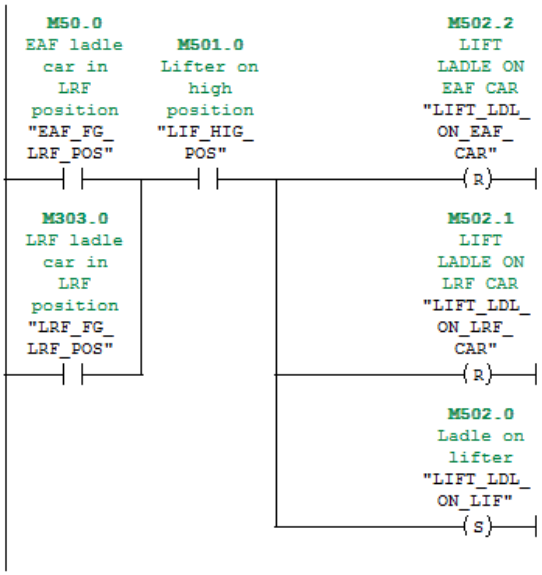
Sl.4.3. Postavljanje lonca na EAF kolica

Programsko rješenje za postavljanje lonca na EAF kolica prikazano je na slici 4.3. Lonac će se postaviti na kolica ukoliko se EAF kolica nalaze na LRF poziciji "EAF_FG_LRF_POS" i ako je dizalo u donjoj poziciji "LIF_LOW_POS". Marker "EAF_FG_LRF_POS" šalje signal iz FC-a za upravljanje EAF kolicima u FC upravljanja dizalom. Marker "LIF_LOW_POS" šalje signal iz istog FC-a gdje je napravljeno upravljanje dizalom. Ako su prethodno opisana dva markera u logičkoj jedinici, lonac će se postaviti na kolica te će se marker "LIFT_LDL_ON_EAF_CAR" postaviti u logičku jedinicu pomoću funkcije SET. Markeri "LIFT_LDL_ON_LIF" i "LIFT_LDL_ON_LRF_CAR" koji služe za postavljanje lonca na LRF kolica i lonca na dizalo postavit će se u logičku nulu.



Sl.4.4. Postavljanje lonca na LRF kolica

Programsko rješenje za postavljanje lonca na LRF kolica prikazano je na slici 4.4. Lonac će se postaviti na kolica ukoliko se LRF kolica nalaze na LRF poziciji "LRF _FG_LRF_POS" i ako je dizalo u donjoj poziciji "LIF_LOW_POS". Marker "LRF_FG_LRF_POS" šalje signal iz FC-a za upravljanje LRF kolicima u FC upravljanja dizalom. Marker "LIF_LOW_POS" šalje signal iz istog FC-a gdje je napravljeno upravljanje dizalom. Ako su prethodno opisana dva markera u logičkoj jedinici, lonac će se postaviti na kolica te će se marker "LIFT_LDL_ON_LRF_CAR" postaviti u logičku jedinicu pomoću funkcije SET. Markeri "LIFT_LDL_ON_LIF" i "LIFT_LDL_ON_EAF_CAR" koji služe za postavljanje lonca na EAF kolica i lonca na dizalo postavit će se u logičku nulu pomoću funkcije RESET.

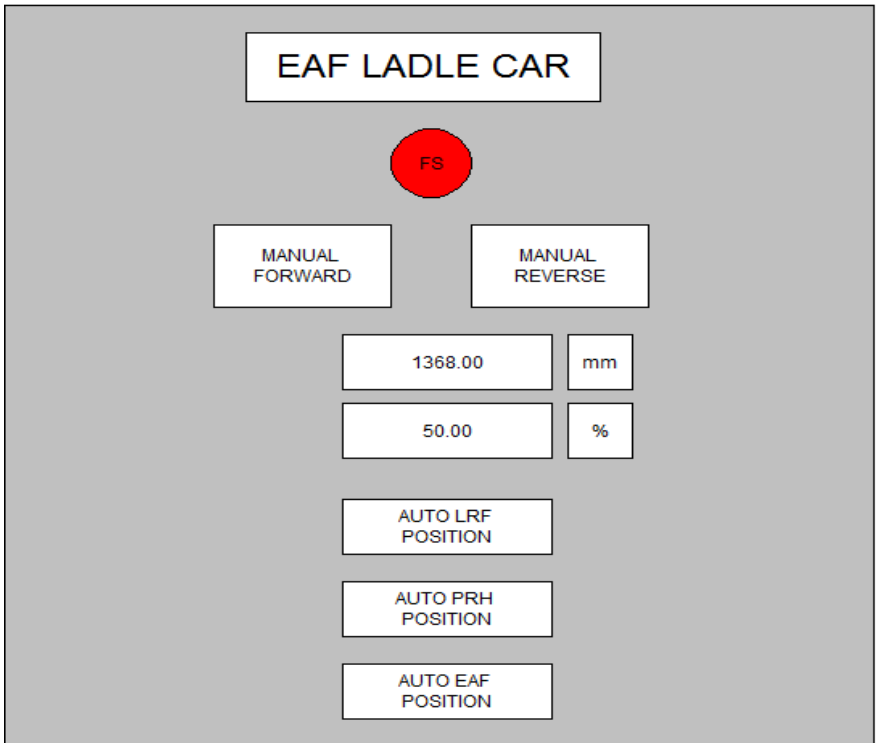


Sl.4.5. Postavljanje lonca na dizalo

Programsko rješenje za postavljanje lonca na dizalo prikazano je na slici 4.5. Lonac će se postaviti na dizalo ukoliko se LRF kolica nalaze na LRF poziciji "LRF _FG_LRF_POS" ili ako se EAF kolica nalaze na LRF poziciji "EAF_FG_LRF_POS". Uz prethodna dva uvjeta dizalo mora biti u gornjoj poziciji "LIF_HIG_POS". Marker "LRF_FG_LRF_POS" šalje signal iz FC-a za upravljanje LRF kolicima u FC upravljanja dizalom. Marker "LIF_HIG _POS" šalje signal iz istog FC-a gdje je napravljeno upravljanje dizalom. Marker "EAF_FG_LRF_POS" šalje signal iz FC-a za upravljanje EAF kolicima. Markeri "LIFT_LDL_ON_EAF_CAR" i "LIFT_LDL_ON_LRF_CAR" koji služe za postavljanje lonca na EAF i LRF kolica postavit će se u logičku nulu pomoću funkcije RESET.

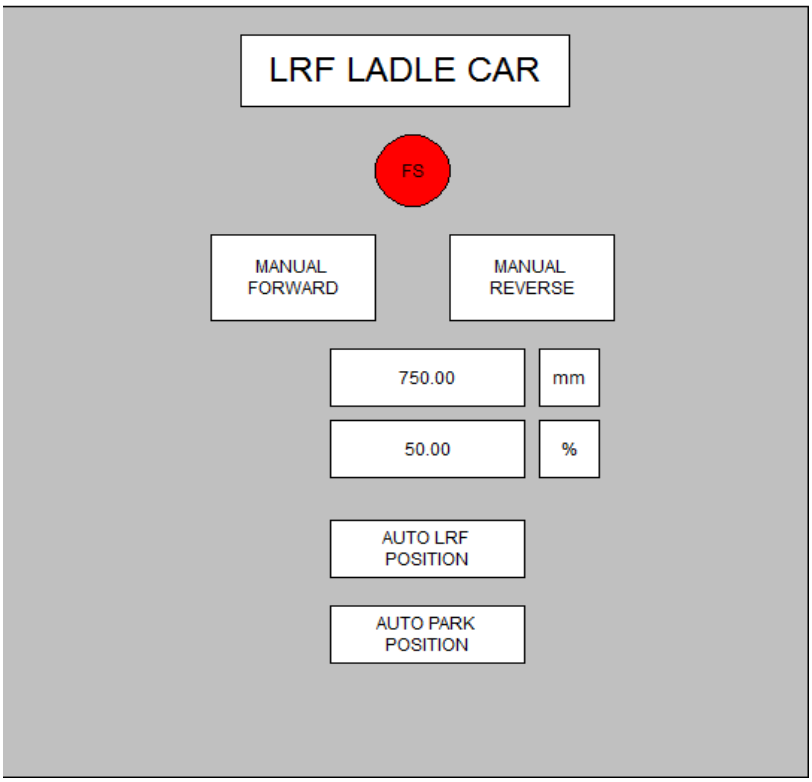
5. VIZUALIZACIJA U CoDeSys-U

Unutar razvojnog programskog okruženja CoDeSys napravljena je vizualizacija sučelja između čovjeka i stroja. EAF kolica i LRF kolica imaju zasebne HMI, te dizalo također ima zaseban HMI.



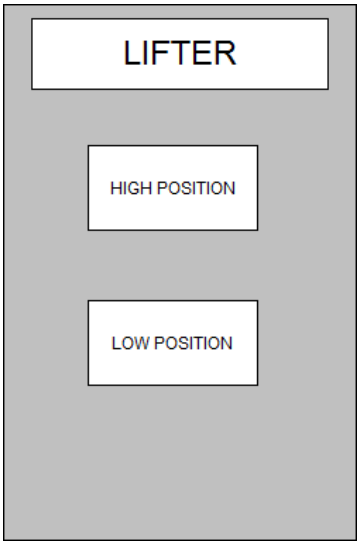
Sl.5.1. Prikaz HMI sučelja EAF kolica

HMI sučelje EAF kolica koje je prikazano na slici 5.1. sadrži naredbe za automatsko pozicioniranje kolica u LRF poziciju (AUTO LRF POSITION), poziciju zagrijavanja (AUTO PRH POSITION) i EAF poziciju (AUTO EAF POSITION). Također sadrži naredbe za ručno pozicioniranje u pozitivnom (MANUAL FORWARD) i negativnom (MANUAL REVERSE) smjeru. Na HMI sučelju ispisuje se pređena udaljenost kolica i brzina. FS označuje naredbu za žurno zaustavljanje kolica u rizičnim situacijama prilikom upravljanja.



Sl.5.2. Prikaz HMI sučelja LRF kolica

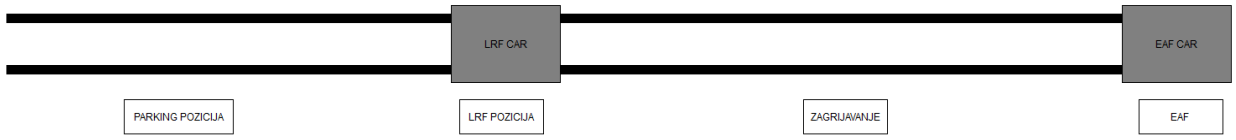
HMI sučelje LRF kolica koje je prikazano na slici 5.2. sadrži naredbe za automatsko pozicioniranje kolica u LRF poziciju (AUTO LRF POSITION) i parking poziciju (AUTO PARK POSITION). Također sadrži naredbe za ručno pozicioniranje u pozitivnom (MANUAL FORWARD) i negativnom (MANUAL REVERSE) smjeru. Na HMI sučelju ispisuje se prijeđena udaljenost kolica i brzina. FS označuje naredbu za žurno zaustavljanje kolica u rizičnim situacijama prilikom upravljanja.



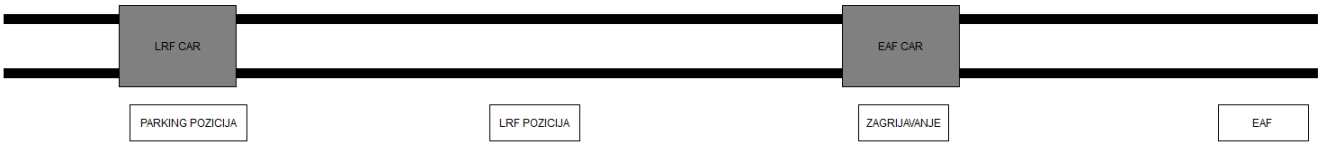
Sl.5.3. Prikaz HMI sučelja LRF kolica

HMI sučelje dizala koje se sastoji od naredbi za postavljanje dizala u gornju i donju poziciju prikazano je na slici 5.3. Pomoću HIGH POSITION postavljamo dizalo u gornju poziciju, pomoću LOW POSITION postavljamo dizalo u donju poziciju.

Na slici 5.4. prikazana je vizualizacija kolica koja se nalaze na početnim pozicijama. Vizualizacija LRF kolica nakon izvršavanja automatskog ciklusa na poziciju parkiranja i EAF kolica na poziciju zagrijavanja prikazan je na slici 5.5.



Sl.5.4. Vizualizacija kolica s pozicijama



Sl.5.5. Vizualizacija kolica nakon pozicioniranja

6. ZAKLJUČAK

U ovom završnom radu cilj je bio izraditi programsku podršku za automatsko upravljanje sustavom dvaju kolica. Za izradu programske podrške korišten je alat Step7 i CoDeSyS.

RAMP funkcija predstavlja jedan od temeljnih dijelova koda jer je korištena za linearno ubrzavanje/usporavanje motora koji pokreće kolica. Prikazano je i objašnjeno programsko rješenje RAMP funkcije.

Automatsko upravljanje kolicima s loncem glavni je dio ovog rada. Opisane su funkcije kolica i sama svrha istih. Objašnjene su funkcije mehaničkih prekidača pomoću kojih se prati pozicija kolica s loncima, slikovno su predstavljene pozicije pojedinog prekidača za EAF i LRF kolica. Praćenje pozicije EAF i LRF kolica moguće je pomoću virtualnog enkodera koji je sastavni dio simulatora pozicije. Automatske sekvence omogućavaju automatsko pozicioniranje kolica te znatno ubrzavaju sam proces pozicioniranja za razliku od ručnog pozicioniranja kolica. Prilikom pozicioniranja kolica treba paziti da ne dođe do kolizije između njih. Kolizije mogu izazvati zaustavljanje rada što izaziva velike novčane gubitke, te ih je potrebno predvidjeti. Objašnjen je način rada dizala i njegove mogućnosti rada. Kolica se izmjenjuju ispod dizala tako što prvo EAF kolica dovezu lonac s materijalom. Zatim dizalo podigne lonac, EAF kolica se izmaknu i LRF kolica se pozicioniraju ispod dizala koje na njega spušta lonac. Također je moguće i suprotno, prvo da LRF kolica dovezu lonac s materijalom. Vizualizacijom su prikazana HMI sučelja kolica i dizala.

LITERATURA

[1] Siemens Step7 Manuals

[2] Siemens, Information and Training, SIMATIC S7

[3] R., Singh PLC E-Learning

http://electrical-engineering-portal.com/resources/plc-programming-training

[4] RAMP funkcija, http://plchowto.com/plc-ramp/

[5] J., Velagić Senzori za mjerenje pozicije, Lekcija 6, Elektrotehnički fakultet Sarajevo

http://people.etf.unsa.ba/~jvelagic/laras/dok/Lekcijam6.pdf, 2012./2013. godine

[6] http://www.plcacademy.com/scan-time-of-the-plc-program/ , svibanj 2015.

SAŽETAK

Izrađen je programski blok za automatsko upravljanje dvaju kolica. Upravljanje kolicima je omogućeno pomoću upravljačkog programa. Upravljački program sastoji se od programskih blokova za automatsko upravljanje dvojih kolica i dizala. Za određivanje pozicije kolica koriste se mehanički prekidači koji šalju digitalne signale u PLC. Prilikom pozicioniranja kolica ne smije doći do kolizije. Upravljanje brzinom vrtnje motora omogućeno je pomoću izrađene RAMP funkcije. Pomoću RAMP funkcije postavljamo željenu vrijednost brzine, tako što RAMP funkcija linearno povećava ili smanjuje brzinu u određenom broju koraka. Izmjenjivanje lonaca je omogućeno pomoću dizala ispod kojeg se kolica izmjenjuju. Programska podrška izrađena je u alatima Step 7 i CoDeSyS. Vizualizacija je izrađena u CoDeSys programskom okruženju.

Ključne riječi: kolica s loncima, RAMP, simulator pozicije, kolizija, dizalo

ABSTRACT

A program block for automatic control of two ladle cars has been developed. Control of the ladle cars is enabled by a control program. The control program consists of program blocks for automated control of the two cars, mechanical switches, car changing, and rotation speed of the motor and the lifter. Mechanical switches that send digital signals to the PLC are used to determine the positions of ladle cars. The ladle cars must not collide during the process of their positioning. Control of the motor speed is enabled by a RAMP function. Using the RAMP function, a speed is set to a desired speed value as the RAMP function linearly increases or decreases the speed in a given number of steps. Changing the ladle is enabled by the lifter under which the cars are switched. The program support was developed in Step 7 and CoDeSyS tools. Visualization was made in the CoDeSys programming environment.

Keywords: ladle cars, RAMP, position simulator, collision, lifter

ŽIVOTOPIS

Davor Brkić rođen je 24.09.1996. u Slavonskom Brodu. Osnovnu školu pohađao je u OŠ "Josip Kozarac" Slavonski Šamac u Kruševici. Oduvijek je pokazivao zanimanje za prirodnim znanostima i tehnici. Srednjoškolsko obrazovanje nastavlja u III. gimnaziji Osijek u Osijeku, koju upisuje 2011. godine. Tijekom srednjoškolskog obrazovanja van prebivališta boravi u učeničkom domu Ugostiteljsko turističke škole u Osijeku. Nakon završetka srednje škole upisuje sveučilišni preddiplomski studiji računarstva na Elektrotehničkom fakultetu Osijek koji se kasnije preimenovao u Fakultet elektrotehnike, računarstva i informacijskih tehnologija Osijek.

PRILOZI:

Priložene datoteke na CD-u:

[1] Završni rad "Upravljanje sustavom dvaju kolica" u .docx formatu

[2] Završni rad "Upravljanje sustavom dvaju kolica" u .pdf formatu

[3] Programska podrška i simulacija izrađena u Step7 u .rar formatu

[4] Programska podrška, simulacija i vizualizacija izrađena u CoDeSys-u u .rar formatu