

Projektiranje upravljačkog sustava automatiziranog regalnog skladišta

Novak, Dino

Master's thesis / Diplomski rad

2024

Degree Grantor / Ustanova koja je dodijelila akademski / stručni stupanj: **University of Zagreb, Faculty of Mechanical Engineering and Naval Architecture / Sveučilište u Zagrebu, Fakultet strojarstva i brodogradnje**

Permanent link / Trajna poveznica: <https://urn.nsk.hr/urn:nbn:hr:235:329237>

Rights / Prava: [In copyright](#)/[Zaštićeno autorskim pravom.](#)

Download date / Datum preuzimanja: **2024-07-14**

Repository / Repozitorij:

[Repository of Faculty of Mechanical Engineering and Naval Architecture University of Zagreb](#)



SVEUČILIŠTE U ZAGREBU
FAKULTET STROJARSTVA I BRODOGRADNJE

DIPLOMSKI RAD

Dino Novak

Zagreb, 2024.

SVEUČILIŠTE U ZAGREBU
FAKULTET STROJARSTVA I BRODOGRADNJE

DIPLOMSKI RAD

Mentor:
prof. dr. sc. Tomislav Staroveški, dipl. ing.

Student:
Dino Novak

Zagreb, 2024.

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

Zahvaljujem se mentoru, profesoru Tomislavu Staroveškom, na mentorstvu, strpljivosti, pruženim prilikama i svom prenesenom znanju.

Zahvaljujem se tvrtci Strojotehnika d.o.o na pruženoj prilici izrade diplomskog rada i osiguranoj opremi. Zahvaljujem se svim kolegama iz tvrtke na pomoći, savjetima, strpljivosti i prenesenom znanju.

Zahvaljujem se majci Sonji, ocu Nikoli i sestri Emili na pomoći i podršci tijekom studiranja.

I na kraju, zahvaljujem se prijateljima koji su mi uljepšali studiranje na svim lijepim trenucima, pomoći i druženjima.

Dino Novak



SVEUČILIŠTE U ZAGREBU
FAKULTET STROJARSTVA I BRODOGRADNJE



Središnje povjerenstvo za završne i diplomske ispite
Povjerenstvo za diplomske ispite studija strojarstva za smjerove:
Proizvodno inženjerstvo, inženjerstvo materijala, industrijsko inženjerstvo i menadžment,
mehatronika i robotika, autonomni sustavi i računalna inteligencija

Sveučilište u Zagrebu Fakultet strojarstva i brodogradnje	
Datum	Prilog
Klasa: 602 - 04 / 24 - 06 / 01	
Ur.broj: 15 - 24 -	

DIPLOMSKI ZADATAK

Student: **Dino Novak**

JMBAG: 0035217873

Naslov rada na hrvatskom jeziku: **Projektiranje upravljačkog sustava automatiziranog regalnog skladišta**

Naslov rada na engleskom jeziku: **Design and commissioning of automated storage rack control system**

Opis zadatka:

U tvrtki Strojotehnika d.o.o. se provodi projekt revitalizacije automatiziranog regalnog skladišta koje je trenutno izvan funkcije. U cilju cjelovite revitalizacije predmetnog sustava, u radu je potrebno:

1. Opisati sustav (regalno skladište) u zatečenom stanju.
2. Projektirati upravljački sustav prema dostupnom budžetu i raspoloživoj opremi te sudjelovati u izradi upravljačkog elektro ormara i ožičenju upravljačkog sustava.
3. Napisati upravljački PLC program i parametrirati sve prigone za pozicioniranje
4. Testirati sustav.
5. Dati zaključke rada.

U radu je potrebno navesti korištenu literaturu i eventualno dobivenu pomoć.

Zadatak zadan:

16. studenoga 2023.

Zadatak zadao:

Izv. prof. dr. sc. Tomislav Staroveški

Datum predaje rada:

18. siječnja 2024.

Predviđeni datumi obrane:

22. – 26 . siječnja 2024.

Predsjednik Povjerenstva:

Prof. dr. sc. Ivica Garašić

SADRŽAJ

SADRŽAJ	I
POPIS SLIKA	II
POPIS TABLICA	IV
POPIS KRATICA	V
SAŽETAK	VI
SUMMARY	VII
1 UVOD	1
2 REGALNO SKLADIŠTE	2
2.1 Zatečeno stanje i uvedene promjene	2
2.2 Mehanički sustav	4
2.3 Ulazi i izlazi	6
2.4 Funkcije i procesi u radu sustava	10
3 UPRAVLJAČKI SUSTAV	15
3.1 ProfiNET i topologija Ethernet mreže [4]	15
3.2 Raspored elemenata u elektroormaru	17
3.3 Napajanje upravljačkog sustava	19
3.4 Programabilni logički kontroler Siemens S7-1214C [8]	20
3.5 Varijabilni frekvencijski pretvarač Mitsubishi FR-E840	20
3.6 Kontroler svjetlosne zavjese DUOmetric SCAN 1.11-min [11]	22
4 UPRAVLJAČKI PLC PROGRAM	24
4.1 Struktura upravljačkog programa	24
4.2 Upravljačka logika stanice za unos robe	26
4.3 Upravljačka logika stanice dizala	33
4.4 Upravljačka logika stanice skladišnog prostora	44
4.5 Sekvence u automatskom režimu rada	49
5 PARAMETRIRANJE PRIGONA	52
5.1 Konfiguriranje komunikacije između PLC-a i VFD-a [13]	52
5.2 Parametriranje prigona vertikalne osi	54
5.3 Parametriranje prigona horizontalne osi	56
6 TESTIRANJE SUSTAVA	57
6.1 Testiranje i parametriranje svjetlosne zavjese	57
6.2 Testiranje sustava u ručnom režimu rada	58
6.3 Testiranje sustava u automatskom režimu rada	59
7 ZAKLJUČAK	60
LITERATURA	61

POPIS SLIKA

1.1	Modicon 084 i njegov razvojni tim [3]	1
2.1	Regalno skladište izvana	2
2.2	HMI	3
2.3	Signalna svjetla	3
2.4	Sučelje za priključivanje	3
2.5	Graničnik	3
2.6	Skica regalnog skladišta	4
2.7	Stalci i spremljena ladica	5
2.8	Nazivne pločice elektromotornih pogona	5
2.9	Ulazi i izlazi regalnog skladišta	7
2.10	Glavni zaslon HMI-a	10
2.11	Zaslon za nadgledanje ulaza i izlaza	11
2.12	Zaslon za evidenciju uskladištene robe	11
2.13	Zaslon za praćenje grešaka	12
2.14	Dijagram toka sekvence <i>Izvadi</i>	13
2.15	Dijagram toka sekvence <i>Spremi</i>	14
3.1	Profinet OSI model [4]	15
3.2	Topologija mreže [6]	16
3.3	Mrežni preklopnik Wago 852-1106 [7]	17
3.4	Raspored elemenata u elektroormaru [6]	17
3.5	Elektroormar	18
3.6	Električna shema spajanja ispravljača napona [6]	19
3.7	PLC Siemens S7-1214C [9]	20
3.8	VFD Mitsubishi FR-E840 [10]	21
3.9	Električna shema spajanja frekvencijskog pretvarača [6]	21
3.10	Kontroler svjetlosne zavjese	22
3.11	Odašiljač svjetlosne zavjese	22
3.12	Električna shema spajanja kontrolera svjetlosne zavjese [6]	23
4.1	Struktura upravljačkog programa	24
4.2	Funkcijski blok senzora	26
4.3	Definicija pozicije ladice u stanici za utovar	27
4.4	FB svjetlosne zavjese: upravljanje visokobrzinskim brojačem	27
4.5	FB svjetlosne zavjese: resetiranje vrijednosti	28
4.6	FB svjetlosne zavjese: kalibracija	29
4.7	FB svjetlosne zavjese: mjerni način rada	30
4.8	FB svjetlosne zavjese: korekcija izmjerene vrijednosti	31
4.9	FB svjetlosne zavjese: Sigurnosna funkcija	31
4.10	FB svjetlosne zavjese: Greška - Teret u nedopuštenoj poziciji	31
4.11	FB svjetlosne zavjese: Greška - Izmjerena vrijednost manja od nule	32
4.12	Funkcijski blok svjetlosne zavjese	32
4.13	Funkcijski blok za upravljanje inkrementalnim enkoderom	33
4.14	Provjera postizanja tražene pozicije	33
4.15	Brojač polica	34
4.16	Određivanje smjera kretanja vertikalne osi	35
4.17	Referenciranje vertikalne osi	36
4.18	FB za upravljanje kočnicom	37
4.19	FB za pisanje parametara VFD-a	38
4.20	Upravljanje promjenom vrijednosti parametra u memoriji VFD-a	38

4.21	Upravljanje kontaktorom prigona vertikalne osi	39
4.22	Upravljanje bitom <i>StartLift</i>	40
4.23	Logika upravljanja smjerom horizontalne osi	41
4.24	Logika upravljanja bitom <i>StartFWD</i>	42
4.25	FB za upravljanje VFD-om	43
4.26	Struktura za spremanje trenutnog stanja skladišnog prostora	44
4.27	Izračun potrebnog broja polica	45
4.28	Prepisivanje prethodne vrijednosti u novu varijablu	48
4.29	Korak sekvence	49
5.1	FR Configurator 2	52
5.2	Hardverska konfiguracija sustava	53
5.3	Mrežne postavke VFD-a	53
5.4	Sinkrona brzina asinkronog motora s obzirom na frekvenciju i broj polova [14]	54
5.5	Karakteristike prigona vertikalne osi izmjerene postupkom <i>Online auto tuning</i>	55
6.1	Rezultati mjerenja	57
6.2	Testiranje vertikalne osi	58
6.3	Okruženje PLCSIM i podatkovni blok s upisanim podacima	59

POPIS TABLICA

2.1	Karakteristike elektromotornih pogona	6
2.2	Popis ulaza	8
2.3	Popis izlaza	9
5.1	Parametri za komunikaciju VFD-a i PLC-a	52
5.2	Parametri prigona vertikalne osi	54
5.3	Parametri prigona horizontalne osi	56

POPIS KRATICA

AC	<i>Alternating Current</i>	izmjenična struja
DB	<i>Data Block</i>	podatkovni blok
DC	<i>Direct Current</i>	istosmjerna struja
FB	<i>Function Block</i>	funkcijski blok
FBD	<i>Function Block Diagram</i>	dijagram funkcijskih blokova
HMI	<i>Human Machine Interface</i>	operatorski panel
HSC	<i>High Speed Counter</i>	visokobrzinski brojač
IP	<i>Internet Protocol</i>	Internet protokol
NC	<i>Normally Closed contact</i>	normalno zatvoren kontakt
NO	<i>Normally Open contact</i>	normalno otvoren kontakt
PLC	<i>Programmable Logic Controller</i>	programabilni logički kontroler
SFC	<i>Sequential Function Chart</i>	dijagram sekvenci funkcija
ST	<i>Structured Text</i>	strukturirani tekst
VFD	<i>Variable Frequency Drive</i>	varijabilni frekvencijski pretvarač

SAŽETAK

U ovom radu opisan je postupak projektiranja upravljačkog sustava automatiziranog regalnog skladišta. Opisani su glavni dijelovi sustava, uvedene su nadogradnje i nove funkcionalnosti u njegovom radu. Prikazane su i opisane komponente novog upravljačkog sustava. Upravljački sustav temeljen je na programabilnom logičkom kontroleru Siemens S7-1214C i varijabilnom frekvencijskom pretvaraču Mitsubishi FR-E840. Napisan je PLC upravljački program korištenjem softverskog paketa za programiranje kontrolera proizvođača Siemens, TIA Portal V17. Prikazan je proces izrade PLC upravljačkog programa te je objašnjeno na koji način on radi. Korištenjem softverskog alata za upravljanje varijabilnim frekvencijskim pretvaračima proizvođača Mitsubishi Electric, FR Configurator 2, parametrirani su prigoni horizontalne i vertikalne osi. Prikazan je postupak parametriranja svjetlosne zavjese i prigona obiju osi. Na kraju, sustav je temeljito testiran te su dani zaključci rada.

Ključne riječi: regalno skladište, upravljački sustav, PLC, VFD, svjetlosna zavjesa

SUMMARY

This paper describes the process of designing the control system for an automated rack storage system. The main components of the system, introduced upgrades, and the improvements of the processes involved in its operation are outlined. The components of the new control system are presented and described. The control system is based on the Siemens S7-1214C programmable logic controller and the Mitsubishi FR-E840 variable frequency drive. A PLC control program has been written using the Siemens controller programming software, TIA Portal V17. The process of creating the PLC control program is shown, along with an explanation of how it works. Using the Mitsubishi Electric variable frequency drive management software tool, FR Configurator 2, the drives for the horizontal and vertical axes are parameterized. The process of parameterizing the light curtain and both drives is presented. At the end of the paper, the thorough testing process of the system is shown and conclusions are provided.

Keywords: rack storage, control system, PLC, VFD, light curtain

1 UVOD

Svijet kakav danas poznajemo temelji se na proizvodnoj industriji. Svaki artefakt kojeg čovjek koristi - od najjednostavnijih predmeta i uređaja, preko potrošnih dobara pa sve do brodova, zrakoplova i svih njihovih sastavnih dijelova - negdje i nekako mora biti proizveden. Kako bi to bilo moguće u razmjerima koje zahtijeva moderno društvo, proizvodna industrija mora biti brza, učinkovita i pouzdana, što znači da se ne može oslanjati na ljudski ručni rad. Ideje o zamjeni ljudskog rada strojnim sežu još u prvo stoljeće prije nove ere kada su za preradu žita korištene vodenice. U osamnaestom stoljeću Oliver Evans projektirao je mlin za proizvodnju brašna. Rad Evansovog mlina smatra se prvim potpuno automatiziranim industrijskim procesom u povijesti. U drugoj polovici devetnaestog stoljeća postavljaju se temelji teorije automatske regulacije. Otkrićem električne struje i razvojem elektromotora, releja i ostalih elektromehaničkih komponenti u dvadesetom stoljeću, do tada razvijena teorija počinje se primjenjivati u proizvodnim procesima. Sustavi temeljeni na relejnoj logici već tada upravljaju procesima cijelih tvornica. Pojava tranzistora i digitalne logike omogućila je razvoj višestruko bržih i pouzdanijih digitalnih kontrolera [1]. 1968. godine tvrtka General Motors razvila je Modicon 084, prvi programabilni logički kontroler, prikazan na slici 1.1. Voditelj tima koji je razvio Modicon 084, Richard Morley, danas se smatra ocem PLC-a [2].



Slika 1.1: Modicon 084 i njegov razvojni tim [3]

Programabilni logički kontroler, skraćeno PLC (engl. *Programmable Logic Controller*), je industrijsko računalo temeljeno na elektroničkom sklopovlju koje se sastoji od memorije, procesora te analognih i digitalnih ulaza i izlaza. Svrha je PLC-a upravljanje stanjem na izlazima s obzirom na stanja na ulazima. Vežu između stanja na izlazu i ulazu definira upravljački program. Spomenuti ulazi mogu biti senzori, tipkala, releji, potencimetri ili bilo kakvi elektronički uređaji koji mijenjaju svoje izlazno stanje na temelju promjene u sustavu kojim se upravlja. Na izlaze se dodaju elementi čije je stanje potrebno promijeniti. To su najčešće uređaji koji upravljaju izvršnim elementima - releji, pneumatski razvodnici, frekvencijski pretvarači, drugi kontroleri i slično. Upravljački PLC programi se u svojoj osnovi temelje na logičkim operacijama I, ILI i NE koje se provode nad ulaznim signalima i na izlaz daju rezultate logičkih operacija (RLO).

Cilj ovog rada je potpuna revitalizacija automatiziranog regalnog skladišta koja uključuje projektiranje novog upravljačkog sustava, izradu novog elektroarmara i razvoj upravljačkog PLC programa. Korišten je PLC Siemens S7-1214C za koji je napisan upravljački program u programskim jezicima FBD (*Function Block Diagram*) i ST (*Structured Text*) korištenjem softverskog paketa TIA Portal V17.

2 REGALNO SKLADIŠTE

2.1 Zatečeno stanje i uvedene promjene

U tvrtki Strojotehnika d.o.o. nabavljeno je rabljeno regalno skladište Megamat Megalift MM450. Pošto je upravljački sustav u trenutku kada se skladište trebalo početi koristiti bio oštećen do mjere da ga nije bilo moguće popraviti, pokrenut je projekt obnove nabavljenog skladišta. Odlučeno je da će se oštećeni upravljački sustav u potpunosti ukloniti i zamijeniti novim sustavom temeljenim na programabilnom logičkom kontroleru Siemens S7-1214C.



(a) Stanica za unos robe



(b) Skladišni prostor

Slika 2.1: Regalno skladište izvana

Slika 2.1 prikazuje regalno skladište. S lijeve strane vidi se donji dio skladišta gdje je smještena stanica za utovar robe, a s desne strane ostatak skladišta u kojem je smješten prostor za skladištenje. Skladište je modularno što omogućuje povećanje kapaciteta dodavanjem skladišnih modula u visinu. Skladište u trenutnoj konfiguraciji sadrži 184 skladišna mjesta raspoređena u dva stupca na 8,5 m visine. Jednom skladišnom mjestu odgovara jedna polica, odnosno mjesto za jednu ladu. Osim same promjene upravljačkog sustava, koji je detaljnije opisan u nastavku rada, uvedene su još neke nadogradnje u svrhu dovođenja sustava na razinu današnjeg industrijskog standarda. Postojeće sučelje za upravljanje skladištem proglašeno je neadekvatnim za korištenje u kombinaciji s novim upravljačkim sustavom pa je umjesto njega ugrađeno standardno industrijsko operatorsko sučelje ili HMI (*Human-Machine Interface*). Uz HMI, ugrađeno je i sučelje s fizičkim tipkama za osnovne funkcije: dvokanalni prekidač s ključem za prebacivanje između ručnog i automatskog režima rada, udarno tipkalo za zaustavljanje u nuždi (ESTOP), tipkalo za resetiranje alarma i tipkalo za puštanje u automatski režim. HMI i tipkala prikazani su na slici 2.2. Dodana su i tri svjetlosna indikatora za indicaciju statusa skladišta, slika 2.4). Nadalje, ugrađeno je sučelje za povezivanje vanjskog računala na upravljački sustav prikazano na slici 2.5. Sučelje sadrži priključak na izmjenični monofazni napon gradske mreže i *Ethernet* priključak. Na prednji rub stanice za utovar ugrađena su dva graničnika s oprugama čija je uloga mekano zaustavljanje ladice uslijed

neželjenog gibanja uzrokovano njegovom inercijom. Graničnik je prikazan na slici 2.3.



Slika 2.2: HMI



Slika 2.3: Signalna svjetla



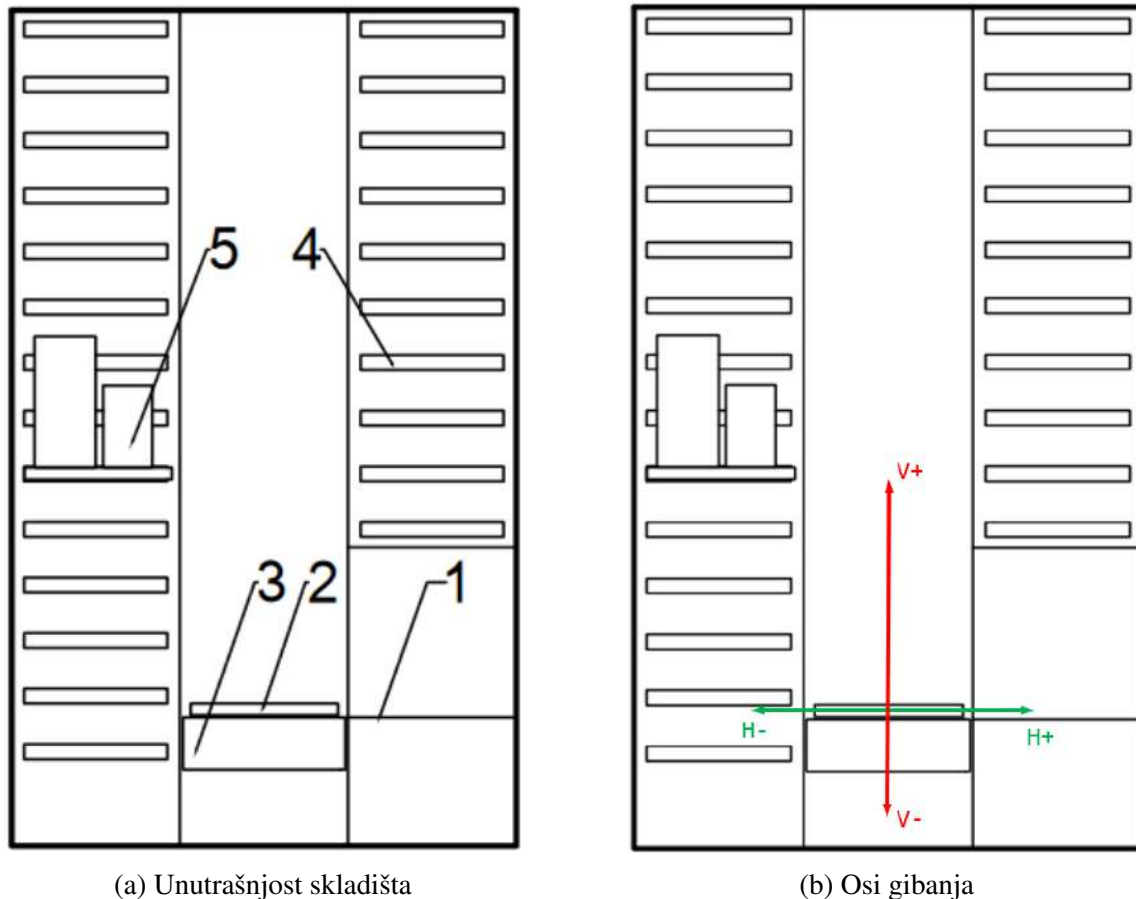
Slika 2.4: Sučelje za priključivanje



Slika 2.5: Graničnik

2.2 Mehanički sustav

Zbog nedostupnosti službene tehničke dokumentacije i nemogućnosti fotografiranja pojedinih dijelova regalnog skladišta, u svrhu prikaza i lakšeg razumijevanja teksta izrađene su skice s označenim dijelovima koji se opisuju. Skice su u odnosu na stvarni proizvod pojednostavljene i nisu u mjerilu.



Slika 2.6: Skica regalnog skladišta

Slika 2.6 (a) prikazuje presjek unutrašnjosti skladišta iz bočnog pogleda. Na slici su brojevima označeni sljedeći dijelovi:

1. Stanica za unos robe,
2. ladica za čuvanje robe,
3. dizalo,
4. stalak za skladištenje ladice i
5. uskladištena ladica s robom.

Kao što je prikazano na slici 2.6 (b), skladište ima dva stupnja slobode gibanja. Dizalo se giba po vertikalnoj osi, na slici označenoj crvenom bojom. Svrha gibanja po vertikalnoj osi jest dolazak dizala u zadanu poziciju. Gibanje u pozitivnom smjeru vertikalne osi odgovara translaciji dizala prema gore. Horizontalnom osi, na slici označenoj zelenom bojom, giba se ladica. Uloga horizontalne osi je pomicanje ladice s dizala na stalak ili obrnuto. Pozitivni smjer horizontalne osi podrazumijeva translaciju ladice prema operateru, odnosno na slici prema desno.

Skladišni prostor sastoji se od dva stupca odvojenih prostorom kroz koji se giba dizalo. Stalak, na slici 2.6 (a) označen brojem 4, element je skladišnog prostora pričvršćen na nosivu konstrukciju u koji se postavlja ladica. Stražnji stupac skladišnog prostora sadrži 101, a prednji 83 stalaka. Prednji stupac ima manji broj stalaka jer je u njemu sadržana i stanica za utovar robe. Stalci i jedna spremljena ladica prikazani su na slici 2.7.

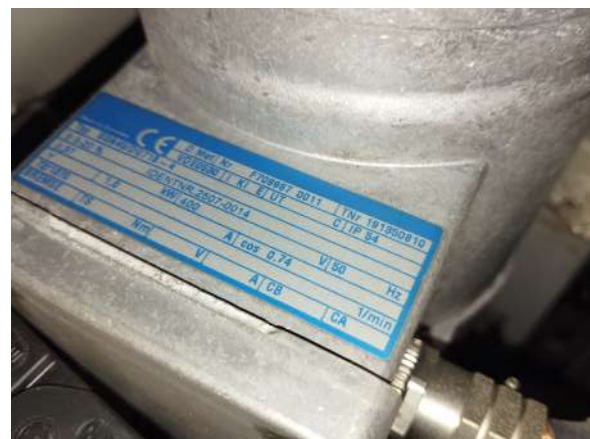


Slika 2.7: Stalci i spremljena ladica

Vertikalna os pogonjena je trofaznim asinkronim elektromotorom FGA332SO/4D200LC-4 nazivne snage 3 kW, a horizontalna trofaznim asinkronim elektromotorom SGA45/DG71B-4 nazivne snage 0,37 kW. Nazivne pločice elektromotornih pogona prikazane su na slici 2.8.



(a) Pogon vertikalne osi



(b) Pogon horizontalne osi

Slika 2.8: Nazivne pločice elektromotornih pogona

Elektromotor vertikalne osi dolazi s elektromehaničkom kočnicom ZFB40SO. U normalnom je stanju zatvorena a otvara se dovođenjem monofaznog izmjeničnog napona iznosa 230 V. Kočnica je nužna jer je motor u svakom trenutku zbog mase dizala opterećen torzijskim momentom oko uzdužne osi vratila. Bez kočnice rotor bi se nakon prestanka dovođenja napona nastavio okretati što bi rezultiralo padom dizala. U tablici 2.1 prikazana je usporedba karakteristika pogona.

Tablica 2.1: Karakteristike elektromotornih pogona

	FGA332SO/4D200LC-4	SGA45/DG71B-4
Nazivna snaga [kW]	3,0	0,37
Napon [V]	400	400
Nazivna struja [A]	6,4	1,0
Nazivna frekvencija [Hz]	50	50
Kutna brzina [min^{-1}]	1430	1870
$\cos \varphi$ [-]	0,82	0,74

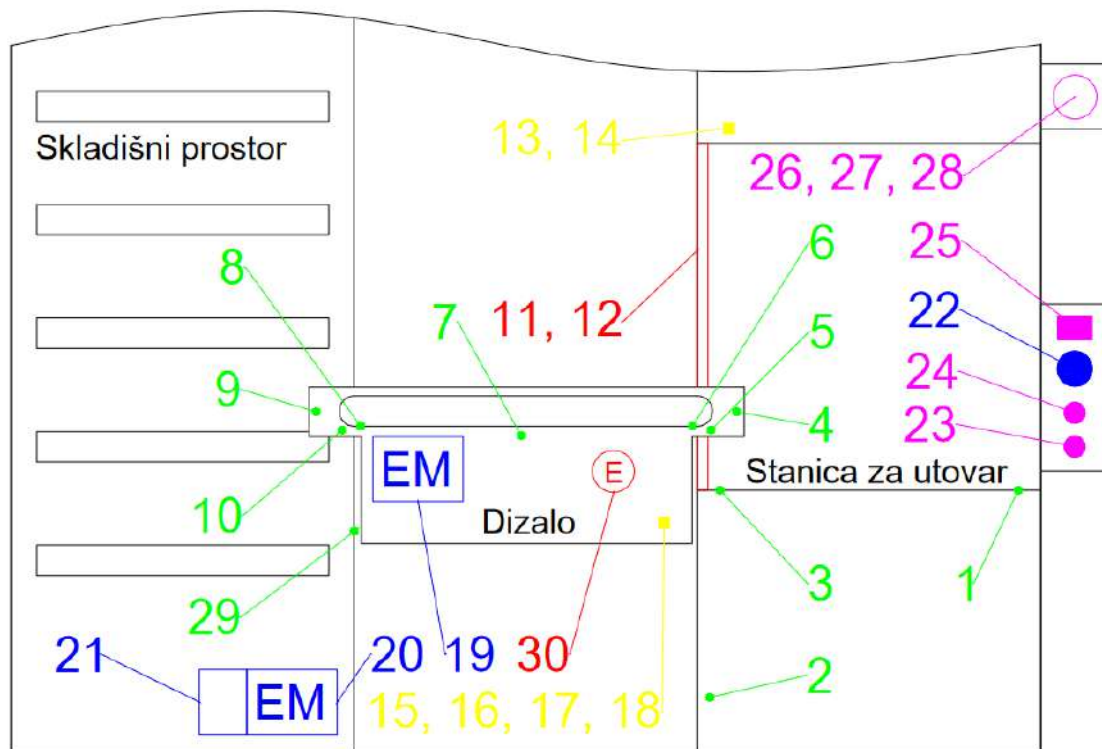
Prijenos snage i gibanja između elektromotora i dizala ostvaruje se preko sustava lančanika i lanaca s bočnih strana skladišta. Dizalo je sa svake strane pričvršćeno na dva lanca koji ga okretanjem lančanika podižu ili spuštaju. Elektromotorni pogon vertikalne osi pričvršćen je na donju gredu nosive konstrukcije skladišta. Za razliku od vertikalne osi, elektromotor horizontalne osi smješten je na donjoj strani platforme dizala. Prijenos gibanja od tog elektromotora do ladice ostvaren je znatno jednostavnijim sustavom koji se sastoji od dva lančanika i lanca s jedne i druge bočne strane platforme dizala. Oba lanca horizontalne osi imaju po dva posebna članka s nastavcima za pomicanje ladice. Ladica se pomiče tako da joj nastavci na lancu zahvate rub i vuku je do jedne od tri moguće pozicije: stražnji stupac skladišta, prednji stupac skladišta ili platforma dizala.

2.3 Ulazi i izlazi

Upravljanje procesima sustava pomoću programabilnog logičkog kontrolera u principu se svodi na definiranje logičkih veza između vrijednosti na ulazima i izlazima kontrolera. Na primjeru regalnog skladišta, kontaktor elektromotora horizontalne osi (izlaz) se mora uključiti ako je dizalo doseglo traženu poziciju (ulaz) i ako je kontaktor elektromotora vertikalne osi isključen (izlaz). Horizontalni elektromotor mora ostati u takvom stanju sve dok ladica nije u traženoj poziciji (ulaz). Kada je ladica dosegla poziciju, kontaktor horizontalnog pogona (izlaz) mora se isključiti. Digitalni ulazi i izlazi mogu biti u jednom od dva stanja: logička nula ili logička jedinica. Logička nula u strujnom krugu upravljačkog sustava odgovara niskoj razini kontrolnog napona (oko 0 V DC), a logička jedinica visokoj razini napona (oko 24 V DC). U ranije opisanom primjeru, uvjet za isključivanje kontaktora horizontalnog pogona je pozicija ladice. Taj uvjet je ispunjen samo ako izlazne vrijednosti svih senzora pozicije ladice indiciraju da je ladica u traženoj poziciji. Normalno otvoreni induktivni senzor zatvara strujni krug ako se u njegovom radnom području nađe metalni dio. Stanica za utovar robe regalnog skladišta ima dva induktivna senzora, jedan na prednjem, a jedan na stražnjem rubu. Kada je ladica u poziciji na stanici za utovar, njezino metalno dno je u radnim područjima obiju navedenih senzora. Strujni krugovi između tih senzora i PLC-a se tako zatvaraju i ulazi prelaze u logičku jedinicu čime PLC upravljački program dobiva potvrdu da je ladica u poziciji. U nastavku na opisani primjer, uključivanje kontaktora elektromotornog pogona podrazumijeva zatvaranje strujnog kruga između PLC-a i kontaktora, odnosno postavljanje određenog izlaza na višu razinu napona ili logičku jedinicu.

Kontaktor je, slično releju, elektromehanička komponenta upravljačkog sustava koji na temelju promjene stanja u kontrolnom strujnom krugu mijenja stanje radnog strujnog kruga. Kontaktori, senzori i prekidači mogu biti normalno otvoreni ili normalno zatvoreni. Strujni krug s normalno

otvorenim kontaktom je otvoren i zatvara se aktiviranjem elementa, a krug s normalno zatvorenim kontaktom je zatvoren i aktiviranjem elementa se otvara. Ova distinkcija je važna jer direktno utječe na logiku u upravljačkom programu. Ulazni i izlazni elementi prisutni kod regalnog skladišta prikazani su na slici 2.9.



Slika 2.9: Ulazi i izlazi regalnog skladišta

Ulazi i izlazi na slici 2.9 prikazani su različitim bojama s obzirom na njihovu vrstu i funkciju. Zelenom bojom označeni su induktivni senzori, žutom granični prekidači, plavom releji i kontaktori, ljubičastom tipkala i svjetla, a crvenom bojom označeni su enkoder i svjetlosna zavjesa. Induktivni senzori služe kao diskretne povratne veze pozicija osi. Granični prekidači su dio sigurnosnog sustava. Njihova je uloga signalizacija upravljačkom programu da je došlo do situacije koja će uzrokovati štetu. Pomoću kontaktora kontroler pokreće i zaustavlja motore, a relej služi za puštanje i zatezanje kočnice pogona vertikalne osi. Tipkala i signalna svjetla služe za interakciju operatera i sustava.

Enkoder i svjetlosna zavjesa razlikuju se od ostalih komponenti po tome što njihov izlazni signal nije konstantni niski ili visoki napon već niz kratkih impulsa visoke razine napona. Standardni ulaz PLC-a nije u stanju uzorkovati takvu vrstu signala pa se za ova dva elementa koristi posebna vrsta ulaza: visokobrzinski brojač (*High-Speed Counter*, HSC). Visokobrzinski brojač detektira brze promjene ulaznog digitalnog signala i upravljačkom programu šalje broj impulsa sadržanih u tom signalu. Enkoder korišten u ovom radu je inkrementalan. Inkrementalni enkoder mjeri relativnu poziciju osi, odnosno za koliko se os pomaknula u odnosu na prošlu poziciju. To znači da ne postoji jedinstvena vrijednost za svaku moguću poziciju osi već samo vrijednost za koju je os zamaknuta u odnosu na poziciju u kojoj je bila na početku rada. Posljedično ovakvom načinu rada, vertikalnu os je nakon svakog ponovnog uključivanja stroja potrebno referencirati. U tablicama i dan je popis svih ulaznih i izlaznih komponenti korištenih u ovom radu. Komponente su u tablicama pridružene brojevima kojima su označene na slici 2.9. Za svaku komponentu navedene su njihova vrsta, tip kontakta, oznaka ulaza na koji su spojeni te je dan kratki opis njihove funkcije.

Tablica 2.2: Popis ulaza

Oznaka	Opis	Tip kontakta	Funkcija	PLC ulaz
1	Induktivni senzor	NO	Pozicija ladice u stanici za utovar	%I11.3
2	Induktivni senzor	NO	Donja referentna pozicija	%I10.4
3	Induktivni senzor	NO	Pozicija ladice u stanici za utovar	%I11.0
4	Induktivni senzor	NO	Brojač polica prednjeg stupca	%I10.1
5	Induktivni senzor	NO	Pozicija ladice na dizalu	%I10.5
6	Induktivni senzor	NO	Pozicija lanca horizontalne osi	%I11.1
7	Induktivni senzor	NO	Pozicija ladice na dizalu	%I10.7
8	Induktivni senzor	NO	Pozicija lanca horizontalne osi	%I11.1
9	Induktivni senzor	NO	Brojač polica stražnjeg stupca	%I10.6
10	Induktivni senzor	NO	Pozicija ladice na dizalu	%I10.2
11	Svjetlosna zavjesa	-	Mjerenje visine tereta (HSC)	%ID104
13	Granični prekidač	NC	Zaustavljanje dizala u slučaju pomicanja sigurnosne poluge	%I13.1
14	Granični prekidač	NC	Zaustavljanje dizala u slučaju pomicanja sigurnosne poluge	%I13.2
15	Granični prekidač	NC	Zaustavljanje dizala ako je nešto ispod	%I12.7
16	Granični prekidač	NC	Zaustavljanje dizala ako je nešto ispod	%I13.0
17	Granični prekidač	NC	Zaustavljanje dizala ako je nešto ispod	%I13.3

Nastavlja se na sljedećoj stranici

Tablica 2.2: Popis ulaza (nastavak)

Oznaka	Opis	Tip kontakta	Definicija	PLC ulaz
18	Granični prekidač	NC	Zaustavljanje dizala ako je nešto ispod	%I13.4
22	Relej	NC	Tipka ESTOP	%I12.1
23	Tipkalo	NO	Resetiranje greške	%I11.7
24	Tipkalo	NO	Početak automatskog rada	%I12.4
25	Dvokanalni prekidač	NO/NC	Prebacivanje između automatskog i ručnog režima	%I11.5, %I11.6
29	Induktivni senzor	NO	Dizalo u stanici za utovar	%I12.0
30	Enkoder	-	Praćenje pozicije dizala	%ID100

Tablica 2.3: Popis izlaza

Oznaka	Opis	Tip kontakta	Funkcija	PLC izlaz
12	Svjetlosna zavjesa	-	Upravljanje svjetlosnom zavjesom (enable signal)	Q10.0
19	Kontaktor	NO	Upravljanje pogonom horizontalne osi	Q1.0
20	Kontaktor	NO	Upravljanje pogonom vertikalne osi	Q1.1
21	Relej	NO	Upravljanje elektromehaničkom kočnicom	Q1.2
26	Signalno svjetlo	-	Crveno svjetlo - greška	Q10.1
27	Signalno svjetlo	-	Žuto svjetlo - upozorenje	Q10.2
28	Signalno svjetlo	-	Zeleno svjetlo - automatski režim rada	Q10.3

2.4 Funkcije i procesi u radu sustava

Operater sustavom upravlja pomoću operatorskog sučelja ili HMI-a. Grafičko korisničko sučelje za HMI izrađeno je na postojećem standardu tvrtke Strojotehnika d.o.o. Na slici 2.10 prikazan je glavni zaslon.



Slika 2.10: Glavni zaslon HMI-a

Svi zaslone s donje strane imaju alatnu traku za kretanje između zaslona. Glavni zaslon sastoji se od tri dijela:

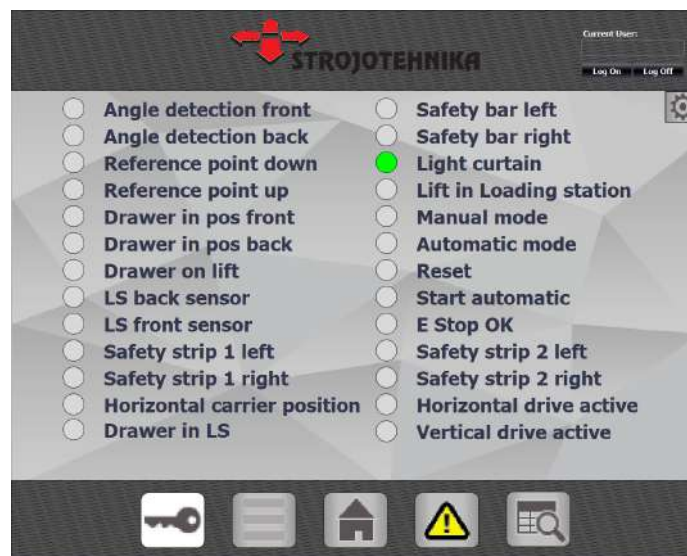
- Sučelje za praćenje stanja sustava - obavještava operatera o trenutnom načinu rada, greškama i funkcijama koje se trenutno izvode;
- Sučelje za ručni režim - služi za ručno upravljanje sustavom;
- Sučelje za automatski režim - služi za pokretanje programa spremanja i vađenja tereta.

Sučelje za praćenje stanja sustava je zamišljeno kao pomoć operateru u korištenju regalnog skladišta. U svakom trenutku na zaslon se ispisuje poruka koja opisuje što sustav trenutno radi na temelju čega operater odlučuje kako će postupiti. Definirano je osam poruka:

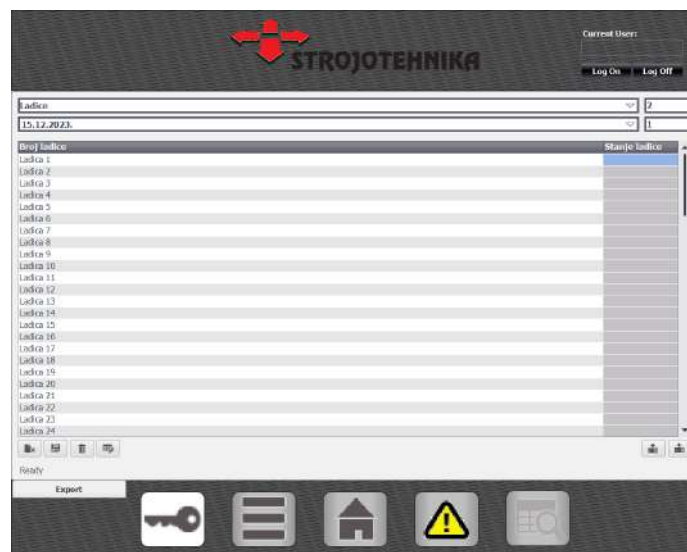
- Automatski način rada;
- Ručni način rada;
- Rad onemogućen;
- Spremanje ladice u tijeku;

- Vađenje ladice u tijeku;
- Referenciranje u tijeku;
- Udarno tipkalo pritisnuto;
- Greška u sustavu.

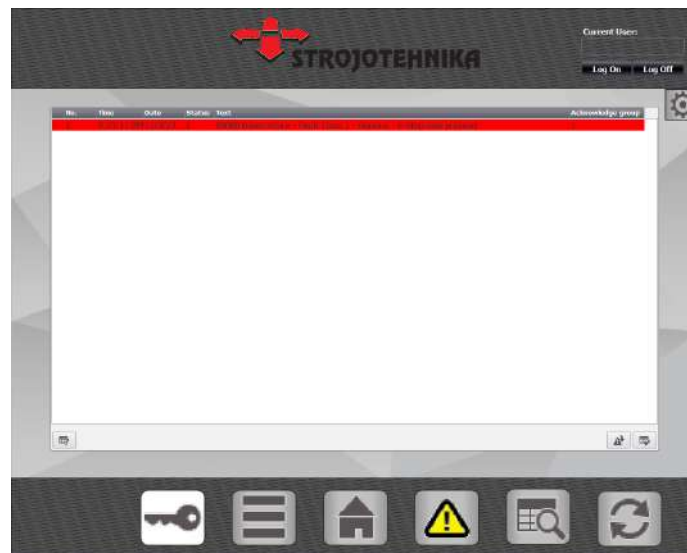
Osim glavnog, HMI nudi još tri zaslona: zaslon za nadgledanje stanja ulaza i izlaza, zaslon za vođenje evidencije uskladištene robe i zaslon za praćenje grešaka. Zaslon za nadgledanje stanja ulaza i izlaza, prikazan na slici 2.11, sadrži popis svih ulaza i izlaza te ikonu koja prikazuje trenutno stanje svakog od njih. Zaslon za vođenje evidencije sadrži tablicu s popisom svih ladica i nudi mogućnost upisivanja uskladištene robe po datumima. Prikazan je na slici 2.12. Zaslon za praćenje grešaka, prikazan na slici 2.13, u slučaju pojave greške ispisuje vrstu i opis greške te gdje je do nje došlo.



Slika 2.11: Zaslon za nadgledanje ulaza i izlaza



Slika 2.12: Zaslon za evidenciju uskladištene robe



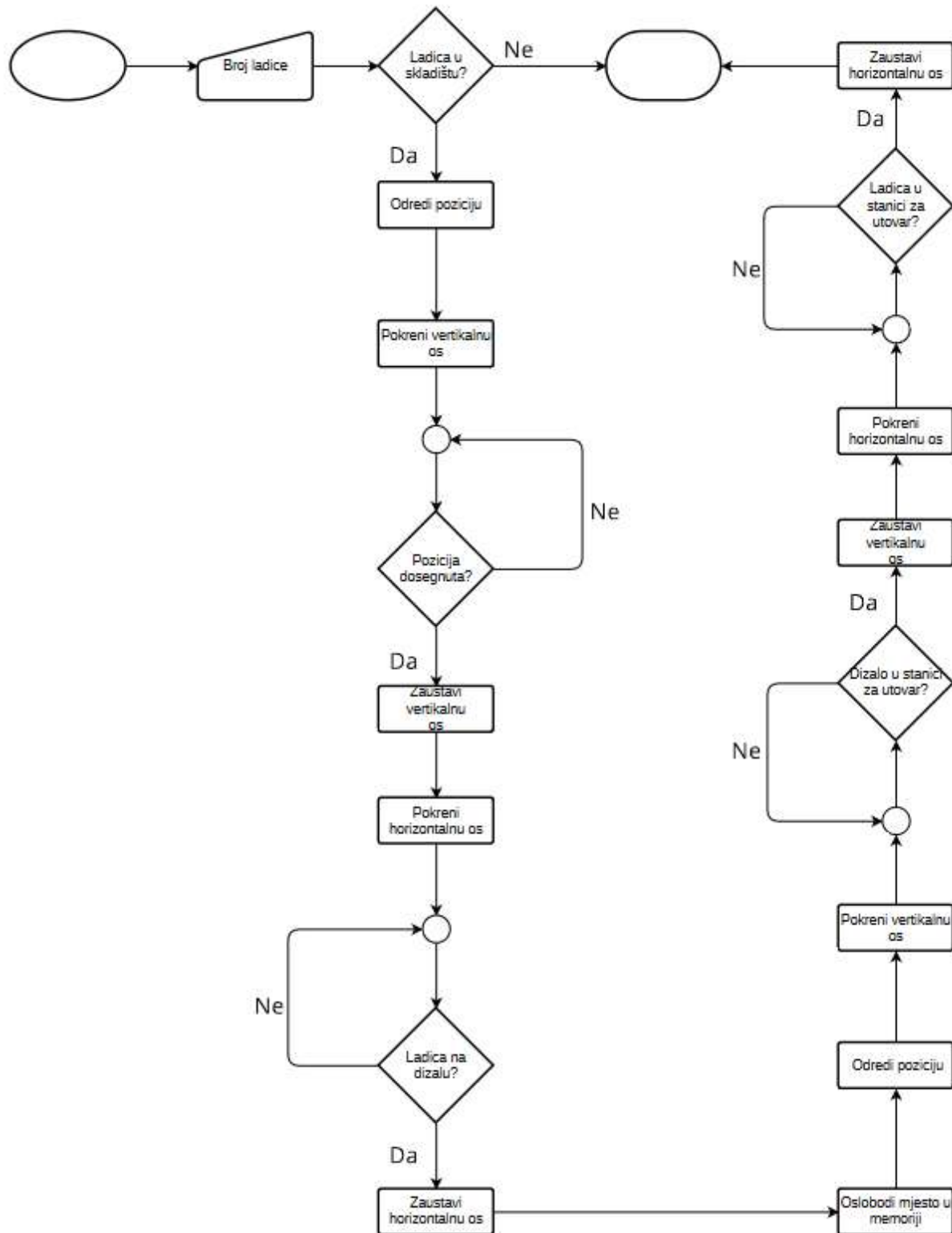
Slika 2.13: Zaslona za praćenje grešaka

Upravljanje automatiziranim regalnim skladištem može se odvijati u jednom od dva režima: ručni i automatski. Ručni režim rada služi za testiranje, dovođenje osi u početnu poziciju u slučaju greške i puštanje sustava u rad. Ručni režim nije namijenjen za izvršavanje funkcija u radu sustava. Omogućuje ručno pomicanje horizontalne i vertikalne osi u oba smjera, promjenu brzine gibanja osi te pokretanje funkcija referenciranja vertikalne osi i kalibracije svjetlosne zavjese. Ako je vertikalna os referencirana i ako su zadovoljeni svi uvjeti sigurnosti, dozvoljava se pokretanje automatskog režima rada. Automatski režim rada zamišljen je kao zadani način u svakodnevnom korištenju sustava. Omogućuje pokretanje četiri funkcije: *Spremi*, *Izvadi*, *Pozovi lift* i *Izbroji police*. Funkcije u automatskom režimu zapravo su sekvence procesa koji se izvode korak po korak, točno određenim redoslijedom. Trenutni korak završava kada su ispunjeni uvjeti za prijelaz na sljedeći korak.

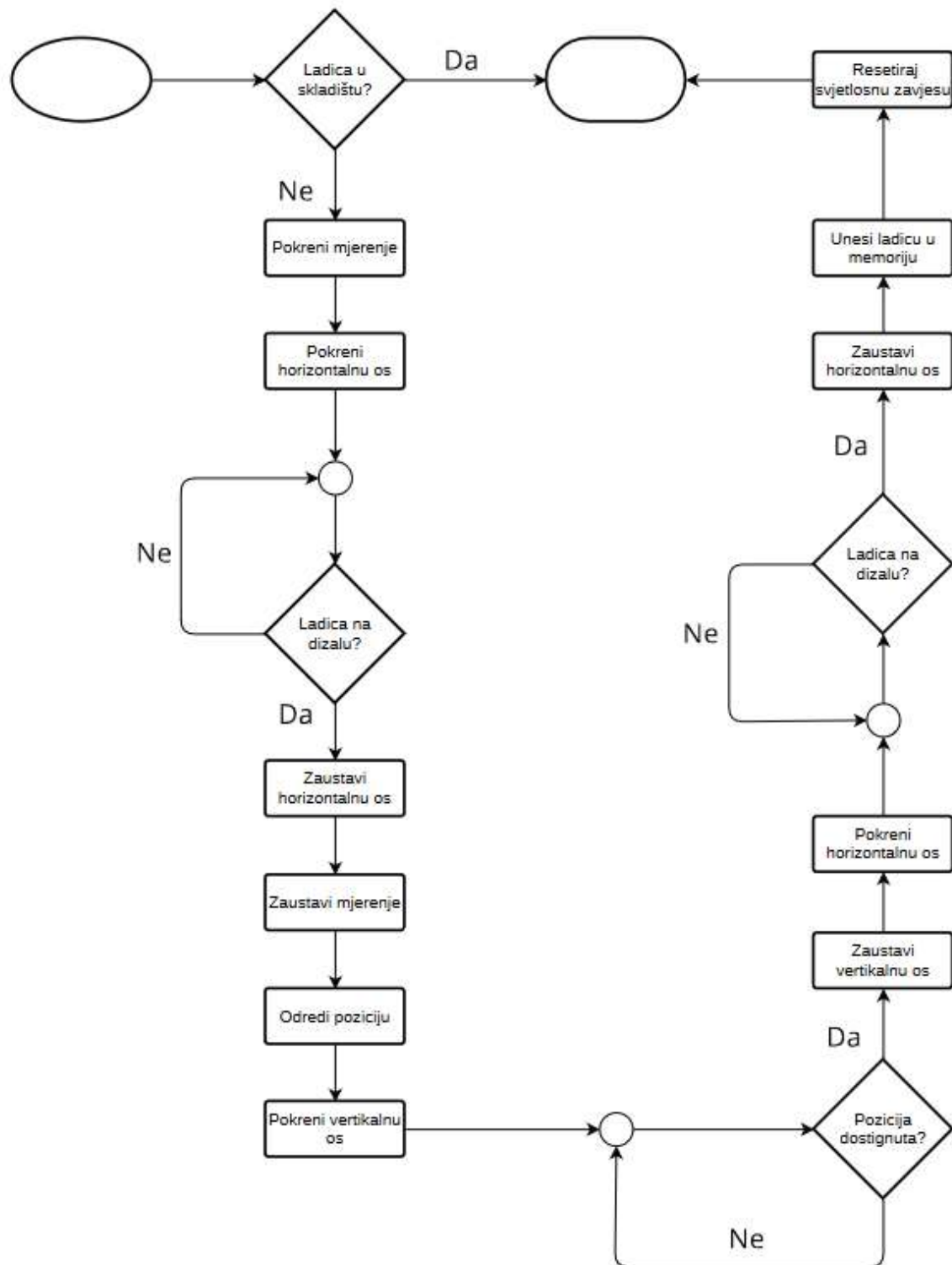
Sekvenca *Izbroji police* služi za mjerenje enkoderskih pozicija polica unutar skladišta. Pokretanjem ove funkcije vertikalna os se pozicionira u donju referentnu točku i iz nje kreće u pozitivnom smjeru. Za svaki rastući i padajući rub signala senzora za brojanje polica stražnjeg stupca (broj 9 u tablici 2.2), u memoriju se privremeno zapisuje vrijednost enkoderske pozicije u tom trenutku. Do rastućeg ruba tog signala dolazi kada je spomenuti senzor na poziciji početka, a do padajućeg kada je u poziciji završetka police. Srednja vrijednost pozicija svakog para padajućeg i rastućeg ruba predstavlja enkodersku poziciju u kojoj se vertikalna os mora zaustaviti kako bi dosegla traženu policu. Izračunate vrijednosti zapisuju se u memoriju PLC-a.

Funkcija *Pozovi lift* služi za dovođenje dizala u stanicu za utovar robe. Pokretanjem funkcije *Pozovi lift*, pokreće se vertikalna os. Na temelju razlike između trenutne pozicije i spremljene enkoderske pozicije stanice za utovar, upravljački program određuje pokreće li se os u pozitivnom ili negativnom smjeru. Sekvenca završava kada je razlika između trenutne i zadane pozicije unutar određene tolerancije i kada je aktivan senzor za poziciju dizala u stanici za utovar (broj 29 u tablici 2.2).

Ove dvije funkcije u pravilu tijekom normalnog svakodnevnog korištenja sustava nisu potrebne već su napravljene kako bi bile dostupne tijekom testiranja i puštanja u rad. Skladištenje robe u sustavu svodi se na pozivanje dviju glavnih funkcija u automatskom režimu: *Spremi* i *Izvadi*. Prije pozivanja ovih funkcija potrebno je u polje *Broj ladice* upisati broj kojim je označena ladica koju operater namjerava spremati ili izvaditi. Opisane sekvence za spremanje i vađenje robe zornije su prikazane dijagramima toka na slikama 2.14 i 2.15.

Slika 2.14: Dijagram toka sekvence *Izvadi*

Pokretanjem sekvence *Izvadi*, upravljački program traži poziciju na kojoj je spremljena tražena ladica. Dizalo odlazi do pronađene pozicije, ladica se povlači s police na dizalo i dovodi se u stanicu za utovar. S ciljem što jednostavnijeg i učinkovitijeg upravljanja sustavom, u slučaju da stanica za utovar u trenutku pozivanja funkcije *Izvadi* već sadrži neku drugu ladice, sustav automatski prije vađenja tražene ladice izvodi sekvencu spremanja trenutne ladice.

Slika 2.15: Dijagram toka sekvence *Spremi*

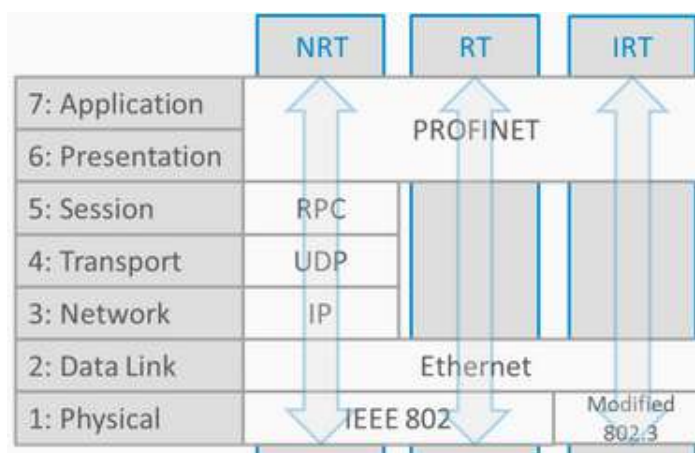
Nakon upisivanja broja ladice, pritiskom na tipku *Spremi* ladica se povlači iz stanice za utovar na dizalo. Na temelju visine tereta koju izmjeri svjetlosna zavjesa, određuje se pozicija unutar skladišta na kojoj ima dovoljno slobodnog prostora za spremanje tog tereta. Dizalo odlazi u tu poziciju i ladica se povlači s dizala na policu. Algoritmi koji traže slobodni prostor u skladištu, određuju lokacije traženih spremljenih ladica i spremaju promjene u memoriju PLC-a prikazani su i opisani u poglavlju *Upravljački PLC program*.

3 UPRAVLJAČKI SUSTAV

3.1 ProfiNET i topologija Ethernet mreže [4]

Većina upravljačkih sustava se, osim od samog programabilnog logičkog kontrolera, sastoji i od različitih perifernih uređaja koji upravljaju različitim elementima sustava. Kako bi PLC mogao upravljati perifernim uređajima, između njih mora postojati neki način prijenosa podataka. Kod jednostavnih uređaja poput induktivnih senzora komunikacija se svodi na dva moguća stanja - viši i niži napon signala. Viši napon predstavlja logičku jedinicu, a niži logičku nulu. Za komunikaciju s kompleksnijim perifernim uređajima poput robotskih kontrolera, varijabilnih frekvencijskih pretvarača, vizijskih sustava i čak drugih programabilnih logičkih kontrolera, koriste se industrijski komunikacijski protokoli i standardi. Komunikacijski protokol je skup pravila koji opisuje na koji se način izmjenjuju podaci između elemenata unutar mreže. Prvi standardizirani industrijski komunikacijski protokol bio je Modbus. Modbus je razvijen za potrebe prvog programabilnog logičkog kontrolera, Modicon 084. Danas postoji velik broj komunikacijskih protokola. Neki od njih jesu Profibus, EtherCAT, EtherNet/IP, ControlNet, CIP i drugi. Jedan od najčešće korištenih protokola u današnjoj industriji automatizacije je ProfiNET.

ProfiNET je komunikacijski standard osnovan na OSI referentnom modelu. OSI model (*Open Systems Interconnection Model*) definira sedam slojeva na temelju kojih uređaji u mreži razmjenjuju podatke. Fizički sloj određuje fizičku vezu između uređaja, konektore, naponske razine i slično. Sloj podatkovne veze omogućuje komunikaciju između uređaja unutar lokalne mreže preko MAC (*Media Access Control*) adresa uređaja. Mrežni sloj dijeli segmente podataka u pakete i šalje ih uređajima preko IP (*Internet Protocol*) adrese. Transportni sloj služi za pouzdani prijenos podataka na TCP (*Transmission Control Protocol*) ili UDP (*User Datagram Protocol*) protokolu. Dijeli podatke na segmente i primljene segmente sastavlja na strani primatelja. Sesijski sloj definira komunikacijske kanale ili sesije između uređaja. Omogućuje stvaranje kontrolnih točaka s kojih se prijenos podataka može nastaviti u slučaju prekida veze. Prezentacijski sloj definira na koji način uređaji trebaju pakirati i komprimirati podatke. Aplikacijski sloj omogućuje korištenje protokola aplikacijama koje prikazuju podatke korisniku [5]. Prikaz Profinet standarda u OSI referentnom modelu dan je na slici 3.1.



Slika 3.1: Profinet OSI model [4]

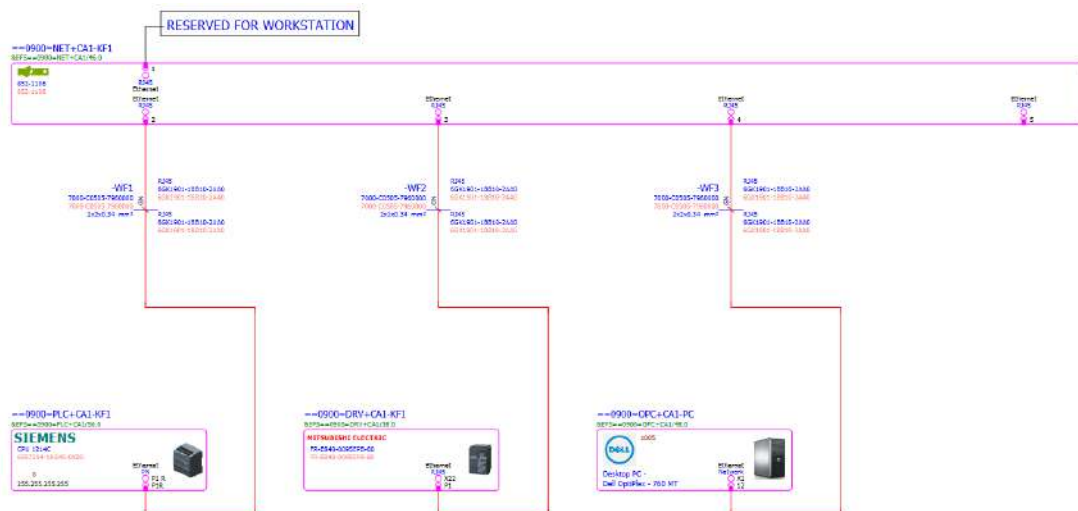
Fizički i podatkovni sloj ProfiNET standarda definirani su standardom IEEE 802.3. Fizička veza ostvaruje se Ethernet kabelom i RJ45 konektorom. Svaki uređaj na Profinet mreži ima svoju MAC adresu preko koje može komunicirati s drugim uređajima na podatkovnom sloju. Ovisno o zahtiji-

jevanoj brzini i latenciji prijenosa podataka, Profinet komunikacija može se odvijati na jednom tri načina:

- asinkroni način (*Non-Real-Time*, NRT),
- komunikacija u stvarnom vremenu (*Real-Time*, RT),
- izokroni način (*Isochronous Real-Time*, IRT).

Komunikacija na asinkronom kanalu odvija se kroz svih sedam slojeva OSI referentnog modela. Za prijenos podataka koriste se protokoli IP, UDP i RPC (*Remote Procedure Call*). Posljedica razmjene podataka kroz sve slojeve je visoka latencija što komunikaciju u asinkronom načinu rada čini nepraktičnom za primjene u situacijama kada podaci moraju biti dostavljeni u točno određeno vrijeme. Razmjena podataka u asinkronom načinu je nedeterministički proces što znači da trenutak u kojem će poruka doći do cilja nije jednoznačno određena već ovisi o drugim razmjenama u mreži. NRT se koristi za uspostavljanje veze između uređaja i dohvaćanje dijagnostičkih podataka. Komunikacija u stvarnom vremenu ne koristi mrežni, transportni i sesijski sloj. Time se smanjuje latencija i razmjena podataka je brža. Nedostatak komunikacije na ovom kanalu je da, zbog zaobilazanja mrežnog sloja, uređaji ne mogu komunicirati preko IP adresa. Komunikacija je zato ograničena samo na lokalnu mrežu, a odvija se u podatkovnom sloju preko MAC adresa. Izokroni način rada pogodan je za primjenu u procesima poput regulacije pozicije, gdje je važno da svaki paket dođe u točno određeno vrijeme. Komunikacija u izokronom načinu čini ProfiNET determinističkim protokolom. Deterministička komunikacija na Ethernet mreži postignuta je dodavanjem novih pravila za mrežne preklopnike (*switch*) u IEEE 802.3 standardu. Prema novom pravilu, preklopnik daje prednost kritičnim podacima dok ostale podatke sprema i propušta tek kasnije. Za razliku od NRT i RT načina rada, IRT način nije podržan na svim ProfiNET uređajima već samo na onim koji zahtijevaju takvu vrstu komunikacije.

U upravljačkom sustavu regalnog skladišta za usmjeravanje prijenosa podataka na Ethernet mreži korišten je mrežni preklopnik Wago 852-1106. Na slici 3.2 prikazana je topologija mreže upravljačkog sustava. Ethernet mrežom povezani su PLC, kontroler elektromotornih pogona i HMI. PLC i kontroler elektromotornih pogona izmijenjuju podatke na ProfiNET protokolu.



Slika 3.2: Topologija mreže [6]

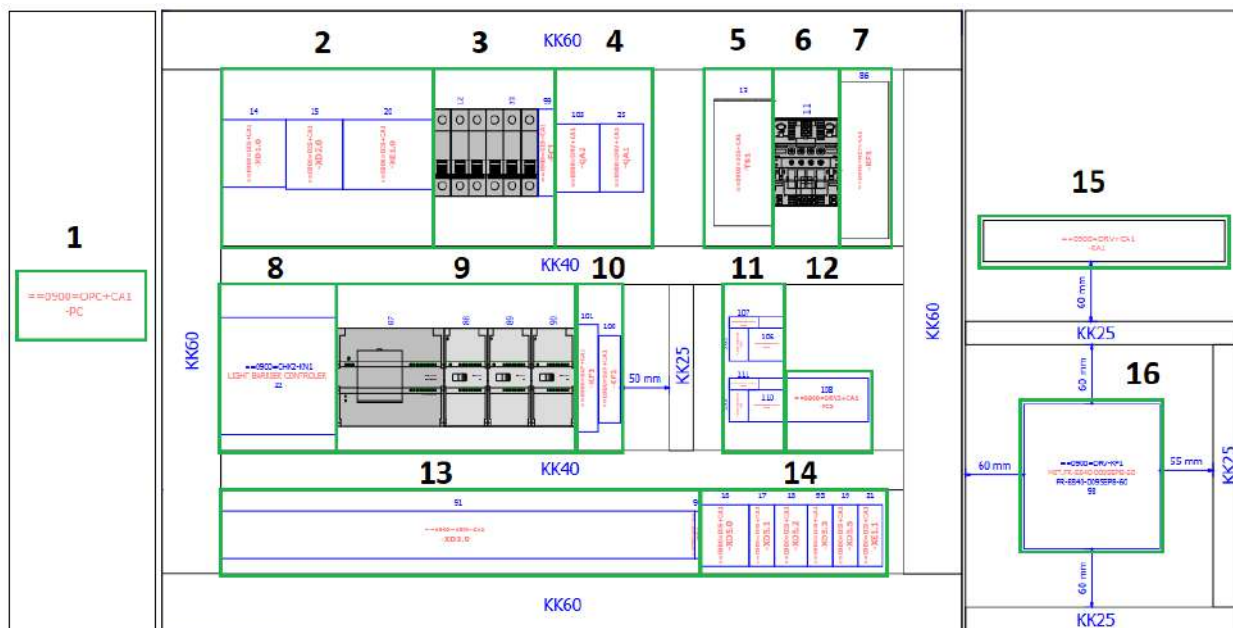
Mrežni preklopnik Wago 852-1106, prikazan na slici 3.3, ima šesnaest Ethernet priključaka i podržava prijenos podataka ProfiNET protokolom.



Slika 3.3: Mrežni preklopnik Wago 852-1106 [7]

3.2 Raspored elemenata u elektroormaru

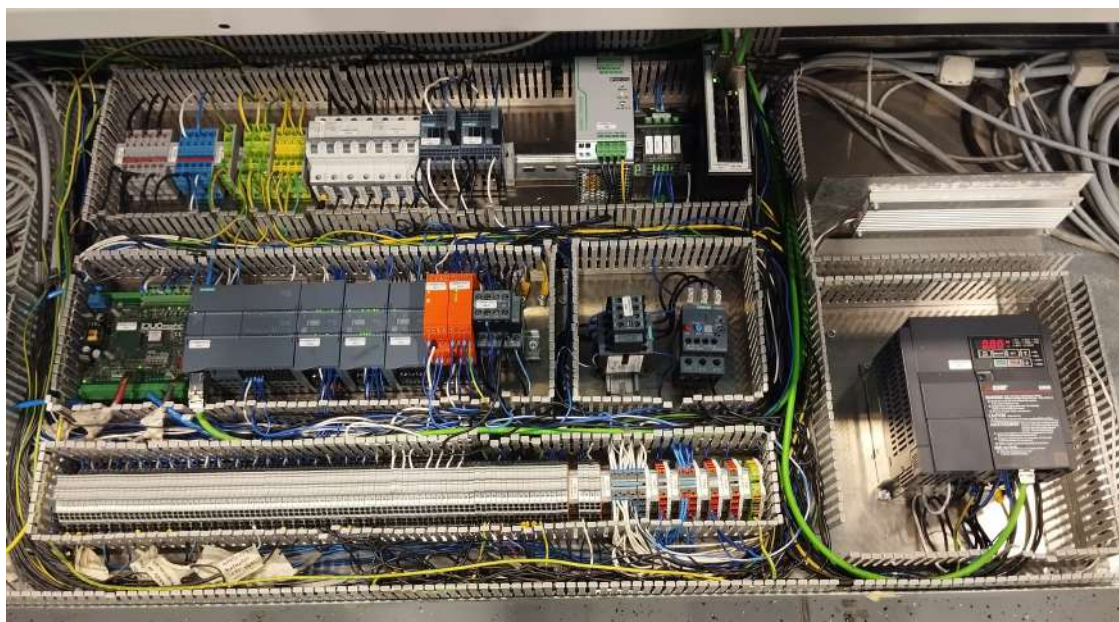
Na slici 3.4 prikazana je shema rasporeda elemenata u elektroormaru regalnog skladišta. Na temeljnu ploču elektroormara najprije su montirane DIN šine za prihvatanje komponenti i kanalice širina 60, 40 i 25 mm za polaganje električnih vodova. Na slici 3.5 prikazan je elektroormar.



Slika 3.4: Raspored elemenata u elektroormaru [6]

Na DIN šine zatim su montirane sve komponente. Brojevima od 1 do 16 na slici 3.4 označene su sljedeće skupine komponenata:

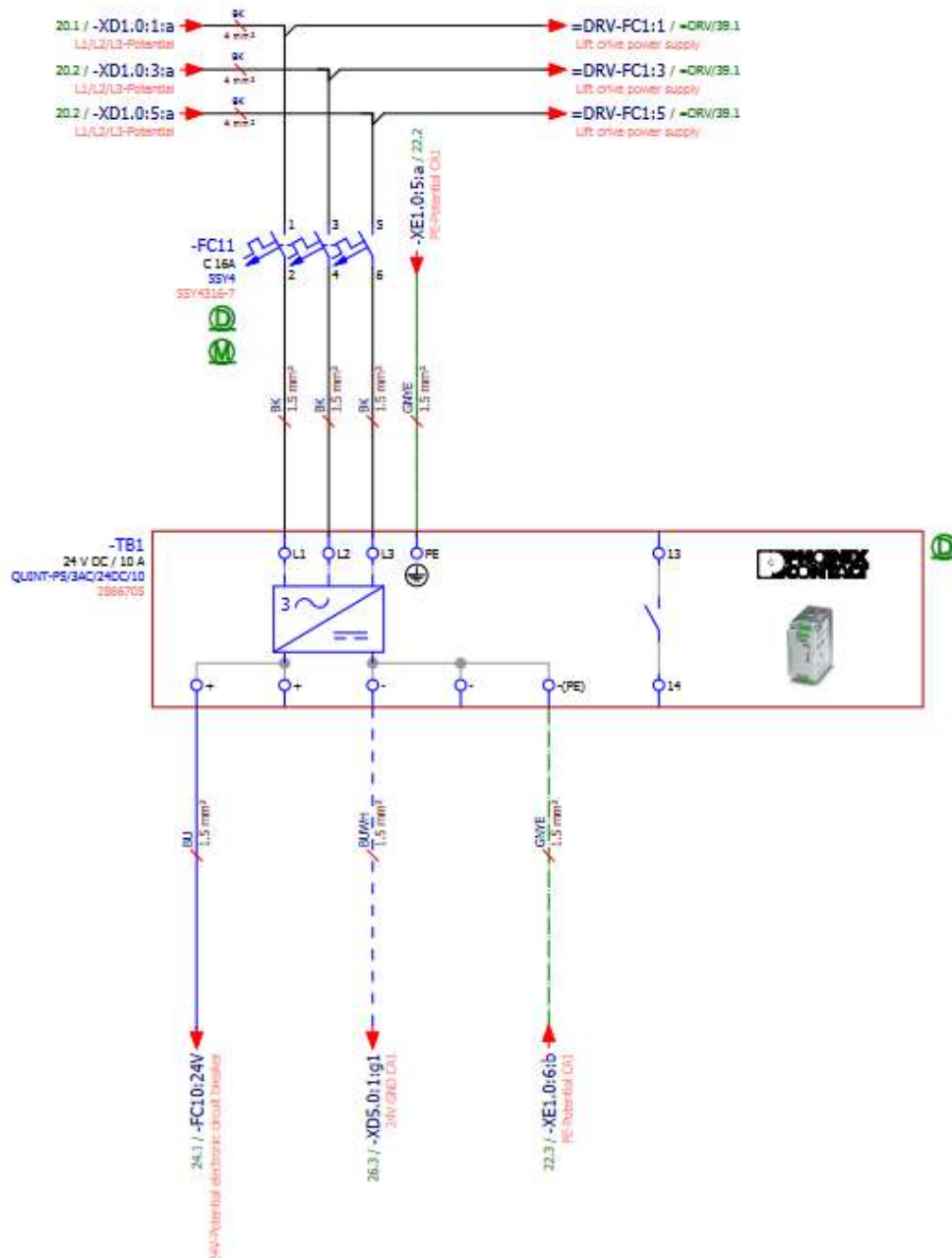
1. Računalo operatorskog sučelja (HMI PC),
2. redne stezaljke za distribuciju radnog napona,
3. glavni osigurači,
4. kontaktori kočnice,
5. ispravljač napona,
6. osigurač DC kruga,
7. mrežni preklopnik,
8. kontroler svjetlosne zavjese,
9. programabilni logički kontroler (PLC),
10. sigurnosni releji,
11. kontaktori elektromotornih pogona,
12. osigurač elektromotornih pogona,
13. redne stezaljke ulaza i izlaza,
14. redne stezaljke za distribuciju kontrolnog napona,
15. otpornik kočnice,
16. varijabilni frekvencijski pretvarač (VFD)



Slika 3.5: Elektroormar

3.3 Napajanje upravljačkog sustava

Sustav se napaja trofaznom strujom napona 400 VAC. Ovakav napon se kroz distribucijski dovodi u VFD te se zatim iz njega odvodi u elektromotore horizontalne i vertikalne osi. Monofazni napon iznosa 230 VAC koristi se za napajanje računala HMI-a, kočnice elektromotora vertikalne osi i osvjjetljenja regalnog skladišta. Osim u VFD, trofazni napon je spojen i na ulazni terminal ispravljača napona Phoenix Contact QUINT-PS/3AC/24DC/10.



Slika 3.6: Električna shema spajanja ispravljača napona [6]

Shema spajanja ispravljača napona u upravljački sustav prikazana je na slici 3.6. Ispravljač pretvara ulazni trofazni napon iznosa 400 VAC u istosmjerni napon iznosa 24 VDC. Istosmjerni se napon s izlaznog terminala ispravljača odvodi u distribucijski sustav kontrolnog strujnog kruga. Ovim se naponom napajaju PLC, mrežni preklopnik, svjetlosna zavjesa, enkoder vertikalne osi, induktivni senzori, granični prekidači, tipkala i signalna svjetla.

3.4 Programabilni logički kontroler Siemens S7-1214C [8]

Za upravljačko računalo sustava odabran je programabilni logički kontroler Siemens S7-1214C DC/DC/DC. Ovaj PLC dio je linije Siemensovih kontrolera S7-1200. Središnji modul (*Central Processing Unit, CPU*) sastoji se od mikroprocesora, memorije, ulaznih i izlaznih priključaka te Ethernet sučelja za komunikaciju preko ProfiNET protokola. Prikazan je na slici 3.7.



Slika 3.7: PLC Siemens S7-1214C [9]

DC/DC/DC u nazivu kontrolera označava da se ovaj PLC te njegovi ulazi i izlazi napajaju istosmjernim naponom. Napon kontrolnog kruga iznosi 24 V. CPU podržava četrnaest digitalnih ulaza, deset digitalnih izlaza i dva analogna ulaza. Može se proširiti s do osam ulaznih ili izlaznih modula, do 3 komunikacijska modula, memorijskom karticom i najviše jednom signalnom ili komunikacijskom pločicom i jednim baterijskim modulom. Komunikacijski moduli dodaju sučelja za komunikaciju preko drugih protokola. U ovom radu CPU je proširen s dva dodatna ulazna modula SM1221 i jednim izlaznim modulom SM1222. Modul SM1221 dodaje šesnaest digitalnih ulaza, a modul SM1222 šesnaest digitalnih izlaza. Memorija CPU-a 1214C sastoji se od 75 KB radne memorije (*work memory*), 4 MB memorije za učitavanje (*load memory*) i 10 KB memorije za zadržavanje (*retentive memory*).

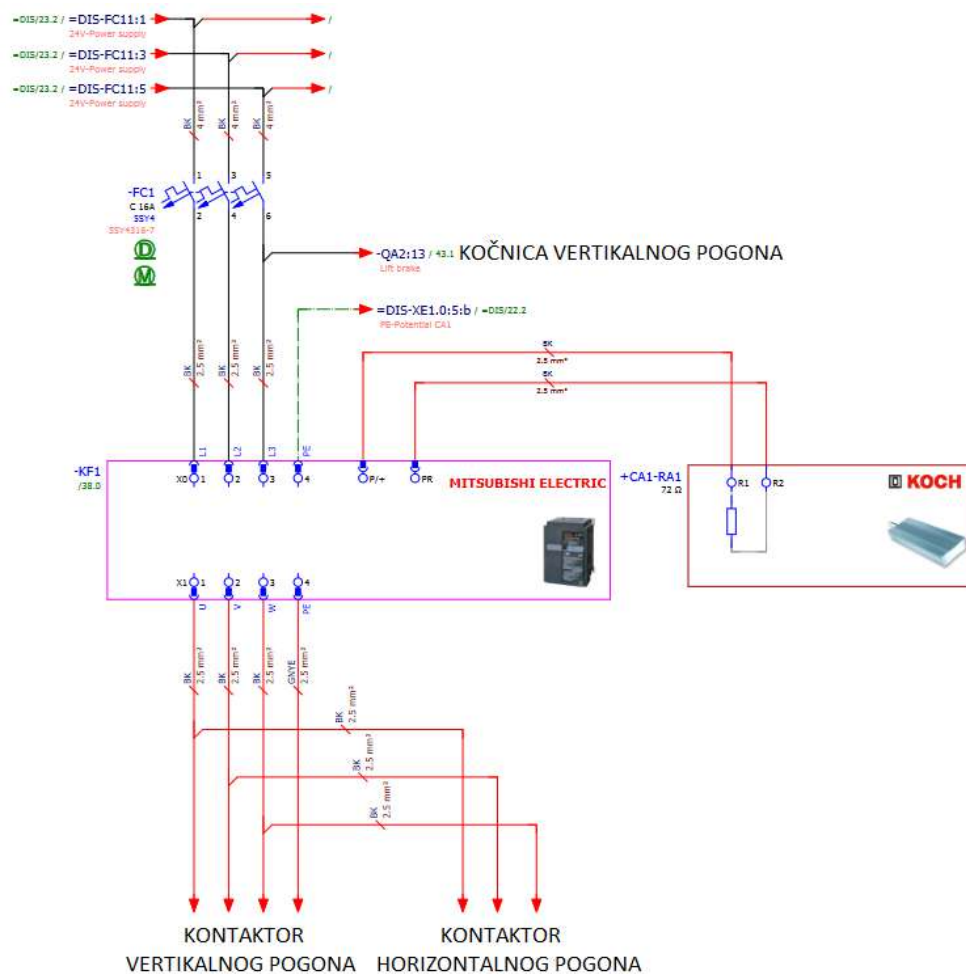
3.5 Varijabilni frekvencijski pretvarač Mitsubishi FR-E840

U svrhu upravljanja elektromotornim pogonima horizontalne i vertikalne osi odabran je varijabilni frekvencijski pretvarač Mitsubishi FR-E840-0095EPB-60. Varijabilni frekvencijski pretvarač (*Variable Frequency Drive, VFD*) je uređaj koji služi za regulaciju brzine asinkronih indukcijskih motora na principu promjene frekvencije ulaznog izmjeničnog napona. Sinkrona brzina vrtnje rotora elektromotora proporcionalna je frekvenciji ulaznog napona. Ako nazivna frekvencija elektromotora iznosi 50 Hz, kao što je ovdje slučaj, njegovom sinkronom brzinom smatra se brzina pri frekvenciji ulaznog napona koja odgovara nazivnoj frekvenciji motora. Osim regulacije brzine, VFD omogućuje promjenu upravljačkih parametara kao što su vrijeme ubrzanja i vrijeme zaustavljanja, automatsko mjerenje i namještanje konstanti motora i druge. Ovaj VFD podržava upravljanje i parametriranje dva elektromotora. Prikazan je na slici 3.8.



Slika 3.8: VFD Mitsubishi FR-E840 [10]

Na slici 3.9 prikazana je električna shema spajanja VFD-a u upravljačkom sustavu. Električni vodovi L1, L2, L3 i PE (tri faze i uzemljenje) spojeni su na ulazne kontakte konektora X0 na VFD-u. Kontakti izlaznog konektora X1 spojeni su na kontakte koji su dalje vodovima U, V i W povezani na ulazne konektore samih elektromotora. Na kontakte P/+ i PR VFD-a spojen je otpornik kočnice elektromotora vertikalne osi.



Slika 3.9: Električna shema spajanja frekvencijskog pretvarača [6]

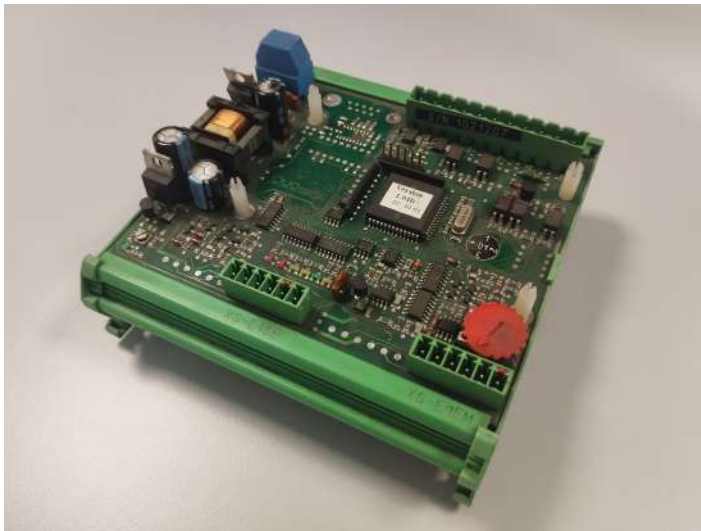
3.6 Kontroler svjetlosne zavjese DUOmetric SCAN 1.11-min [11]

Sustav svjetlosne zavjese sastoji se od kontrolera, odašiljača i prijamnika. Odašiljač ima 64 izvora svjetlosti u infracrvenom spektru. Snopovi svjetlosti usmjereni su prema prijamniku koji ih očitava. Odašiljač i prijamnik povezuju se na kontroler, a kontroler se povezuje na PLC i istosmjerni napon iznosa 24 V. PLC za komunikaciju s kontrolerom koristi tri ulaza i jedan izlaz. Svjetlosna zavjesa ima tri načina rada:

- normalni rad,
- kalibracija,
- mjerenje.

U normalnom načinu rada odašiljač šalje snopove svjetlosti prijamniku koji kontroleru daje signal kojim potvrđuje da svi snopovi dolaze od odašiljača do prijamnika. Kontroler za to vrijeme na ulaz PLC-a šalje konstantni signal na višoj razini napona (24 V) ili logičku jedinicu. U slučaju dolaska nekog objekta u prostor svjetlosne zavjese, određeni broj snopova se prekida i prijamnik zaustavlja slanje signala kontroleru. Razina napona signala na ulazu PLC-a pada, odnosno kontroler PLC-u šalje logičku nulu. Normalni način rada obavlja sigurnosnu funkciju. Kada PLC na ulazu zaprimi logičku nulu, sve funkcije sustava se zaustavljaju.

Kalibracijski način aktivira se slanjem signala u trajanju između 90 i 110 ms s izlaza PLC-a na ulaz kontrolera. Nakon što je kalibracija gotova, svjetlosna zavjesa je spremna za mjerenje. Mjerni način aktivira se slanjem konstantnog signala s izlaza PLC-a kontroleru. Signal je aktivan tako dugo dok traje mjerni način. Prestankom slanja aktivacijskog signala kontroler na ulaz PLC-a šalje signal s određenim brojem impulsa i frekvencijom 1 kHz. Svjetlosna zavjesa se vraća u normalni način. Na slici 3.10 je prikazan kontroler svjetlosne zavjese, a na slici 3.11 odašiljač.

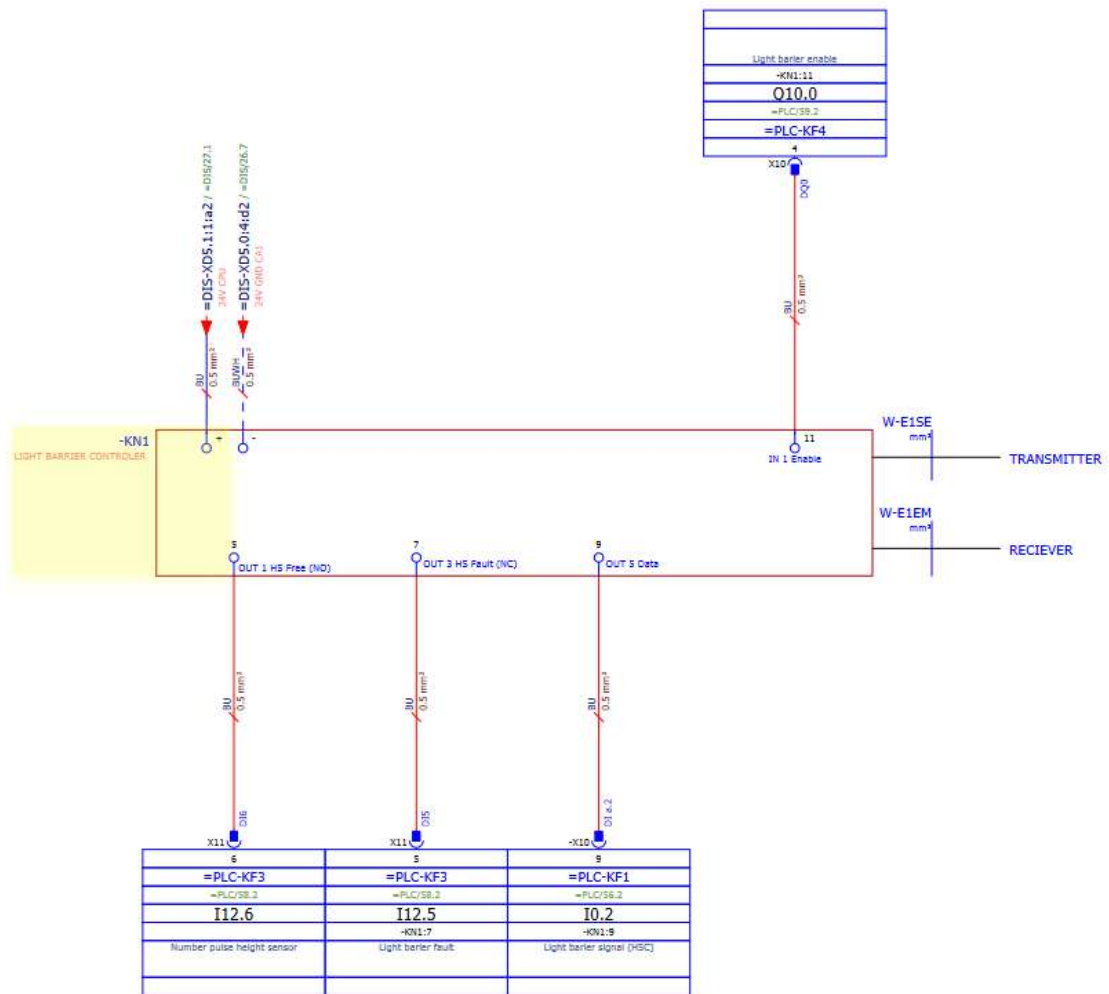


Slika 3.10: Kontroler svjetlosne zavjese



Slika 3.11: Odašiljač svjetlosne zavjese

Na slici 3.12 prikazana je električna shema spajanja kontrolera svjetlosne zavjese u elektroormaru upravljačkog sustava. Istosmjerni napon iznosa 24 VDC dovodi se na kontakte 1(+) i 2(-). Kontakti 5, 7 i 9 su izlazi kontrolera svjetlosne zavjese spojeni na PLC ulaze kao što je prikazano u tablici 2.2. Na ulazni kontakt 11 spojen je PLC izlaz prema tablici 2.3. Odašiljač je na kontroler spojen preko konektora W-E1SE a prijamnik preko konektora W-E1EM.

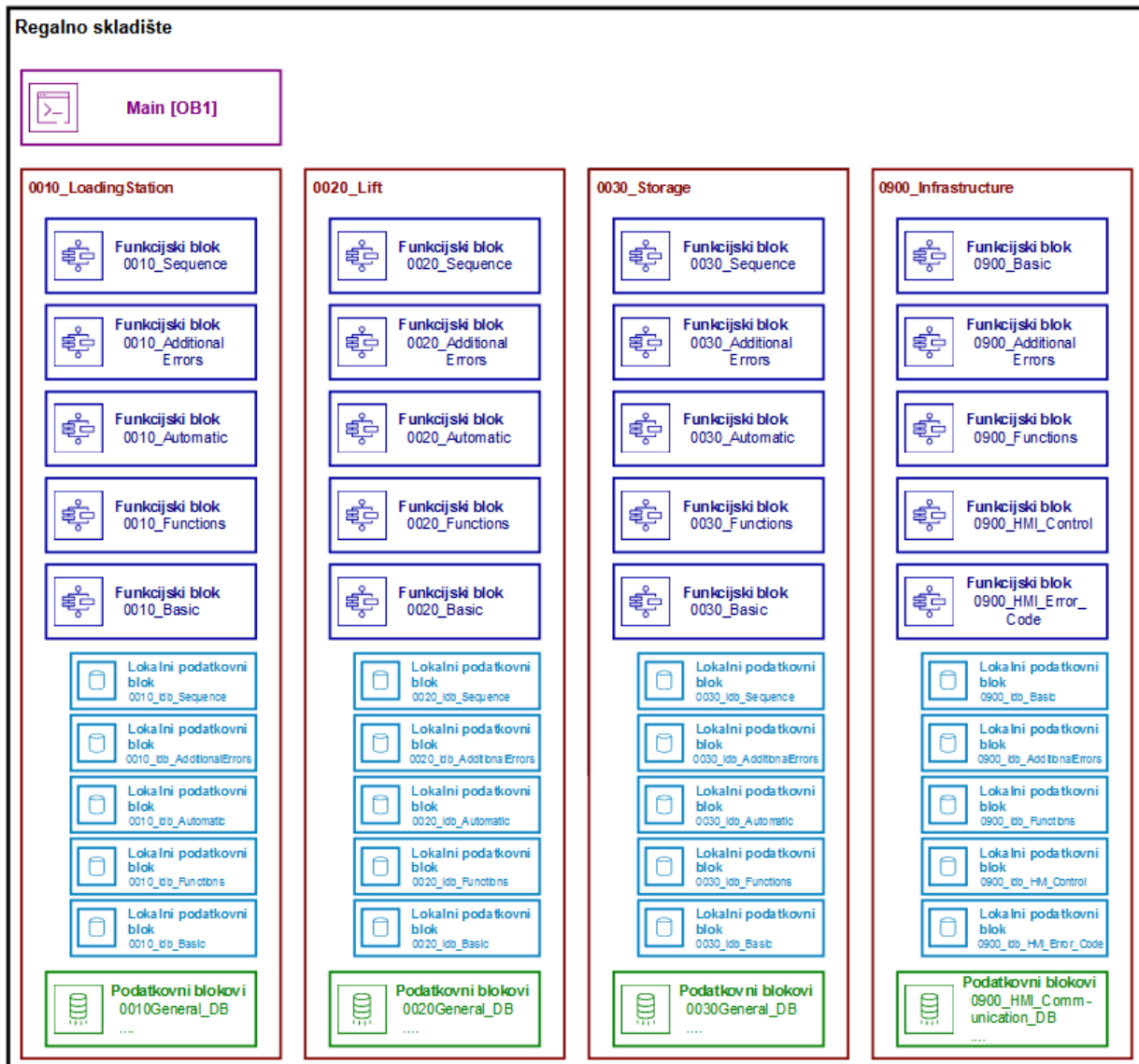


Slika 3.12: Električna shema spajanja kontrolera svjetlosne zavjese [6]

4 UPRAVLJAČKI PLC PROGRAM

4.1 Struktura upravljačkog programa

Upravljački PLC program regalnog skladišta napisan je u postojećem standardu razvijenom u tvrtki Strojotehnika d.o.o. Standard definira strukturu upravljačkog programa te sadrži funkcijske i podatkovne blokove zadužene za infrastrukturu programa, za sigurnosne funkcije, za standardne kupovne komponente poput senzora i pneumatskih cilindara te za definiranje koraka sekvenci. Struktura upravljačkog programa prikazana je na dijagramu na slici 4.1.



Slika 4.1: Struktura upravljačkog programa

Sustav je u svrhu organizacije i lakše čitljivosti programskog koda podijeljen na tri fizikalne i logičke cjeline ili stanice:

- Stanica za unos robe - *0010>LoadingStation*
- Dizalo - *0020>Lift*
- Skladišni prostor - *0030>Storage*

U programskom jeziku FBD, funkcijski blok (FB) je organizacijska struktura koja sadrži dijelove programskog koda koji su na neki način povezani i za koje se očekuje da će se koristiti više puta. Da bi se funkcijski blok izvršio, mora biti pozvan jednom ili više puta negdje u upravljačkom programu. Svakom funkcijskom bloku mogu se definirati ulazi i izlazi pomoću kojih se ostvaruje interakcija s drugim funkcijskim blokovima ili varijablama. Prema standardu u kojem je napisan upravljački program, svaka stanica ima pet glavih funkcijskih blokova:

- *_Basic*
- *_Functions*
- *_Automatic*
- *_Sequence*
- *_Additional_Errors*

FB *_Functions* svake stanice poziva se u glavnom organizacijskom bloku *Main*. Unutar ovog FB-a pozivaju se ostali glavni FB-ovi te se u njega dodaje programski kod kojim se definira logika upravljanja elementima stanice. U FB-u *_Basic* definirani su osnovni elementi stanice: broj stanice, referentna pozicija, sigurnost, resetiranje grešaka i pokretanje automatskog režima. Unutar FB-a *_Sequence* definiraju se koraci sekvenci u automatskom režimu rada, a FB *_Automatic* kontrolira pokretanje i zaustavljanje sekvenci te ponašanje sekvenci tijekom eventualnih grešaka. FB *_Additional_Errors* služi za definiranje grešaka na razini stanice.

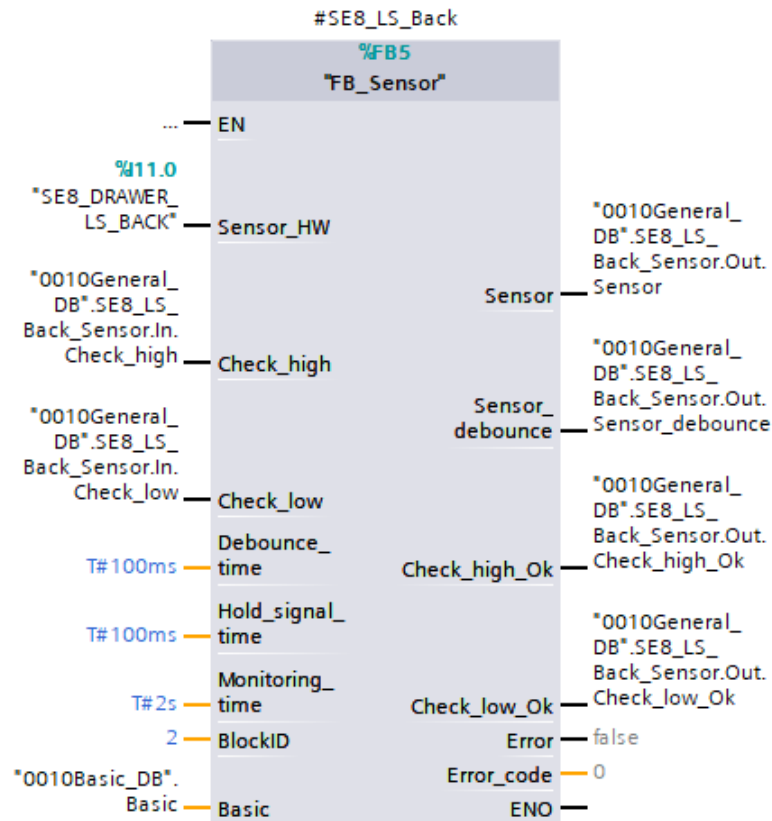
Za razliku od funkcija (FC), svaki FB ima svoj lokalni podatkovni blok (*Instance Data Block*, IDB) unutar kojeg se lokalno spremaju sve varijable korištene unutar tog FB-a i FB-ova pozvanih u njemu. Varijable spremljene u lokalni podatkovni blok čitati i pisati može samo FB kojem taj IDB pripada. Ako se nekoj varijabli pristupa iz više mjesta u upravljačkom programu, ona se sprema u podatkovni blok (*Data Block*, DB). Ovim varijablama može se pristupiti u svakom bloku. Za svaku stanicu definirano je pet DB-ova:

- *Basic_DB* - varijable FB-a *_Basic*;
- *Additional_Errors_DB* - varijable FB-a *_Additional_Errors*;
- *Parameters_DB* - vrijednosti koje korisnik može promijeniti u slučaju promjene na sustavu, ne mijenja ih upravljački program;
- *Register_DB* - praćenje vrijednosti poput lokacija i količina obradaka;
- *General_DB* - sve ostale varijable

Stanica *0900_Infrastructure* koristi se za definiranje funkcionalnosti na razini cijelog sustava. FB *0900_Basic* objedinjuje izlaze svih ostalih *Basic* FB-ova i na temelju njih ažurira stanje sustava. U ovom FB-u definirane su tri klase grešaka: klasa 1 odnosi se na greške na razini cijelog sustava, klasa 2 na greške na razini stanice a klasa 3 su upozorenja. Stanica *0900_Infrastructure* sadrži i FB-ove *HMI_Control* i *HMI_Error_Code* koji sadrže programski kod koji upravlja slanjem poruka o statusu skladišta, omogućavanjem pozivanja funkcija sa zasolna HMI-a i slanjem poruka o greškama s PLC-a na HMI.

4.2 Upravljačka logika stanice za unos robe

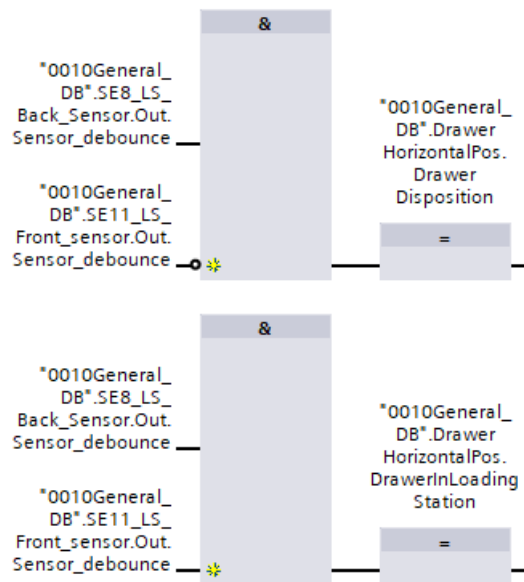
Stanica za unos robe sadrži svjetlosnu zavjesu i tri induktivna senzora: dva senzora za poziciju ladice u stanici za unos i senzor za poziciju dizala na visini stanice za unos. U funkcijskom bloku *0010_Functions* pozvani su standardni FB-ovi za upravljanje prisutnim induktivnim sensorima, definirana je logika za praćenje pozicije ladice u stanici za utovar te je napravljen FB za upravljanje svjetlosnom zavjesom. Slika 4.2 prikazuje funkcijski blok stražnjeg senzora za poziciju ladice.



Slika 4.2: Funkcijski blok senzora

Na ulaz *Sensor_HW* spaja se PLC oznaka (PLC tag) koja predstavlja fizički ulaz na kojem je taj senzor pripojen na PLC. Ostali ulazi služe za testiranje senzora. Kada se u programu koristi vrijednost koju daje taj senzor, koriste se izlazne vrijednosti *Sensor* ili *Sensor_debounce* ovog FB-a. Te varijable spremljene su u podatkovni blok *0010_GeneralDB*. Vrijednost varijable *Sensor* uvijek je jednaka samom PLC ulazu (u ovom slučaju *%I11.0*), a *Sensor_debounce* poprima stvarnu vrijednost senzora nakon 100 ms (ulazi *Debounce_time* i *Hold_Signal_Time*). Svrha izlaza *Sensor_debounce* je filtriranje slučajnih i neželjenih aktiviranja senzora. Na isti način inicijalizirani su svi senzori i graični prekidači u sustavu, razlika je jedino u varijablama koje se spajaju na ulaze i izlaze standardnog FB-a.

Ovaj senzor, u kombinaciji s prednjim sensorom za poziciju ladice, daje upravljačkom programu informaciju o poziciji ladice tijekom njenog ulaženja, boravka i izlaženja iz stanice za unos robe. Ta informacija je bitna jer se dizalo ne smije početi gibati ako ladica cijelom svojom površinom nije u jednoj ili drugoj stanici. Isto tako, ako je ladica u stanici za utovar, ne smije se dozvoliti vađenje neke druge ladice iz skladišta jer će doći do kolizije. Na slici 4.3 prikazan je programski kod kojim se definira pozicija ladice s obzirom na ta dva senzora.

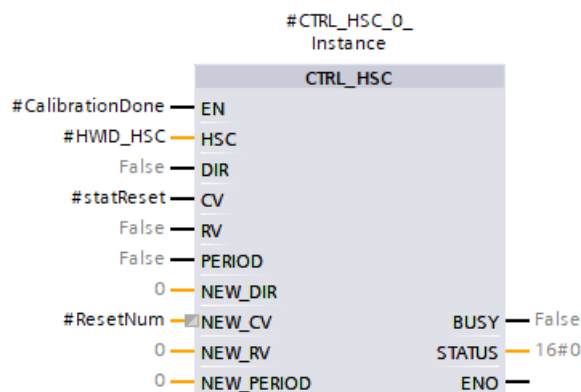


Slika 4.3: Definicija pozicije ladice u stanici za utovar

Ako je stražnji senzor pozicije aktivan, a prednji nije, smatra se da je ladica u nedopuštenoj poziciji, odnosno ladica je djelomično na dizalu a djelomično u stanici za utovar. U toj situaciji bit *DrawerDisposition* postavlja se na 1. Ako su oba senzora aktivna, ladica je u stanici za utovar i bit *DrawerInLoadingStation* postavlja se na 1.

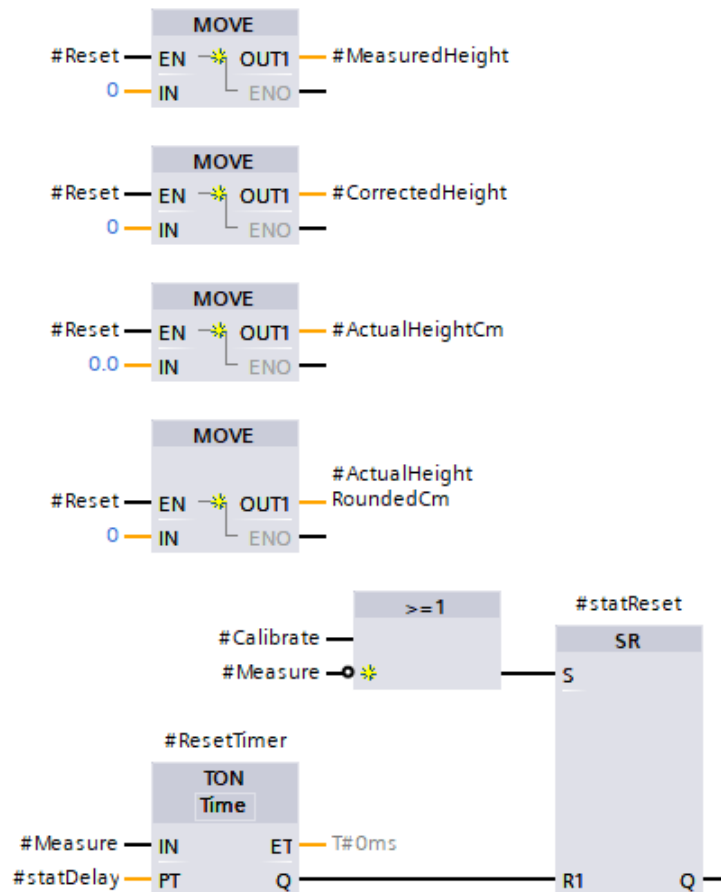
U svrhu upravljanja svjetlosnom zavjesom, izrađen je funkcijski blok koji objedinjuje sve funkcije svjetlosne zavjese definirane ranije u poglavlju 3.6 *Kontroler svjetlosne zavjese DUOmetric SCAN 1.11-min.* Programski kod ovog FB-a podijeljen je na osam dijelova.

U prvom dijelu pozvan je predefimirani funkcijski blok CTRL_HSC koji služi za parametrisiranje visokobrzinskog brojača. Prikazan je na slici 4.4. Na ulaz EN dodaje se signal *#CalibrationDone* kako bi se osiguralo da mjerenje nije moguće prije nego je provedena kalibracija. Na ulaz HSC dodana je varijabla *#HWID_HSC* koja sadrži hardverski identifikator visokobrzinskog brojača. To je interna vrijednost definirana u postavkama CPU modula. Bit *#statReset* na ulazu CV naređuje CTRL_HSC funkcijskom bloku da od visokobrzinskog brojača zatraži novu vrijednost. Na ulaz NEW_CV dodana je varijabla koja određuje na koju se vrijednost postavlja brojač nakon resetiranja.



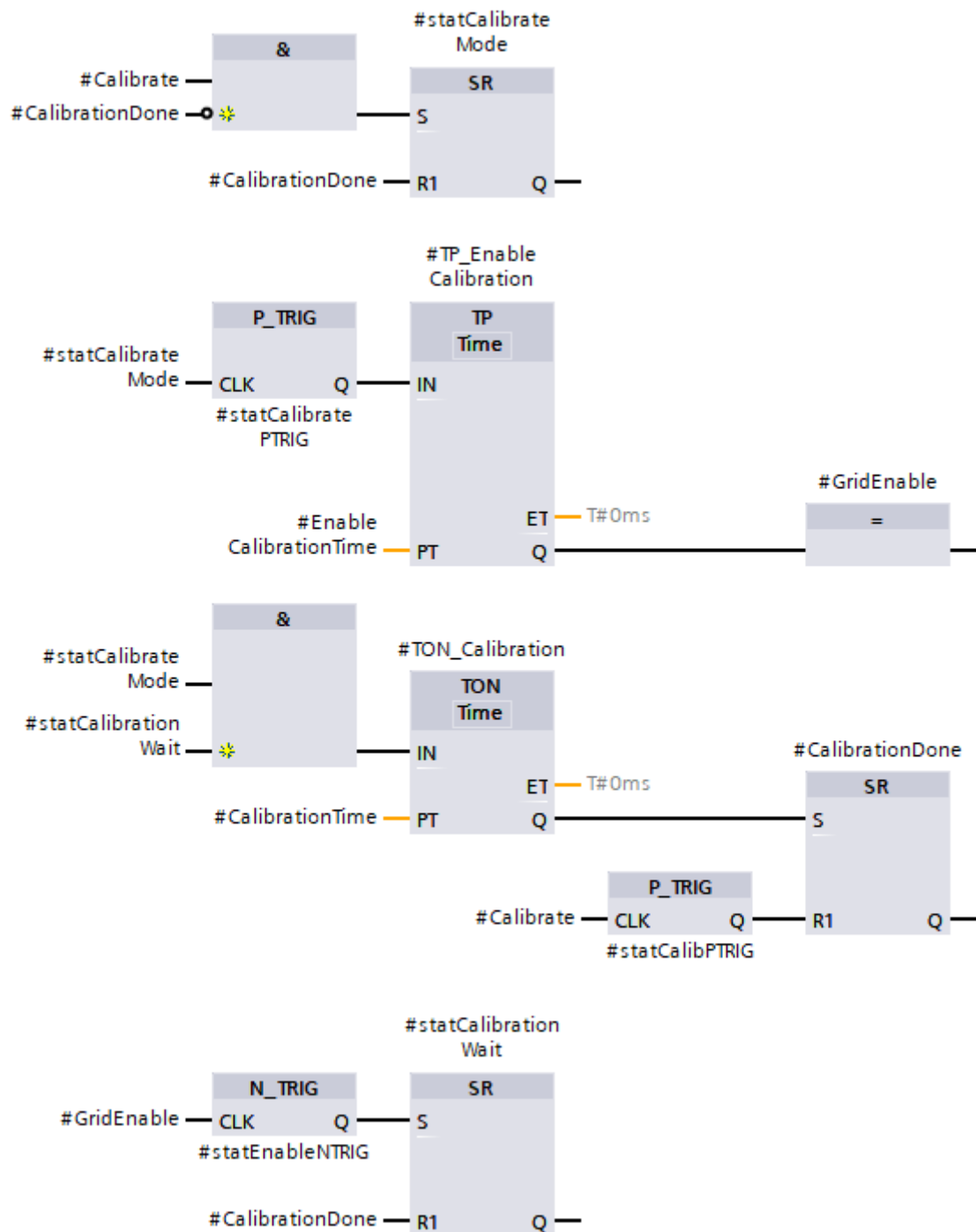
Slika 4.4: FB svjetlosne zavjese: upravljanje visokobrzinskim brojačem

Drugi dio služi za postavljanje prethodno izmjerenih vrijednosti u nulu u slučaju novog mjerenja. Programski kod prikazan je na slici 4.5. Aktivacijom bita *#Reset*, naredbom MOVE vrijednosti varijabli u koje su spremljene vrijednosti izmjerene, korigirane i preračunate visine postavljaju se na nula. Zadnja linija dodana je u svrhu filtriranja vrijednosti koju kontroler svjetlosne zavjese iz neutvrđenih razloga šalje PLC-u kada nije uključen mjerni način.



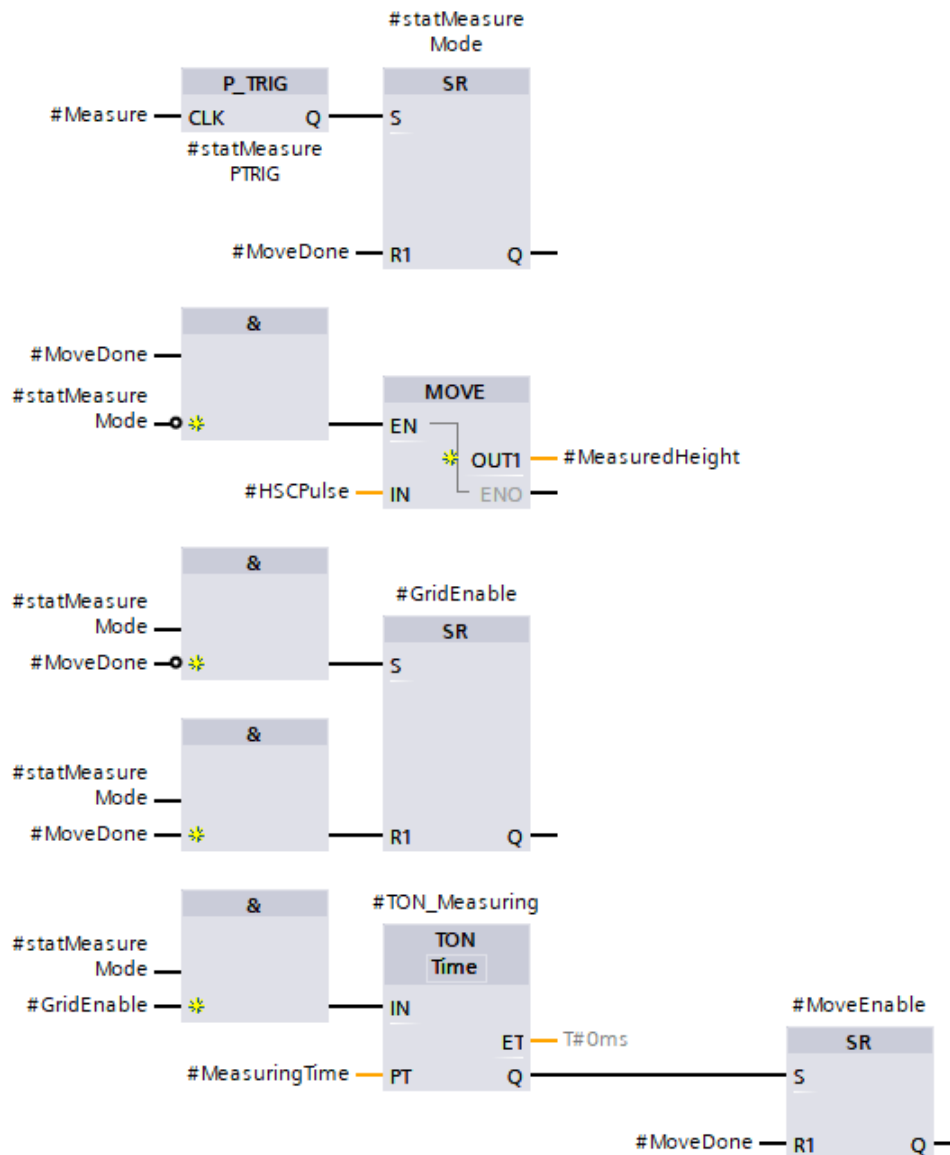
Slika 4.5: FB svjetlosne zavjese: resetiranje vrijednosti

Treći dio, prikazan na slici 4.6, određuje ponašanje sustava tijekom kalibracije svjetlosne zavjese. Kada se bit *#Calibrate* postavi na vrijednost 1, i ako kalibracija još nije provedena, aktivira se statička varijabla *#statCalibrateMode*, čija je svrha da unutar programa označava da je u tijeku kalibracija. Vrijednost tog bita se vraća na 0 kada je kalibracija gotova. Kada se aktivira *#statCalibrateMode*, vremenski brojač vrste TP, koji se aktivira kada mu je na ulazu IN vrijednost 1 i ostaje aktivan za vrijeme definirano na ulazu PT, kontroleru svjetlosne zavjese šalje aktivacijski signal u trajanju od 100 ms. Nakon što se kalibracija pokrene, vremenski brojač vrste TON, koji se aktivira kada istekne vrijeme definirano na ulazu PT nakon aktiviranja ulaza IN, odbrojava vrijeme u iznosu od 1,3 s prije nego što se bit *#CalibrationDone* postavi na 1. Bit *#CalibrationDone* se postavlja na 0 kada se sljedeći puta pokrene kalibracija. U posljednjoj liniji ovog dijela postavlja se bit *#statCalibrationWait* koji u prethodnoj liniji služi kao potvrda da je uključen aktivacijski signal.



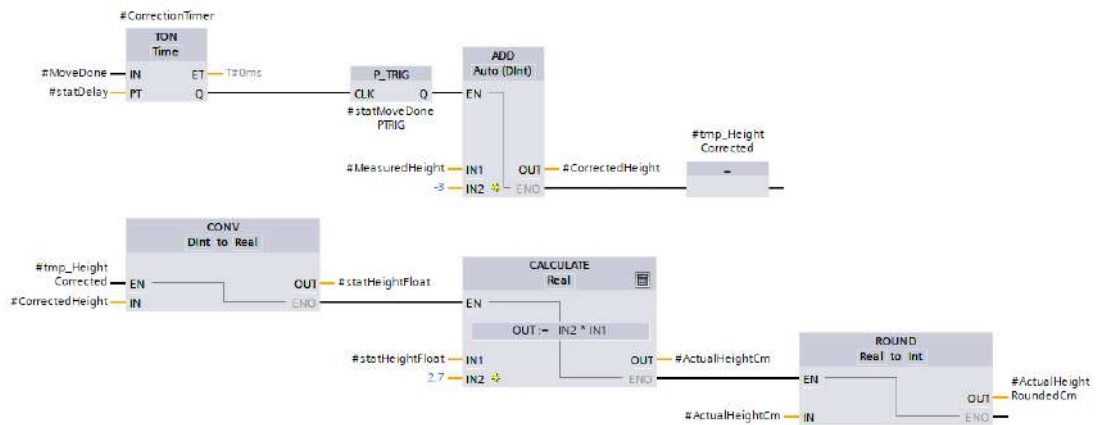
Slika 4.6: FB svjetlosne zavjese: kalibracija

Četvrti dio, prikazan na slici 4.7, definira mjerni način rada svjetlosne zavjese. Slično kao kod kalibracije, vanjskim postavljanjem bita *#Measure* na 1 aktivira se bit *#statMeasureMode*. Isti bit se postavlja na 0 kada je izvana pomoću bita *#MoveDone* primljena potvrda da je objekt prošao kroz svjetlosnu zavjesu. Ako je objekt prošao kroz zavjesu i mjerenje je gotovo, naredbom MOVE vrijednost visokobrzinskog brojača *#HSCPulse* zapisuje se u varijablu *#MeasuredHeight*. Ako je u tijeku mjerenje i objekt se trenutno kreće kroz svjetlosnu zavjesu, kontroleru svjetlosne zavjese šalje se aktivacijski signal *#GridEnable*. Aktivacijski signal se prestaje slati kada je mjerenje gotovo. U zadnjoj liniji ove mreže vremenski brojač TON onemogućuje kretanje kroz zavjesu prvih 150 ms nakon pokretanja mjernog načina kako bi se osiguralo da je mjerenje u tijeku.



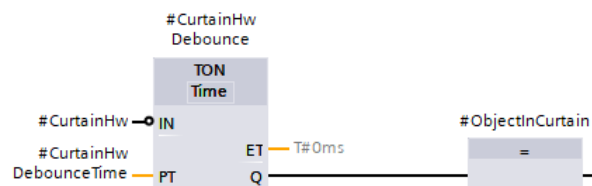
Slika 4.7: FB svjetlosne zavjese: mjerni način rada

Peti dio služi za korekciju izmjerene vrijednosti i pretvaranje iste u vrijednost izraženu u centimetrima. Prikazan je na slici 4.8. Nakon što je mjerenje gotovo i objekt je prošao kroz zavjesu, vremenski brojač TON čeka 200 ms kako bi se osiguralo da je izmjerena vrijednost zapisana u odgovarajuću varijablu. Najprije se provodi korekcija vrijednosti tako da se od izmjerene vrijednosti naredbom ADD oduzme pomak iznosa 3 definiran u dokumentaciji kontrolera svjetlosne zavjese. Korigirana vrijednost pretvara se iz tipa podatka DInt (cijeli broj dvostruke preciznosti) u tip Real (realni broj) kako bi ju se moglo množiti drugim realnim brojevima. U sljedećem koraku provodi se pretvaranje vrijednosti HSC ulaza u smislenu vrijednost visine mjerenog predmeta pomoću relacije utvrđene tijekom testiranja svjetlosne zavjese. Naredbom CALCULATE računa se stvarna visina. Stvarna visina se zbog jednostavnosti na kraju zaokružuje na cijeli broj, pretvara u tip podatka Int i sprema u varijablu *#ActualHeightRoundedCm*.



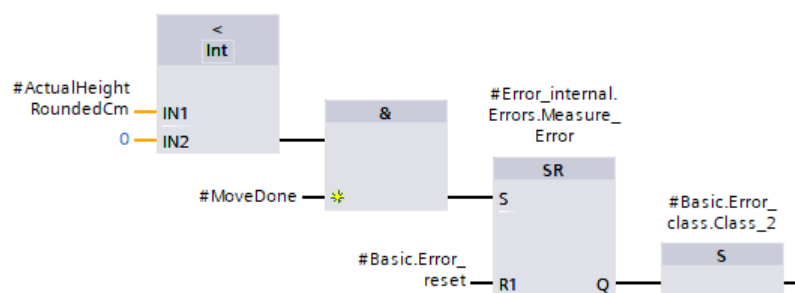
Slika 4.8: FB svjetlosne zavjese: korekcija izmjerene vrijednosti

Šesti dio u svrhu sigurnosnih funkcija u ostalim dijelovima upravljačkog sustava provjerava nalazi li se trenutno u radnom području svjetlosne zavjese neki objekt. Dio je prikazan na slici 4.9. Ako bit *#CurtainHw*, koji ima vrijednost 1 ako svjetlosna zavjesa nije prekinuta, poprmi vrijednost 0 i zadrži se određeno vrijeme (*#CurtainHwDebounceTime*), vrijednost bita *#ObjectInCurtain* postavlja se na 1.

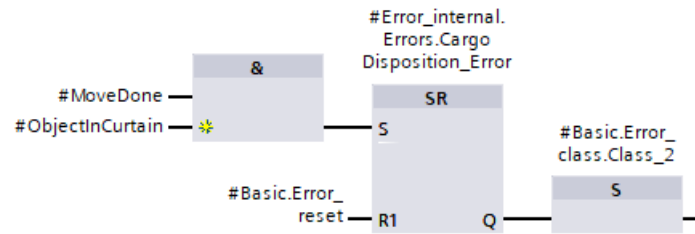


Slika 4.9: FB svjetlosne zavjese: Sigurnosna funkcija

Sljedeća dva dijela, prikazana na slikama 4.10 i 4.11, definiraju greške koje se mogu pojaviti tijekom rada svjetlosne zavjese. Ovi dijelovi su potrebni da dio upravljačkog programa za upravljanje greškama može poduzeti potrebne mjere u ostatku upravljačkog programa. Do greške *Cargo disposition error* dolazi ako je sustav javio da je provlačenje objekta kroz svjetlosnu zavjesu gotovo, a svjetlosna zavjesa je još uvijek prekinuta. Može se zaključiti da je objekt u nepravilnoj poziciji te se daljnji rad sustava mora onemogućiti. Greška *Measure error* javlja se ako je izmjerena vrijednost manja od nule. U tom slučaju nešto očito nije u redu s radom svjetlosne zavjese i spremanje tereta u skladište ne smije se nastaviti, jer nije moguće izračunati prostor potreban za njegovo spremanje.

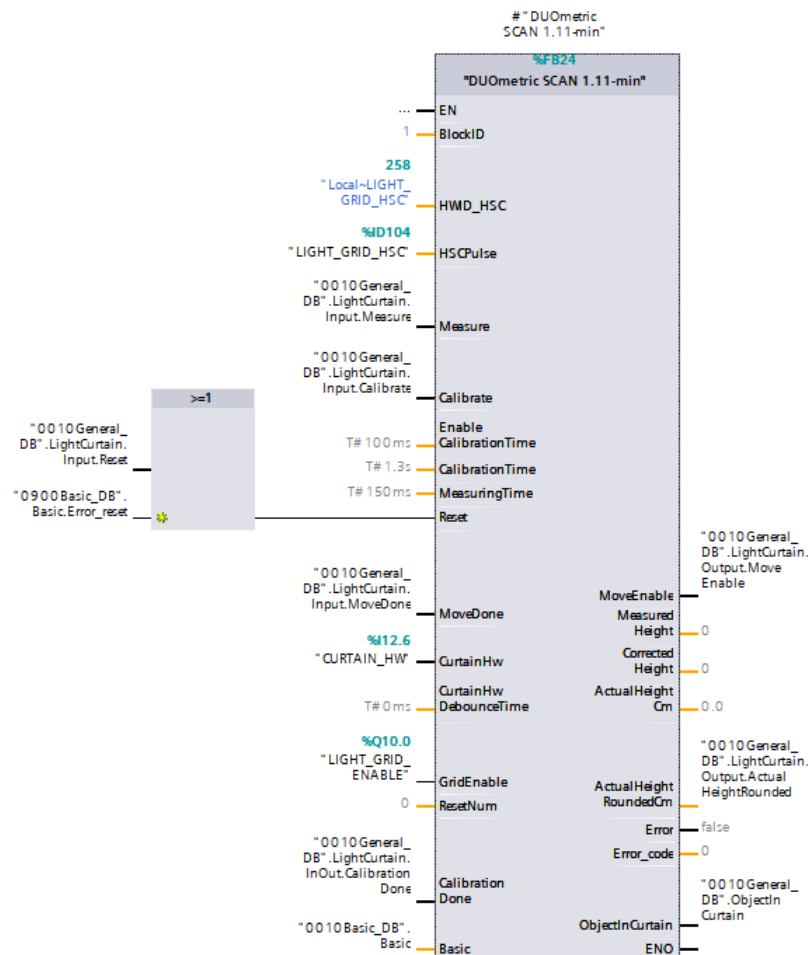


Slika 4.10: FB svjetlosne zavjese: Greška - Teret u nedopuštenoj poziciji



Slika 4.11: FB svjetlosne zavjese: Greška - Izmjerena vrijednost manja od nule

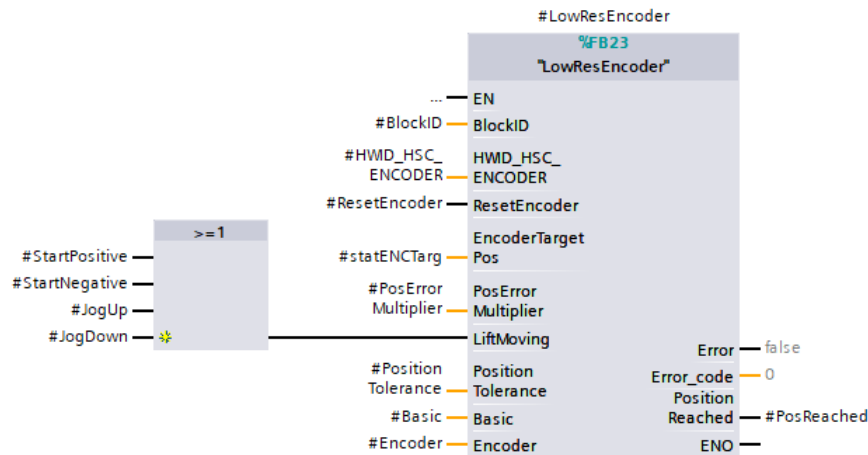
Slika 4.12 prikazuje kako je FB svjetlosne zavjese pozvan u FB-u *0010_Functions*. Na ulaze FB-a dodane su PLC oznake koje predstavljaju hardver svjetlosne zavjese i varijable kojima se pokreću funkcije mjerenja, kalibracije i resetiranja izmjerenih vrijednosti. Izmjerene vrijednosti se resetiraju kada je aktivan bit *Reset* ili kada operater pritisne tipkalo za resetiranje grešaka. Izlaz *MoveEnable* je aktivan samo kada je u tijeku mjerenje i koristi se kao dozvola za pomicanje ladice iz stanice za utovar na dizalo. Zaokružena izmjerena vrijednost dolazi na izlaz *ActualHeightRoundedCm* i koristi se za izračun broja polica potrebnih za spremanje ladice.



Slika 4.12: Funkcijski blok svjetlosne zavjese

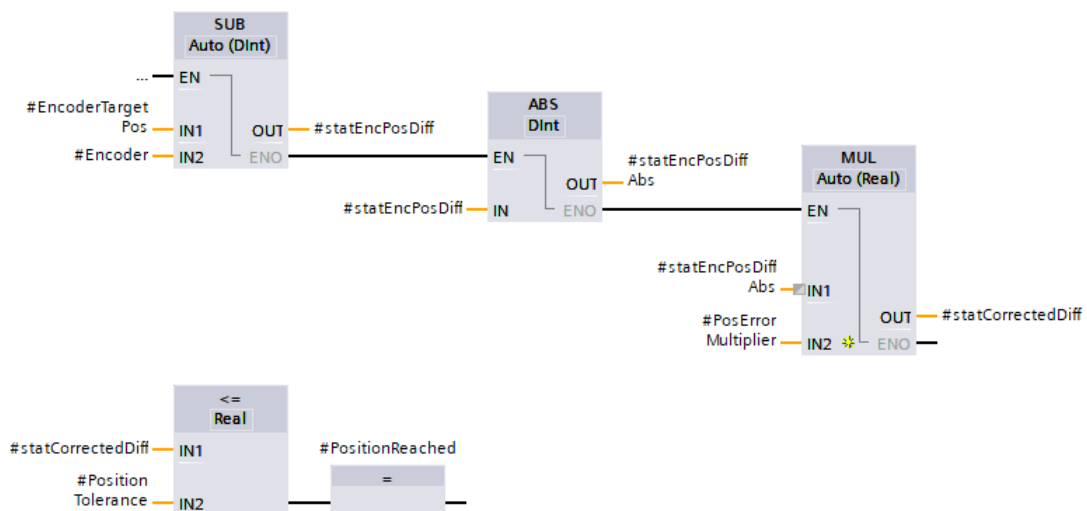
4.3 Upravljačka logika stanice dizala

Stanica dizalo sastoji se od vertikalne i horizontalne osi te svih senzora smještenih na platformi dizala. Vertikalna os ima inkrementalni enkoder kojim se zatvara regulacijska petlja po poziciji. Za dohvaćanje i interpretaciju vrijednosti inkrementalnog enkodera napravljen je funkcijski blok prikazan na slici 4.13.



Slika 4.13: Funkcijski blok za upravljanje inkrementalnim enkoderom

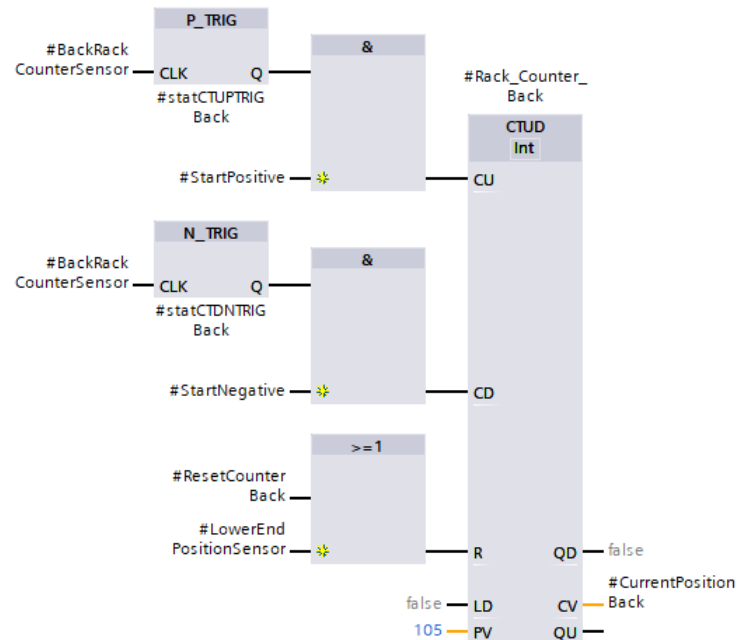
Uloga ovog FB-a je određivanje je li vertikalna os postigla traženu poziciju na temelju vrijednosti enkodera. Enkoder je, slično svjetlosnoj zavjesi, spojen na HSC ulaz. FB kao ulazne vrijednosti uzima vrijednost enkodera, signal da je dizalo u pokretu, toleranciju postizanja pozicije i signal za resetiranje vrijednosti HSC ulaza. Izlazna vrijednost je potvrda da je pozicija postignuta koja se određuje kodom prikazanim na slici 4.14.



Slika 4.14: Provjera postizanja tražene pozicije

Najprije se određuje apsolutna vrijednost razlike zadane pozicije i trenutne pozicije, odnosno udaljenost dizala od cilja. Ta se razlika množi faktorom *#PosErrorMultiplier* koji ovisi o rezoluciji enkodera i u ovom slučaju iznosi 1. Ako je dobivena razlika manja ili jednaka određenoj maksimalnoj dozvoljenoj razlici *#PositionTolerance*, smatra se da je dizalo postiglo zadanu poziciju i bit

#PosReached na izlazu FB-a postaje aktivan. FB za upravljanje enkoderom pozvan je unutar FB za kontrolu pozicije vertikalne osi. Zadaća FB-a za kontrolu pozicije je određivanje smjera kretanja dizala, referenciranje vertikalne osi i praćenje pozicije dizala s obzirom na redni broj police unutar skladišnog prostora. Brojanje polica jer se na temelju tog broja indeksira struktura kojom se u memoriju PLC-a spremaju informacije o zauzetosti polica. Brojač polica prikazan je na slici 4.15.

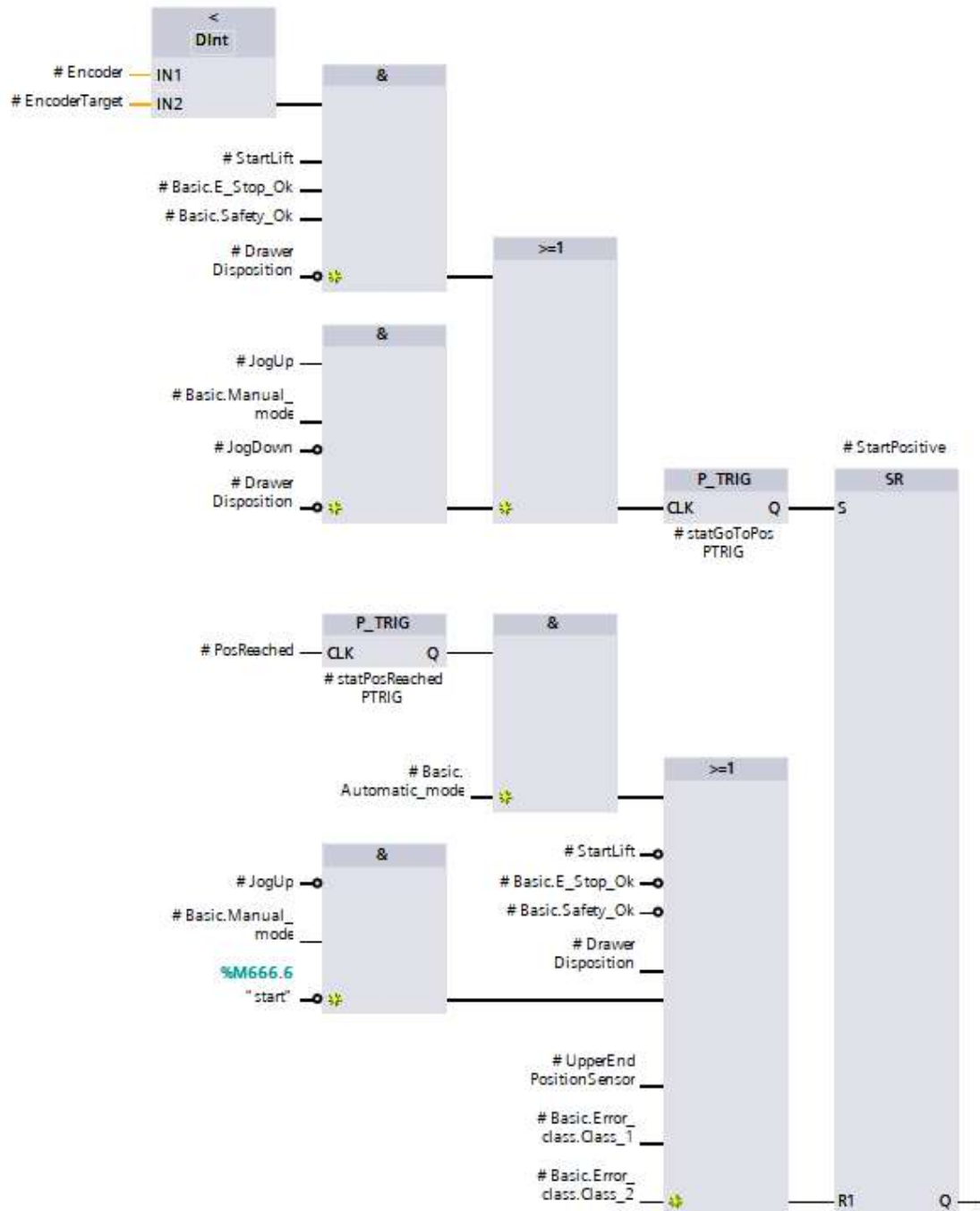


Slika 4.15: Brojač polica

Brojanje se odrađuje pomoću naredbe CTUD koja svoju izlaznu vrijenost povećava za 1 kada je aktivan ulaz CU i smanjuje za 1 kada je aktivan ulaz CD. Postavljeni su uvjeti da se brojač povećava za 1 na svaki rastući brid signala senzora za brojanje polica kada se dizalo kreće prema gore, a smanjuje za 1 na svakom padajućem bridu senzora kada se dizalo kreće u negativnom smjeru. Trenutno stanje se sprema u varijablu *#CurrentPositionBack* koja je jedna od izlaznih vrijenosti FB-a. Na isti način broje se i police u prednjem stupcu skladišta. Ako ova varijabla poprimi vrijednost veću od ukupnog broja polica ili manju od nule, zaključuje se da nešto nije u redu s brojačem i javlja se greška *Couter Error*.

Funkcijski blok za kontrolu pozicije određuje i smjer gibanja vertikalne osi. Na slici 4.16 prikazano je na koji se način određuje da dizalo treba krenuti u pozitivnom smjeru. Bit *#StartPositive* je jedna od izlaznih vrijednosti FB-a za kontrolu pozicije. Kada je taj bit aktivan, FB za upravljanje VFD-om dobiva signal da VFD pokrene motor vertikalne osi u pozitivnom smjeru, što je prikazano kasnije. Prikazanim kodom definirano je da vertikalna os treba krenuti u pozitivnom smjeru kada je aktivan bit *#StartLift* a trenutna enkoderska pozicija je manja od tražene pozicije ili kada je u ručnom načinu rada aktivan bit *#JogUp*. Bit *#StartLift* poziva se u sekvencama automatskog režima rada, a *JogUp* pritiskom na tipku za ručno pokretanje dizala na HMI-u. *#StartPositive* se resetira, što rezultira zaustavljanjem dizala, ako FB enkodera javi da je pozicija postignuta, ako operater pusti tipku za ručno pokretanje ili ako padnu uvjeti sigurnosti.

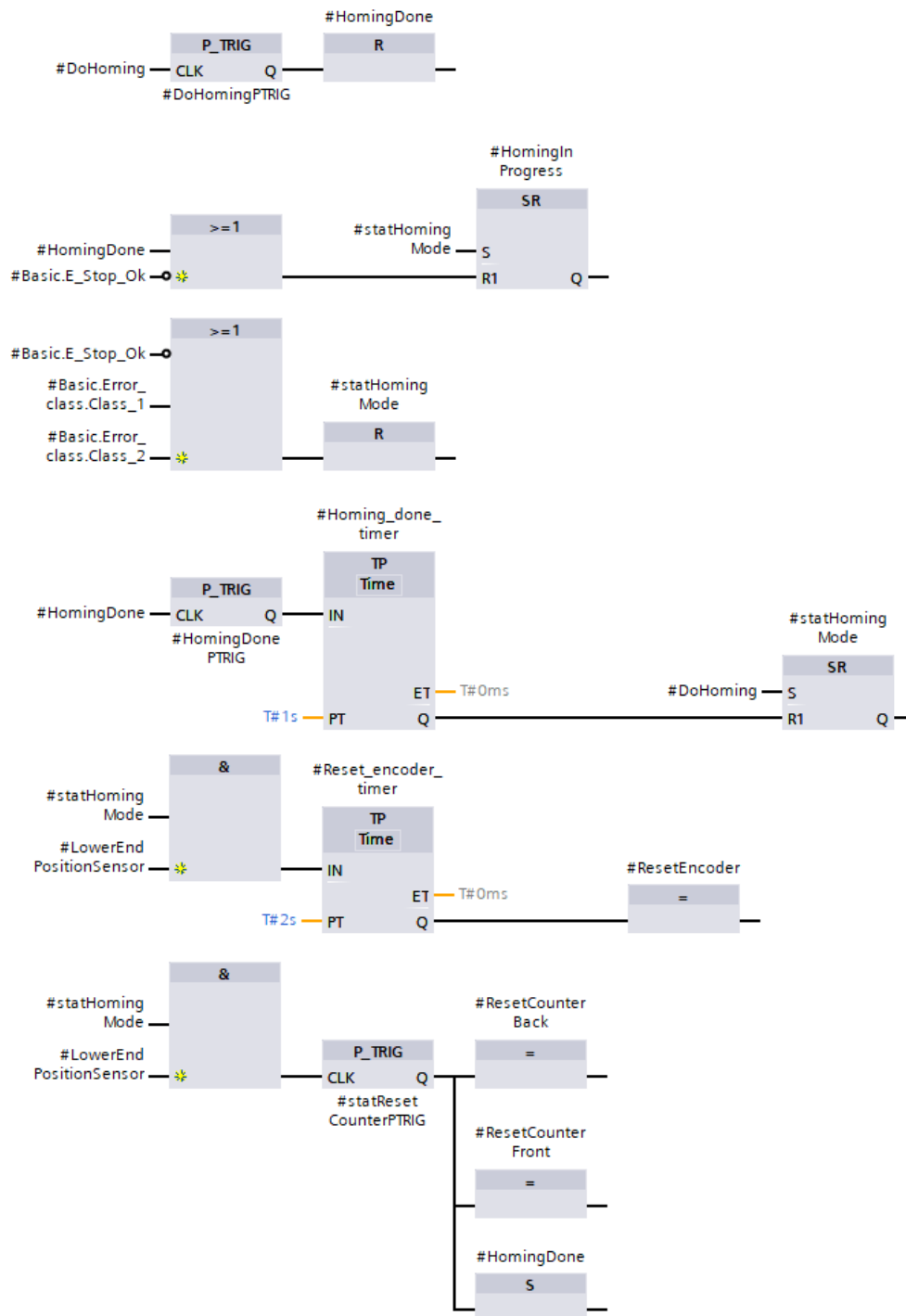
Svrha naredbe P_TRIG koja dolazi između uvjeta za pokretanje i samog pokretanja je da ulaz S naredbe SR bude aktivan samo u trenutku rastućeg brida signala na koji su spojeni definirani uvjeti. Na taj način osigurava se da se bit *#StartPositive* neće nepoželjno aktivirati nakon resetiranja eventualne greške. Na isti način određeni su i uvjeti za pokretanje dizala u negativnom smjeru.



Slika 4.16: Određivanje smjera kretanja vertikalne osi

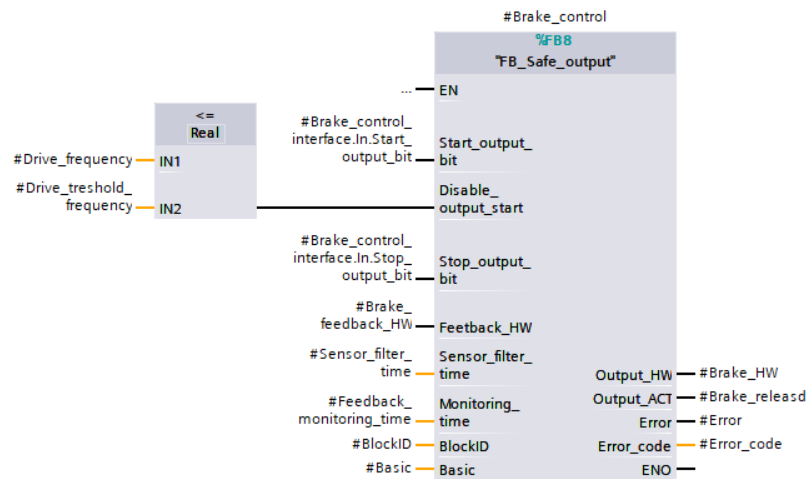
Na slici 4.17 prikazan je dio funkcijskog bloka za kontrolu pozicije kojim se odrađuje referenciranje vertikalne osi. Funkcija referenciranja poziva se dodiranjem tipke *Homing* na glavnom zaslonu HMI-a. Pokretanjem funkcije referenciranja aktivira se bit *#DoHoming* na ulazu FB-a *Position-Control*. *#DoHoming* pomoću naredbe SR aktivira bit *#statHomingMode* koji se koristi kao jedan od uvjeta za pokretanje elektromotora vertikalne osi u negativnom smjeru. Također se aktivira bit *#HomingInProgress* koji je spojen na izlaz FB-a i služi za signaliziranje drugim dijelovima upravljačkog programa da je u tijeku referenciranje. Dizalo se počinje gibati prema dolje sve dok ne aktivira senzor donje referentne pozicije. Kada dizalo dosegne senzor, vrijednosti ulaza na kojem je enkoder te prednjeg i stražnjeg brojača polica postavljaju se na nulu. Aktivira se bit *#HomingDone* koji je također izlazna vrijednost FB-a i služi za javljanje sustavu da je os referencirana.

Kada je referenciranje gotovo, resetiraju se bitovi `#statHomingMode` i `#HomingInProgress`. Dizalo se zaustavlja. `#HomingDone` je jedan od uvjeta koji moraju biti ispunjeni kako bi sustav mogao ući u automatski režim rada.



Slika 4.17: Referenciranje vertikalne osi

Elektromehanička kočnica prigona vertikalne osi kritični je element sustava. Mora se otpustiti svaki puta kada su aktivni bitovi `#StartPositive` i `#StartNegative` na izlazu FB-a `PositionControl` te zategnuti svaki puta kada se oni deaktiviraju, odnosno kada se prigon vertikalne osi zaustavi. Osobito je važno da se kočnica ne otpusti kada elektromotor nije aktivan jer bi to rezultiralo nekontroliranim padom dizala. Upravljački program kočnicom upravlja pomoću standardnog funkcijskog bloka za upravljanje relejima prikazanim na slici 4.18.

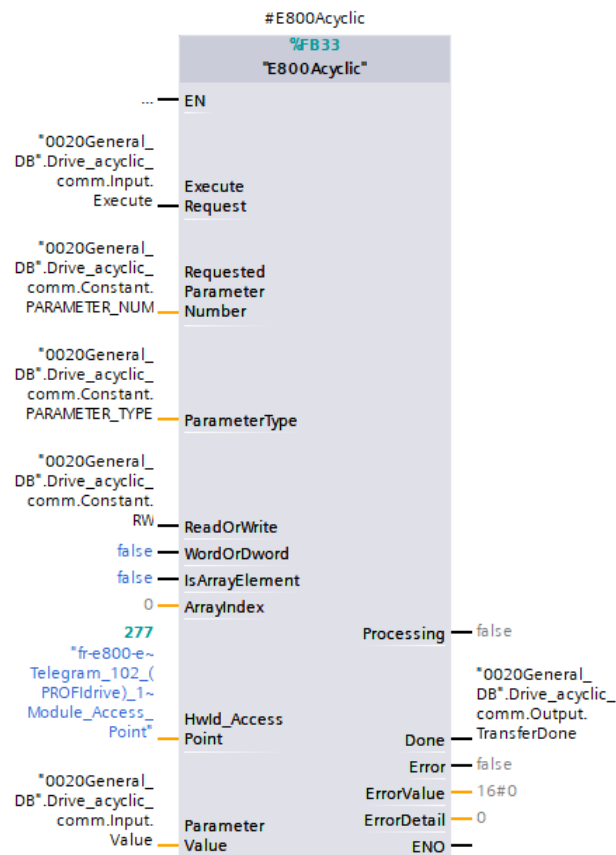


Slika 4.18: FB za upravljanje kočnicom

Kočnica se otpušta kada je aktivan ulaz *Start_output_bit* uz uvjet da je ulaz *Disable_output_bit* neaktivan. *Start_output_bit* je aktivan kada su aktivni *#StartPositive* ili *#StartNegative* i ako je uklopljen kontaktor prigona vertikalne osi. Na ulazu *Disable_output_bit* postavljen je uvjet da se kočnica otpušta tek kada frekvencija na izlazu VFD-a dostigne prag *#Drive_treshold_frequency*. Vrijednost tog praga utvrđena je tijekom testiranja sustava. Na ovaj način osigurano je da se kočnica ne može otpustiti ako motor nije pod naponom. Ako je aktivan ulaz *Stop_output_bit*, kočnica se zateže bez obzira na sve ostale ulaze. *Stop_output_bit* će se aktivirati ako dođe do bilo kakvog neočekivanog ili nekontroliranog gibanja dizala. Nekontrolirano gibanje otkriva se na temelju informacija koje sustav dobiva od kontrolera i VFD-a.

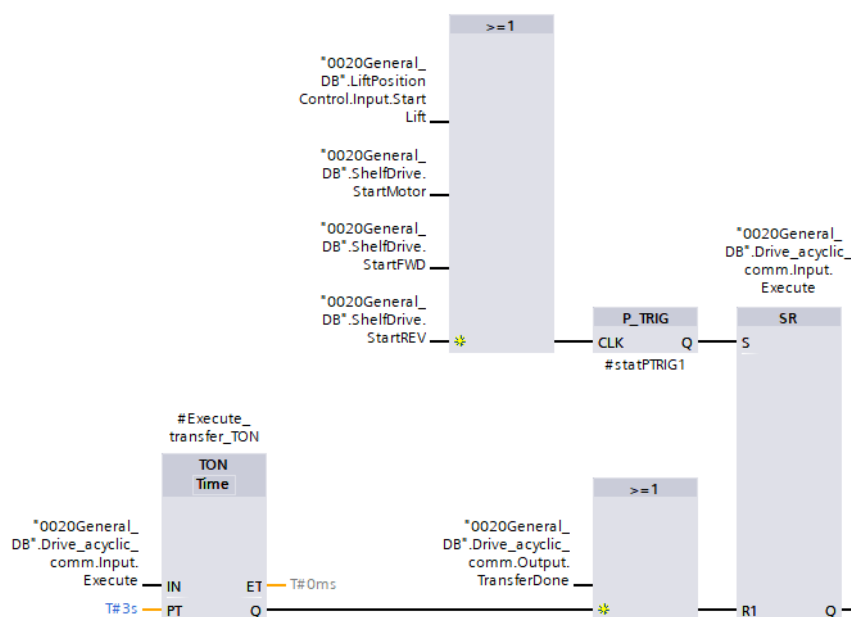
Korištenjem taktnog signala, svakih 400 ms računa se apsolutna vrijednost razlike između trenutne enkoderske pozicije i one od prije 400 ms. Ako je ta vrijednost veća od najveće dopuštene vrijednosti, utvrđene testiranjem, kočnica se mora zategnuti jer se dizalo giba brzinom većom od maksimalne što znači da dizalo pada. Ako je trenutna vrijednost enkodera veća od prethodne, dakle ako se dizalo giba u pozitivnom smjeru, a aktivan je bit *#StartNegative*, smatra se da je dizalo u nekontroliranom gibanju. Isto vrijedi i za obrnutu situaciju, dakle ako se dizalo giba u negativnom smjeru a aktivan je *#StartPositive*. U bilo kojoj od opisanih situacija aktivira se *Stop_output_bit* i kočnica se aktivira.

S obzirom na to da se za upravljanje prigonima obiju osi koristi jedan zajednički frekvencijski pretvarač (VFD), prije svakog pokretanja elektromotora PLC mora VFD-u prenijeti informaciju o tome koji elektromotor će se pokrenuti. To je potrebno zato jer elektromotori nisu jednake veličine što znači da su im svi parametri koje VFD koristi za upravljanje različiti. Prema [12], funkcije za upravljanje drugim motorom aktiviraju se kada se učita vrijednost 3 u jedan od parametara između Pr.178 i Pr.189 u memoriji VFD-a. Funkcije drugog motora se deaktiviraju ako se u isti parametar upiše vrijednost 0. Određeno je da se prvim motorom smatra prigon vertikalne osi, a drugim prigon horizontalne osi. Koristi se parametar Pr.185. Kada se VFD-om upravlja preko Profinet standarda, kao što je ovdje slučaj, vrijednost parametra mijenja se pomoću FB-a *E800Acyclic* prikazanog na slici 4.19. Ovaj FB dio je standardne biblioteke za upravljanje VFD-ovima serije E800 izrađene od strane proizvođača VFD-a. Na ulaze FB-a dodaje se broj parametra koji se mijenja, vrsta parametra, vrijednost koju je potrebno upisati i hardverski identifikator za pristup memoriji VFD-a. Upisivanje nove vrijednosti izvodi se svaki puta kada se aktivira bit na ulazu *ExecuteRequest*. Nakon što je upisivanje izvršeno, aktivira se izlaz *Done*.



Slika 4.19: FB za pisanje parametara VFD-a

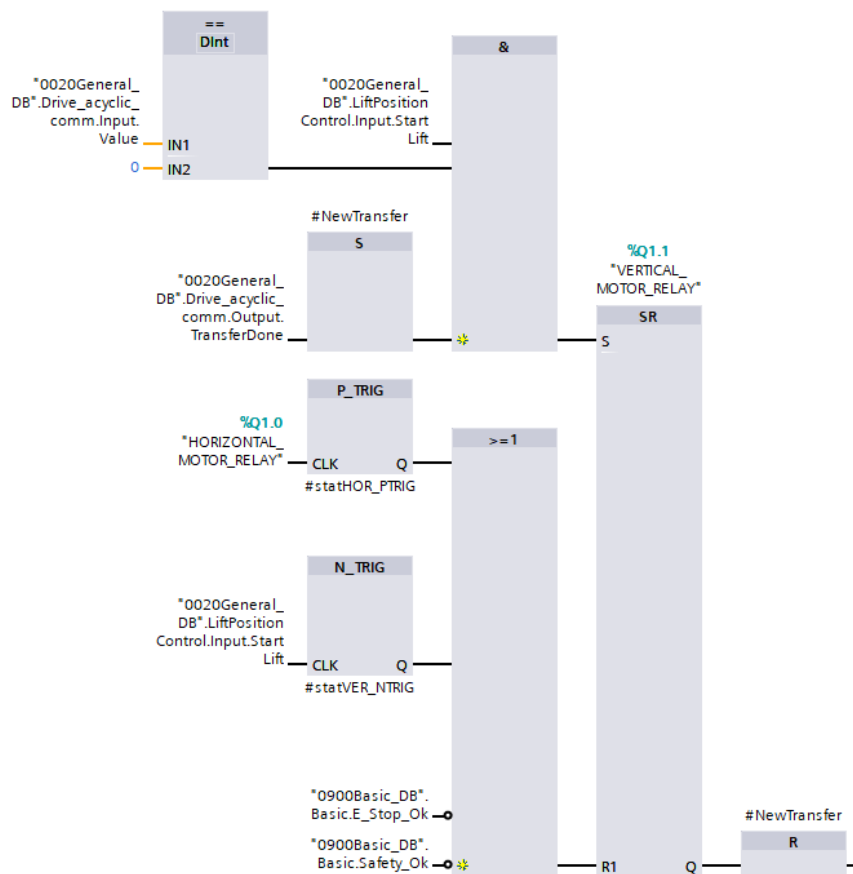
Na slici 4.20 prikazan je kod koji upravlja upisivanjem vrijednosti u parametar Pr.185. Bit *Execute*, koji pokreće novo upisivanje, aktivira se svaki puta kada se aktivira neki od bitova za pokretanje motora. Kao što je prikazano u nastavku, svaki od tih bitova može se aktivirati tek nakon što je u varijablu *InputValue* upisana ispravna vrijednost parametra, a motor može se pokrenuti tek nakon što je upisivanje izvršeno.



Slika 4.20: Upravljanje promjenom vrijednosti parametra u memoriji VFD-a

Bit *Execute* vraća se na nulu kada FB *E800Acyclic* javi da je upisivanje dovršeno, odnosno kada je aktivan bit *TransferDone* ili kada istekne vrijeme od 3 s. Vremenski brojač ovdje je dodan jer je testiranje pokazalo da FB *E800Acyclic* ponekad ne aktivira bit *TransferDone* iako je upisivanje uspješno izvršeno.

Kada su ispunjeni svi uvjeti za pokretanje elektromotora, uključuje se kontaktor tog motora tako da se aktivira odgovarajući izlaz PLC-a. Na slici 4.21 je prikazan kod koji upravlja kontaktorom prigona vertikalne osi.

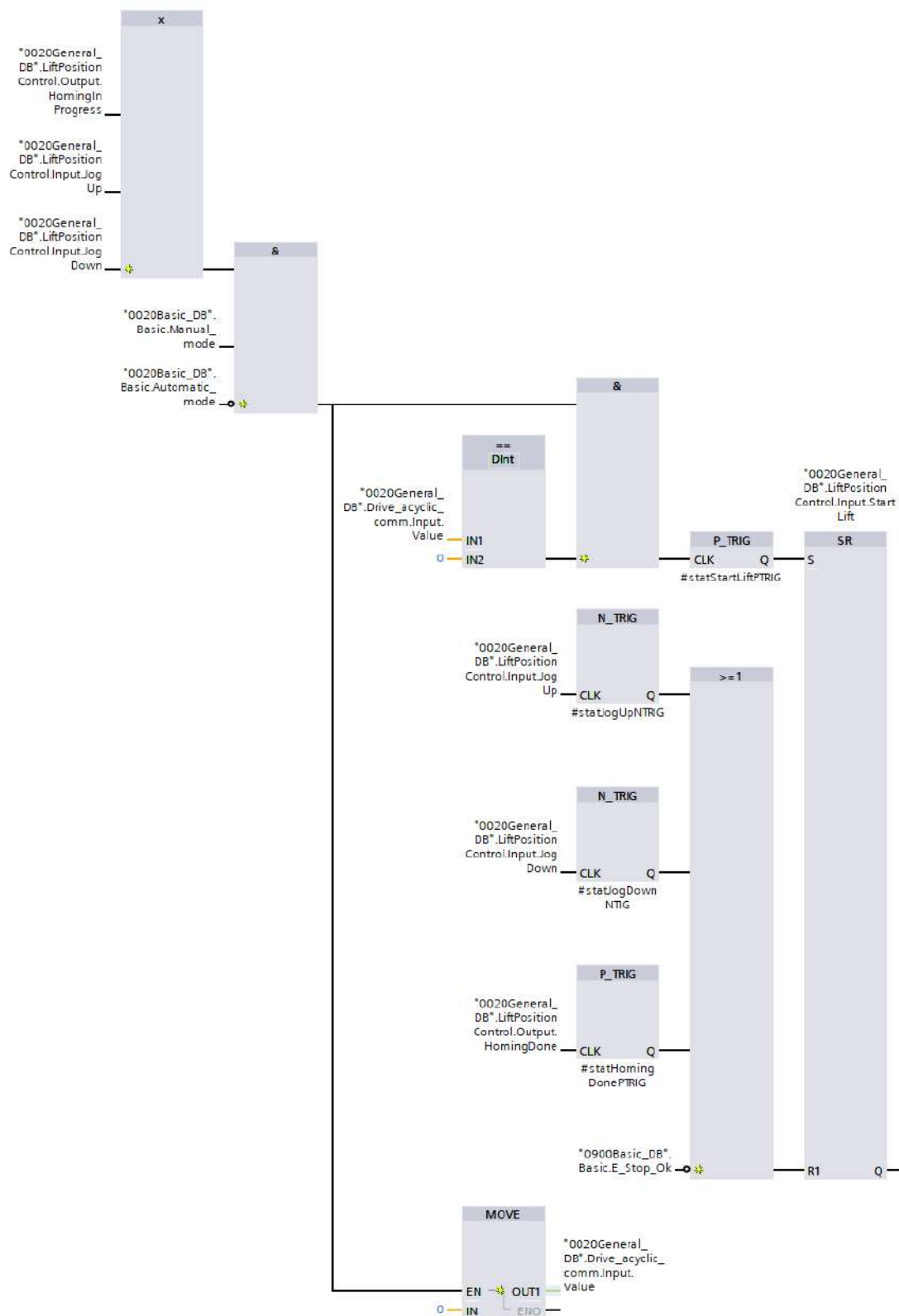


Slika 4.21: Upravljanje kontaktorom prigona vertikalne osi

Izlaz *%Q1.1* koji upravlja kontaktorom prigona vertikalne osi aktivira se ako je aktivan bit *StartLift*, ako je spremljena ispravna vrijednost parametra Pr.185 i ako je upisivanje izvršeno. Pošto je signal *TransferDone* u obliku kratkog impulsa, dodan je novi bit *#NewTransfer* koji se postavlja na vrijednost 1 sve dok se kontaktor ne aktivira. Kontaktor se deaktivira ako se aktivirao kontaktor prigona horizontalne osi, ako je pao signal za pokretanje prigona, ako je pritisnut ESTOP ili ako je iz nekog drugog razloga aktiviran sigurnosni logički krug. Na isti način upravlja se i kontaktorom prigona horizontalne osi.

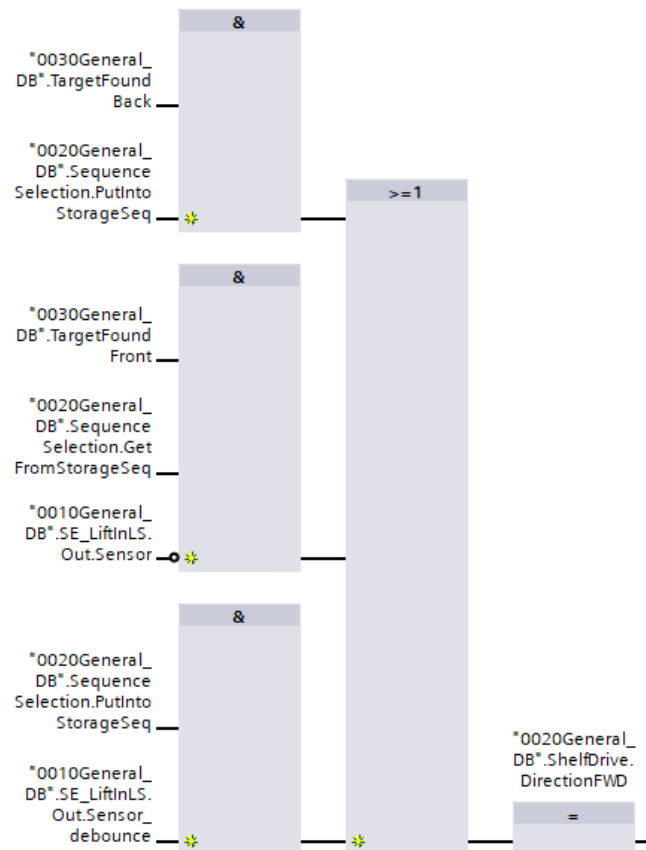
Do sada nekoliko puta naveden bit *StartLift* aktivira se svaki puta kada se treba pokrenuti prigon vertikalne osi. Služi kao okidač pomoću kojeg sekvenca u automatskom režimu rada od programa traži da provede sve do sada opisane korake potrebne za puštanje prigona pod napon. Logika upravljanja ovim bitom prikazana je na slici 4.22. *StartLift* aktivira se, osim u sekvenci, ako operater dodirne tipku za ručno pokretanje dizala ili ako je pozvana funkcija referenciranja. Prije samog aktiviranja bita, korištenjem naredbe *MOVE* upisuje se 0 u varijablu koja sprema vrijednost koja se mora upisati u memoriju VFD-a. *StartLift* se deaktivira ako je operater pustio tipku za ručno

vođenje, ako je završilo referenciranje ili ako je pritisnut ESTOP. Kao što je do sada prikazano, nakon aktiviranja ovog bita najprije se određuje smjer gibanja dizala (*StartPositive* ili *StartNegative*), zatim se izvršava upisivanje vrijednosti 0 u parametar Pr.185, uključuje se kontaktor prigona vertikalne osi i otpušta se kočnica.



Slika 4.22: Upravljanje bitom *StartLift*

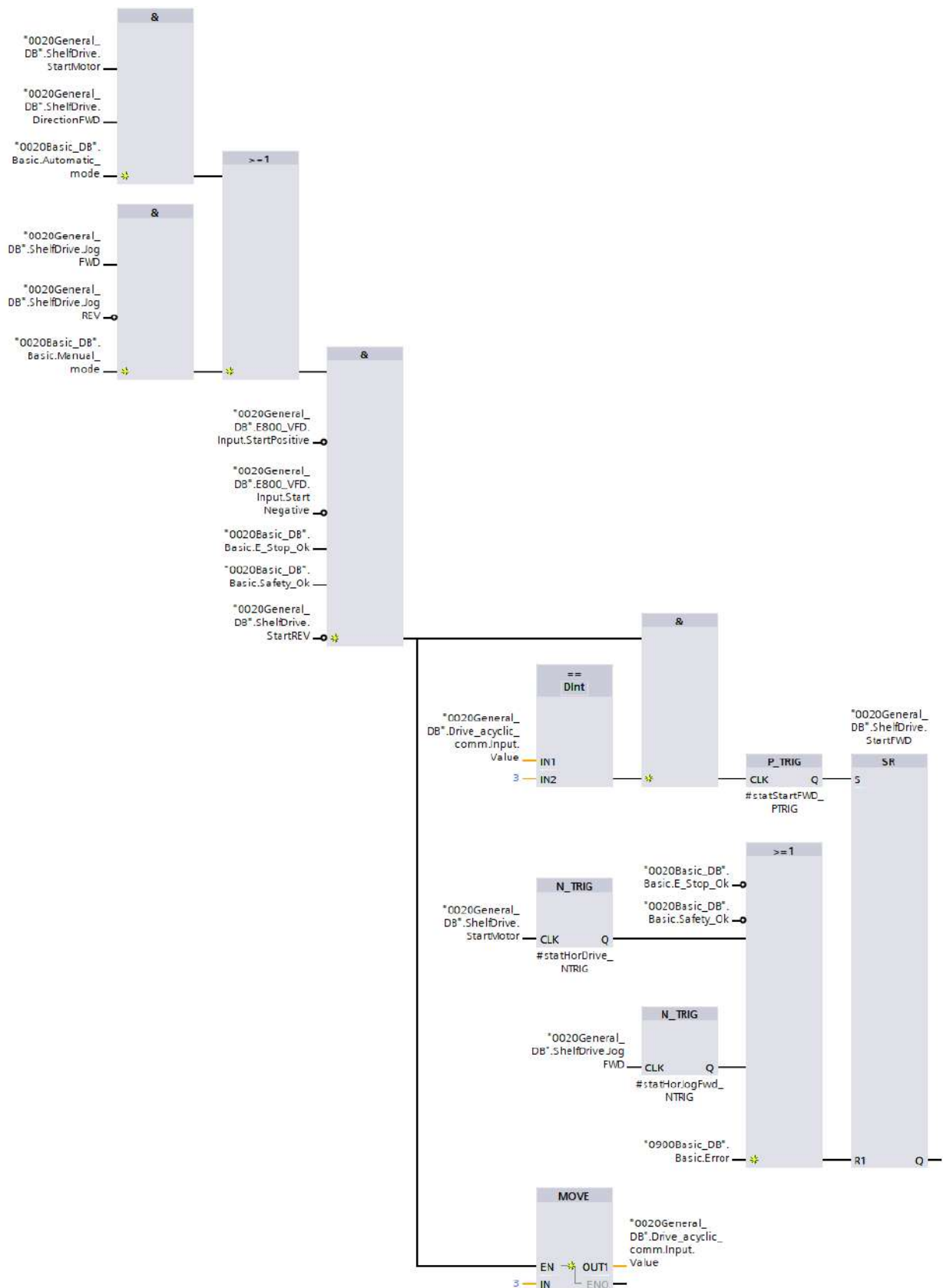
Do sada je prikazano upravljanje vertikalnom osi. Upravljanje horizontalnom osi znatno je jednostavnije jer prigon nema ni kočnicu ni enkoder. Iz tog razloga se za upravljanje horizontalnom osi ne koristi FB *PositionControl*. Slično kao kod vertikalne osi, prigon horizontalne osi pokreće se aktivacijom bita *ShelfDrive.StartMotor*. Smjer kretanja horizontalne osi određuje se jednostavnom logikom, prikazanoj na slici 4.23, koja se sastoji od definicije svake moguće situacije u kojoj horizontalni pogon mora krenuti u jednom ili drugom smjeru.



Slika 4.23: Logika upravljanja smjerom horizontalne osi

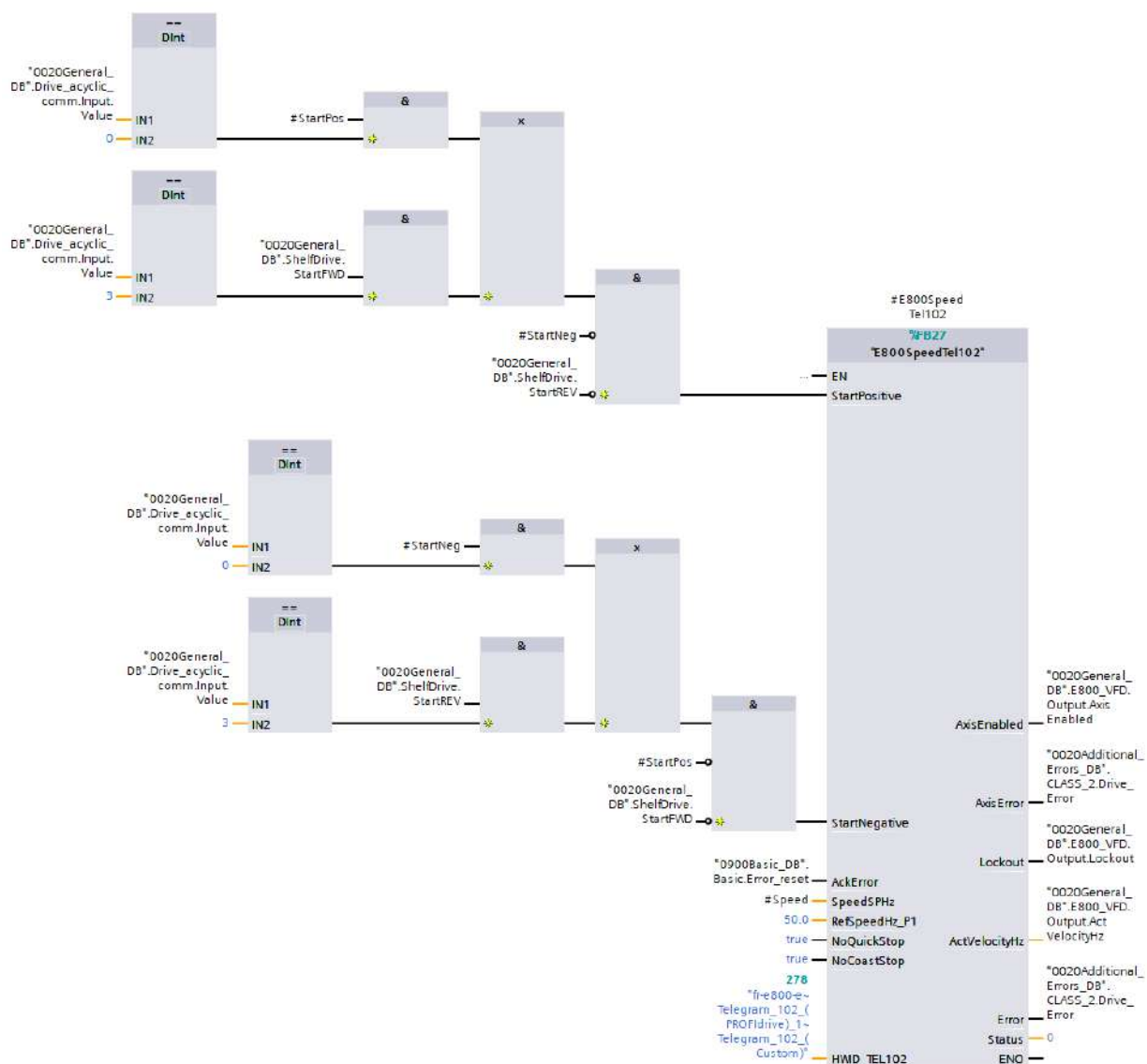
Horizontalna os treba se kretati u pozitivnom smjeru (*ShelfDrive.DirectionFWD*) ako se ladica sprema u stražnji stupac skladišnog prostora, ako se ladica vadi iz prednjeg stupca ili ako se ladica pomiče is stanice za utovar na dizalo. Na isti način definirano je kretanje ladice u negativnom smjeru. Nakon što je određen smjer, preostalo je aktivirati bitove *ShelfDrive.StartFWD* ili, ako je smjer negativan, *ShelfDrive.StartREV* koji služe za aktiviranje funkcija promjene vrijednosti parametra Pr.185 i uključivanje kontaktora prigona horizontalne osi.

Logika upravljanja bitom *ShelfDrive.StartFWD* prikazana je na slici 4.24. *StartFWD* aktivira se ako je u automatskom režimu pozvan *StartMotor* i ako je određen pozitivan smjer ili ako je na HMI-u dodirnutu tipka za ručno vođenje horizontalne osi u pozitivnom smjeru. Dodani su i uvjeti da bitovi za pokretanje prigona vertikalne osi moraju biti neaktivni i da se horizontalna os u tom trenutku ne smije gibati u negativnom smjeru. Kada su svi uvjeti ispunjeni, naredbom MOVE upisuje se vrijednost 3 u varijablu *InputValue* te se nakon toga aktivira bit *StartFWD*. Bit se deaktivira na padajući brid signala *StartMotor*, kada je na HMI-u pušten gumb za ručno vođenje, ako je pritisnut ESTOP ili ako se iz nekog drugog razloga aktivira sigurnosni logički krug.

Slika 4.24: Logika upravljanja bitom *StartFWD*

Proces je dalje jednak kao kod vertikalne osi. *StartFWD* aktivira proces upisivanja vrijednosti 3 u parametar Pr.185 VFD-a. Nakon što je parametar upisan te je VFD aktivirao funkcije za upravljanje drugim motorom, uključuje se kontaktor prigona horizontalne osi.

Komponenta koja povezuje svu do sada prikazanu logiku za upravljanje osima je funkcijski blok *E800SpeedTel102*. Ovaj FB dio je standardne biblioteke za upravljanje VFD-om i služi kao sučelje za komunikaciju PLC-a i VFD-a Profinet standardom. Prikazan je na slici 4.25.



Slika 4.25: FB za upravljanje VFD-om

Na ulaz *StartPositive* dodani su bitovi za pokretanje objiju osi u pozitivnom smjeru, a na *StartNegative* bitovi za pokretanje objiju osi u negativnom smjeru. Kada je jedan od tih ulaza aktivan, VFD će pustiti napon na svoj izlazni terminal. Ulaz *AckError* služi za resetiranje grešaka i na njega je dodan signal koji se aktivira pritiskom tipkala za resetiranje.

Osim samih aktivacijskih signala, PLC VFD-u šalje i podatak o traženoj brzini osi izraženoj u obliku frekvencije. U ručnom načinu rada, operater može unijeti željenu brzinu u obliku postotka maksimalne brzine koja odgovara frekvenciji iznosa 50 Hz. Za automatski režim rada, definirano je da je brzina horizontalne osi uvijek 50%. Brzina vertikalne osi postavljena je na 90% i smanjuje se na 10% kada je razlika između trenutne i zadane enkoderske pozicije manja od 100, dakle kada se dizalo približava cilju. Razlog tome je što se dizalo, koje zajedno sa samom robom ima relativno veliku masu, zbog svoje inercije neće zaustaviti točno u traženoj poziciji ako se zakoči prenglo.

Horizontalna os smije se pokrenuti samo ako je dizalo u poziciji iz koje je ladicu moguće spremiti u policu ili u stanici za utovar. Ako se ladica pokrene u bilo kojoj drugoj poziciji, doći će do kolizije što će rezultirati materijalnom štetom. Kako bi se to spriječilo, postavljen je uvjet da se automatski režim rada ne može pokrenuti ako dizalo nije u dozvoljenoj poziciji. U nastavku je prikazan programski kod kojim se provjerava je li dizalo u poziciji neke police.

```
FOR #counter_enc_pos_back := 1 TO #MAX_RACKS DO

    IF "0020General_DB".LiftPositionControl.Output.EncoderPos > "0030General_DB".Storage.
        BackRacks[#counter_enc_pos_back].Rack.EncoderPos - "0020General_DB".PositionTolerance
        AND "0020General_DB".LiftPositionControl.Output.EncoderPos < "0030General_DB".Storage.
        BackRacks[#counter_enc_pos_back].Rack.EncoderPos + "0020General_DB".PositionTolerance THEN

        #In_allowed_enc_pos := TRUE;

    END_IF;

END_FOR;

IF "0020General_DB".E800_VFD.Input.StartPositive = TRUE
OR "0020General_DB".E800_VFD.Input.StartNegative = TRUE THEN

    #In_allowed_enc_pos := FALSE;

END_IF;
```

Brojač iterira kroz spremljene enkoderske pozicije svih polica. Ako je trenutna enkoderska pozicija jednaka nekoj od spremljenih pozicija i unutar granica tolerancije, smatra se da je dizalo u dozvoljenoj poziciji i aktivira se bit `#In_allowed_enc_pos`. Ako se dizalo ponovo počelo gibati, više nije u toj poziciji, `#In_allowed_enc_pos` se resetira i provjera se radi ponovo nakon zaustavljanja.

4.4 Upravljačka logika stanice skladišnog prostora

Stanica skladišnog prostora sastoji se od prednjeg i stražnjeg stupca skladišnog prostora. U funkcijskom bloku `0030_Functions` definirana je logika koja izračunava broj polica potrebnih za spremanje nekog tereta, traži dostupan prostor i upisuje promjene u memoriju PLC-a. Na slici 4.26 prikazano je na koji se način u memoriji PLC-a sprema trenutno stanje skladišnog prostora.

0030General_DB					
	Name	Data type	Start value	Snapshot	Retain
1	Static				<input type="checkbox"/>
2	Storage	"Storage"			<input checked="" type="checkbox"/>
3	BackRacks	Array[1..105] of "Rack"			<input checked="" type="checkbox"/>
4	BackRacks[1]	"Rack"			<input checked="" type="checkbox"/>
5	Rack	Struct			<input checked="" type="checkbox"/>
6	IsAvailable	Bool	FALSE	FALSE	<input checked="" type="checkbox"/>
7	StoredDrawerID	Int	54	54	<input checked="" type="checkbox"/>
8	StoredCargoHeight	Int	1	1	<input checked="" type="checkbox"/>
9	EncoderPos	Dint	92	92	<input checked="" type="checkbox"/>

Slika 4.26: Struktura za spremanje trenutnog stanja skladišnog prostora

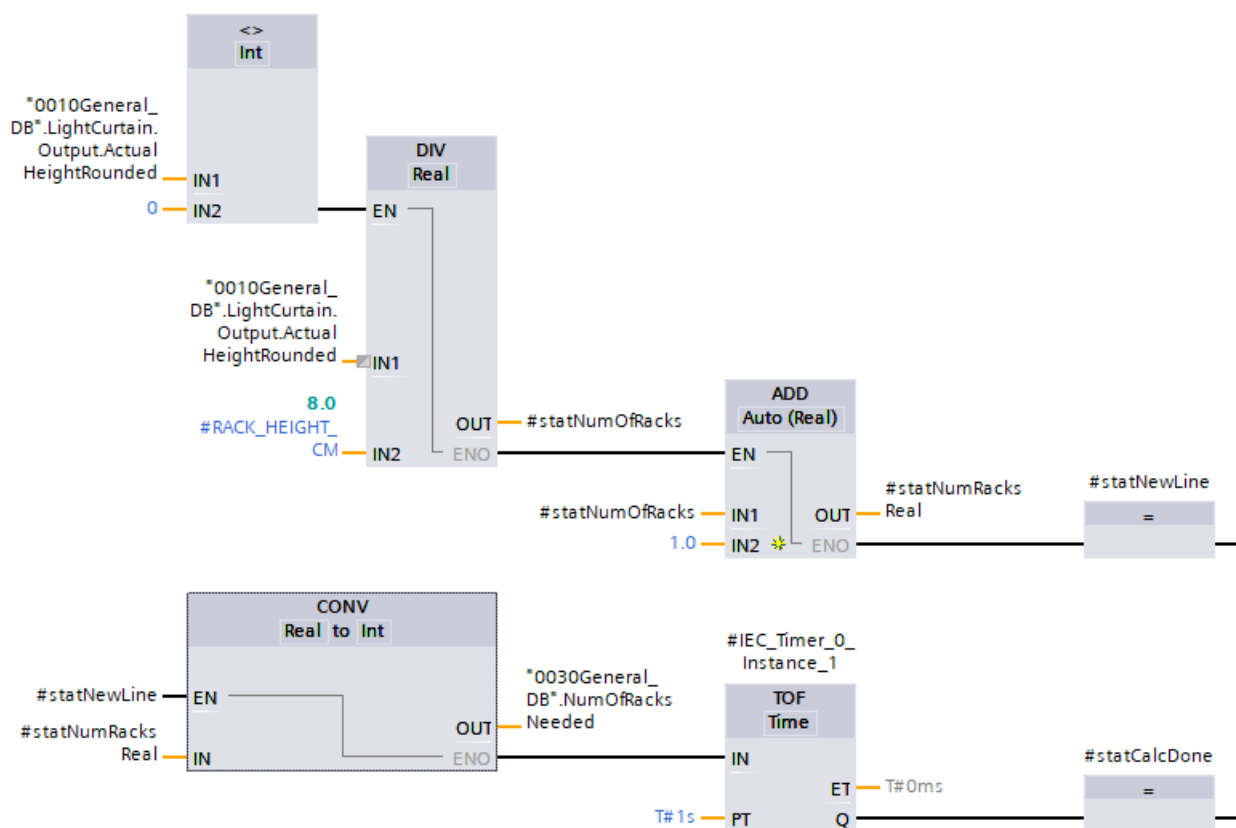
DB-u `0030General_DB` inicijalizirana je varijabla posebnog tipa podatka "Storage". Sastoji se od dvije liste, `BackRacks` i `FrontRacks`, koje se sastoje od 105 varijabli posebnog tipa podatka "Rack". Svaka policica u sustavu ovdje ima svoju strukturu tipa "Rack". "Rack" je struktura koja se sastoji od četiri podatka kojima je stanje te police potpuno opisano:

- `IsAvailable` - ako je vrijednost 0, policica je zauzeta, inače je slobodna;

- StoredDrawerID - ako je na polici spremljena ladica, ovdje se sprema broj kojim je ta ladica označena;
- StoredCargoHeight - ako je na polici spremljena ladica, ovdje se sprema koliko sljedećih polica zauzima visina spremljenog tereta;
- EncoderPos - enkoderska pozicija police.

Osim ove strukture, definirana je još i struktura "Drawers". To je struktura koja se sastoji od liste u kojoj je spremljena brojčana oznaka svake ladice unutar sustava.

Nakon što ladica tijekom spremanja prođe kroz svjetlosnu zavjesu, najprije je potrebno izračunati koliko je polica potrebno rezervirati za spremanje te ladice. Na slici 4.27 prikazan je kod koji računa broj potrebnih polica.



Slika 4.27: Izračun potrebnog broja polica

Ako je izmjerena visina različita od nule, dijeli se s visinom jedne police koja iznosi 8 cm. Na dobiveni rezultat se zbog sigurnosti dodaje 1. Rezultat se pretvara u tip podatka Int i sprema u varijablu *NumOfRacksNeeded*. Nakon što je spremanje dovršeno, vrijednosti koje sudjeluju u ovom izračunu se resetiraju na 0:

```
IF NOT "0020General_DB".SequenceSelection.PutIntoStorageSeq THEN

    #statNumRacksReal := 0;
    #statNumOfRacks := 0;
    "0030General_DB".NumOfRacksNeeded := 0;

END_IF;
```

Nakon što je određeno koliko je prostora potrebno za spremanje ladice, potrebno je proći kroz trenutnu konfiguraciju skladišnog prostora i pronaći lokaciju na kojoj ima dovoljno mjesta da bi se ladica spremila. U nastavku je prikazan algoritam za traženje prostora.

```
#statNoSpaceBack := FALSE;
#statNoSpaceFront := FALSE;

IF "0020General_DB".SequenceSelection.PutIntoStorageSeq = TRUE AND "0010General_DB".LightCurtain.
Input.MoveDone = TRUE AND #statCalcDone = TRUE THEN

  FOR #counter_of_racks := 1 TO "0030Register_DB".NumOfRacks DO

    IF "0030General_DB".RackFound = FALSE THEN

      IF "0030General_DB".Storage.BackRacks[#counter_of_racks].Rack.IsAvailable = TRUE THEN

        #statNoSpaceBack := FALSE;

        FOR #counter_of_free_racks := #counter_of_racks TO
          #counter_of_racks + "0030General_DB".NumOfRacksNeeded - 1 DO

          IF #counter_of_free_racks <= "0030Register_DB".NumOfRacks THEN

            IF "0030General_DB".Storage.BackRacks[#counter_of_free_racks].Rack.
              IsAvailable = FALSE THEN

              #statNoSpaceBack := TRUE;

            END_IF;
          ELSE

            #statNoSpaceBack := TRUE;

          END_IF;
        END_FOR;

        IF #statNoSpaceBack = FALSE THEN

          "0020General_DB".LiftPositionControl.Input.TargetPosition
            := #counter_of_racks;
          "0020General_DB".LiftPositionControl.Input.EncoderTarget := "0030General_DB".
            Storage.BackRacks[#counter_of_racks].Rack.EncoderPos;
          "0030General_DB".Storage.BackRacks[#counter_of_racks].Rack.StoredCargoHeight
            := "0030General_DB".NumOfRacksNeeded;
          "0030General_DB".RackFound := TRUE;
          "0030General_DB".TargetFoundBack := TRUE;
          "0030General_DB".TargetFoundFront := FALSE;

        END_IF;
      END_IF;
    END_IF;
  END_FOR;

```

Na početku se inicijaliziraju dvije varijable, *#statNoSpaceBack* i *#statNoSpaceFront*. Postavljaju se na FALSE, dakle pretpostavlja se da u skladištu ima prostora. Započinje iteracija po stražnjem stupcu skladišta. Varijabla *RackFound* prati je li prostor pronađen. Ako nije, iteracija se zaustavlja prvi puta kada se u ranije prikazanoj strukturi pronađe polica kod koje bit *IsAvailable* iznosi 1, odnosno kada se pronađe slobodna polica. Nakon toga započinje nova iteracija koja ide od pronađene police do visine potrebne za spremanje tereta. Ako se u cijelom tom prostoru pronađe polica koja nije slobodna, zaključuje se da polica pronađena prvom iteracijom, iako je prazna, nije dobar kandidat za spremanje te ladice jer postoji ladica iznad koja smeta. *#statNoSpaceBack* se postavlja na TRUE. Prvi iterator kreće dalje i postupak se ponavlja tako dugo dok nije pronađeno *NumOfRacksNeeded* slobodnih polica u nizu. Ako je prostor pronađen, *#statNoSpaceBack* je ostao na FALSE i trenutna vrijednost prvog brojača, koja odgovara rednom broju pronađene police, postaje zadana pozicija (*TargetPosition*) vertikalne osi. Iz memorije se očitava enkoderska pozicija spremljena na toj lokaciji u listi i ta vrijednost postaje zadana enkoderska pozicija *EncoderTarget*. U memoriju se zapisuje visina tereta koji se sprema. *RackFound* se postavlja na TRUE što omogućuje izlaz

iz petlje i prestanak izvođenja ovog algoritma. Aktivira se bit *TargetFoundBack* koji se koristi za određivanje smjera kretanja horizontalne osi. Ako u cijelom stražnjem stupcu nije pronađeno dovoljno prostora, na isti način se pretražuje stražnji stupac. U slučaju da je cijeli skladišni prostor ispunjen, algoritam neće pronaći slobodnu policu. I *#statNoSpaceBack* i *#statNoSpaceFront* ostat će na TRUE što će aktivirati grešku *No space in storage Error*. Spremanje će se zaustaviti i dizalo se neće maknuti iz stanice za utovar.

Nakon što je ladica spremljena ili izvađena, preostalo je novo stanje zapisati u memoriju. Sljedeći kod evidentira novu ladicu nakon što je spremljena.

```

IF #statPutDrawerToBack THEN

    FOR #counter_reserved_racks := "0020General_DB".LiftPositionControl.Input.TargetPosition TO "
    0020General_DB".LiftPositionControl.Input.TargetPosition + "0030General_DB".NumOfRacksNeeded
    - 1 DO
        IF "0030General_DB".Storage.BackRacks[#counter_reserved_racks].Rack.IsAvailable
            = TRUE THEN

            "0030General_DB".Storage.BackRacks[#counter_reserved_racks].Rack.IsAvailable
                := FALSE;
            "0030General_DB".Storage.BackRacks["0020General_DB".LiftPositionControl.Input.
            TargetPosition].Rack.StoredDrawerID := "0020General_DB".ManualInputDrawerID;
            "0030General_DB".Storage.BackRacks["0020General_DB".LiftPositionControl.Input.
            TargetPosition].Rack.StoredCargoHeight := "0030General_DB".NumOfRacksNeeded;

            END_IF;
        END_FOR;

    ELSIF #statPutDrawerToFront THEN

        FOR #counter_reserved_front_racks := "0020General_DB".LiftPositionControl.Input.
        TargetPosition TO "0020General_DB".LiftPositionControl.Input.TargetPosition + "0030General_DB
        ".NumOfRacksNeeded - 1 DO
            IF "0030General_DB".Storage.FrontRacks[#counter_reserved_front_racks].Rack.IsAvailable
                = TRUE THEN

                "0030General_DB".Storage.FrontRacks[#counter_reserved_front_racks].Rack.IsAvailable
                    := FALSE;
                "0030General_DB".Storage.FrontRacks["0020General_DB".LiftPositionControl.Input.
                TargetPosition].Rack.StoredDrawerID := "0020General_DB".ManualInputDrawerID;
                "0030General_DB".Storage.FrontRacks["0020General_DB".LiftPositionControl.Input.
                TargetPosition].Rack.StoredCargoHeight := "0030General_DB".NumOfRacksNeeded;

                END_IF;
            END_FOR;

        END_IF;
    
```

Ako je ladica spremljena u stražnji stupac, što je određeno bitom *#statPutDrawerToBack*, započinje iteracija po policama koje su sada zauzete. Podatak *IsAvailable* svake od njih postavlja se na FALSE. Ovime se postiže da algoritam za traženje prostora zanemari ove police kada će se spremati sljedeća ladica. U varijable *StoredDrawerID* i *StoredCargoHeight* police u kojoj je ladica spremaju se broj te ladice i visina tereta u njoj. Isti proces se provodi ako je ladica spremljena u prednji stupac, odnosno ako je aktivan bit *#statPutDrawerToFront*.

Kada se spremljena ladica vadi iz skladišta, provodi se proces suprotan gore opisanom. Prikazan je u nastavku.

```

IF #statTakeDrawerFromBack OR #statTakeDrawerFromFront THEN

    FOR #rack_num := 1 TO "0030Register_DB".NumOfRacks DO

        IF "0020General_DB".ManualInputDrawerID = "0030General_DB".Storage.
        BackRacks[#rack_num].Rack.StoredDrawerID THEN

            FOR #counter_of_racks_reset := #rack_num TO #rack_num + "0030General_DB".Storage.
            BackRacks[#rack_num].Rack.StoredCargoHeight - 1 DO

                "0030General_DB".Storage.BackRacks[#counter_of_racks_reset].Rack.IsAvailable
            
```

```

:= TRUE;
"0030General_DB".Storage.BackRacks[#counter_of_racks_reset].Rack.StoredDrawerID
:= 0;
"0030General_DB".Storage.BackRacks[#counter_of_racks_reset].Rack.
  StoredCargoHeight := 0;

END_FOR;
END_IF;

IF "0020General_DB".ManualInputDrawerID = "0030General_DB".Storage.FrontRacks[#rack_num].
  Rack.StoredDrawerID THEN

FOR #counter_of_front_racks_reset := #rack_num TO #rack_num +
  "0030General_DB".Storage.FrontRacks[#rack_num].Rack.StoredCargoHeight - 1 DO

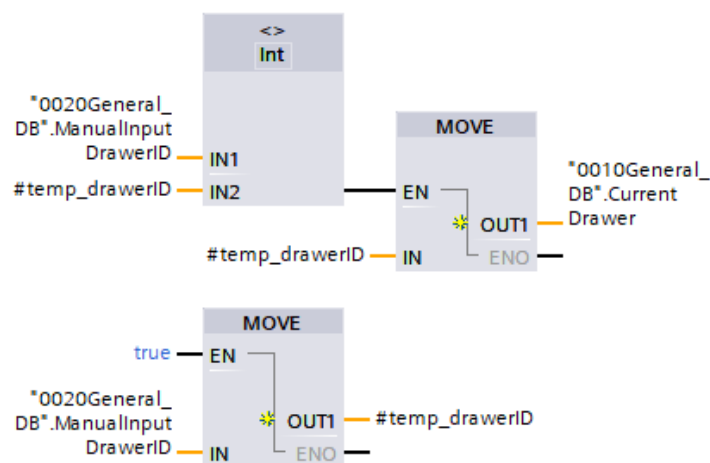
  "0030General_DB".Storage.FrontRacks[#counter_of_front_racks_reset].Rack.
    IsAvailable := TRUE;
  "0030General_DB".Storage.FrontRacks[#counter_of_front_racks_reset].Rack.
    StoredDrawerID := 0;
  "0030General_DB".Storage.FrontRacks[#counter_of_front_racks_reset].Rack.
    StoredCargoHeight := 0;

END_FOR;
END_IF;
END_FOR;
END_IF;

```

Nakon što je ladica izvađena iz police, započinje iteracija po policama. Ako je pronađena polica kojoj je podatak *StoredDrawerID* jednak broju ladice koja je upravo izvađena, varijable *IsAvailable*, *StoredDrawerID* i *StoredCargoHeight* postavljaju se na vrijednost 0 za sve police koje su bile zaokupljene teretom na izvađenoj ladici.

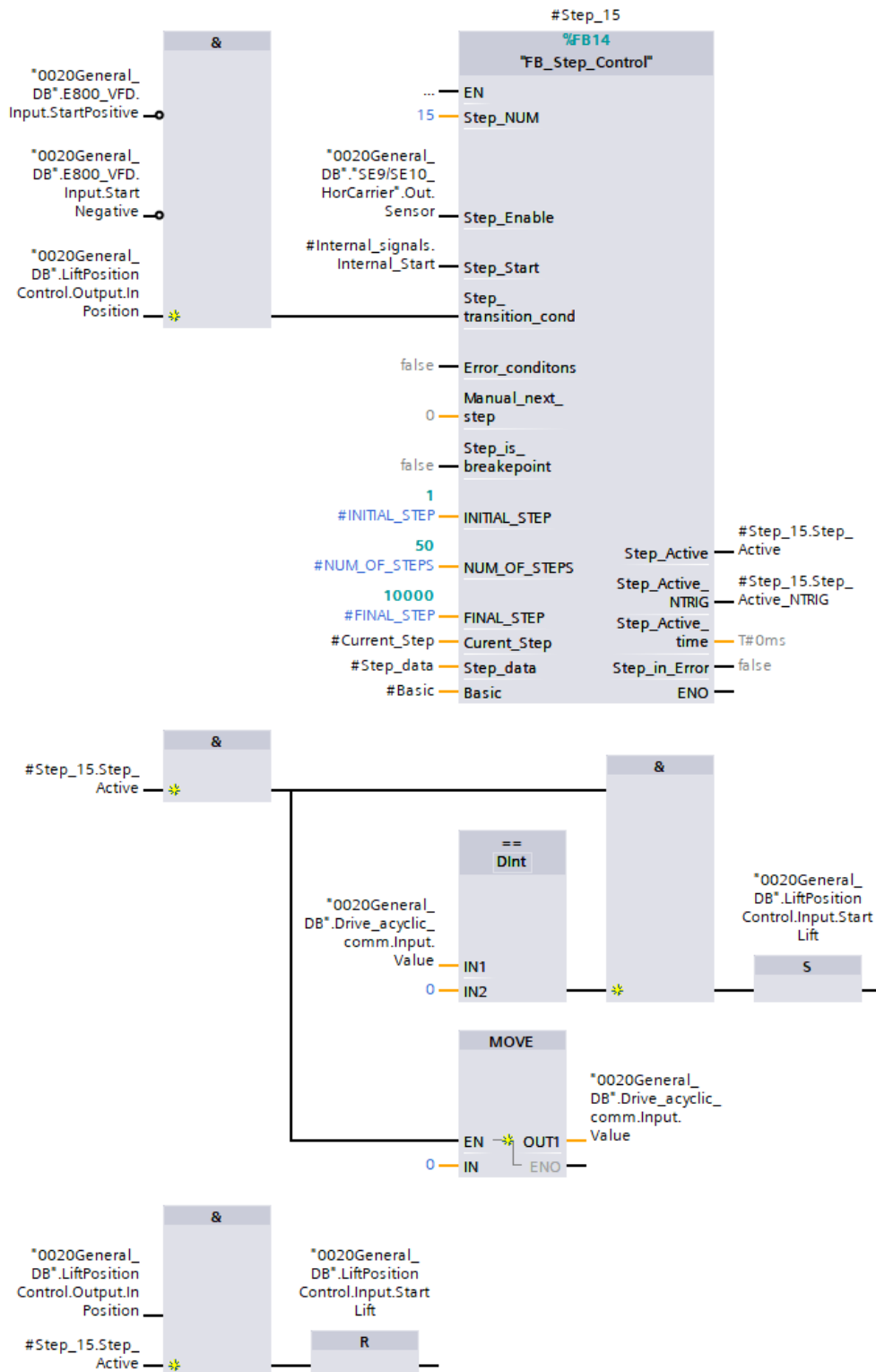
Kao što je opisano u poglavlju 2.4 *Funkcije i procesi u radu sustava*, ako je pokrenuta funkcija *Izvadi* dok je u stanici za utovar neka druga ladica, ta ladica će se prvo spremiti a tek nakon toga će se izvaditi ladica koju operater traži. Na primjer, u stanici za utovar je ladica broj 10. Operater želi izvaditi ladicu broj 15, taj broj upisuje u polje za unos broja ladice na zaslonu HMI-a i pokreće funkciju izvadi. Kada se ladica broj 10 spremi, upravljački program mora znati da sprema ladicu broj 10, a ne ladicu broj 15 čiji je broj operater unio kako se u varijablu *StoredDrawerID* ne bi upisala pogrešna vrijednost. Ako je u stanici za utovar bila ladica broj 10, može se zaključiti da je prethodna operacija bila vađenje te ladice, dakle prethodna vrijednost na HMI-u je bila 10. Prema tome, svaki puta kada operater upiše novi broj ladice, prethodni broj ladice mora ostati zapisan u memoriji kako bi se osigurala točnost podataka u memoriji PLC-a. Na slici 4.28 prikazan je kod koji zapisuje broj prethodne ladice svaki puta kada se upiše nova vrijednost.



Slika 4.28: Prepisivanje prethodne vrijednosti u novu varijablu

4.5 Sekvence u automatskom režimu rada

Svaki korak sekvence definiran je pomoću funkcijskog bloka *FB_Step_Control* koji je dio standarda u kojem je napisan upravljački program. Na slici 4.29 prikazan je primjer jednog koraka sekvence.



Slika 4.29: Korak sekvence

U ovom primjeru prikazan je korak u kojem dizalo odlazi do zadane pozicije tijekom spremanja ladice. Svaki korak sastoji se od uvjeta pokretanja koraka (ulaz *Step_Enable*), rednog broja koraka (*Step_NUM*), uvjeta za prijelaz na sljedeći korak (ulaz *Step_transition_cond*) i ostalih podataka potrebnih za kontrolu automatskog režima. Nakon pozivanja FB-a, definira se što taj korak radi.

U ovom primjeru, uvjet za početak izvođenja koraka je stanje senzora pozicije lanca horizontalne osi. Korak se može izvesti samo ako je horizontalna os u poziciji u kojoj se vertikalna os može pokrenuti bez da dođe do kolizije. Kada ovaj korak postane aktivan, u varijablu čija se vrijednost prepisuje u parametar Pr.185 VFD-a se zapisuje vrijednost 0 i aktivira se bit *StartLift*. Kada dizalo dosegne traženu poziciju, *StartLift* se resetira i dizalo se zaustavlja. Ispunjeni su uvjeti za prijelaz na sljedeći korak i ovaj korak prestaje biti aktivan.

Na ovaj način definirani su svi koraci sekvence. Pošto ovdje korišteni PLC ne podržava jezik SFC, koji je idealan za pisanje sekvenci, nije moguće na jednostavan način postići grananje. Iz tog razloga, sve četiri sekvence su zapravo jedna sekvenca, a grananje se postiže na način prikazan u nastavku.

```

IF #Current_Step = #INITIAL_STEP AND #Basic.Automatic_mode = TRUE THEN

    "0020General_DB".LiftPositionControl.Input.TargetPosition := 0;
    "0020General_DB".LiftPositionControl.Input.EncoderTarget := 0;
    "0020General_DB".SequenceSelection.PutIntoStorageSeq := FALSE;
    "0020General_DB".SequenceSelection.GetFromStorageSeq := FALSE;
    "0020General_DB".SequenceSelection.CountRacksSeq := FALSE;
    "0020General_DB".SequenceSelection.GoToLoadingStationSeq := FALSE;
    "0020General_DB".StartDrawerMotor := FALSE;
    "0010General_DB".LightCurtain.Input.Measure := FALSE;
    "0010General_DB".LightCurtain.Input.MoveDone := FALSE;
    "0030General_DB".TargetFoundBack := FALSE;
    "0030General_DB".TargetFoundFront := FALSE;
    "0030General_DB".StoringDone := FALSE;
    "0030General_DB".ExtractionDone := FALSE;
    "0030General_DB".RackFound := FALSE;
    "0020General_DB".Acquire_target := FALSE;
    "0020General_DB".Target_acquired := FALSE;
    #Step_19.Step_Enable := FALSE;

    CASE "0020General_DB".SequenceSelection.SelectNum OF

        1: // PutIntoStorage Sequence
            #Current_Step := 3;

        2: // GetFromStorage Sequence
            IF "0020General_DB".SequenceSelection.SelectNum = 2 AND "0010General_DB".
                DrawerHorizontalPos.DrawerInLoadingStation = TRUE THEN

                "0020General_DB".Extract_while_drawer_in_LS_req := TRUE;
                #Current_Step := 3;
            ELSE
                #Current_Step := 20;
            END_IF;

        3: // CountRacks Sequence
            #Current_Step := 33;
            "0020General_DB".SequenceSelection.CountRacksSeq := TRUE;

        4: // GoToLoadingStation Sequence
            #Current_Step := 37;
            "0020General_DB".SequenceSelection.GoToLoadingStationSeq := TRUE;

    END_CASE;
END_IF;

IF #Current_Step >= 3 AND #Current_Step <= 19 THEN

    "0020General_DB".SequenceSelection.GetFromStorageSeq := FALSE;
    "0020General_DB".SequenceSelection.PutIntoStorageSeq := TRUE;

```

```
ELSIF #Current_Step >= 20 AND #Current_Step <= 29 THEN  
    "0020General_DB".SequenceSelection.PutIntoStorageSeq := FALSE;  
    "0020General_DB".SequenceSelection.GetFromStorageSeq := TRUE;  
END_IF;
```

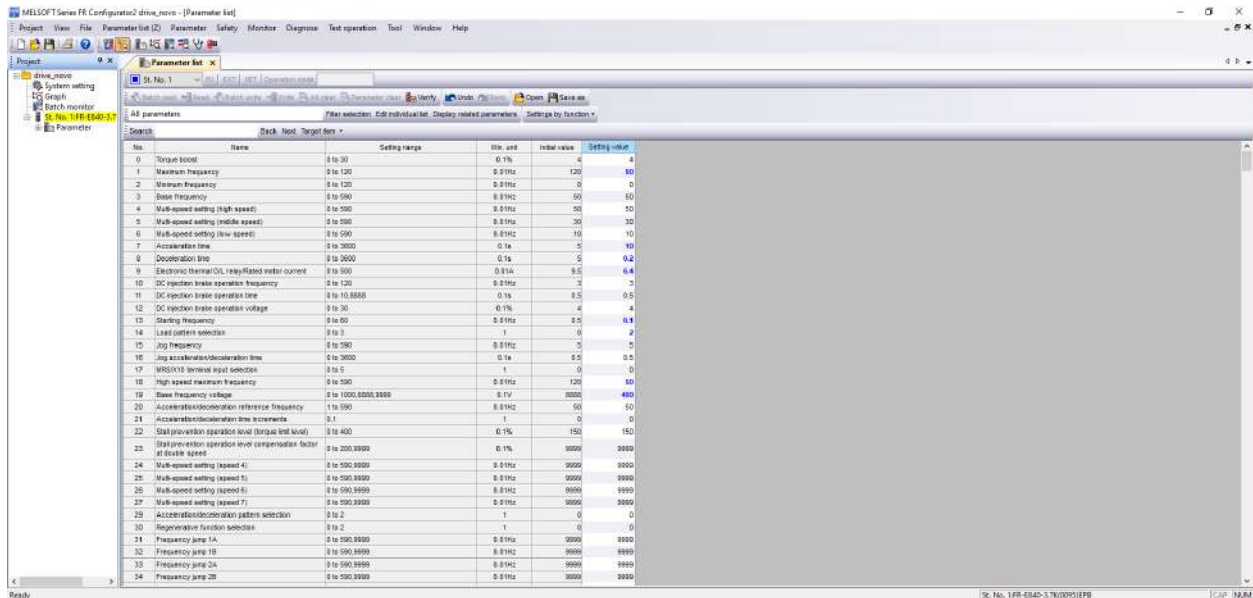
Nakon što je pokrenut automatski režim rada, program se nalazi u početnom koraku. U ovom koraku definirani su uvjeti grananja na način da se poziva korak u kojem započinje tražena sekvenca. Svaki puta kada se program vrati u ovaj korak, sve varijable koje sudjeluju u izvođenju funkcija *Spremi* i *Izvadi* postavljaju se na nula. Ovo je potrebno jer svaka varijabla zadržava svoju vrijednost tako dugo dok se u nju ne upiše neka nova vrijednost. Ako se neki korak sekvence izvede s pogrešnim vrijednostima, može doći do greške u radu sustava.

Dodirom gumba za pokretanje određene sekvence na glavnom zaslonu HMI-a, u varijablu *SequenceSelection.SelectNum* upisuje se cijeli broj između 1 i 4, ovisno o tome koja funkcija je pozvana. Na temelju tog broja, naredba CASE poziva korak u kojem počinje tražena sekvenca. Aktivira se bit koji ostatku programa javlja da je u tijeku ta sekvenca. Na primjer, ako operater dodirne gumb *Spremi*, *SelectNum* poprima vrijednost 1. Pokreće se korak 3 i aktivira se bit *PutIntoStorageSeq*. Nakon što se izvrši korak 19, sekvenca se vraća u početni korak.

5 PARAMETRIRANJE PRIGONA

5.1 Konfiguriranje komunikacije između PLC-a i VFD-a [13]

Pošto se obadvama prigonima upravlja zajedničkim VFD-om, njihovo parametriranje svodi se na upisivanje potrebnih parametara u memoriju VFD-a. Oba prigona parametriraju se na jednak način pomoću računalnog alata FR Configurator 2 proizvođača Mitsubishi Electric.



Slika 5.1: FR Configurator 2

Na slici 5.1 prikazano je sučelje *Parameter List* programa FR Configurator 2. *Parameter List* sadrži popis svih parametara u memoriji VFD-a. Odabirom funkcije *Batch read* prvo se čitaju sve vrijednosti trenutno upisane u VFD. Nakon što su unesene sve potrebne promjene, funkcijom *Batch write* novi podaci upisuju se u memoriju VFD-a. Prvi korak je konfiguriranje parametara za komunikaciju s PLC-om. U tablici 5.1 prikazano je koji parametri su uneseni.

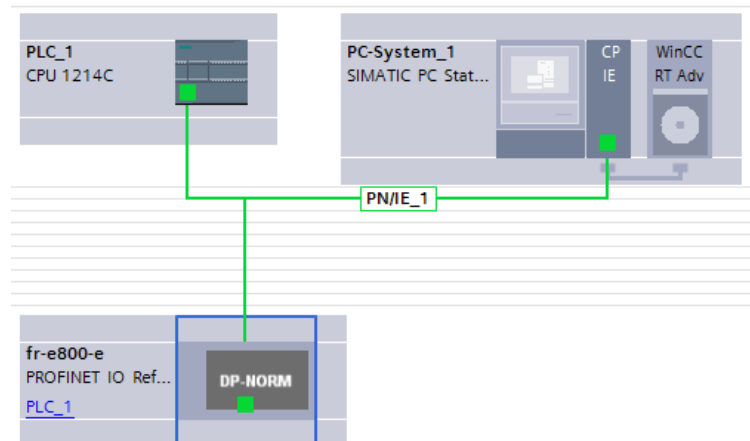
Tablica 5.1: Parametri za komunikaciju VFD-a i PLC-a

Parametar	Naziv	Vrijednost
1429	Ethernet function selection 3	34962
1449	Ethernet command source selection IP address 1	10
1450	Ethernet command source selection IP address 2	100
1451	Ethernet command source selection IP address 3	10
1452	Ethernet command source selection IP address 4	11
1320	User defined cyclic communication input 1 mapping	5
1330	User defined cyclic communication output 1 mapping	6

Unosom vrijednosti 34962 u parametar 1429 definira se da će se komunikacija provoditi na Profinet standardu. Parametri 1449 - 1452 sadrže IP adresu PLC-a. IP adresa je u ovom slučaju

10.100.10.11. Ostala dva parametra, 1320 i 1330 moraju biti postavljeni na ove vrijednosti kako bi bilo moguće koristiti FB *E800SpeedTel102* za regulaciju prigona po brzini.

Preostalo je dodati VFD u hardversku konfiguraciju sustava. U sučelju *Devices & networks* programa TIA Portal, na Ethernet mrežu dodan je VFD. U postavkama VFD-a unesena je njegova IP adresa i naziv uređaja na Profinet mreži. Hardverska konfiguracija prikazana je na slici 5.2, a unesene postavke na slici 5.3.



Slika 5.2: Hardverska konfiguracija sustava

Slika 5.3: Mrežne postavke VFD-a

5.2 Parametriranje prigona vertikalne osi

Prigon vertikalne osi parametriran je unosom potrebnih vrijednosti u memoriju VFD-a pomoću programa FR Configurator 2. Definirani su parametri poput nazivnih vrijednosti motora, metode regulacije i vremena ubrzavanja i usporavanja. Popis svih promijenjenih parametara dan je u tablici 5.2.

Tablica 5.2: Parametri prigona vertikalne osi

Parametar	Naziv	Vrijednost
1	Maximum frequency	50
7	Acceleration time	10
8	Deceleration time	0.2
9	Electornic thermel O/L relay / Rated motor current	6,4
13	Starting frequency	0,1
14	Load pattern selection	2
19	Base frequency voltage	400
71	Applied motor	5
80	Motor capacity	3,0
81	Number of motor poles	4
84	Rated motor frequency	50
245	Rated slip	4,67
800	Control method selection	11

Parametri 9, 19, 80 i 84 odnose se na vrijednosti s nazivne pločice motora. Kao što je ranije prikazano na slici 2.8 (a), na nazivnoj pločici prigona vertikalne osi vidi se da mu snaga iznosi 3 kW, nazivna struja 6,4 A, nazivni napon 400 V, nazivna frekvencija 50 Hz a kutna brzina 1430 min^{-1} . Iz ovih podataka i podataka u tablici prikazanoj na slici određeni su broj polova i klizanje. Parametrom 71 *Applied motor* označeno je da je prigon vertikalne osi indukcijski motor u zvjezda spoju.

Frequency - f - (Hz)	Shaft rotation - n - (rev/min, rpm)					
	Number of poles - p -					
	2	4	6	8	10	12
10	600	300	200	150	120	100
20	1200	600	400	300	240	200
30	1800	900	600	450	360	300
40	2400	1200	800	600	480	400
50 ¹⁾	3000	1500	1000	750	600	500
60 ²⁾	3600	1800	1200	900	720	600
70	4200	2100	1400	1050	840	700
80	4800	2400	1600	1200	960	800
90	5400	2700	1800	1350	1080	900
100	6000	3000	2000	1500	1200	1000

Slika 5.4: Sinkrona brzina asinkronog motora s obzirom na frekvenciju i broj polova [14]

Klizanje se određuje prema izrazu

$$s = \frac{N_s - N}{N_s} \cdot 100\%,$$

gdje su:

- s [%] - klizanje,
- N_s [min^{-1}] - sinkrona brzina motora,
- N [min^{-1}] - asinkrona brzina motora.

Iz tablice na slici 5.4 za sinkronu brzinu odabrana je vrijednost najbliža nazivnoj brzini u retku koji se odnosi na frekvenciju od 50 Hz. Odabrana je vrijednost $N_s = 1500 \text{ min}^{-1}$, a prema tome broj polova prigona vertikalne osi iznosi 4. Iz ovih podataka dobiva se klizanje:

$$s = \frac{1500 - 1430}{1500} \cdot 100\% = 4,67\%.$$

Dobivene vrijednosti klizanja i broja polova upisane su u parametre 245 i 81. Parametar 14 *Load pattern selection* odabran je iz dokumentacije [12]. Prema dokumentaciji, vrijednost 2 parametra 14 predviđena je za prigone dizala. Parametar 13 *Starting frequency* određuje da će VFD tijekom pokretanja motora pustiti motor pod napon tek kada dostigne frekvenciju 0,1 Hz. Parametrom 1 *Maximum frequency* određeno je da VFD neće povećati frekvenciju iznad nazivne frekvencije motora.

Parametar 800 *Control method selection* služi za odabir metode regulacije motora. Iako vertikalna os ima enkoder kao povratnu vezu, on se ne koristi u regulacijskoj petlji VFD-a. Razlog tome je što je enkoder montiran na samom dizalu, a prema [12], enkoder mora biti montiran direktno na vratilo motora kako bi se mogao koristiti kao povratna veza u PID regulacijskog petlji. Ako enkoder nije direktno povezan na vratilo motora, zatvaranje regulacijske petlje njegovom povratnom vezom nije preporučeno zbog greške koja se može pojaviti zbog zračnosti u elementima za prijenos gibanja. Iz tog razloga, unosom vrijednosti 11 u parametar 800 odabrana je regulacijska metoda Real sensorless vector control.

Vektorska regulacija bez povratne veze je regulacijska metoda otvorenog kruga koja upravlja brzinom motora modulacijom vrijednosti amplitude, faznog pomaka i frekvencije napona na temelju resultantnog vektora magnetskog toka i momenta motora [15]. Iako povratna veza zapravo postoji, smatra se da je ova regulacijska petlja otvorena jer povratna veza ne dolazi iz sustava već iz senzora unutar samog VFD-a. Kako bi ovaj način regulacije bio moguć, prije pokretanja motora potrebno je provesti postupak mjerenja konstanti motora. Unosom vrijednosti 1 u parametar 95 *Online auto tuning selection* pokreće se funkcija mjerenja konstanti motora. Nakon što je postupak proveden, prigon vertikalne osi je parametriran. Na slici 5.5 prikazane su karakteristike motora koje je VFD izmjerio.

82	Motor excitation current	0 to 500,9999	0.01A	9999	2,79
83	Rated motor voltage	0 to 1000	0.1V	400	400
84	Rated motor frequency	10 to 400,9999	0.01Hz	9999	50
89	Speed control gain (Advanced magnetic flux vector)	0 to 200,9999	0.1%	9999	9999
90	Motor constant (R1)	0 to 50,9999	0.001Ω	9999	1,719
91	Motor constant (R2)	0 to 50,9999	0.001Ω	9999	1,46
92	Motor constant (L1)/d-axis inductance (Ld)	0 to 50,9999	0.001Ω	9999	2,71
93	Motor constant (L2)/q-axis inductance (Lq)	0 to 50,9999	0.001Ω	9999	2,71
94	Motor constant (X)	0 to 500,9999	0.01Ω	9999	79,91

Slika 5.5: Karakteristike prigona vertikalne osi izmjerene postupkom *Online auto tuning*

5.3 Parametriranje prigona horizontalne osi

Slično kao kod prigona vertikalne osi, korištenjem programa FR Configurator 2 potrebni parametri upisani su u memoriju VFD-a. U tablici 5.3 dan je popis izmijenjenih parametara.

Tablica 5.3: Parametri prigona horizontalne osi

Parametar	Naziv	Vrijednost
44	Second acceleration time	10
45	Second deceleration time	0,1
51	Rated second motor current	1,0
450	Second applied motor	5
453	Second motor capacity	0,37
454	Number of second motor poles	4
456	Rated second motor voltage	400
457	Second motor rated frequency	50
743	Second motor maximum frequency	50

Podaci upisani u parametre 51, 453, 456 i 457 isčitani su s nazivne pločice motora. Snaga mu iznosi 0,37 kW, nazivni napon 400 V, nazivna frekvencija 50 Hz, nazivna struja 1 A, a asinkrona brzina 1370 min^{-1} . Iz ovih podataka je, na temelju tablice na slici 4.23, određeno da je prigon horizontalne osi četveropolni motor s klizanjem:

$$s = \frac{1500 - 1370}{1500} \cdot 100\% = 8,67\%.$$

Kako je ovo motor s visokim klizanjem i , u odnosu na prigon vertikalne osi, relativno malom snagom, nije moguće provođenje automatskog mjerenja konstanti motora. Zbog toga nije moguće korištenje ni jedne od dostupnih metoda regulacije brzine ili momenta. Iako ovaj način upravljanja motorom nije idealan, brzinu mu je i dalje moguće mijenjati promjenom frekvencije napona. Prema izrazu

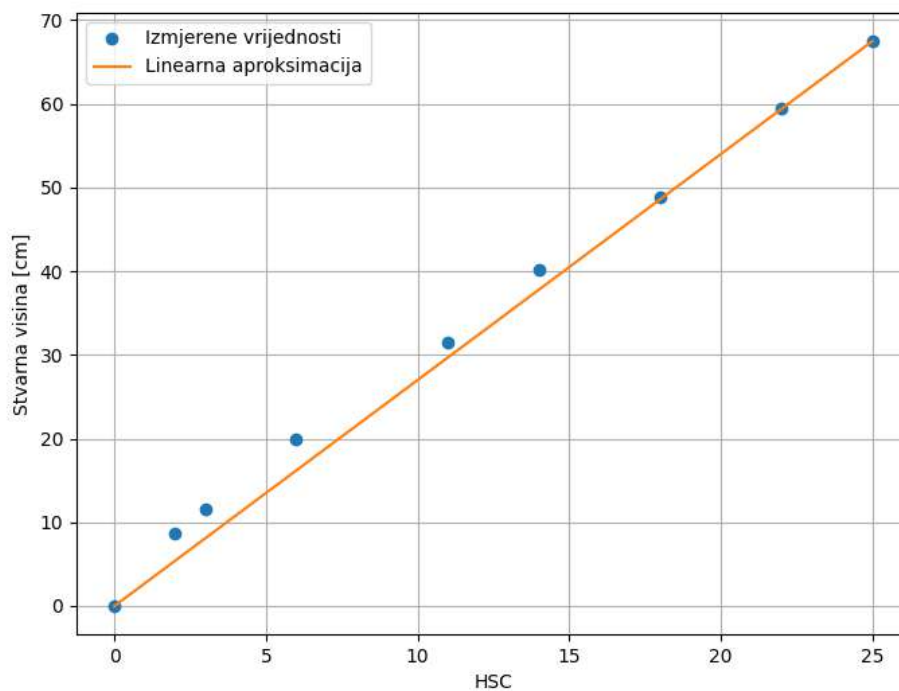
$$N = \frac{120f}{p},$$

pošto je broj polova p konstantan i iznosi 4, brzina motora proporcionalna je frekvenciji napona na ulazu.

6 TESTIRANJE SUSTAVA

6.1 Testiranje i parametrisiranje svjetlosne zavjese

Kako vrijednost visine mjerenog tereta koju kontroler svjetlosne zavjese šalje PLC-u nije izražena u nekoj smisljenoj mjernoj jedinici već u broju impulsa na ulazu visokobrzinskog brojača, potrebno je eksperimentalnom metodom odrediti relaciju koja povezuje te dvije veličine kako bi se taj podatak mogao koristiti za određivanje broja polica potrebnih za spremanje robe. Eksperimentalni podaci dobiveni su propuštanjem predmeta poznate visine kroz svjetlosnu zavjesu. Nakon svakog mjerenja zabilježena je vrijednost na ulazu s visokobrzinskim brojačem. Dobiveni podaci prikazani su u grafu na slici 6.1.



Slika 6.1: Rezultati mjerenja

Iako odnos stvarne visine i vrijednosti HSC-a nije linearan, zbog štednje vrlo ograničenih resursa PLC-a korištena je linearna aproksimacija. Uvrštavanjem dobivenih podataka u jednadžbu pravca određena je tražena relacija:

$$Visina, cm = 2,7 \cdot HSC$$

Ova relacija uvrštena je u naredbu CALCULATE unutar FB-a za kontrolu svjetlosne zavjese kao što je prikazano na slici 4.8. Daljnjim testiranjem svjetlosne zavjese potvrđeno je da je relacija točna i rezultati mjerenja su ispravni. Eventualne greške u mjerenju, najviše izražene kod predmeta manjih visina, kompenziraju se pribrajanjem dodatne police nakon dijeljenja visine s visinom police.

6.2 Testiranje sustava u ručnom režimu rada

Prije bilo kakvog pokretanja prigona proveden je postupak provjere svih ulaza. Za svaki senzor i granični prekidač provjereno je jesu li funkcionalni, aktiviraju li se u situacijama u kojima se očekuje da se aktiviraju i registrira li PLC pravovremeno njihove signale. U slučaju da se neki senzor ne aktivira kada bi se trebao, potrebno ga je namjestiti u pravilnu poziciju.

Nakon što je utvrđeno da su svi senzori ispravni, testirano je ručno vođenje vertikalne osi. Brzina ručnog vođenja za početak je postavljena na 10 % najveće brzine. Na početku testiranja dizalo je pomicano samo u kratkim pomacima kako bi se, u slučaju da se kočnica ne aktivira, spriječio nastanak štete. Najvažniji dio testiranja vertikalne osi je provjera rada kočnice kao najkritičnijeg dijela sustava. Provjereno je aktivira li se kočnica u slučaju pojave greške, pritiska tipke ESTOP i puštanja gumba za ručno vođenje osi na HMI-u. Nakon što je utvrđeno da kočnica i prigon vertikalne osi ispravno rade, dizalo je pokretano cijelim putem od donje do gornje referentne točke i provjereno je zaustavlja li se kada dođe do senzora referentnih točaka. Pokrenuta je funkcija referenciranja te je nakon toga testiran brojač polica. Ako je brojač ispravno napisan, očekuje se da će njegova vrijednost biti jednaka nuli svaki puta kada dizalo dođe u donju referentnu točku.

Sljedeći korak je testiranje horizontalne osi. Prije pokretanja prigona horizontalne osi, provjereno je upisivanje parametra 185 za aktiviranje funkcija drugog motora na VFD-u. Nakon što je PLC VFD-u poslao vrijednost 3, na zaslonu VFD-a provjereno je promijenila li se doista vrijednost tog parametra. Isto kao i kod vertikalne osi, horizontalna os vođena je ručno pri niskoj brzini dodiranjem tipke na HMI-u. Provjereno je aktiviraju li se senzori za poziciju lanca vertikalne osi u ispravnim trenutcima. Nakon toga u stanicu za utovar robe stavljena je ladica. Provjereno je završava li ladica na ispravnoj poziciji nakon povlačenja iz stanice za utovar na dizalo i s dizala na policu. Tijekom svakog dijela testiranja pritisnuta je tipka za zaustavljanje rada sustava kako bi se potvrdilo da se sve funkcije doista zaustavljaju kada se to očekuje. Na slici 6.2 prikazan je sustav u trenutku testiranja vertikalne osi.



Slika 6.2: Testiranje vertikalne osi

6.3 Testiranje sustava u automatskom režimu rada

Nakon što je utvrđeno da je sva upravljačka logika ispravna, sustav je prebačen u automatski režim rada. U automatskom režimu, provjereno je poziva li se ispravan korak sekvence pozivanjem svake od funkcija na zaslonu HMI-a. Pokrenuta je sekvenca *Izbroji police* te su u memoriju PLC-a upisane enkoderske pozicije svih policia.

U simulacijskom okruženju PLCSIM testiran je algoritam za traženje prostora u skladištu te je provjereno upisuju li se podaci o spremljenim ladicama na ispravan način u memoriju PLC-a. Na slici 6.3 prikazan je DB sa spremljenim enkoderskim pozicijama i rezultatima simulacije.

	Name	Data type	Start value	Monitor value	Retain	Accessible f...	Write...	Visible in ...	Setpoint
1	Static								
2	Storage	"Storage"							
3	BackRacks	Array[1..105] of "Re..."							
4	BackRacks[1]	"Rack"							
5	Rack	Struct							
6	isAvaila...	Bool	FALSE	FALSE					
7	StoreID...	Int	54	54					
8	StoreIDC...	Int	1	1					
9	Encoder...	DInt	92	92					
10	BackRacks[2]	"Rack"							
11	Rack	Struct							
12	isAvaila...	Bool	FALSE	FALSE					
13	StoreID...	Int	26	26					
14	StoreIDC...	Int	1	1					
15	Encoder...	DInt	270	270					
16	BackRacks[3]	"Rack"							
17	Rack	Struct							
18	isAvaila...	Bool	FALSE	FALSE					
19	StoreID...	Int	58	58					
20	StoreIDC...	Int	1	1					
21	Encoder...	DInt	450	450					
22	BackRacks[4]	"Rack"							
23	Rack	Struct							
24	isAvaila...	Bool	FALSE	FALSE					
25	StoreID...	Int	57	57					
26	StoreIDC...	Int	1	1					
27	Encoder...	DInt	631	631					
28	BackRacks[5]	"Rack"							
29	Rack	Struct							
30	isAvaila...	Bool	FALSE	FALSE					
31	StoreID...	Int	28	28					
32	StoreIDC...	Int	1	1					
33	Encoder...	DInt	809	809					

Slika 6.3: Okruženje PLCSIM i podatkovni blok s upisanim podacima

Nakon što je upravljački program testiran u simulacijskom okruženju, provedeno je testiranje sekvencu *Spremi* i *Izvadi*. Testiranje sekvenci ponavljano je tako dugo dok se sa sigurnošću nije moglo zaključiti da sustav radi na očekivani način. Nakon što je sustav u potpunosti testiran i sve eventualne greške u upravljačkom programu ispravljene, sustav je pušten u rad.

7 ZAKLJUČAK

U tvrtki Strojotehnika d.o.o. nabavljeno je rabljeno automatizirano regalno skladište. S obzirom da sustav nije bio u potpunosti funkcionalan, pokrenut je projekt revitalizacije cijelog sustava. Regalno skladište je sustav koji se sastoji u dva stupca skladišnog prostora u koje se spremaju ladice s robom. Između dva stupca smješteno je dizalo koje služi za dovođenje ladica do polica unutar skladišta. Roba se u ladice sprema u stanici za utovar robe. Nakon što se roba smjesti u ladicu, horizontalna os povlači je do dizala. Dizalo je odvodi do određene police gdje ju lanac horizontalne osi povlači s dizala na policu.

U sklopu projekta projektiran je novi upravljački sustav. Upravljački sustav temeljen je na programabilnom logičkom kontroleru Siemens S7-1214C DC/DC/DC. Za upravljanje prigonima horizontalne i vertikalne osi korišten je varijabilni frekvencijski pretvarač Mitsubishi FR-E840. Osim navedenih komponenti, nabavljen je novi ispravljač napona i komponente za distribuciju napona. Izrađen je novi elektroormar. Sustav je nadograđen operatorskim sučeljem za koje je napravljeno korisničko sučelje pomoću kojeg operater na jednostavan i intuitivan način upravlja sustavom i prati njegovo stanje.

Na temelju postojećeg standarda tvrtke, izrađen je upravljački PLC program. Sustav je zamišljen tako da se proces korištenja svodi na dvije glavne funkcije: *Spremi* i *Izvadi*. Upravljački program koncipiran je tako da se od operatera traži samo da unese broj ladice koju sprema ili želi izvaditi, pritiskom na gumb pozove željenu funkciju te eventualno vodi evidenciju o sadržaju ladica. Sve ostale funkcije provode se automatski unutar upravljačkog programa. Upravljački program sastoji se od logike za upravljanje horizontalnom i vertikalnom osi, upravljanje funkcijama svjetlosne zavjese, izračunavanje prostora potrebnog za spremanje robe, traženje pozicije slobodnih polica te sekvenci za upravljanje automatskim režimom rada.

Prije pokretanja, provedeno je parametriranje prigona obiju osi. Prigoni su parametrirani unosom potrebnih parametara u memoriju VFD-a pomoću računalnog alata FR Configurator 2. Nakon što je izrađen upravljački program, sustav je temeljito testiran. Testiranje se sastojalo od parametriranja svjetlosne zavjese, testiranja kočnice vertikalne osi, testiranja komunikacije između PLC-a i VFD-a, testiranja prigona obiju osi i na kraju testiranja sekvenci u automatskom režimu rada. Nakon što su otklonjene sve greške, sustav je pušten u rad te se aktivno koristi. U budućnosti je planirana nadogradnja sustava s čitačem bar kodova. Ova nadogradnja ukloniti će potrebu za ručnom evidencijom robe u ladici. Planirano je da svaki proizvod u skladištu ima jedinstveni bar kod. Na taj način, evidencija će biti svedena na skeniranje koda koji će se nakon toga automatski zapisati u memoriju.

LITERATURA

- [1] <https://www.progressiveautomations.com/blogs/news/the-evolution-of-automation>. Pristupljeno 6.12.2023.
- [2] <https://control.com/technical-articles/the-origin-story-of-the-plc/>. Pristupljeno 6.12.2023.
- [3] <https://www.engineering.com/story/programmable-logic-controllers-the-evolution-of-a-disruptive-technology>. Pristupljeno 27.11.2023.
- [4] <https://profinetuniversity.com/profinet-basics/profinet-communication-channels/>. Pristupljeno 11.12.2023.
- [5] <https://www.imperva.com/learn/application-security/osi-model/>. Pristupljeno 11.12.2023.
- [6] *Megamat Megalift*. tehnička dokumentacija. Strojotehnika d.o.o. 2023.
- [7] <https://www.wago.com/global/industrial-switches/industrial-switch/p/852-1106>. Pristupljeno 11.12.2023.
- [8] *S7-1200 Programmable Controller*. System manual. Verzija V4.3.0 02/2019. Siemens.
- [9] <https://www.automation.siemens.com/bilddb/search.aspx>. Pristupljeno 10.12.2023.
- [10] https://emea.mitsubishielectric.com/fa/products/drv/inv/fr_e800/fre800/fr-e840-0095epb-60.html. Pristupljeno 10.12.2023.
- [11] *DUOmetric SCAN 1.11-min*. dokumentacija. DUOmetric GmbH. 1999.
- [12] *FR-E800 Instruction Manual (Function)*. dokumentacija. Mitsubishi Electric. 2023.
- [13] *FR-E800 Instruction Manual (Communication)*. dokumentacija. Mitsubishi Electric. 2023.
- [14] https://www.engineeringtoolbox.com/synchronous-motor-frequency-speed-d_649.html. Pristupljeno 18.12.2023.
- [15] <https://www.practicalmachinist.com/forum/threads/what-really-is-vector-sensorless-drive.276144/>. Pristupljeno 18.12.2023.