



UNIVERZITET U SARAJEVU
ELEKTROTEHNIČKI FAKULTET
ODSJEK ZA AUTOMATIKU I ELEKTRONIKU

Praktikum za korištenje programabilnog logičkog kontrolera TM200CE24T u procesima upravljanja

ZAVRŠNI RAD
- PRVI CIKLUS STUDIJA -

Student:
Anela Mutilović

Mentor:
Docent dr. sc. Nedim Osmić, dipl. ing. el.

Sarajevo,
juli 2024.

Sažetak

Rad je baziran na opisu korištenja programabilnog logičkog kontrolera TM200CE24T u procesima upravljanja. Na samom početku rada navedene su osnovne karakteristike programabilnih logičkih kontrolera, opisan princip rada te njihove prednosti i mane. Detaljnije je opisan kontroler proizvođača Schneider Electric koji će se koristiti za upravljanje procesima. Upravljanje navedenim kontrolerom moguće je pomoću softvera EcoStruxure Machine Expert - Basic. Detaljno je opisano okruženje softvera, kao i mogućnost konfiguracije hardvera, te simulacije programiranog procesa. Obradeno je pet načina programiranja koje podržava kontroler posredstvom navedenog softvera. Za svaki tip programiranja su urađeni primjeri zadataka koji su odabrani s ciljem obuhvatanja što većeg broja komponenti određenog programskog jezika. Time su prikazane mogućnosti kontrolera, kao i dato detaljno uputstvo korištenja istog. U posljednjem poglavljju ovog rada realiziran je sistem linije za sortiranje. Prikazan je prijedlog rješenja u skladu sa tehničkim opisom sistema. Priložena je lista signala, izbor komponenti i popis opreme, kao i predračun korištenih komponenti. Prikazane su sheme djelovanja, priključni plan i nacrt razvodnog ormara. Izvršena je softverska realizacija korištenjem softvera EcoStruxure Machine Expert - Basic, programiranjem na bazi sekvencijalnih funkcijskih grafikona i fukcionalnih blok dijagrama.

Ključne riječi: TM200CE24T, PLC, EcoStruxure Machine Expert - Basic

Abstract

This paper is based on the description of the use of the program logic controller TM200CE24T in control processes. At the very beginning of the work, the basic characteristics of programmable logic controllers are listed, the principle of operation and their advantages and disadvantages are described. The Schneider Electric controller that will be used for process management is described in more detail. Management of the mentioned controls is possible using EcoStruxure Machine Expert - Basic software. The software environment is described in detail, as well as the possibility of hardware configuration and simulation of the programmed process. Five programming modes supported by the controller through the mentioned software are covered. Examples of tasks were created for each type of programming, which were selected with the aim of including as many components as possible of a specific programming language. This shows the possibilities of the controller, as well as detailed instructions for its use. In the last chapter of this work, the sorting line system was realized. A proposed solution is presented in accordance with the technical description of the system. A list of signals, a selection of components and a list of equipment, as well as an estimate of the components used, are attached. Action schemes, connection plan and distribution cabinet plan are shown. The software implementation was completed using EcoStruxure Machine Expert - Basic software, programming based on sequential functional charts and functional block diagrams.

Keywords: TM200CE24T, PLC, EcoStruxure Machine Expert - Basic

Elektrotehnički fakultet, Univerzitet u Sarajevu
Odsjek za automatiku i elektroniku
Docent dr. sc. Nedim Osmić dipl. ing. el.
Sarajevo, januar 2024.

Postavka zadatka završnog rada I ciklusa:

U završnom radu je potrebno predstaviti opštu arhitekturu programabilnih logičkih kontrolera te ukratko objasniti princip rada programabilnog logičkog kontrolera. Potrebno obraditi osnovni rad sa softverom koji će se korisiti (EcoStruxure Machine Expert - Basic) i načine programiranja programabilnog logičkog kontrolera koje podržava navedeni softver. U okviru rada potrebno je objasniti način kreiranja projekata te objasniti različite tipove programiranja (Ljestvičasti dijagrami, FBD, Grafcet (List), Grafcet (SFC)). Dati osnovna pojašnjenja najčešće korištenih komponenti koje se koriste pri programiranju te je potrebno demonstrirati korištenje navedenih komponenti kroz par primjera. Od predstavljenih načina programiranja odabrati najpodesniji pomoću koga će se riješiti problem upravljanja sistema linije za sortiranje sa detaljnim objašnjenjem načina realizacije rješenja.

Polazna literatura:

- [1] EcoStruxure Machine Expert - Basic Operating Guide, Schneider Electric, France, 2022.
- [2] Schneider Electric, Product data sheet – TM200CEC24T. Schneider Electric, Le Creusot, France, 2023.
- [3] K. Kamel and E. Kamel, eds., Programmable logic controllers: Industrial control. New York, USA: McGraw-Hill Education, 2013

Docent dr. sc. Nedim Osmić dipl. ing. el.

Izjava o autentičnosti radova

Završni rad I ciklusa studija

Ime i prezime: Anela Mutilović

Naslov rada: Praktikum za korištenje programabilnog logičkog kontrolera TM200CE24T u procesima upravljanja

Vrsta rada: Završni rad Prvog ciklusa studija

Broj stranica: 84

Potvrđujem:

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

Sarajevo, juli 2024.

Potpis:

Anela Mutilović

Sadržaj

Popis slika	viii
Popis tabela	ix
1 Uvod	1
1.1 Obrazloženje teme	1
1.2 Struktura rada	3
2 Programabilni logički kontroleri - PLC	4
2.1 Arhitektura PLC	4
2.2 Princip rada	6
2.3 Načini programiranja	7
2.4 Tipovi PLC	9
2.5 Schneider Electric TM200CE24T	9
3 EcoStruxure Machine Expert - Basic	11
3.1 Uvod u EcoStruxure Machine Expert - Basic	11
3.2 Kreiranje projekta	13
3.3 Konfiguracija hardvera	14
3.4 Radni prostor za programiranje	15
3.4.1 Posebne funkcije	18
3.5 Prozor za puštanje u rad	20
3.6 Simulacija programa	23
4 Uputstvo za korištenje kontrolera TM200CE24T	27
4.1 Programiranje ljestvičastim dijagramima	27
4.2 Primjeri ljestvičastog programiranja	30
4.2.1 Primjer 1.	30
4.2.2 Primjer 2.	31
4.2.3 Primjer 3.	32
4.3 Programiranje instrukcijskim listama	35
4.4 Primjeri programiranja instrukcijskim listama	36
4.4.1 Primjer 1.	36
4.4.2 Primjer 2.	37
4.4.3 Primjer 3.	38
4.5 FBD programiranje	40
4.6 Primjeri FBD programiranja	42
4.6.1 Primjer 1.	42
4.6.2 Primjer 2.	44

4.6.3	Primjer 3	46
4.7	GRAFCET Lista programiranje	48
4.8	GRAFCET (SFC) progamiranje	51
4.8.1	Principi Grafcet (SFC)	52
4.8.2	SFC način programiranja	53
4.9	Primjeri SFC progamiranja	55
4.9.1	Primjer 1	55
4.9.2	Primjer 2	59
4.9.3	Primjer 3	61
5	Primjer realizacije linije za sortiranje	68
5.1	Tehnički opis	68
5.2	Prijedlog rješenja	69
5.3	Lista signala	69
5.4	Izbor komponenti i opis opreme	69
5.5	Predračun opreme	71
5.6	Grafički dio	72
5.6.1	Principijelna shema	72
5.6.2	Sheme djelovanja	73
5.7	Priklučni plan	76
5.8	Nacrt ormara	77
5.9	Softversko rješenje	78
6	Zaključak	83
Literatura		84

Popis slika

2.1	Osnovne komponente PLC uređaja	5
2.2	Princip rada PLC-a	6
2.3	Zastupljenost proizvođača PLC (istraživanje 2021. godina) [1]	7
2.4	Kompaktni PLC [2]	9
2.5	Modularni PLC [3]	9
2.6	Kartična izvedba PLC [4]	9
2.7	Programabilni logički kontroler TM200CE24T (Schneider Electric) [5]	10
3.1	Prikaz faza izrade projekta u EcoStruxure Machine Expert - Basic.	12
3.2	Kreiranje novog projekta u EcoStruxure Machine Expert - Basic.	13
3.3	Postavke novog projekta u EcoStruxure Machine Expert - Basic.	13
3.4	Konfiguracija hardvera u EcoStruxure Machine Expert - Basic.	14
3.5	Izbor hardvera u EcoStruxure Machine Expert - Basic.	14
3.6	Radni prostor za programiranje u EcoStruxure Machine Expert - Basic.	15
3.7	Kreiranje "POU" u EcoStruxure Machine Expert - Basic.	15
3.8	Kreiranje korisnički definirane funkcije u EcoStruxure Machine Expert - Basic.	16
3.9	Prozor korisnički definirane funkcije u EcoStruxure Machine Expert - Basic.	16
3.10	Kreiranje korisnički definiranog funkcionalnog bloka u EcoStruxure Machine Expert - Basic.	17
3.11	Prozor korisnički definiranog funkcionalnog bloka u EcoStruxure Machine Expert - Basic.	17
3.12	Primjer reverzibilnosti programa - Ladder dijagram.	19
3.13	Primjer reverzibilnosti programa - Instrukcijske liste.	19
3.14	Prikaz prozora za puštanje u rad.	20
3.15	Prikaz prozora za ažuriranje firmware-a logičkog kontrolera.	20
3.16	Prikaz prozora za upravljanje memorijom logičkog kontrolera.	21
3.17	Prikaz prozora za informacije trenutnog stanja logičkog kontrolera.	22
3.18	Prikaz prozora za upravljanje satom realnog vremena logičkog kontrolera.	23
3.19	Prikaz prozora za manipulaciju stanja ulaza i izlaza logičkog kontrolera.	24
3.20	Prikaz iskočnog prozora prije pokretanje simulacije.	24
3.21	Prikaz prozora za upravljanje vremenom simulatora logičkog kontrolera.	25
3.22	Prikaz prozora za upravljanje analognim ulazima logičkog kontrolera.	25
3.23	Prikaz prozora za konfiguraciju praćenja izlaza logičkog kontrolera.	26
3.24	Prikaz prozora za praćenje stanja izlaza logičkog kontrolera.	26
4.1	Rješenje primjera pomoću ljestvičastog programiranja.	30
4.2	Rješenje drugog primjera pomoću ljestvičastog programiranja.	31
4.3	Prikaz hemijskog postrojenja.	32
4.4	Rješenje trećeg primjera pomoću ljestvičastog programiranja.	34

4.5 Rješenje prvog primjera pomoću instrukcijske liste.	36
4.6 Rješenje drugog primjera pomoću instrukcijske liste.	37
4.7 Prikaz hemijskog postrojenja.	38
4.8 Rješenje trećeg primjera pomoću instrukcijske liste.	39
4.9 Rješenje prvog primjera pomoću funkcionalnih blok dijagrama.	42
4.10 Simboli dodijeljeni ulaznim i izlaznim elementima.	43
4.11 Rješenje prvog primjera pomoću funkcionalnih blok dijagrama sa definisanim simbolima.	43
4.12 Prikaz sistema skladišta.	44
4.13 Rješenje drugog primjera pomoću funkcionalnih blok dijagrama.	45
4.14 Korišteni ulazi i izlazi.	46
4.15 Rješenje trećeg primjera pomoću FBD programiranja.	47
4.16 Primjer programiranja GRAFCET (Lista).	48
4.17 Primjer sekvencijalnog programiranja - Ladder dijagrami.	49
4.18 Primjer sekvencijalnog programiranja - GRAFCET (Lista).	50
4.19 Primjer post-obrade.	50
4.20 Otvaranje prozora za SFC programiranje.	53
4.21 Kreiranje i povezivanje step blokova.	53
4.22 Prikaz svih step blokova i tranzicija.	54
4.23 Prozor za programiranje tranzicija.	54
4.24 Ilustrativni prikaz sistema.	55
4.25 Prikaz koda za početni step 1.	56
4.26 Prikaz koda za step 2.	56
4.27 Prikaz koda za step 3.	56
4.28 Prikaz koda za prvu tranziciju.	56
4.29 Prikaz koda za drugu tranziciju.	57
4.30 Prikaz koda za treću tranziciju.	57
4.31 Prikaz koda za četvrту tranziciju.	57
4.32 Prikaz realizacije transportne trake.	58
4.33 Graf automata.	59
4.34 Realizacija Mooreovog automata pomoću SFC.	60
4.35 Prikaz predloženog rješenja.	61
4.36 Korišteni ulazi i izlazi.	62
4.37 Prikaz realizacije kafe aparata.	63
4.38 Prvi step (unos novca).	64
4.39 Stepovi 3, 7 i 8 (pravljenje napitka).	64
4.40 Step 4 (proračun kusura).	65
4.41 Step 4 i 5 (vraćanje kusura).	65
4.42 Pomoći POU.	66
4.43 Prikaz nekih tranzicija.	67
5.1 Linijska sortiranje.	68
5.2 Principijelna shema povezivanja.	72
5.3 Shema povezivanja napajanja.	73
5.4 Shema povezivanja motora.	73
5.5 Shema povezivanja ulaza PLC-a.	74
5.6 Shema povezivanja izlaza PLC-a.	74
5.7 Shema povezivanja senzora.	75

5.8	Shema povezivanja pneumatskih elemenata.	75
5.9	Shema povezivanja stezaljki.	76
5.10	Prikaz razvodnog ormara.	77
5.11	Realizacija linije za sortiranje pomoću SFC.	78
5.12	Prikaz početnog stanja.	79
5.13	Prikaz stanja detekcije.	79
5.14	Prikaz stanja klipa A.	80
5.15	Prikaz stanja aktiviranja prese i klipa B.	80
5.16	Prikaz stanja zaglavljenog klipa.	80
5.17	Prikaz prve tranzicije.	81
5.18	Prikaz druge tranzicije.	81
5.19	Prikaz treće tranzicije.	81
5.20	Prikaz četvrte tranzicije.	81
5.21	Prikaz pete tranzicije.	81
5.22	Prikaz šeste tranzicije.	81
5.23	Prikaz sedme tranzicije.	81
5.24	POU - Hitan stop.	82

Popis tabela

4.1	Grafički elementi za ljestvičasto programiranje.	29
4.2	Lista signala za drugi primjer.	31
4.3	Podržani tipovi instrukcija.	35
4.4	Modifikatori koji se dodjeljuju zagradama.	35
4.5	Lista signala za drugi primjer.	37
4.6	Grafički elementi za FBD programiranje.	41
4.7	Lista signala za prvi primjer	42
4.8	Sistemski biti za Grafset kontrolu.	49
4.9	Kontrolni biti u Grafset (SFC).	52
4.10	Tablica signala za prvi primjer.	55
4.11	Lista signala za Mooreov automat.	59
5.1	Lista signala linije za sortiranje.	69
5.2	Predračun opreme.	71

Poglavlje 1

Uvod

Nekada, u prošlosti ljudi su morali samostalno izvršavati razne upravljačke zadatke. Razvojem i napretkom tehnologije, pojavila se ideja za izradu upravljačkog uređaja koji bi imao mogućnost jednostavnog reprogramiranja za upravljanje i obavljanje zadataka. Jedan od najefikasnijih pristupa upravljanju industrijskim postrojenjima je pomoću programabilnih logičkih kontrolera - PLC (eng. *PLC- programmable logic controller*). U ovom završnom radu prikazat će se mogućnosti upravljanja i obavljanja raznih zadataka pomoću EcoStruxure Machine Expert - Basic i programabilnog logičkog kontrolera TM200CE24T. U nastavku rada će detaljnije, kroz poglavlja, biti pojašnjena navedena tema.

1.1 Obrazloženje teme

EcoStruxure Machine Expert - Basic je softversko okruženje koje omogućava razvoj, konfiguraciju i puštanje u pogon cijelog razmatranog sistema uključujući kontrolu motora i logiku. Navedeno softversko okruženje omogućava programiranje programabilnih logičkih kontrolera proizvođača Schneider Electric, te predstavlja optimalno upravljačko rješenje za većinu industrijskih procesa. PLC omogućava praćenje i upravljanje sistema u realnom vremenom, što obezbeđuje precizan i tačan rad upravljanja procesa. Programabilni logički kontroleri su u širokoj upotrebi u sistemima industrijske automatizacije za upravljanje mašinama i procesima. Nude brojne prednosti u odnosu na tradicionalne upravljačke sisteme zasnovane na relejima, uključujući povećanu fleksibilnost, pouzdanost i lakoću održavanja [6]. PLC-ovi su programirani pomoću specijalizovanog programskega jezika i mogu se programirati za obavljanje širokog spektra zadataka, uključujući upravljanje motorima, relejima i drugim vrstama električnih i mehaničkih uređaja [7]. Glavne prednosti korištenja PLC za automatizaciju procesa su [8]:

- **Fleksibilnost:** PLC sistemi se mogu programirati za obavljanje širokog spektra zadataka, od jednostavne kontrole uključivanja/isključivanja do složene kontrole kretanja i automatizacije procesa. To ih čini vrlo fleksibilnim i prilagodljivim promjenjivim uslovima i zahtjevima.
- **Pouzdanost:** Dizajnirani su da rade u zahtjevnim industrijskim okruženjima, sa karakteristikama kao što je ugrađena zaštita od električne buke i ekstremnih temperatura. To ih čini pouzdanim od tradicionalnih kontrolnih sistema baziranih na relejima.
- **Brže vrijeme odziva:** Kontroleri su sposobni da reaguju na ulaze i proizvode izlaze mnogo brže od tradicionalnih sistema zasnovanih na relejima.

- **Preciznost:** PLC-ovi su sposobni za vrlo preciznu kontrolu, što ih čini idealnim za aplikacije kao što su automatizacija procesa i robotika.
- **Jednostavnost:** Kontroleri su modularnog dizajna, što znači da se neispravne komponente mogu lako zamijeniti bez uticaja na ostatak sistema. Imaju i ugrađene dijagnostičke funkcije koje olakšavaju rješavanje problema.
- **Skalabilnost:** PLC sistemi se mogu lako proširiti dodavanjem dodatnih modula ili kontrolera, što ih čini visoko skalabilnim.
- **Isplativost:** Iako početni trošak PLC sistema može biti veći od tradicionalnog sistema baziranog na relejnoj logici, često su isplativiji na duge staze. To je zato što zahtijevaju manje održavanja i imaju duži vijek trajanja.
- **Sigurnost:** Programabilni logički kontroleri se mogu programirati da obavljaju sigurnosne funkcije kao što su isključenja u nuždi i zaštita mašine. Navedeno poboljšava ukupnu sigurnost u industrijskom okruženju.

Sve navedene prednosti čine programabilne logičke kontrolere odličnim izborom za upravljanje industrijskim procesima. Naravno kontroleri kako imaju prednosti imaju i svoje nedostatke. Nedostaci korištenja programabilnih logičkih kontrolera su [7]:

- **Složenost:** PLC-ovi mogu biti složeni za programiranje i održavanje, posebno za korisnike koji nisu upoznati sa programskim jezikom koji se koristi za njihovo programiranje.
- **Ograničena procesorska snaga:** Programabilni logički kontroleri nisu sposobni kao neki drugi tipovi sistema kao što je DCS, te zbog toga nisu u stanju da obrađuju složene zadatke ili velike količine podataka. [7].
- **Trošak:** Dok su PLC-ovi generalno isplativiji od drugih tipova industrijskih kontrolnih sistema, i dalje mogu biti skupi za kupovinu i održavanje.
- **Ograničena proširivost:** PLC-ovi su dizajnirani za specifične zadatke i procese i možda se neće moći lako proširiti ili modificirati za obavljanje novih zadataka.

Posebna vrsta kontrolera je programabilni logički kontroler TM200CE24T sa 24 ulaza/izlaza, te sa kontrolom kretanja po dvije ose što je posebno pogodno prilikom rukovanja, pakovanja i podizanja raznih materijala. Upravljanje navedenim PLC pomoću softvera predstavlja najpogodnije rješenje za automatske sisteme koji u svom radu zahtijevaju jednostavno kretanje i veliku fleksibilnost.

Cilj praktikuma je omogućiti studentima da se stekne praktično iskustvo u programiranju, konfiguraciji i upravljanju PLC-ovima kao ključnom tehnologijom u industrijskoj automatizaciji. Kroz praktične primjere i simulacije rješenja, studenti mogu naučiti kako implementirati kontrolu motora, logiku i upravljanje pokretnim sistemima korištenjem softvera EcoStruxure Machine Expert - Basic i PLC-a TM200CE24T. Ovaj završni rad pomaže u razvijanju vještina programiranja koje su neophodne za rješavanje raznih zadataka u industriji i optimizaciji proizvodnih procesa.

1.2 Struktura rada

Završni rad je organiziran na sljedeći način:

U prvom poglavlju dat je kratak uvod u temu koja će biti temeljito analizirana u nastavku. U prvom poglavlju dato je i detaljnije obrazloženje teme, te opisan razlog korištenja programabilnog logičkog kontrolera za rješenje industrijskih procesa.

U nastavku rada, u drugom poglavlju, detaljno je opisan programabilni logički kontroler, ističući njegove ključne karakteristike, arhitekturu i specifičnosti modela TM200CE24T proizvođača Schneider Electric. Naveden je princip rada i načini programiranja koji čine osnovu za njegovu praktičnu primjenu u različitim industrijskim sistemima.

Tematika trećeg poglavlja je analiza softvera EcoStruxure Machine Expert-Basic, koji predstavlja ključni alat za programiranje i upravljanje kontrolerom TM200CE24T. Objasnjeno je kako okruženje softvera, tako i njegove mogućnosti za integraciju, dijagnostiku i simulaciju, što omogućava korisnicima efikasno razvijanje i optimizaciju industrijskih automatskih sistema.

Zbog boljeg razumjevanja izabrane teme i korištenih komponenti, date su, u četvrtom poglavlju, praktične smjernice za korištenje kontrolera TM200CE24T u različitim procesima upravljanja. Kroz konkretno postavljene primjere dato je objašnjenje za implementaciju naprednih funkcija, te je prikazano automatizirano softversko rješenje za industrijske procese.

Fokus petog poglavlja se odnosi na praktičnu primjenu kontrolera kroz realizaciju projekta upravljanja linijom za sortiranje. Priložen je kompletan opis projekta, uključujući tehničke i grafičke aspekte projekta, kao i implementaciju softverskog rješenja unutar EcoStruxure Machine Expert - Basic koristeći kontroler TM200CE24T. Ovaj dio rada pruža uvid u konkretne primjene PLC-a u industriji, ističući tako sve njegove prednosti, čime se podstiče modernizacija industrijskih procesa.

U posljednjem šestom poglavlju, dat je zaključak i smjernice za budući rad.

Poglavlje 2

Programabilni logički kontroleri - PLC

Automatizacija industrijskih procesa i sistema danas je gotovo nemoguća bez programabilnih uređaja. Najrasprostranjenija je upotreba programabilnih logičkih kontrolera, ili kraće PLC, koji svoje korijene imaju od kraja šesdesetih i ranih sedamdesetih godina [9]. Razvojem PLC cijeli proces je olakšan, zbog njegove mogućnosti jednostavnog reprogramiranja, fleksibilnosti i veoma visoke pouzdanosti upravljanja.

PLC je prvi put razvijen u automobilskoj industriji, zbog potrebe za fleksibilnim, lako programabilnim i robusnim kontrolerima koji će zamijeniti žičane releje, tajmere i sekvene. Prvi programabilni logički kontroler potiče iz 1968. godine [10], a za taj veliki napredak zaslužna je firma Modicon pod vodstvom Dick Morley-a. Firma Modicon je dobila ime po engleskoj riječi za modularni digitalni kontroler koja glasi *M*ODular *D*igital *C*ONtroller. Sve do razvoja prvog personalanog računara nazivao se samo programabilni kontroler (PC), a nakon toga je promijenio ime u ono koje se koristi i danas - PLC.

Po definiciji, PLC je digitalni elektronski uređaj koji instrukcije napisane bilo kojim načinom programiranja smješta u programabilnu memoriju, u cilju efikasnog upravljanja različitim procesima i obavljanja različitih funkcija kao što su brojanje, tajming, logičke i aritmetičke operacije [11].

Također, PLC se definiše i kao industrijski računar prilagođen radu u industrijskim uslovima, otporan na prašinu i vlagu, a koji se može lako ugrađivati u već postojeće ili u nove sisteme [12].

Neke od prednosti korištenja PLC su: jednostavno programiranje i komunikacija, standardizacija, reprogramiranje te jako velika pouzdanost za ispravno upravljanje.

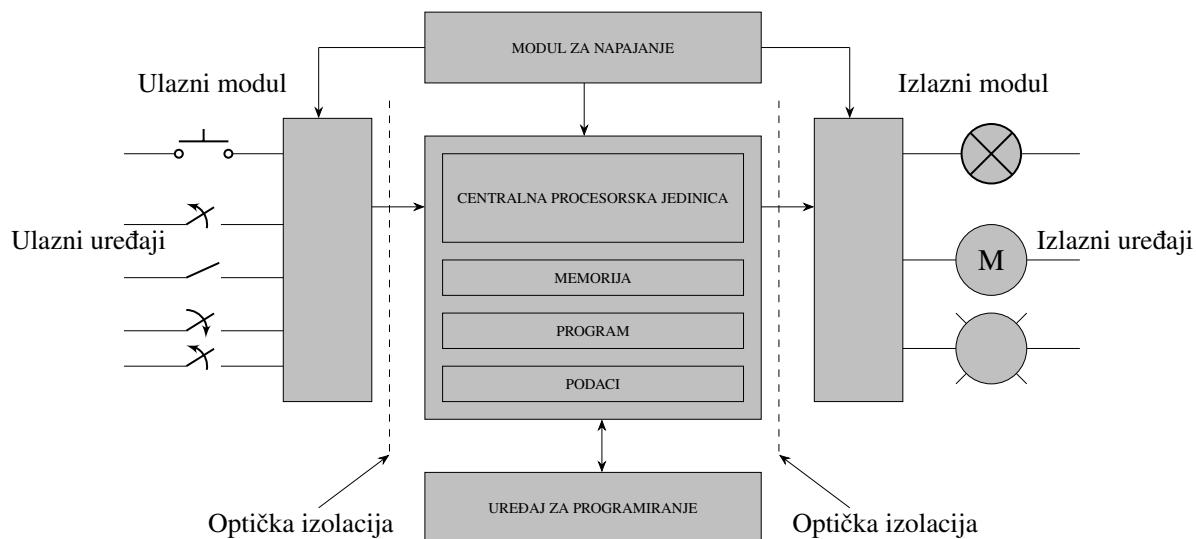
Budućnost razvoja PLC je teško predvidjeti, ali pretpostavlja se da će se u narednom periodu napredovati ubrzanim tempom, u skladu sa napretkom ostale tehnologije.

2.1 Arhitektura PLC

Svaki PLC se u osnovi sastoji od istih hardverskih komponenti. Osnovne komponente koje čine arhitekturu PLC prikazana je na slici 2.1, te se može uočiti da se sastoje od:

- Modula za napajanje
- Centralna procesorska jedinica (CPU - eng. *Central Processing Unit*)
- Ulaznih modula (digitalni ili analogni)
- Izlaznih modula (digitalni ili analogni), te

- Uređaja za programiranje (PC)



Slika 2.1: Osnovne komponente PLC uređaja.

Centralna procesorska jedinica ili kraće CPU predstavlja glavni dio svakog PLC-a. Obrađuje logički sve analogne ili digitalne ulaze u skladu sa programom koji je upisan od strane korisnika, te prema dobivenim rezultatima upravlja izlaznim uređajima. CPU upravlja komunikacijom sa ostalim komponentama sistema, kao i načinima donošenja logičkih odluka pa se iz tog razloga često naziva i mozak PLC-a.

Memorija, program i podaci nisu fizički vidljivi, ali su neophodni za rad PLC uređaja. Postoje dvije vrste memorija koje se koriste pri radu sa PLC, a to su RAM (eng. *Random Access Memory* - radna memorija), ROM (eng. *Read Only Memory*) i EEPROM (eng. *Electricaly Erasable Programmable Read Only Memory* - trajno pamti podatke). Programi se izvršavaju iz RAM memorije, koja je kod većine PLC uređaja vezana sa baterijom radi osiguravanja rada programa pri prestanku napajanja. U ROM memoriji smješten je operativni sistem koji se ne može mijenjati, dok su u RAM memoriji pored programa, smješteni brojači, vremenske varijable i I/O.

Modul za napajanje predstavlja osnovni dio za rad PLC. To je najrobustniji modul neosjetljiv na smetnje koje dolaze iz mreže. Modul se napaja mrežnim izmjeničnim naponom 230V AC koji se pretvara u istosmjerni napon, najčešće 24V DC potreban za rad kontrolera i drugih komponenti u sistemu.

Uređaj za programiranje omogućava korisniku unošenje programa koji će se izvršavati, te komunikacijskim kanalom omogućava prenos informacija i tako program upisuje u memoriju PLC-a za izvršavanje. Programatori mogu biti industrijski računari, personalni računari, ručni terminali i slično.

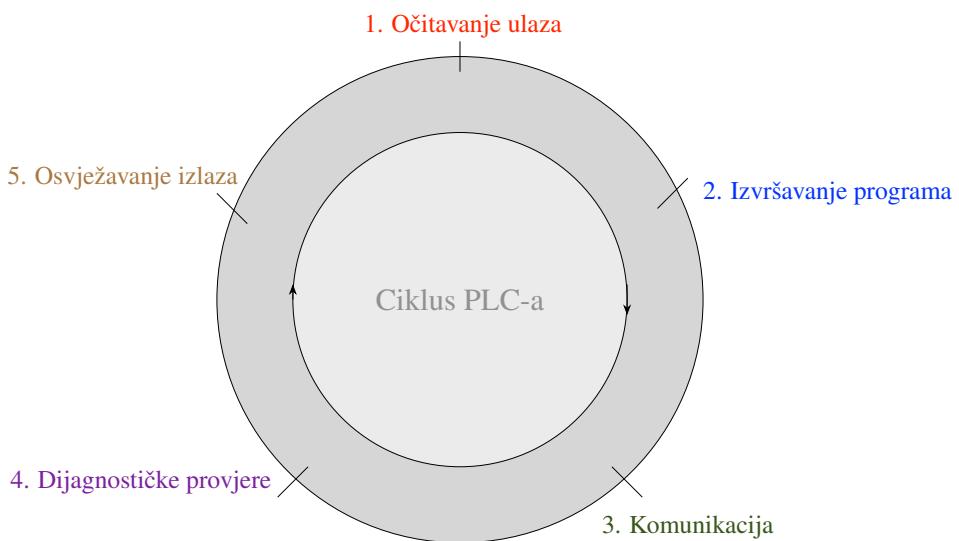
Ulazni modul procesira digitalne ili analogne ulazne signale, pretvara ih u signale pogodne za PLC i prenosi ih do njega. Dodatna uloga modula je da osigura zaštitu kontrolera od svih vanjskih utjecaja i smetnji tako što su ulazni signali pomoću optičkih izoliranih krugova odvojeni od direktnе veze sa PLC. Digitalni ulazni signali su sa sklopki, senzora, tastera i slično,

dok analogni ulazni signali mogu biti naponski signal 0-10 V DC s mjernog pretvarača pritiska, temperature i sl.

Izlazni modul ostvaruje komunikaciju između PLC-a sa izlaznim uređajima i jedinicama uz obaveznu konverziju signala sa kontrolera u signal koji je pogodan za upravljanje mašinama ili procesima. Kao i ulazni modul, izlazni modul posjeduje optičku izolaciju za zaštitu PLC od vanjskih nepoželjnih djelovanja. Na izlazne stezaljke spajaju se izvršni uređaji kojima PLC šalje digitalne ili analogne signale te pomoću istih upravlja procesima. Na digitalne izlaze moguće je spojiti releje, startere motora, ventila, pumpi, signalne lampice i slično, dok se na analogne izlaze mogu spojiti strujni signali potrebni za prikazivanje veličina na instrumentima, reference brzine potrebne za frekventni pretvarač, PID regulirane veličine i slično.

2.2 Princip rada

Princip rada kontrolera najlakše je opisati dijagramom prikazanim na slici 2.2.



Slika 2.2: Princip rada PLC-a.

Programabilni logički kontroler svakom promjenom ulaznog signala mora korigovati izlazne signale u skladu sa novim ulazom. Obrada podataka se vrši ciklički u beskonačnoj petlji. Sve radnje koje PLC ciklično izvršava nazivamo programskim ciklusom.

Jedan programske ciklus sačinjen je iz 5 osnovnih faza:

1. **Očitavanje ulaza** je prva faza u programskom ciklusu. Procesor konstantno očitava stanja ulaznih analognih ili digitalnih signala i upisuje ih u memoriju za ulazne podatke.
2. **Izvršavanje programa** predstavlja drugu fazu u radu PLC. Procesor u ovoj fazi izvršava upisani program od strane korisnika, u skladu sa ulaznim podacima sniljenim u memoriji. Rezultati dobiveni izvršavanjem programa pohranjuju se u memoriju koja je određena za izlazne podatke. Procesor upisani program izvršava sekvencijalno, naredbu po naredbu i u skladu s tim ažurira memoriju.
3. **Komunikacija** je treća faza u programskom ciklusu rada PLC-a. U ovoj fazi ostvaruje komunikacija između procesora i uređaja povezanih na kontroler.

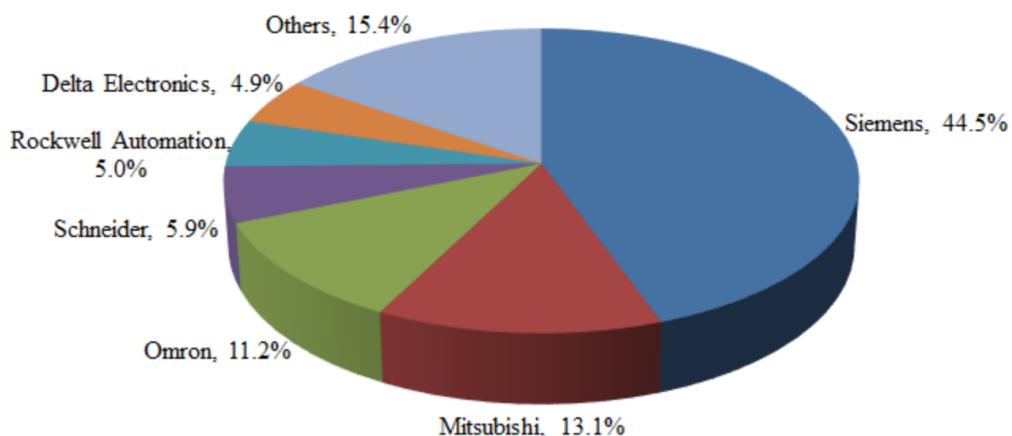
4. **Dijagnostičke provjere** obuhvataju provjeru operativnog sistema, memorije i statusa ulaznih i izlaznih modula, te čine četvrту fazu u ciklusu. U ovoj fazi se ažuriraju tajmeri i brojači potrebni za upravljanje sistemom.

5. **Osvježavanje izlaza** je posljednja faza u ciklusu rada. Rezultati iz memorije za izlazne podatke upisuju se na izlazne module, te završetkom upisa pokreće se proces upravljanja.

Trajanje jednog programskog ciklusa je jako bitno pri izboru PLC-a. Pretežno, dužina trajanja jednog programskog ciklusa iznosi od 0.1ms do desetak milisekundi. Za oko 500 programske naredbi ciklus iznosi oko 1,5 ms [6].

2.3 Načini programiranja

Proizvođači programabilnih logičkih kontrolera uz svoj uređaj isporučuju i programski jezik koji je u skladu sa standardom IEC 61131-1 (eng. *IEC - International Electrotechnical Commission*). Danas, u svijetu postoji mnogo proizvođača PLC-a, neki od najpoznatijih su Schneider Electric, Siemens, Omron, Mitsubishi itd. Na slici 2.3 prikazana je zastupljenost proizvođača na tržištu.



Slika 2.3: Zastupljenost proizvođača PLC (istraživanje 2021. godina) [1].

Postoji mnogo vrsta PLC, stoga je potrebno da programiranje istih raznim programskim jezicima bude standardizovano. Po ranije spomenutom IEC 61131-1 standardu za programiranje PLC, postoji osnovna podjela programskih jezika na tekstualne i grafičke.

Tekstualni programski jezici su:

- Instrukcijske liste (IL – eng. *Instruction List*)
- Struktuirani tekst (ST – eng. *Structured Text*)

Dok, grafički programski jezici su:

- Ljestvičasto programiranje (LD – eng. *Ladder Diagram*)
- Funkcionalni blok dijagrami (FBD – eng. *Function Block Diagram*)
- Sekvencijalni funkcijalni dijagrami (SFC - eng. *Sequential Function Chart*)

Pojedini proizvođači PLC nude mogućnost programiranja procesa u manje zastupljenim jezicima za ovu primjenu, kao što su BASIC i C programske jezike. U nastavku će detaljnije biti pojašnjen svaki od načina programiranja.

Instrukcijska lista predstavlja tip tekstualnog programskog jezika za PLC. U upotrebi je za procese koje je teško riješiti preko drugih programskih jezika. Instrukcijske liste su slične asembleru, naredbe se obavljaju redoslijedom koji je naveden u programu, sekvensijalno, od vrha prema dnu, a zatim ponovo kreće ispočetka. Najpogodniji je način programiranja za iskusne programere.

Struktuirani tekst spada u tekstualne programske jezike za programiranje kontrolera. Koristi se kao veoma koristan alat za složenije zadatke. Prilikom pisanja koda omogućeno je korištenje algoritama i matematičkih funkcija.

Ljestvičasto programiranje je jedan od tipova grafičkog programiranja. Osnovni dio programa je građen od mreža, dok se svaka mreža sastoji od simbola koji predstavljaju instrukcije i upravljačke komande. Program se izvodi paralelno, s lijeva na desno. Tačnije, odozgo prema dolje, mreža po mrežu, zatim odozgo prema dolje u mreži, te konačno s lijeva na desno u jednoj mreži.

Drugi tip grafičkog programiranja predstavljaju **funkcionalni blok dijagrami**. Programiranje u FBD-u je zasnovano na blokovima. Dakle, svi ulazi, izlazi i naredbe su predstavljene blokovima. Programiranje pomoću FBD svodi se na međusobno povezivanje blokova da bi se ostvarila potrebna funkacionalnost. Na ulazu blokova su veličine potrebne za izvršenje zadatka, dok na izlazu iz bloka su generirani izlazni signali - rezultati izvršene operacije u bloku.

Sekvensijalni funkcijski dijagrami su još jedan tip grafičkog načina programiranja. Glavna prednost SFC programiranja je što je omogućeno paralelno izvođenje programskih operacija tokom jednog ciklusa PLC-a. Glavne komponente ovog načina programiranja su programski koraci, tranzicije između programskih koraka, te direktnе veze između koraka i tranzicija. Također, bitno je napomenuti da se svaki blok unutar programa može programirati bilo kojim tipom programiranja.

2.4 Tipovi PLC

Trenutno na tržištu postoji veliki broj PLC-ova koji su podešeni prema potrebama korisnika, te je moguće naručiti odgovarajući PLC za bilo koju primjenu. Osnovna razlika u izvedbama PLC-a je u načinu izrade, te postoje kompaktni i modularni PLC-ovi.

Kompaktni kontroleri u svom kućištu imaju ulazni i izlazni modul i centralnu upravljačku jedinicu. Za razliku od kompaktnih, **modularni** ili proširivi PLC-ovi potrebni su za primjenu neovisno od digitalnih ulazno/izlaznih modula. Također, kod modularnog PLC-a individualni moduli se povezuju pomoću sabirničkog sistema, poznat kao serijska tehnologija.

Posebna vrsta modularnog PLC-a je **kartična izvedba**, koja je svoj razvoj doživjela tokom posljednjih nekoliko godina. Različiti modeli imaju razne mogućnosti proširenja modula, svi moduli se povezuju na sabirnice. Na slikama 2.4, 2.5 i 2.6 prikazane su sve tri izvedbe programabilnih logičkih kontrolera.



Slika 2.4: Kompaktni PLC [2]. Slika 2.5: Modularni PLC [3]. Slika 2.6: Kartična izvedba PLC [4].

2.5 Schneider Electric TM200CE24T

U završnom radu koristi se PLC jednog od globalno poznatnog proizvođača Schneider Electric, TM200CE24T, te spada u grupu Easy Modicon M200. Kontroler posjeduje 24 digitalna ulaza/izlaza, sa nazivnim naponom napajanja od 24V. Dakle, posjeduje 8 brzih ulaza, 6 regularnih ulaza, 8 tranzistorskih izlaza i 2 brza izlaza (PLS/PWM/PTO režim). Izlazna struja je 0.5 A, dok potrošnja energije iznosi 16W pri napajanju 24V DC. PLC posjeduje stepen zaštite IP20, što znači da je otporan na prašinu, te posjeduje i zaštitni poklopac. Otporan je na udare, mikroprekide i vibracije [5].

Upotrebljava se za jednostavne mašine, idealan je i za ugradnju u zidna i podna kućišta upravljačkog sistema. Navedeni kontroler ima ugrađeni Ethernet port, što znači da se lako može integrisati u arhitekture sistema upravljanja, za daljinsko upravljanje i održavanje mašina pomoću raznih aplikacija za tablete i računare.

Funkcije ugrađene u grupi Easy Modicon M200, u koje spada i TM200CE24T kontroler, minimiziraju troškove mašine tako što posjeduju Modbus serijsku vezu, USB port koji se koristi za programiranje i jednostavne funkcije kontrole položaja (brzi brojači i izlazi niza impulsa sa trapezoidnim profilom i S krivom).

Brzina prenosa se razlikuje u odnosu na način kako je ostvarena komunikacija. Najbrži prenos je preko Ethernet Modbus TCP/IP koja iznosi 10/100 Mbit/s za dužinu bus-a od 100 m, dok brzina prenosa podataka preko USB iznosi 12 Mbit/s.

Glavna odlika navedenog PLC-a je jednostavnost. Jednostavan je za pristup informacijama skeniranjem QR koda na prednjoj strani kontrolera, povezanog s web bazom podataka u realnom vremenu sa karakteristikama, dijagnostikom, održavanjem, priključcima i slično. Zatim, jednostavan je za postavljanje i programiranje zahvaljujući EcoStruxure Machine Expert - Basic softveru, koji će biti objašnjen u nastavku rada. Jednostavan je i za održavanje i ažuriranje sa uklonjivim terminalnim blokom, USB preuzimanjem bez napajanja iz mreže i Micro SD memorijskom karticom.

Kontroler TM200CE24T posjeduje visok nivo robusnosti u zahtjevnim okruženjima zahvaljujući dizajniranim ulazima koji pomažu u zaštiti od prenapona, nadzoru izlaza tranzistora radi zaštite od kratkih spojeva, kontrolisanom DC napajanju zbog zaštite od obrnutog polariteta, te obloženoj elektronici za povećanu robusnost u zagađenom okruženju.

Dizajniran je za rad u industriji tekstila, pakovanje, rukovanje raznim vrstama materijala, podizanje, razne mašinske alate i za određene električke montažne mašine. Certificiran je od strane CE, CSA, IACS E10, RCM i CULUS, te podržava standarde IEC 61010-2-201 i IEC 61131-2. Važno je napomenuti da predstavlja odličan izbor za male sisteme koji zahtijevaju automatizaciju sistema koji trebaju biti fleksibilni i pokretni. Posjeduje i veoma dobar odnos performansi sa cijenom, što je velika prednost pri izboru PLC-a. Na slici 2.7 prikazan je opisani programabilni logički kontroler.



Slika 2.7: Programabilni logički kontroler TM200CE24T (Schneider Electric) [5].

* * *

Nakon istraživanja programabilnog logičkog kontrolera (PLC-a), njegove arhitekture, različitih tipova, načina programiranja te specifične analize modela TM200CE24T proizvođača Schneider Electric, pruženo je bazno razumijevanje funkcionalnosti i primjene PLC u industrijskim procesima, naglašavajući prednost korištenja ovih tehnologija za automatizaciju i optimizaciju proizvodnih pogona. U narednom poglavlju, fokus će biti na softveru EcoStruxure Machine Expert-Basic, ključnom alatu za programiranje i upravljanje PLC kontrolerom TM200CE24T.

Poglavlje 3

EcoStruxure Machine Expert - Basic

Korištenje programabilnog logičkog kontrolera TM200CE24T u procesima upravljanja omogućeno je korištenjem softvera EcoStruxure Machine Expert - Basic.

3.1 Uvod u EcoStruxure Machine Expert - Basic

EcoStruxure Machine Expert - Basic je alat za grafičko programiranje dizajniran tako da olakša konfiguraciju, razvoj i puštanje u rad programa za programabilne logičke kontrolere proizvođača Schneider Electric.

EcoStruxure Machine Expert - Basic omogućava kreiranje softverskih rješenja za kontrolere Modicon M221, te Modicon M200 i M300. PLC koji se koristi za izradu je TM200CE24T i spada u grupu kontrolera Easy Modicon M200.

Prednosti korištenja softvera EcoStruxure Machine Expert-Basic:

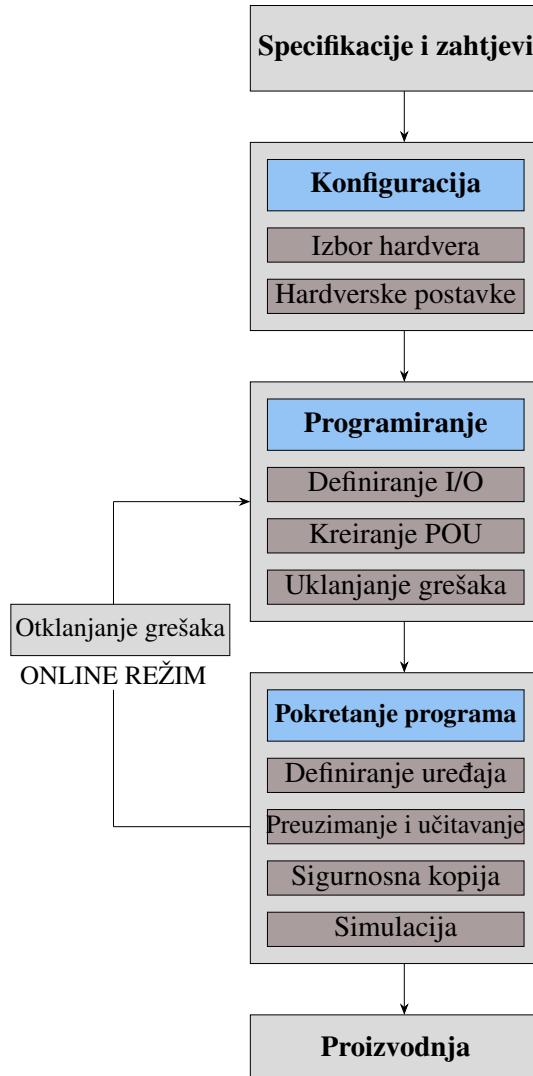
- Organizacija prema ciklusu razvoja projekta
- Laka i intuitivna navigacija softverom
- Efikasnost - veliki broj dostupnih funkcija
- Jednostavnost i brzina

Programabilni logički kontroler čita vrijednosti na ulazu, upisuje vrijednosti na izlaz i rješava logiku u skladu sa upisanim programom. Kreiranje programa koji se izvršava zasnovano je na pisanju instrukcija u jednom od podržanih programske jezike.

EcoStruxure Machine Expert - Basic, prema standardu IEC-61131-3, podržava programske jezike:

- Ljestvičaste dijagrame
- Instrukcijske liste
- Grafet (List)
- Grafet (SFC)

Na dijagramu 3.1 prikazane su faze kroz koje se prolazi prilikom izrade projekta u EcoStruxure Machine Expert - Basic, a to su konfiguracija, programiranje i pokretanje programa.



Slika 3.1: Prikaz faza izrade projekta u EcoStruxure Machine Expert - Basic.

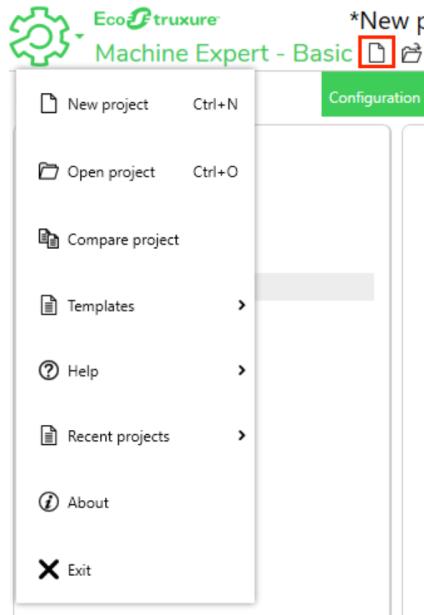
Različiti načini rada pružaju kontrolu nad procesima razvoja, dijagnosticiranja grešaka, praćenja stanja i izmjena aplikacije, bez obzira na to je li kontroler trenutno povezan s EcoStruxure Machine Expert - Basic ili nije.

EcoStruxure Machine Expert - Basic može raditi u tri režima, i to su:

- Online način rada
- Offline način rada
- Simulacijski način rada

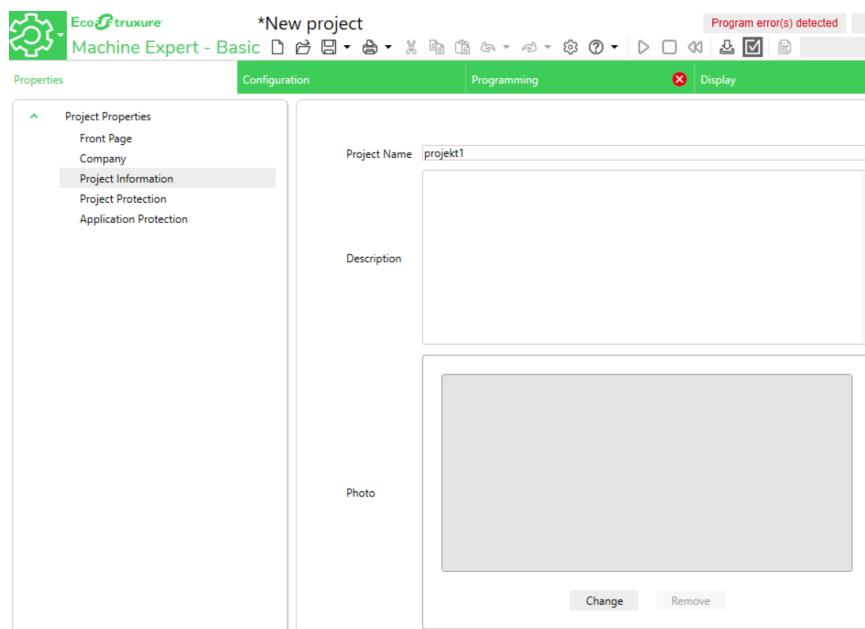
3.2 Kreiranje projekta

Nakon otvaranja programa potrebno je kreirati novi projekt ili otvoriti postojeći EcoStruxure Machine Expert - Basic, TwidoSoft ili TwidoSuite projekt za dalji rad. Otvara se početni prozor prikazan na slici 3.2, te je za kreiranje projekta potrebno iz padajućeg menija odabrati opciju "New project" ili kliknuti na ikonu  označenu crvenom bojom.



Slika 3.2: Kreiranje novog projekta u EcoStruxure Machine Expert - Basic.

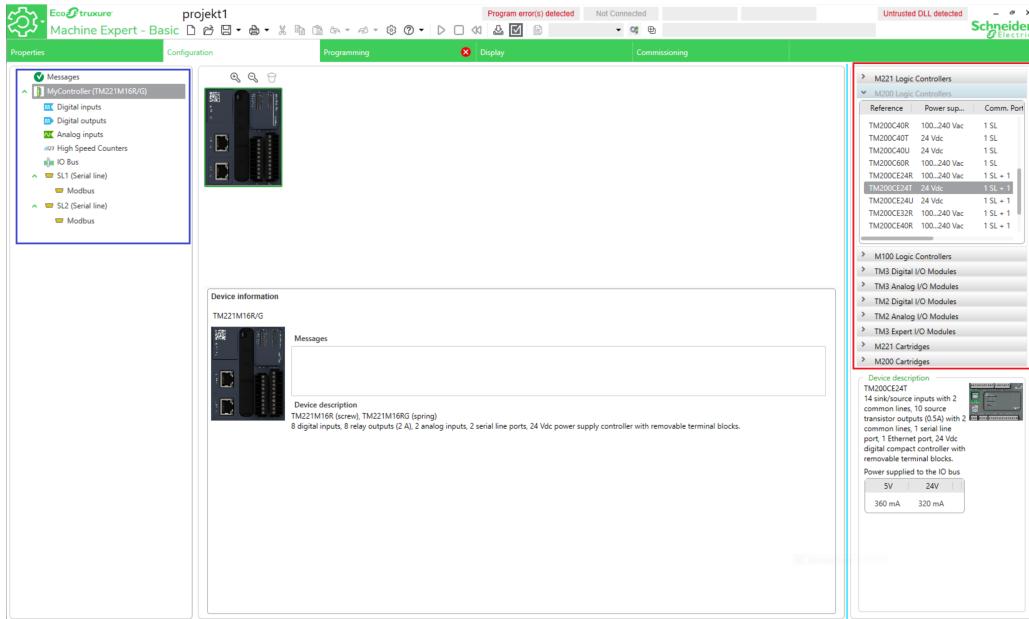
Nakon što se kreira novi projekt, potrebno je u "Properties" prozoru izabrati "Project Information" što je prikazano na slici 3.3. Omogućen je unos imena projekta, opisa i dodavanja fotografije, nakon unosa potrebno je kliknuti "Apply". U "Properties" prozoru moguće je urediti naslovnu stranu, unijeti podatke o kompaniji, te zaštititi projekt i aplikaciju lozinkom.



Slika 3.3: Postavke novog projekta u EcoStruxure Machine Expert - Basic.

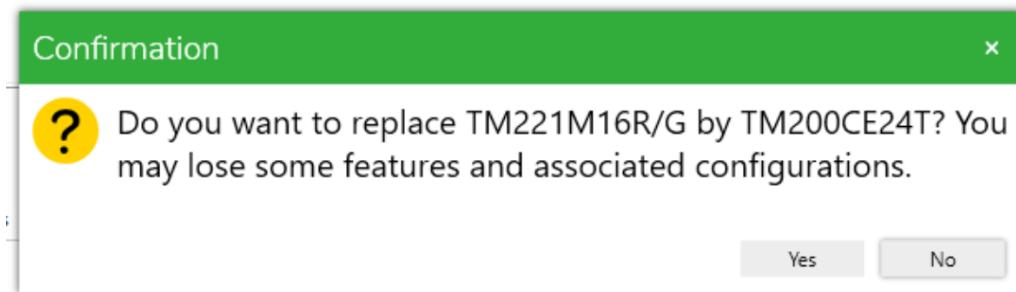
3.3 Konfiguracija hardvera

Kada se završi kreiranje novog projekta i podešavanje postavki, potrebno je izvršiti konfiguraciju hardvera za dalji rad. Izgled prozora za konfiguraciju hardvera prikazan je na slici 3.4.



Slika 3.4: Konfiguracija hardvera u EcoStruxure Machine Expert - Basic.

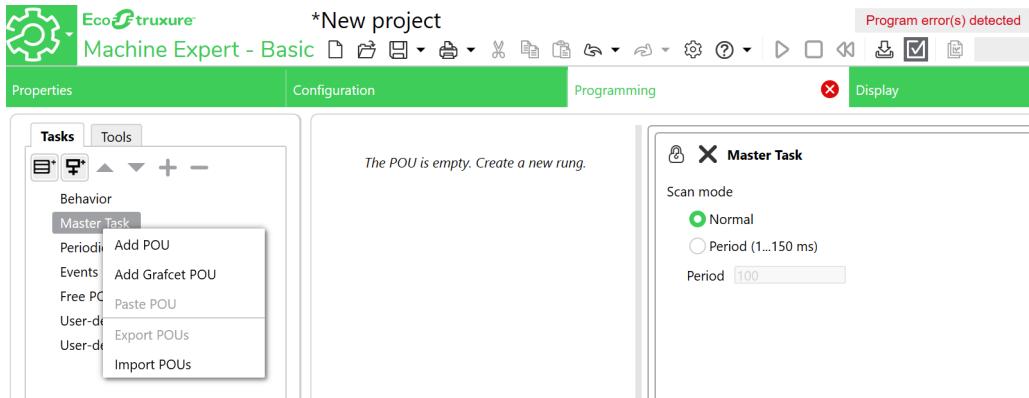
Potrebno je izabrati odgovarajući programabilni logički kontroler, te izabrani iz padajućeg menija prevući ("drag and drop") na već po *default*-u prikazani kontroler. Izbor PLC se vrši u prozoru označenom crvenom bojom na slici 3.4, dok prozor označen plavom bojom predstavlja struktuirani prikaz konfiguracije hardvera. Nakon izbora PLC po ranije objašnjrenom postupku otvara se iskočni prozor (slika 3.5) na kojem klikom na "yes" završava proces konfiguracije.



Slika 3.5: Izbor hardvera u EcoStruxure Machine Expert - Basic.

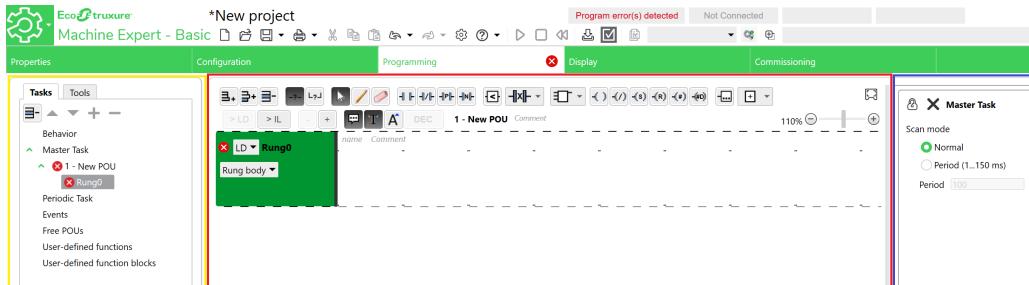
3.4 Radni prostor za programiranje

Radni prostor za programiranje PLC u EcoStruxure Machine Expert - Basic prikazan je na slici 3.6.



Slika 3.6: Radni prostor za programiranje u EcoStruxure Machine Expert - Basic.

Da se omogući programiranje kontrolera potrebno je u padajućem meniju sa lijeve strane izabrati "Add POU" ili kliknuti na ikonu . Nakon dodavanja "POU" otvara se okruženje kao na slici 3.7.



Slika 3.7: Kreiranje "POU" u EcoStruxure Machine Expert - Basic.

Prostor za programiranje je podijeljen u tri glavna dijela označena bojama. Žutom bojom označen je meni programiranja koji omogućava izbor svojstava programa, njegovih funkcija i objekata, te niz alata koji se mogu koristiti za praćenje i otklanjanje potencijalnih grešaka u programu.

"Master Task" predstavlja glavni zadatak koji program izvršava. On kontroliše kontinuiranim cikličnim skeniranjem (u normalnom režimu skeniranja) ili određivanjem perioda skeniranja od 1 do 150 ms (podrazumijevano 100 ms) u režimu periodičnog skeniranja. Sastoji se od sekcija i potprograma kreiranih unutar programske organizacione jedinice (POU). Svaki POU može se programirati u bilo kojem od podržanih programske jezika.

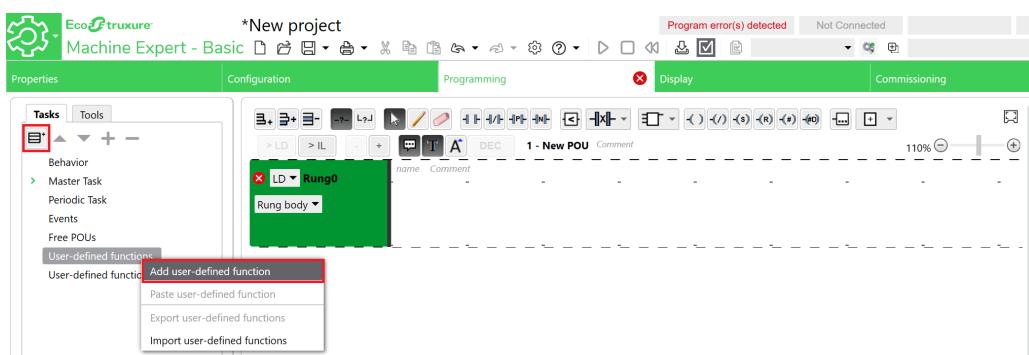
Periodični zadatak (eng.: "Periodic task") predstavlja kratkotrajni potprogram koji se periodično obrađuje. Periodični zadaci se konfigurišu navođenjem perioda skeniranja od 1-255 ms (podrazumijevano 255 ms). U EcoStruxure Machine Expert - Basic, ovaj potprogram je implementiran kao Free POU, te se može napisati na bilo kojem od programskih jezika koji su podržani.

Događaji (eng.: "Events") su potprogrami vrlo kratkog trajanja sa smanjenim vremenom odgovora aplikacije. Pokreću se fizičkim ulazima ili HSC funkcionalnim blokovima. Ovi događaji su povezani sa ugrađenim digitalnim ulazima (I0.2. -I0.5) ili sa brojačima velike brzine (HSC0 i HSC1). Moguće je konfigurisati dva događaja za svaki HSC funkcionalni blok. Ima veći prioritet od glavnog programa, te proizvodi brzo vrijeme odziva, omogućavajući tako smanjenje ukupnog vremena odziva sistema.

Programska organizaciona jedinica ("POU") je objekt koji se koristi u programu i sastoji se od deklaracija varijabli i skupa instrukcija napisanih u izvornom kodu na podržanom jeziku programiranja. Jedan POU mora uvek da postoji i da bude povezan s glavnim zadatkom programa.

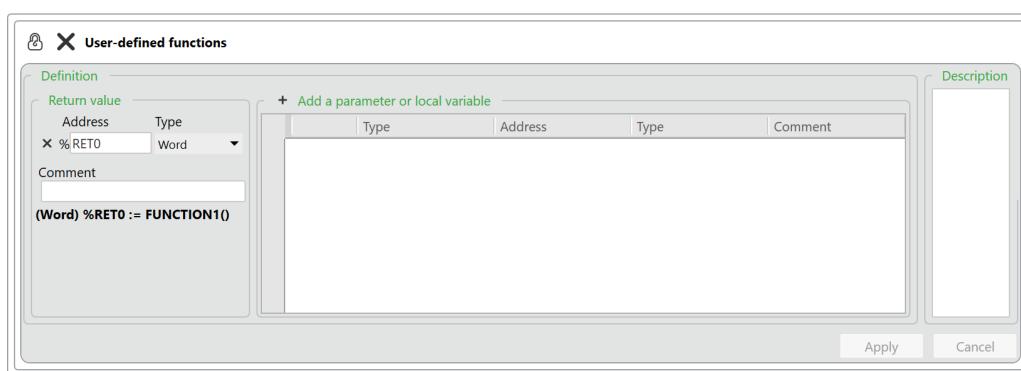
"Free POUs" predstavlja posebnu vrstu programske organizacione jedinice koja nije eksplicitno povezana sa zadatkom. Koriste se kada se pozivaju potprogrami iz POU te za periodične potprograme.

Korisnički definirane funkcije ("User-defined functions") omogućavaju korisnicima kreiranje nove funkcije s određenim ulaznim parametrima, lokalnim varijablama i povratnim vrijednostima. Pohranjene su kao dio EcoStruxure Machine Expert - Basic projekta, i mogu se koristiti u glavnom zadatku, periodičnim zadacima i događajima. Za kreiranje funkcije potrebno je desnim klikom otvoriti padajući meni iz kojeg se izabere *Add user-defined function* kao na slici 3.8 ili jednostavnije kliknuti na ikonu  označenu crvenom bojom.



Slika 3.8: Kreiranje korisnički definirane funkcije u EcoStruxure Machine Expert - Basic.

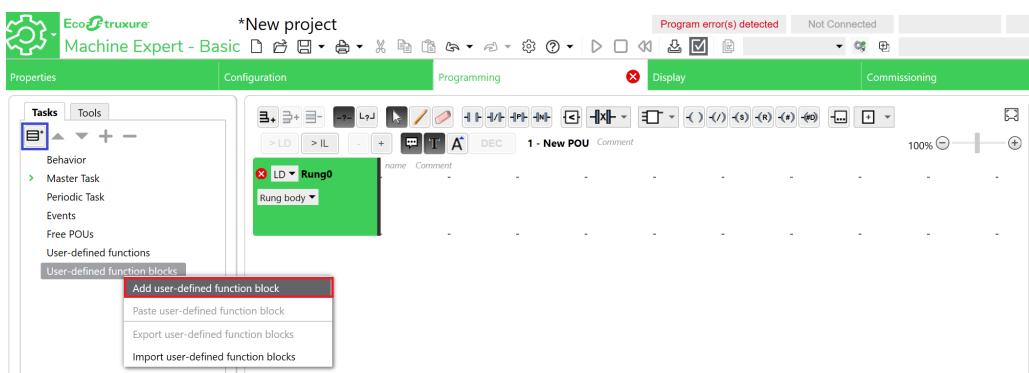
Nakon što se kreira korisnički definirana funkcija, otvara se prozor kao na slici 3.9.



Slika 3.9: Prozor korisnički definirane funkcije u EcoStruxure Machine Expert - Basic.

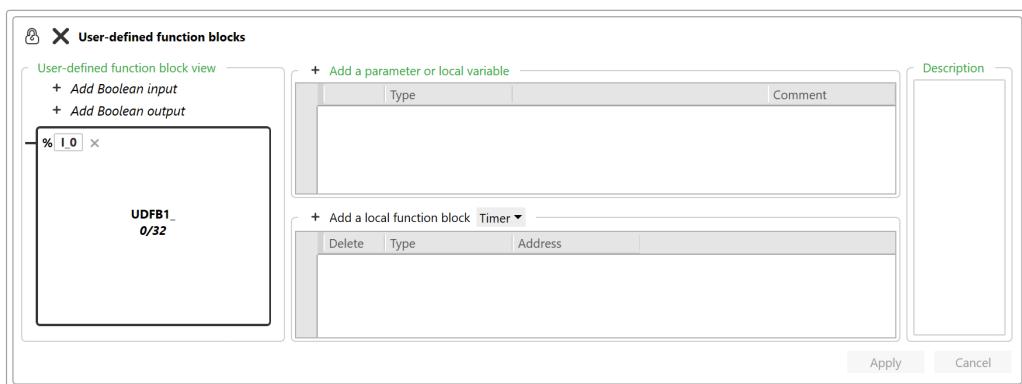
Jedna od karakteristika je opis korisnički definirane funkcije, što olakšava korisniku pri velikom broju funkcija, na način da ne mora zapamtiti primjenu svake od njih. Nakon definiranja svih parametara i opisa funkcije potrebno je kliknuti na ("Apply").

Korisnički definirani funkcionalni blokovi (eng.: "*User-defined function blocks*") omogućavaju kreiranje novih blokova s jednim ili više ulaznih i izlaznih parametara i sa lokalnim varijablama. Isto kao i korisnički definirane funkcije pohranjeni su kao dio projekta i mogu se koristiti u glavnom zadatku, periodičnim zadacima i događajima. Za kreiranje funkcije potrebno je desnim klikom otvoriti padajući meni iz kojeg se izabere *Add user-defined function block* kao na slici 3.10 označeno crvenom bojom ili jednostavnije kliknuti na ikonu  označenu plavom bojom na slici 3.10.



Slika 3.10: Kreiranje korisnički definiranog funkcionalnog bloka u EcoStruxure Machine Expert - Basic.

Nakon što se kreira korisnički definirani funkcionalni blok, otvara se prozor prikazan na slici 3.11.



Slika 3.11: Prozor korisnički definiranog funkcionalnog bloka u EcoStruxure Machine Expert - Basic.

Slično kao i za korisnički definirane funkcije i funkcionalni blokovi daju mogućnost opisivanja istih, te kreiranje ulaznih i izlaznih signala, lokalnih varijabli i lokalnih funkcionalnih blokova.

Plavom bojom na slici 3.7 označen je prostor koji omogućava da se detaljnije vide i konfigurišu svojstva stavke odabrane u meniju programiranja (žuta boja), dok oblast označena crvenom bojom omogućava unos izvornog koda.

3.4.1 Posebne funkcije

Objekti

U softveru EcoStruxure Machine Expert - Basic, objekt se koristi za predstavljanje područja memorije logičkog kontrolera koje je rezervirano za korištenje od strane aplikacije.

Objekti mogu biti:

- Jednostavne softverske varijable, kao što su memorijski bitovi i riječi
- Adrese digitalnih ili analognih ulaza i izlaza
- Interne varijable kontrolera, kao što su sistemske riječi i sistemske bitovi
- Unaprijed definirane sistemske funkcije ili funkcionalni blokovi, kao što su tajmeri i brojači

Memorija kontrolera je ili unaprijed dodijeljena za određene tipove objekata, ili se automatski dodjeli kada se izvrši preuzimanje aplikacije na logički kontroler. Objekti se adresiraju pomoću prefiksa % od strane programa samo kada je memorija dodijeljena. Na primjer, %MW12 je adresa memorijske riječi, %Q0.3 je adresa ugrađenog digitalnog izlaza, a %TM0 je adresa funkcionskog bloka tajmera.

Simboličko adresiranje

EcoStruxure Machine Expert - Basic podržava simboličko adresiranje objekata, tačnije indirektno adresiranje objekata imenom. Upotreba simbola omogućava brzo ispitivanje i analizu programske logike, te pojednostavljuje razvoj i testiranje aplikacije.

Na primjer, CYCLE_END je simbol koji se može koristiti za identifikaciju instance funkcionskog bloka tajmera (*eng. Timer*) koji predstavlja kraj nekog ciklusa. Olakšava korisniku pri razumijevanju u kojem stanju se nalazi program, na način da je lakše upamtiti simboličku adresu nego zapamtiti ulogu adrese programa kao što je naprimjer %TM3. Simboli su dio neprogramskih podataka, pohranjeni su u logičkom kontroleru kao dio EcoStruxure Machine Expert - Basic aplikacije.

Alokacija memorije

EcoStruxure Machine Expert - Basic omogućava da se unaprijed alociraju (rezervišu) blokovi memorije logičkog kontrolera za korištenje od strane određenih tipova objekata koji se koriste u programu, uključujući jednostavne objekte (memorijske riječi, konstantne riječi) i softverske objekte (funkcionalne blokove). U offline modu se određuje način alokacije memorije za svaki tip objekta.

Postoje dva načina alokacije memorije i to su:

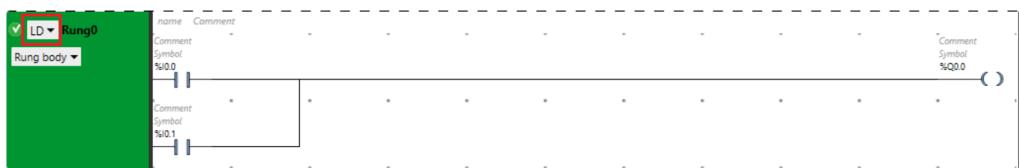
- Automatski - Svi objekti od pomaka 0 do najviše memorijske adrese koja se koristi u programu, ili povezane sa simbolom, automatski se alociraju u memoriju logičkog kontrolera. Dakle, ako se u programu koristi memorijska riječ %MW10, svi objekti od %MW0 do %MW10 uključujući i jedanaesti objekt se automatski alociraju u memoriju. Ako se prebaci na online način rada, ne može se alocirati novi memorijski objekt s adresama višim od najviše adrese koja je korištena u offline modu.

- Manuelno - Potrebno je odrediti broj objekata koji će se alocirati u memoriji. Kada se koristi online način rada, omogućeno je dodavanje novih kontakata, releja ili jednačine u program (do granice alocirane memorije) bez potrebe odjavljivanja iz logičkog kontrolera, modificiranja programa, ponovne prijave i preuzimanja aplikacije. EcoStruxure Machine Expert - Basic prikazuje ukupan broj alociranih memorijskih objekata i ukupan broj memorijskih objekata dostupnih u logičkom kontroleru.

Reverzibilnost programa

EcoStruxure Machine Expert - Basic podržava konverziju iz Ladder dijagrama u liste instrukcija i iz liste instrukcija u Ladder dijagrame. Omogućeno je prebacivanje dijelova programa između programske jezike u bilo koje vrijeme po potrebi korisnika. Dakle, mogu se prikazati programi sa dijelom isprogramiranim pomoću Ladder dijagrama dok su ostali dijelovi programirani listom instrukcija. Nije moguće pretvoriti programe programirane sa Ladder dijogramima i instrukcijskim listama u programe u Grafset (SFC), ili obrnuto, a nije moguća ni konverzija Grafset (Liste) u Grafset (SFC).

Na slikama 3.12 i 3.13 prikazano je programiranje pomoću Ladder dijagrama (slika 3.12) i njegovu ekvivalentnu programsku logiku izraženu kao niz instrukcija programiranih pomoću instrukcijskih listi (slika 3.13).



Slika 3.12: Primjer reverzibilnosti programa - Ladder dijagram.

Za prikaz programske logike kao niza instrukcija moguće je iz menija promijeniti prikaz LD u prikaz IL (označeno crvenom bojom na slici 3.12), ili jednostavnije klikom na ikonu .

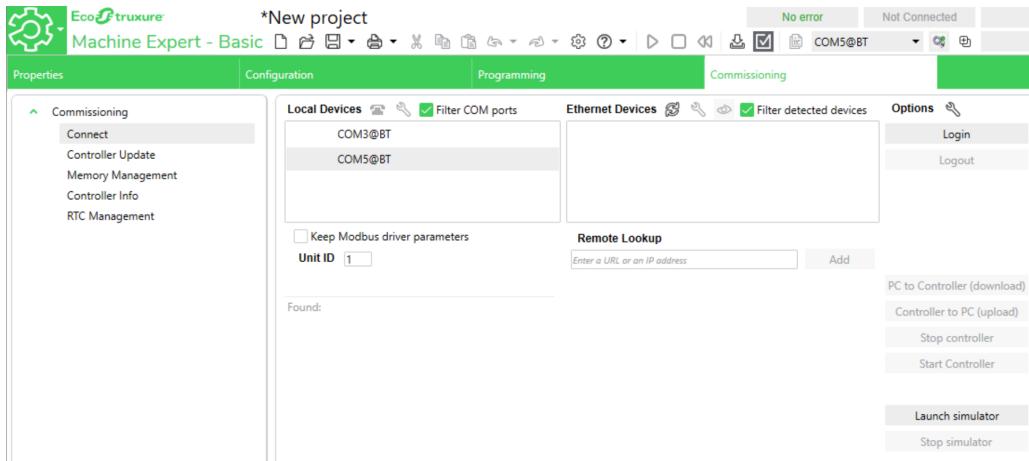


Slika 3.13: Primjer reverzibilnosti programa - Instrukcijske liste.

Program se uvijek interno pohranjuje kao lista instrukcija, bez obzira na to da li je originalno napisan u jeziku ljestvičastih dijagrama ili liste instrukcija.

3.5 Prozor za puštanje u rad

Prozor za puštanje u rad omogućava prijavljivanje ili odjavljivanje sa logičkog kontrolera, nadogradnju firmware-a, upravljanje memorijom kao i prikaz informacija o kontroleru i korištenim modulima za proširenje. Omogućeno je i upravljanje satom realnog vremena (RTC) logičkog kontrolera, te pokretanje, kao i zaustavljanje simulacije. Prozor je prikazan na slici 3.14.

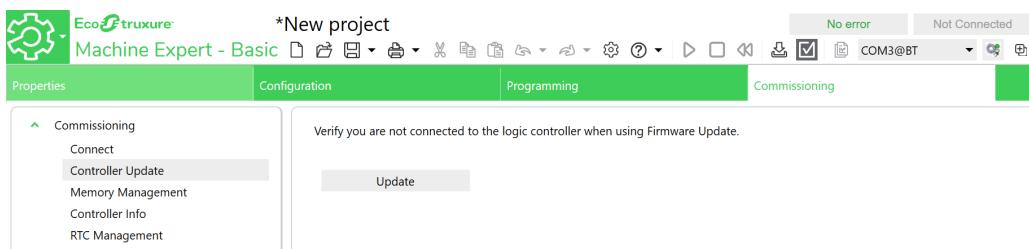


Slika 3.14: Prikaz prozora za puštanje u rad.

Prikazane su dvije vrste logičkih kontrolera:

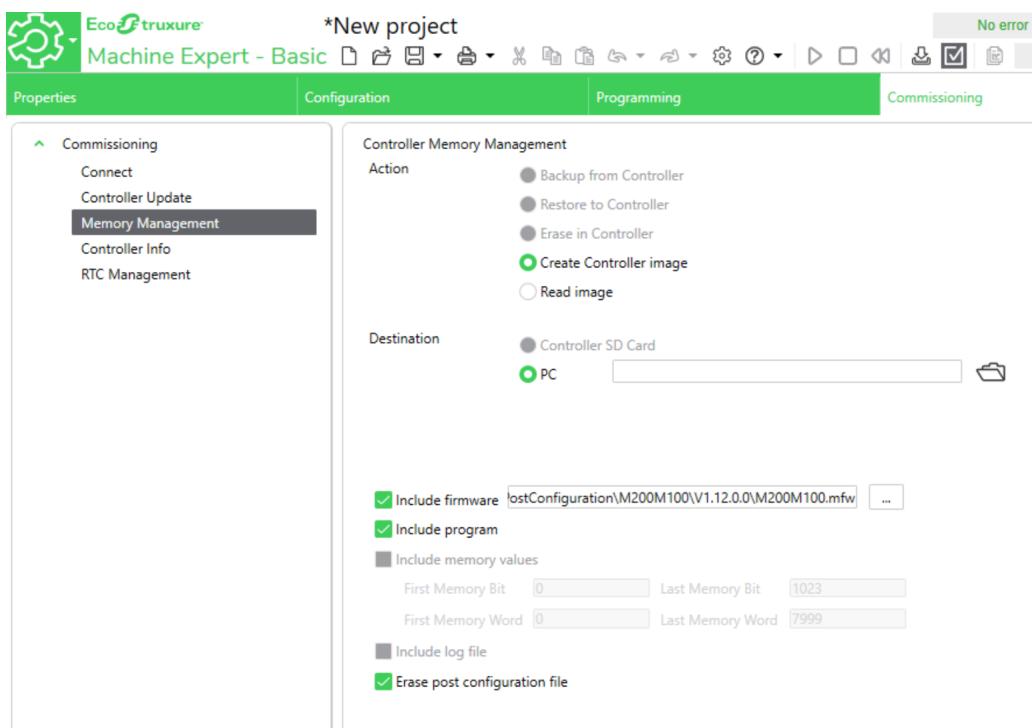
- Lokalni uređaji - prikazuju se svi logički kontroleri povezani na PC
- Ethernet uređaji - prikazuju se svi logički kontroleri kojima je dostupan Ethernet

Klikom na *Controller Update* otvara se prozor kao na slici 3.15. Moguće je preuzeti ažuriranja firmware-a na logički kontroler ili direktno sa EcoStruxure Machine Expert - Basic ili koristeći SD karticu. Izvođenje ažuriranja firmware-a čuva aplikacijski program koji je prisutan u kontroleru.



Slika 3.15: Prikaz prozora za ažuriranje firmware-a logičkog kontrolera.

Klikom na *Memory Management* otvara se prozor kao na slici 3.16. Moguće je napraviti sigurnosnu kopiju, vratiti ili izbrisati različite elemente sa ili na logičkom kontroleru koji je povezan. Opcije sigurnosne kopije, vraćanja i brisanja dostupne su samo u online načinu rada.



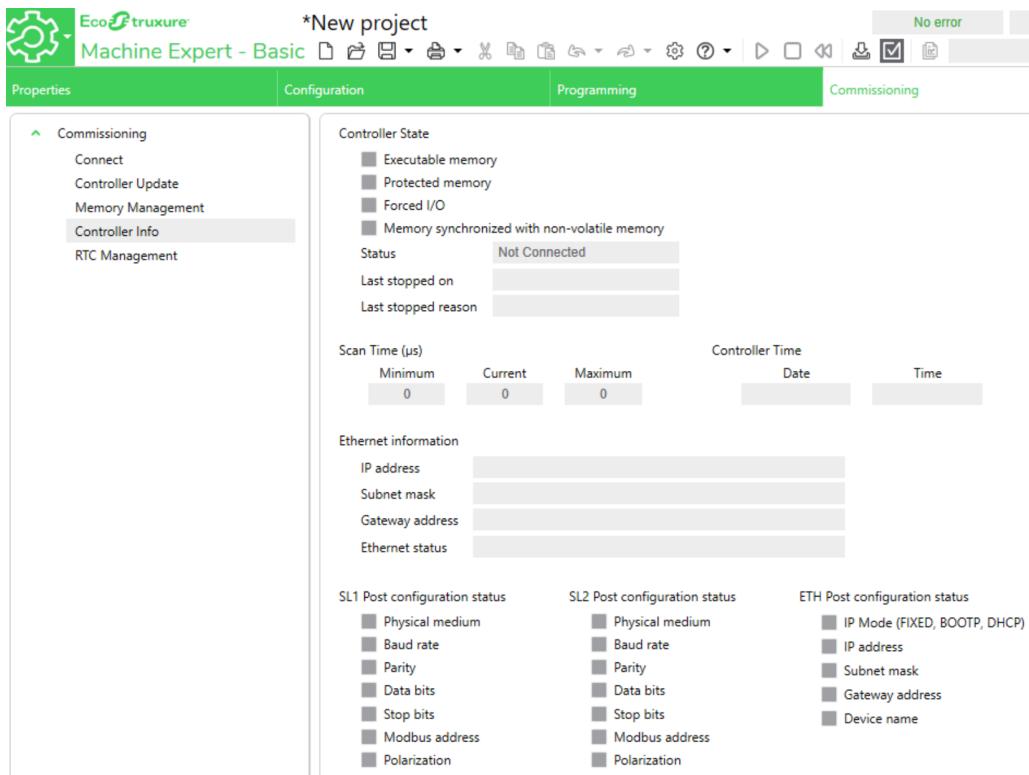
Slika 3.16: Prikaz prozora za upravljanje memorijom logičkog kontrolera.

Izabrana opcija je *Create Controller image* koja omogućava da se firmware, program i post konfiguracijski fajl prenese u logički kontroler. Prilikom kreiranja slike logičkog kontrolera, moguće je izabrati SD karticu kao odredište, te istu koristiti u PLC-u. Navedena opcija se može koristiti i u offline režimu rada.

Preostala opcija je *Read image* koja se kao i prethodna koristi u offline režimu rada. Omogućava da se otvori .smbk fajl kao projekt unutar EcoStruxure Machine Expert - Basic.

Klikom na *Controller Info* otvara se prozor kao na slici 3.17. Na ovom prozoru prikazane su informacije o trenutnom stanju logičkog kontrolera.

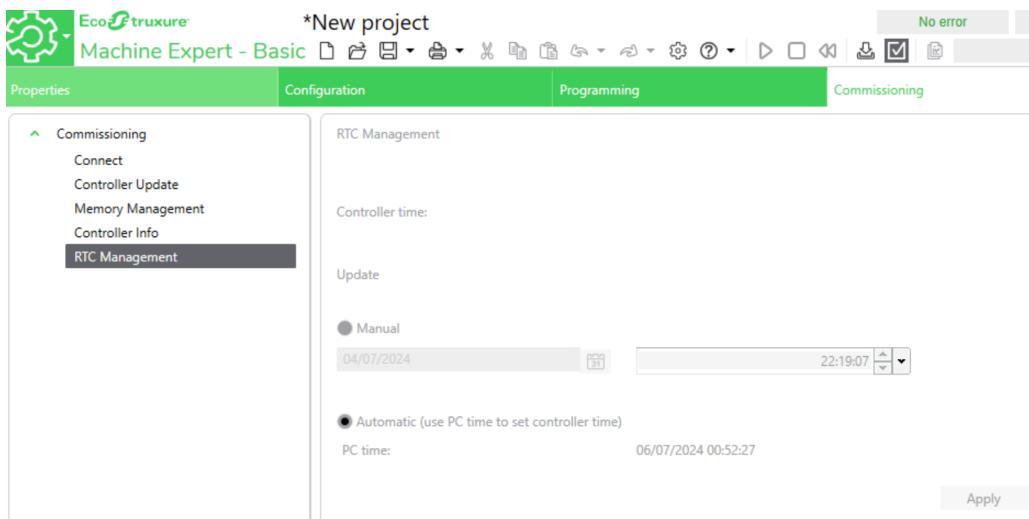
- *Executable memory* ili izvršna memorija provjerava da li je važeća aplikacija snimljena u memoriji logičkog kontrolera.
- *Protected memory* ili zaštićena memorija je označena ako je aplikacija u RAM memoriji logičkog kontrolera zaštićena lozinkom.
- *Forced I/O* je opcija koja se provjerava ako se jedan ili više digitalnih ulaza ili izlaza na logičkom kontroleru prisiljava na neku određenu vrijednost.
- *Memory synchronized with non-volatile memory* je označena ako aplikacija snimljena u nepromjenjivoj memoriji nije identična aplikaciji u RAM memoriji logičkog kontrolera.
- *Status* predstavlja trenutno stanje logičkog kontrolera.
- *Last stopped on* prikazuje datum i vrijeme kada je logički kontroler posljednji put zauzavljen.
- *Last stopped reason* prikazuje razlog posljednjeg zaustavljanja logičkog kontrolera.



Slika 3.17: Prikaz prozora za informacije trenutnog stanja logičkog kontrolera.

- *Scan Time* ili vrijeme skeniranja u mikrosekundama podijeljeno je u tri kategorije.
 - *Minimum* (u mikrosekundama) prikazuje najkraće vrijeme skeniranja od posljednjeg uključivanja logičkog kontrolera.
 - *Current* (u mikrosekundama) prikazuje vrijeme skeniranja.
 - *Maximum* (u mikrosekundama) prikazuje najduže vrijeme skeniranja od posljednjeg uključivanja logičkog kontrolera.
- *Controller Time* ili vrijeme kontrolera se prikazuje samo ako logički kontroler ima sat realnog vremena (RTC). U vrijeme kontrolera spadaju:
 - Datum (DD/MM/GGGG) prikazuje datum pohranjen u logičkom kontroleru
 - Vrijeme (HH:MM:SS) prikazuje vrijeme pohranjeno u logičkom kontroleru
- *Ethernet Information* prikazuju se samo ako logički kontroler ima ugrađenu Ethernet vezu. Prikazuje se IP adresa, maska, adresa gateway-a, te status ethernet veze.
- *SL1 Post Configuration status*, *SL2 Post Configuration status* i *ETH Post Configuration status* čekiraju navedene parametre definirane post konfiguracijskim fajlom.

Klikom na *RTC Management* otvara se prozor prikazan na slici 3.18. Na slici 3.18 se vide mogućnosti podešenja sata realnog vremena (RTC) logičkog kontrolera, koje je moguće samo ako je EcoStruxure Machine Expert - Basic povezan na logički kontroler koji podržava RTC. Ako je u online modu, prikazuje se trenutno vrijeme kontrolera, dok se podešavanje može vršiti manuelno ili automatski.



Slika 3.18: Prikaz prozora za upravljanje satom realnog vremena logičkog kontrolera.

3.6 Simulacija programa

Simulator predstavlja repliku ponašanja stvarnog logičkog kontrolera. Dakle, to je virtualni logički kontroler koji se povezuje sa softverom EcoStruxure Machine Expert - Basic. Simulator omogućava:

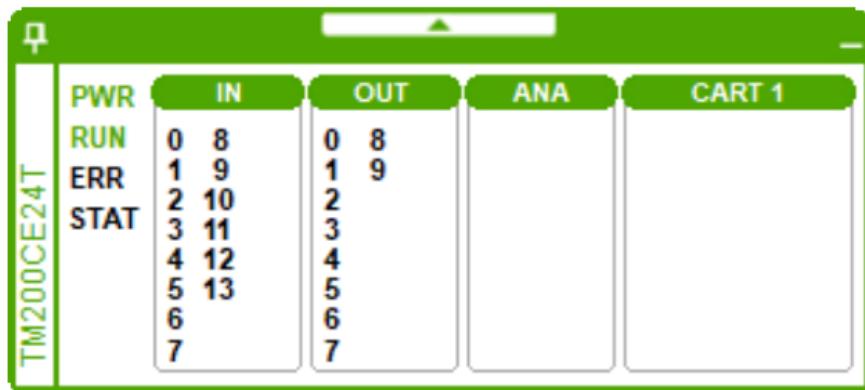
- Simuliranje veze između PC-a, logičkog kontrolera i svih modula za proširenje.
- Pokretanje i testiranje programa bez logičkog kontrolera i modula za proširenje.

Prije pokretanja simulatora potrebno je provjeriti da li u programu postoje greške, što znači da je potrebno program prvenstveno kompajlirati. Kompajliranje se vrši klikom na ikonu . Ukoliko je kompajliranje uspješno izvršeno, klikom na *Launch simulator* u prozoru za puštanje u rad (Slika 3.14) ili jednostavnije klikom na ikonu pokreće se simulacija.

Nakon pokretanja simulacije otvaraju se dva prozora:

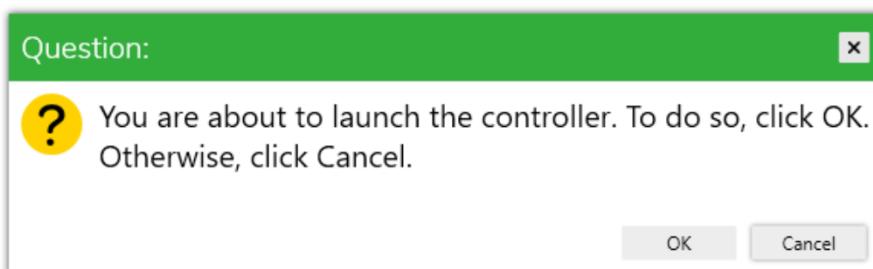
- Prozor za upravljanje vremenom simulatora koji omogućava da se podešava RTC kontrolera.
- Prozor ulaza i izlaza kontrolera koji omogućava upravljanje stanjem ulaza/izlaza kontrolera i modula za proširenje.

Na slici 3.19 prikazan je prozor za manipulaciju stanja ulaza i izlaza kontrolera.



Slika 3.19: Prikaz prozora za manipulaciju stanja ulaza i izlaza logičkog kontrolera.

PWR, RUN, ERR i STAT LED su diode simulirane onako kako bi se pojavile na povezanom stvarnom kontroleru. PWR pokazuje da li je logički kontroler uključen ili ne. RUN označava stanje pokrenutnog logičkog kontrolera. Da bi se kontroler pokrenuo potrebno je kliknuti na ikonu ili izabrati *Start Controller* na slici 3.14. Nakon pritiska na navedenu ikonu, otvara se iskočni prozor kao na slici 3.20, na kojem je potrebno izabrati OK, da bi se simulacija pokrenula.



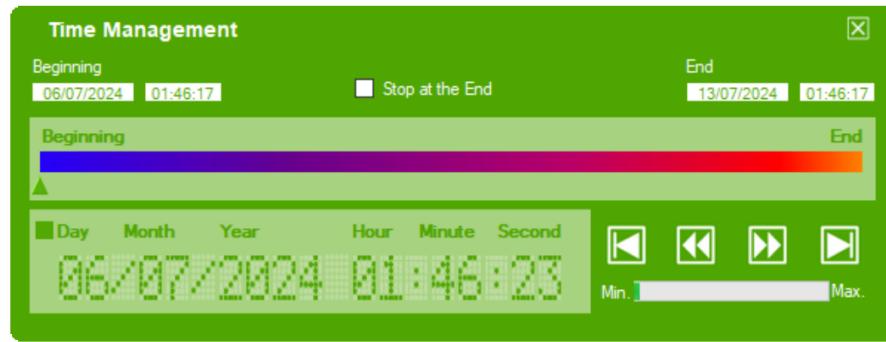
Slika 3.20: Prikaz iskočnog prozora prije pokretanje simulacije.

ERR označava stanje greške simuliranog kontrolera, dok je STAT LED definirana korisničkom logikom.

Ulazi i izlazi su prikazani u listi brojeva. Veličina liste ovisi o ulazima i izlazima odabranog kontrolera i modula proširenja. Za kontroler sa n digitalnih ulaza, lista brojeva će prikazati broj koji počinje od 0 do vrijednosti $(n-1)$, gdje svaki broj odgovara digitalnom ulazu na odgovarajućem ulaznom kanalu. Na slici 3.19 prikazani su *IN* - digitalni ulazi, *OUT* - digitalni izlazi, te *ANA* - analogni ulazi. Ukoliko se dodaju moduli za proširenje, pored navedenih prikaza ulaza i izlaza prikazuju se i digitalni/analogni ulazi i izlazi modula za proširenje.

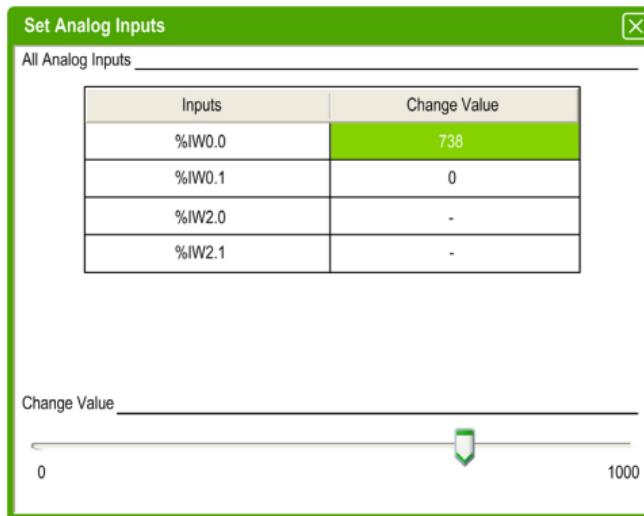
Trenutno stanje digitalnih ulaza i izlaza se identificira pomoću boje. Zelenom bojom su označeni aktivni ulazi i izlazi, postavljeni na 1. Crnom bojom su označeni neaktivni digitalni ulazi i izlazi, postavljeni na 0. Kod analognih ulaza je drugačija situacija, zbog toga što se stanje identificuje po vrijednosti. Dakle, ukoliko je prikazan broj, on predstavlja trenutnu vrijednost, dok ukoliko je prikazana crtica (-) znači da analogni ulaz/izlaz nije konfigurisan.

Prozor za upravljanje vremenom simulatora moguće je otvoriti desnim klikom na prozor sa slike 3.19 i biranjem *Time Management*. Otvoreni prozor prikazan je na slici 3.21. Omogućeno je postavljati datum i vrijeme početka i završetka, zaustaviti program kada se dođe do kraja, te ručno pomicanje protoka vremena naprijed ili nazad.



Slika 3.21: Prikaz prozora za upravljanje vremenom simulatora logičkog kontrolera.

EcoStruxure Machine Expert - Basic u prozoru osnovnog simulatora I/O menadžera omogućava izmjenu vrijednosti ulaza, te praćenje stanja izlaza. Izmjena vrijednosti ulaza omogućena je klikom na broj ulaza koji je potrebno promijeniti. Kako je i ranije navedeno, broj koji je zelene boje označava aktivan ulaz, dok broj crne boje je neaktiviran. Kada se aktivira ulaz, njegova deaktivacija se vrši ponovnim klikom na isti. Za promjenu vrijednosti analognih ulaza klikom na isti otvara se prozor prikazan na slici 3.22. Promjena je omogućena na dva načina. Prvi način predstavlja unos vrijednosti u kolonu *Change Value*, dok drugi način predstavlja pomjeranje klizača od minimalne do maksimalne vrijednosti.



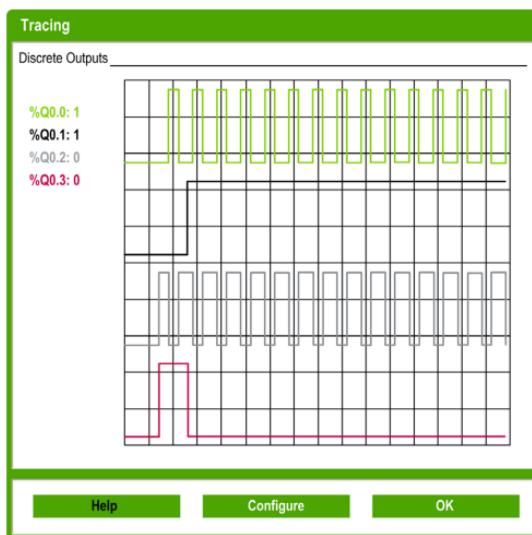
Slika 3.22: Prikaz prozora za upravljanje analognim ulazima logičkog kontrolera.

Praćenje stanja digitlanih i analognih izlaza moguće je u prozoru osnovnog simulatora u koloni OUT, ili u prozoru *Tracing* koji se otvara desnim klikom na osnovni simulator. Prvenstveno se otvori prozor *Tracing*, prikazan na slici 3.23. Nakon otvaranja prozora potrebno je izvršiti konfiguraciju klikom na *Configure*, nakon čega se otvara prozor za konfiguraciju prikazan na slici ispod. Potrebno je izabrati izlaze koji se žele pratiti, te izabrati vremenski interval uzorka za praćenje izlaza.



Slika 3.23: Prikaz prozora za konfiguraciju praćenja izlaza logičkog kontrolera.

Odabrani izlazi se dodaju u prozor *Tracing* koji prikazuje praćenje izlaza sa simuliranim vrijednostima. Prozor *Tracing* prikazan je na slici 3.24.



Slika 3.24: Prikaz prozora za praćenje stanja izlaza logičkog kontrolera.

* * *

EcoStruxure Machine Expert-Basic je softver korišten za programiranje i upravljanje programabilnim logičkim kontrolerom TM200CE24T. Prikazan je način konfiguracije hardvera, koji omogućava integraciju i konfiguraciju različitih komponenti u industrijskim okruženjima. Radni prostor za programiranje nudi širok spektar posebnih funkcija koje olakšavaju razvoj i dijagnostiku programa, dok prozor za puštanje u rad i simulacija omogućavaju testiranje rješenja prije stvarne implementacije. Ovaj softverski alat efikasan je za upravljanje i optimizaciju industrijskih procesa, a prednost mu je jednostavnost korištenja. U nastavku će biti prikazana praktična primjena programabilnog logičkog kontrolera TM200CE24T kroz podržane načine programiranja.

Poglavlje 4

Uputstvo za korištenje kontrolera TM200CE24T

Kako je i ranije navedeno, za programiranje kontrolera TM200CE24T potrebno je koristiti softver EcoStruxure Machine Expert - Basic. Programabilni logički kontroler čita ulaze, upisuje izlaze i rješava logiku na osnovu kontrolnog programa. Kreiranje kontrolnog programa za logički kontroler sastoji se od pisanja niza instrukcija na jednom od podržanih programske jezika. EcoStruxure Machine Expert - Basic podržava sljedeće IEC-61131-3 programske jezike:

- Jezik ljestvičastih dijagrama (eng. *Ladder Diagram language*)
- Jezik liste instrukcija (eng. *Instruction List language*)
- Grafset (Lista)
- Grafset (SFC)

Za programiranje logičkog kontrolera potrebno je prvo kreirati projekt, te konfigurisati hardver koji će se koristiti. U ovom slučaju hardver koji se konfiguriše je kontroler TM200CE24T.

4.1 Programiranje ljestvičastim dijagramima

Ladder dijagrami su slični dijagramima relejne logike koji predstavljaju upravljanje procesima pomoću releja. Glavne razlike između ova dva tipa su sljedeće karakteristike programiranja ljestvičastim dijagramima koje se ne nalaze u logičkim dijagramima releja:

- Svi ulazi i binarni logički bitovi su predstavljeni kontaktima simbolima (- Svi izlazi i binarni logički bitovi su predstavljeni simbolima releja (

Program napisan u jeziku Ladder dijagrama sastoji se od mreža koje su skupovi grafičkih instrukcija nacrtanih između 2 vertikalne potencijalne trake. Mreže se izvršavaju sekvenčno od strane logičkog kontrolera.

Skup grafičkih instrukcija predstavlja sljedeće funkcije:

- Ulazi/izlazi kontrolera (tasteri, senzori, releji, svjetla i slično)
- Funkcije kontrolera (tajmeri, brojači)

- Matematičke i logičke operacije (sabiranje, dijeljenje, AND, XOR i slično)
- Operatori poređenja i druge numeričke operacije ($A < B$, $A = B$, pomak, rotacija)
- Interne varijable u kontroleru (bitovi, riječi)

Navedene grafičke instrukcije su raspoređene s vertikalnim i horizontalnim vezama koje na kraju vode do jednog ili više izlaza i/ili radnji. Jedna mreža ne može podržavati više od jedne grupe povezanih instrukcija.

Za ljestvičasto programiranje potrebno je poznavati grafičke elemente koji se koriste, a prikazani su u tabeli 4.1. Instrukcije u ljestvičastim dijagramima se ubacuju prevlačenjem i ispuštanjem grafičkih elemenata sa trake, koja se pojavljuje iznad radnog prostora za programiranje, u ćeliju mreže.

Tabela 4.1: Grafički elementi za ljestvičasto programiranje.

Naziv elementa	Funkcija	Izgled elementa
Serijska veza	Omogućava postavljanje elemenata za programiranje u serijskoj vezi do izlaza	
Paralelna veza	Omogućava postavljanje elemenata za programiranje u paralelnoj vezi do izlaza	
NO kontakt	Aktivan kontakt kada je objekt kontrolnog bita u stanju 1	
NC kontakt	Aktivan kontakt kada je objekt kontrolnog bita u stanju 0	
Kontakt rastuće ivice	Aktivan kontakt kada se desi promjena kontrolnog bita iz stanja 0 u stanje 1	
Kontakt opadajuće ivice	Aktivan kontakt kada se desi promjena kontrolnog bita iz stanja 1 u stanje 0	
Blok za poređenje	Upoređuje 2 operanda, te izlaz mijenja u 1 nakon provjere rezultata	
XOR instrukcije	Izvodi XOR operaciju između operanda i rezultata prethodne instrukcije	
Funckionalni blokovi	Svaki od funkcionalnih blokova koristi ulaze i izlaze koji omogućavaju veze sa ostalim grafičkim elementima, opisani u dijelu FBD programiranja.	
Direktna špula	Prikazuje vrijednost rezultata instrukcija lijevo od nje.	
Inverzna špula	Prikazuje inverznu vrijednost rezultata instrukcija lijevo od nje.	
Špula za postavljanje vrijednosti	Prikazuje vrijednost rezultata 1 kada se ispunе instrukcije lijevo od nje.	
Špula za resetovanje vrijednosti	Prikazuje vrijednost rezultata 0 kada se ispunе instrukcije lijevo od nje.	
Aktivacija koraka GRAFCET	Deaktivira trenutni korak ili aktivira drugi korak u GRAFCET programu.	
Deaktivacija koraka GRAFCET	Deaktivira korak u GRAFCET programu.	
Operativni blok	Dodavanje operacija u mrežu.	
Ostali Ladder elementi	Dodavanje for petlje, dodavanje if, bloka za negiranje i slično.	

4.2 Primjeri ljestvičastog programiranja

4.2.1 Primjer 1.

Implementacija logičkih funkcija

Potrebno je implementirati sljedeće logičke funkcije:

$$Q1 = I1 \wedge I2$$

$$Q2 = I1 \vee I2$$

$$Q3 = \overline{I1} \wedge \overline{I2}$$

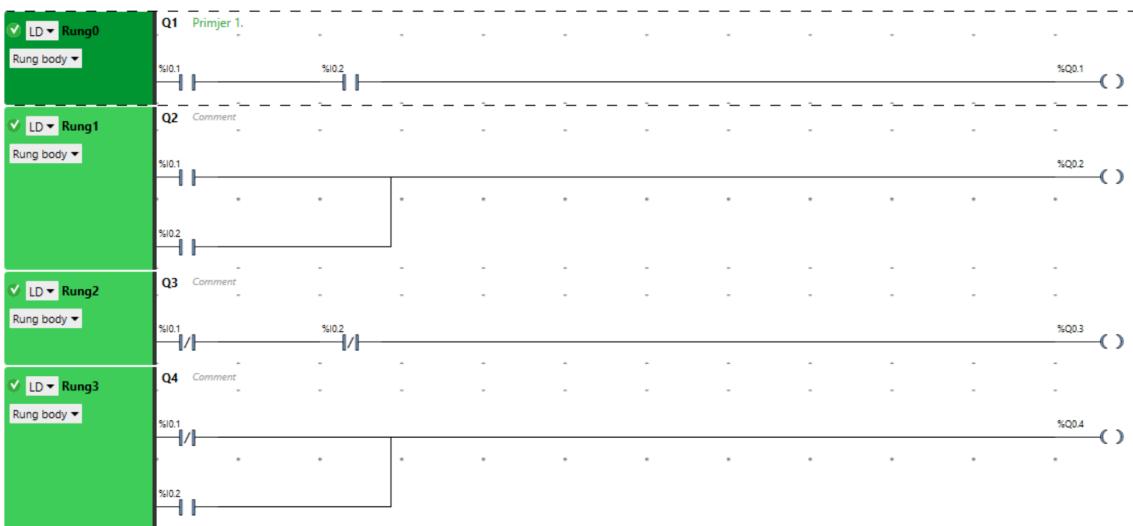
$$Q4 = \overline{I1} \vee I2$$

Prilikom rješavanja problema potrebno je koristiti 2 digitalna ulaza ($I1, I2$) i 4 digitalna izlaza ($Q1, Q2, Q3, Q4$) [12].

U ovom primjeru je obrađen jednostavan zadatak za koji je potrebno implementirati osnovne logičke operacije.

Za logičku operaciju "AND" je potrebno koristiti serijsku vezu dva kontakta. Za logičku operaciju "OR" je potrebno koristiti paralelnu vezu dva kontakta. Ukoliko je varijabla negirana, korišten je negirani kontakt. Špulama su dodijeljene izlazne varijable.

Rješenje navedenog primjera prikazano je na slici 4.1.



Slika 4.1: Rješenje primjera pomoću ljestvičastog programiranja.

4.2.2 Primjer 2.

Aktivacija i deaktivacija mašine

Potrebno je realizirati aktivaciju i deaktivaciju mašine korištenjem "START" i "STOP" tastera. Prilikom pokretanja i zaustavljanja mašina mora ispuniti sljedeće funkcionalnosti:

- nakon pritiska na "START" taster mašina se aktivira istog trenutka,
- nakon pritiska na "STOP" taster mašina se zaustavlja nakon 4 sekunde [12].

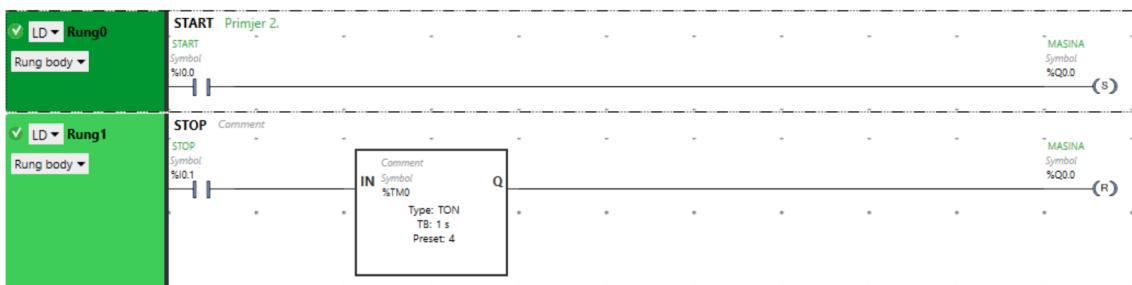
Tabela 4.2: Lista signala za drugi primjer.

Naziv	Tip signala
START	digitalni ulaz
STOP	digitalni ulaz
Mašina	digitalni izlaz

Za rješavanje ovog primjera, korišten je tajmer tipa TON, kod kojeg prednja ivica izlaznog signala kasni za zadato vrijeme koje u ovom slučaju iznosi 4 sekunde.

S obzirom da se aktivacija mašine obavlja tasterom, korištena je špula koja obavlja funkciju RS flip-flopa (upotrijebljeni elementi *Set coil* (S) i *Reset coil* (R)).

Na slici 4.2 prikazano je rješenje navedenog primjera ljestvičastim načinom programiranja.



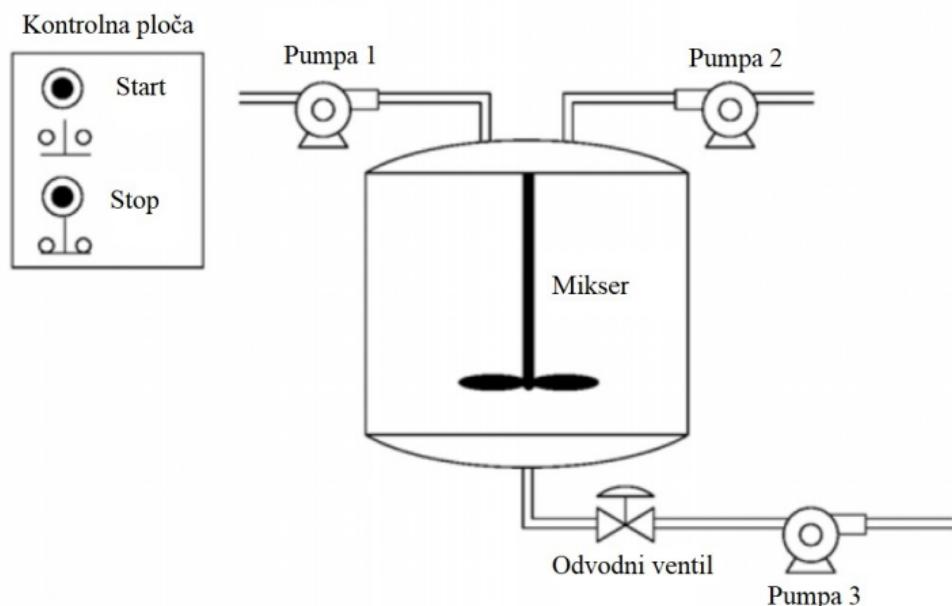
Slika 4.2: Rješenje drugog primjera pomoću ljestvičastog programiranja.

4.2.3 Primjer 3.

Hemijsko postrojenje

Napisati program za PLC koji realizuje proces prikazan na slici 4.7. Proces se odvija na sljedeći način:

- Proces miješanja započinje pritiskom na taster start, nakon čega se uključenjem pumpe 1 u trajanju od 5 sekundi puni rezervoar prvom hemikalijom.
- Nakon toga se uključenjem pumpe 2 u trajanju od 4 sekunde puni rezervoar drugom hemikalijom.
- Nakon toga se ponovnim uključenjem pumpe 1 u trajanju od 5 sekunde puni do vrha rezervoar i drugom hemikalijom.
- Nakon punjenja uključuje se motor mješalice u trajanju 20 sekundi radi dobijanja homogenog rastvora, nakon čega se uključuje ventil i pumpa 3 u trajanju 15 sekundi.
- Rezervoar se prazni i proces zaustavlja.
- Proces se može prekinuti i ranije pritiskom na taster stop. [12]



Slika 4.3: Prikaz hemijskog postrojenja.

Za rješenje ovog zadatka pomoću Ladder dijagrama potrebno je koristiti pet tajmera zbog različitih perioda uključenja. Prvi tajmer je pet sekundi i označen je sa T1, drugi tajmer T2 je četiri sekunde, treći tajmer T3 je 14 sekundi, tajmer T4 je 34 sekunde, i konačno tajmer T5 je 49 sekundi. Realizirani su i tasteri START i STOP. Tasterom START se pomoću *Set coil* aktivira sistem ON/OFF, dok taster STOP pomoću *Reset coil* zaustavlja sistem ON/OFF. Proces se zaustavlja i ako se izvrše svi koraci, dakle ako isteknu svi tajmeri.

Logičke funkcije kojim se ostvaruje proces navedene su u nastavku :

$$\text{ON/OFF} \wedge ((\overline{T1} \wedge \overline{T2}) \vee (T1 \wedge T2)) \wedge \overline{T3} \wedge \overline{T4} \wedge \overline{T5} = \text{PUMPA-1}$$

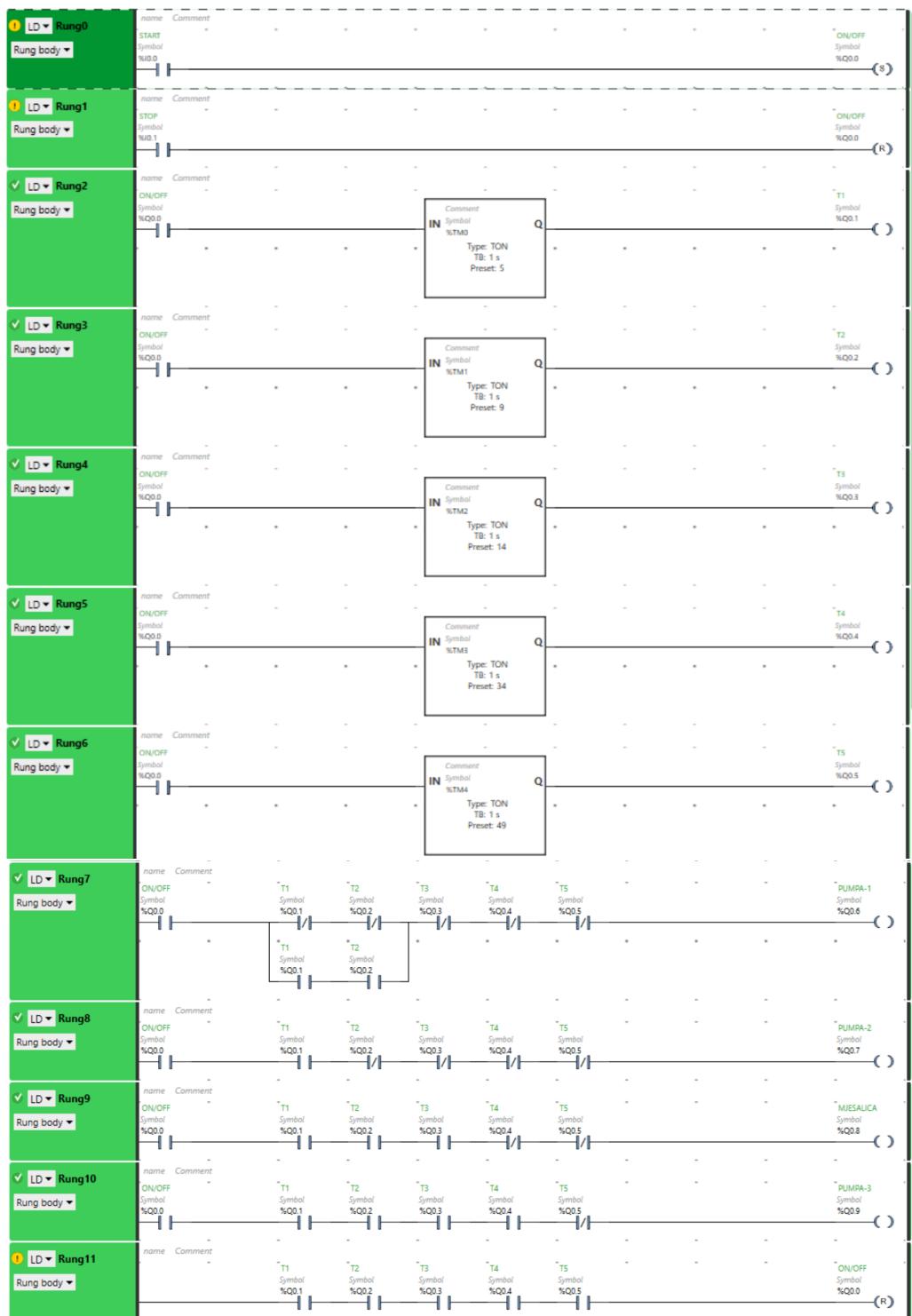
$$\text{ON/OFF} \wedge T1 \wedge \overline{T2} \wedge \overline{T3} \wedge \overline{T4} \wedge \overline{T5} = \text{PUMPA-2}$$

$$\text{ON/OFF} \wedge T1 \wedge T2 \wedge T3 \wedge \overline{T4} \wedge \overline{T5} = \text{MJEŠALICA}$$

$$\text{ON/OFF} \wedge T1 \wedge T2 \wedge T3 \wedge T4 \wedge \overline{T5} = \text{PUMPA-3}$$

$$T1 \wedge T2 \wedge T3 \wedge T4 \wedge T5 = \text{RESET} - \text{ON/OFF}$$

Rješenje navedenog primjera, sa opisanom logikom prikazano je na slici 4.4.



Slika 4.4: Rješenje trećeg primjera pomoću ljestvičastog programiranja.

4.3 Programiranje instrukcijskim listama

Program napisan pomoću liste instrukcija sastoји se od niza instrukcija koje se sekvencialno izvršavaju od strane logičkog kontrolera. Ovaj način programiranja podsjeća na asembler. Svaka instrukcija je predstavljena jednom linijom programa i sastoји se od sljedećih komponenti:

- Broj linije
- Trenutna vrijednost (samo u online modu)
- Operator - koji identificuje operaciju koja se izvodi pomoću operanada
- Operand (i)
- Opcioni komentar

Programiranje pomoću liste instrukcija podržava nekoliko tipova instrukcija koje su prikazane u tabeli 4.3.

Tabela 4.3: Podržani tipovi instrukcija.

Tip instrukcije	Primjer	Funkcija
Logička instrukcija	LD %M10	Učitava vrijednost internog bita %M10
Blok instrukcija	IN %TM0	Pokreće tajmer %TM0
Instrukcija riječi	[%MW10 := %MW50 + 100]	Operacija sabiranja
Programska instrukcija	SR5	Poziva potprogram #5

Sa logičkim operatorima AND i OR koriste se zgrade za realizaciju serijskih i paralelnih veza logičkih instrukcija. Zgrade su povezane s instrukcijama na sljedeći način:

- Otvaranje zagrada je povezano sa AND ili OR operatorom
- Zatvaranje zagrada je instrukcija (operator bez operanda) koja je potrebna za svaku otvorenu zgradu

Zagradama se mogu dodijeliti modifikatori navedeni u tabeli 4.4.

Tabela 4.4: Modifikatori koji se dodjeljuju zagradama.

Modifikatori	Funkcija	Primjeri
N	Negacija	AND(N ili OR(N
F	Opadajuća ivica	AND(F ili OR(F
R	Rastuća ivica	AND(R ili OR(R
[Poređenje	Uz instrukcije za poređenje

Za programiranje pomoću instrukcijskih listi i za korištenje grananja potrebno je pridržavati se sljedećih pravila:

- Svaka otvorena zgrada mora imati odgovarajuću zatvorenu zgradu
- Oznake (%Li:), potprogrami (SRi:), JMP instrukcije (JMP) i instrukcije funkcionalnog bloka ne smiju se stavljati u izraze između zagrada
- Instrukcije za skladištenje (ST, STN, S i R) ne smiju biti programirane između zagrada
- Instrukcije steka (MPS, MRD i MPP) se ne mogu koristiti između zagrada

4.4 Primjeri programiranja instrukcijskim listama

Kako je ranije navedeno u poglavlju 3.4.1, moguće je izvršiti konverziju iz ljestvičastog programiranja u programiranje instrukcijskim listama. Dakle, za prikaz programske logike kao niza instrukcija moguće je iz menija promijeniti prikaz LD u prikaz IL, ili jednostavnije klikom na ikonu . Primjeri prikazani u poglavlju 4.2 će se riješiti konverzijom iz ljestvičastog programiranja u instrukcijsku listu.

4.4.1 Primjer 1.

Implementacija logičkih funkcija

Potrebno je implementirati sljedeće logičke funkcije:

$$Q1 = I1 \wedge I2$$

$$Q2 = I1 \vee I2$$

$$Q3 = \overline{I1} \wedge \overline{I2}$$

$$Q4 = \overline{I1} \vee I2$$

Prilikom rješavanja problema potrebno je koristiti 2 digitalna ulaza (I1, I2) i 4 digitalna izlaza (Q1, Q2, Q3, Q4) [12].

Rješenje navedenog primjera programirano listom instrukcija prikazano je na slici 4.5.



Slika 4.5: Rješenje prvog primjera pomoću instrukcijske liste.

4.4.2 Primjer 2.

Aktivacija i deaktivacija mašine

Potrebno je realizirati aktivaciju i deaktivaciju mašine korištenjem "START" i "STOP" tastera. Prilikom pokretanja i zaustavljanja mašina mora ispuniti sljedeće funkcionalnosti:

- nakon pritiska na "START" taster mašina se aktivira istog trenutka,
- nakon pritiska na "STOP" taster mašina se zaustavlja nakon 4 sekunde [12].

Tabela 4.5: Lista signala za drugi primjer.

Naziv	Tip signala
START	digitalni ulaz
STOP	digitalni ulaz
Mašina	digitalni izlaz

Rješenje navedenog primjera programirano listom instrukcija prikazano je na slici 4.6.



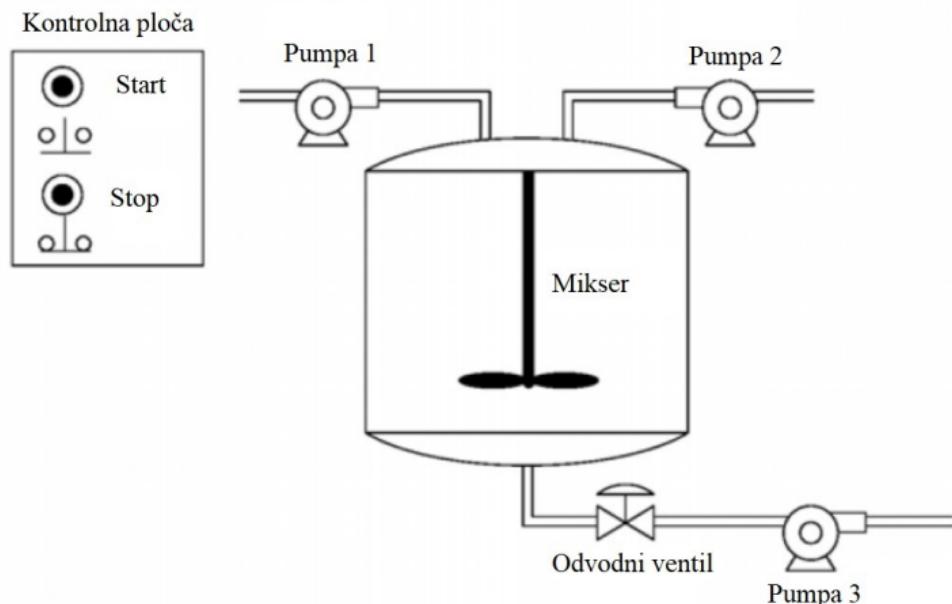
Slika 4.6: Rješenje drugog primjera pomoću instrukcijske liste.

4.4.3 Primjer 3.

Hemijsko postrojenje

Napisati program za PLC koji realizuje proces prikazan na slici 4.7. Proces se odvija na sljedeći način:

- Proces miješanja započinje pritiskom na taster start, nakon čega se uključenjem pumpe 1 u trajanju od 5 sekundi puni rezervoar prvom hemikalijom.
- Nakon toga se uključenjem pumpe 2 u trajanju od 4 sekunde puni rezervoar drugom hemikalijom.
- Nakon toga se ponovnim uključenjem pumpe 1 u trajanju od 5 sekunde puni do vrha rezervoar i drugom hemikalijom.
- Nakon punjenja uključuje se motor mješalice u trajanju 20 sekundi radi dobijanja homogenog rastvora, nakon čega se uključuje ventil i pumpa 3 u trajanju 15 sekundi.
- Rezervoar se prazni i proces zaustavlja.
- Proces se može prekinuti i ranije pritiskom na taster stop [12].



Slika 4.7: Prikaz hemijskog postrojenja.

Rješenje navedenog primjera programirano listom instrukcija prikazano je na slici 4.8.

Rung0	name Comment 0000 LD %I0.0 0001 S %Q0.0	Rung7	name Comment 0000 LD %Q0.0 0001 AND (N) %Q0.1 0002 ANDN %Q0.2 0003 OR (%Q0.1 0004 AND %Q0.2 0005) 0006) 0007 ANDN %Q0.3 0008 ANDN %Q0.4 0009 ANDN %Q0.5 0010 ST %Q0.6
Rung1	name Comment 0000 LD %I0.1 0001 R %Q0.0	Rung8	name Comment 0000 LD %Q0.0 0001 AND %Q0.1 0002 ANDN %Q0.2 0003 ANDN %Q0.3 0004 ANDN %Q0.4 0005 ANDN %Q0.5 0006 ST %Q0.7
Rung2	name Comment 0000 BLK %TM0 0001 LD %Q0.0 0002 IN 0003 OUT_BLK 0004 LD Q 0005 ST %Q0.1 0006 END_BLK	Rung9	name Comment 0000 LD %Q0.0 0001 AND %Q0.1 0002 AND %Q0.2 0003 AND %Q0.3 0004 ANDN %Q0.4 0005 ANDN %Q0.5 0006 ST %Q0.8
Rung3	name Comment 0000 BLK %TM1 0001 LD %Q0.0 0002 IN 0003 OUT_BLK 0004 LD Q 0005 ST %Q0.2 0006 END_BLK	Rung10	name Comment 0000 LD %Q0.0 0001 AND %Q0.1 0002 AND %Q0.2 0003 AND %Q0.3 0004 AND %Q0.4 0005 ANDN %Q0.5 0006 ST %Q0.9
Rung4	name Comment 0000 BLK %TM2 0001 LD %Q0.0 0002 IN 0003 OUT_BLK 0004 LD Q 0005 ST %Q0.3 0006 END_BLK	Rung11	name Comment 0000 LD %Q0.1 0001 AND %Q0.2 0002 AND %Q0.3 0003 AND %Q0.4 0004 AND %Q0.5 0005 R %Q0.0
Rung5	name Comment 0000 BLK %TM3 0001 LD %Q0.0 0002 IN 0003 OUT_BLK 0004 LD Q 0005 ST %Q0.4 0006 END_BLK		
Rung6	name Comment 0000 BLK %TM4 0001 LD %Q0.0 0002 IN 0003 OUT_BLK 0004 LD Q 0005 ST %Q0.5 0006 END_BLK		

Slika 4.8: Rješenje trećeg primjera pomoću instrukcijske liste.

4.5 FBD programiranje

Funkcionalni blok dijagrami (FBD) nude grafički pristup programiranju PLC-ova koristeći posebno definisane funkcionalne blokove. Neke od prednosti ovog načina programiranja su:

- blokovi su modularni i pogodni su za višekratnu upotrebu;
- imaju i podesive parametre koji se dalje mogu koristiti u različitim dijelovima programa, poboljšavajući efikasnost koda i mogućnost održavanja;
- omogućavaju implementiranje složenijih funkcionalnosti u sistemima sa obradom podataka i složenim algoritmima upravljanja

Koristi blokove koji obuhvataju ulazne/izlazne signale, logičke operacije, tajmere, brojače i druge funkcije. Kao i kod ljestvičastih dijagrama, ovi blokovi su međusobno povezani linijama koje označavaju putanju signala, a program se izvršava:

- odozgo prema dole unutar mreža po mreža,
- odozgo prema dole unutar mreže, i
- s lijeva na desno unutar mreže.

I za ovo programiranje potrebno je poznavati grafičke elemente koji se koriste, prikazani su u tabeli 4.6. Slično kao i kod ljestvičastog programiranja, blokovi se ubacuju pritiskom na željeni blok i ispuštanjem na radni prostor za programiranje u ćeliju mreže. Unutar korištenog programskog okruženja ovi blokovi se biraju pritiskom na ikonu .

Tabela 4.6: Grafički elementi za FBD programiranje.

Naziv elementa	Funkcija	Izgled elementa
Tajmer	Koristi se za određivanje vremenskog perioda prije nego što se nešto učini, na primjer, pokretanje događaja.	
Registrar	Memorijski blok koji može pohraniti do 16 riječi od 16 bita	
Pomični registar	Omogućava lijevi/desni pomak bita podataka	
Brojač	Broji diskretne događaje kao broj promjena stanja ulaznog signala	
Brzi brojači	Broje događaje koji se događaju velikom brzinom. Hardverski su bazirani i mogu raditi na mnogo višim frekvencijama od standardnih brojača.	
Step brojač	Prate napredak kroz sekvenčne faze ili korake u procesu.	
Drum	Koristi se za kontrolu sekvenčnog izvršavanja niza operacija ili koraka u procesu	
Sat u realnom vremenu (RTC)	Prati vrijeme u realnom vremenu ili ažurira vrijeme u PLC-u sa korisnički definiranim vremenom i datumom	
Generator impulsa	Generiše digitalne impulse određene frekvencije i trajanja	
PWM	Omogućava promjenu širine impulsa u periodičnom signalu (radi kontrole isporučene snage)	
Zapisivanje podataka (data logging)	Omogućava trajno pohranjivanje podataka iz objekata ili nizova	
PTO	Generiše precizne sekvence digitalnih impulsa sa podešivim parametrima.	
Drive objekti	Kontrolisu ATV pogone i druge uređaje konfigurisane na Modbus Serial IOScanner ili Modbus TCP IOScanner	
Komunikacije	Koriste se za komunikaciju sa Modbus uređajima, slanje/primanje ASCII i SMS poruka	
Korisnički definisane funkcije	Omogućavaju kreiranje blokova koji obavljaju neku logiku	

4.6 Primjeri FBD programiranja

4.6.1 Primjer 1.

Pokretna traka za brojanje proizvoda

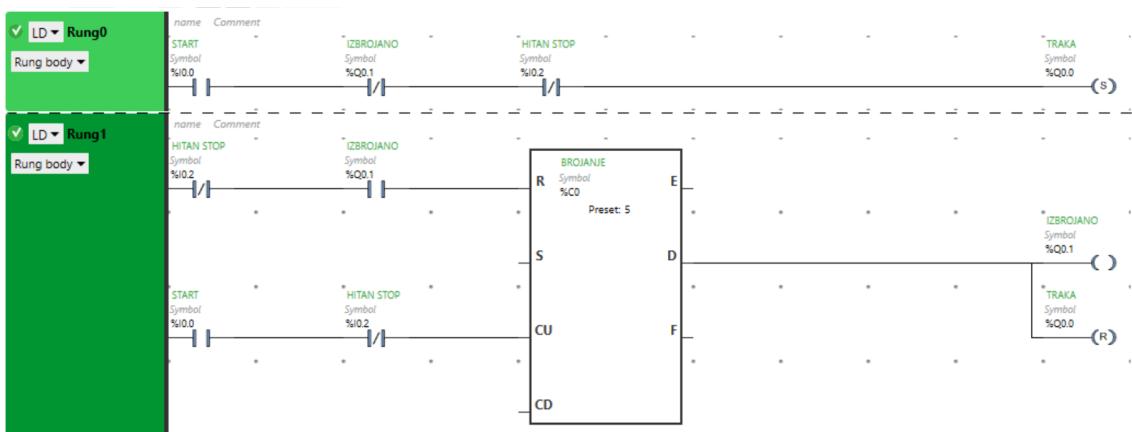
Potrebno je realizovati brojač proizvoda koji se detektuju na pokretnoj traci uz uslove:

- traka se pokreće tasterom START, a zaustavlja se svaki put kada je izbrojano 5 proizvoda,
- dodati taster "HITAN STOP" koji omogućava hitno zaustavljanje trake i onemogućavanje daljeg brojanja, kao i onemogućavanje njenog ponovnog uključenja tasterom "START",
- ponovnim pritiskom na taster "HITAN STOP", program nastavlja s radom [12].

Tabela 4.7: Lista signala za prvi primjer

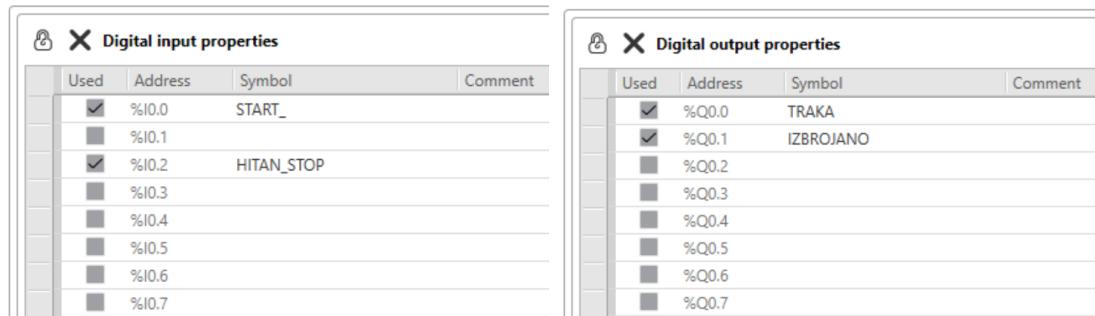
Naziv	Tip signala
START	digitalni ulaz
HITAN STOP	digitalni ulaz
senzor	digitalni ulaz
traka	digitalni izlaz

Traka se pokreće pritiskom na taster "START" te je u tu svrhu potrebno koristiti RS flip-flop. Za brojanje proizvoda je korišten brojač koji daje logičku vrijednost "TRUE" na izlazu kada se izbroji pet proizvoda (IZBROJANO). Nakon što se proizvodi izbroje, traka se zaustavlja i brojač se resetuje (resetovanje se vrši varijablom IZBROJANO) te se traka može ponovo pokrenuti pritiskom na taster "START". Rješenje navedenog primjera pomoću funkcionalnih blok dijagrama prikazano je na slici 4.9.



Slika 4.9: Rješenje prvog primjera pomoću funkcionalnih blok dijagrama.

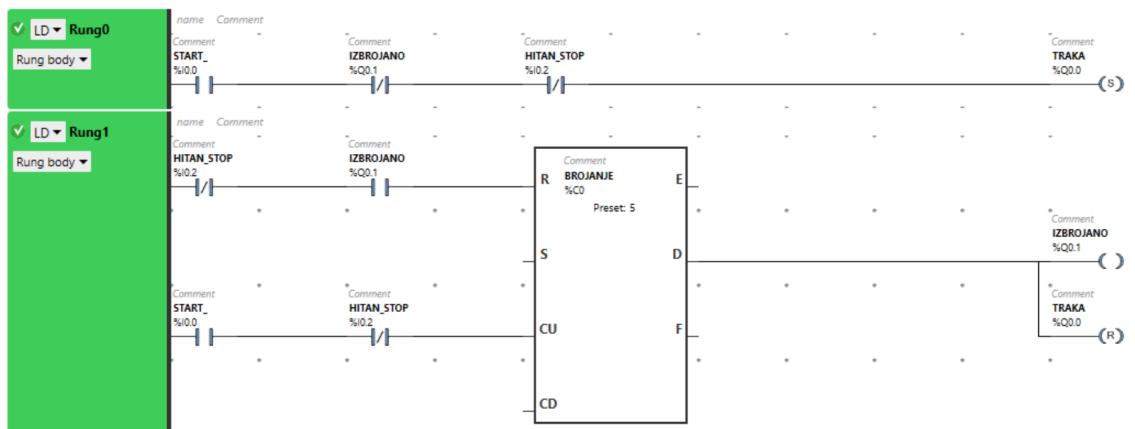
Na slici 4.9 je moguće primijetiti da je za oznake elemenata korišten komentar. Za složenije zadatke potrebno je umjesto komentara koristiti simbole zbog lakše manipulacije varijablama. Dakle, ukoliko se komentari zamijene sa simbolima moguće je novi dodani element dodijeliti samo pretragom naziva simbola, umjesto pamćenjem ulaza i izlaza. Simboli dodijeljeni ulazima i izlazima prikazani su na slici 4.10. Dodjela simbola vrši se dvoklikom na kolonu simbol u redu za koji je potrebno unijeti isti. Važno je napomenuti da je dodavanje simbola omogućeno za sve blokove (u ovom primjeru za brojač), a ne samo za ulaze i izlaze sistema.



The image shows two dialog boxes side-by-side. The left box is titled 'Digital input properties' and the right box is titled 'Digital output properties'. Both boxes have columns for 'Used', 'Address', 'Symbol', and 'Comment'. In the 'Digital input properties' box, rows 1, 3, and 5 have checked 'Used' boxes and contain symbols '%I0.0', '%I0.2', and '%I0.6' respectively, with comments 'START_', 'HITAN_STOP', and '%I0.6'. Rows 2, 4, 6, 7, and 8 have unchecked 'Used' boxes and contain addresses '%I0.1', '%I0.3', '%I0.4', '%I0.5', and '%I0.7'. In the 'Digital output properties' box, rows 1 and 2 have checked 'Used' boxes and contain symbols 'TRAKA' and 'IZBROJANO' respectively, with comments '%Q0.0' and '%Q0.1'. Rows 3 through 8 have unchecked 'Used' boxes and contain addresses '%Q0.2' through '%Q0.7'.

Slika 4.10: Simboli dodijeljeni ulaznim i izlaznim elementima.

Izgled programa nakon unosa simbola umjesto komentara za svaku instanciranu varijablu prikazan je na slici 4.11.



Slika 4.11: Rješenje prvog primjera pomoću funkcionalnih blok dijagrama sa definisanim simbolima.

4.6.2 Primjer 2.

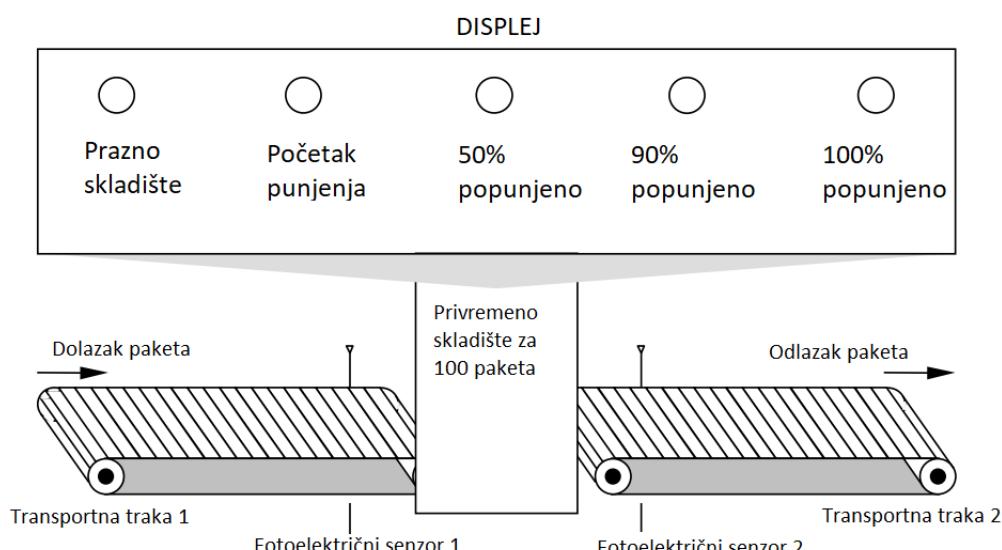
Skladište

Napisati program za PLC koji realizuje proces prikazan na slici 4.12. Slika prikazuje sistem sa dvije transportne trake i privremenim skladišnim prostorom između njih.

Transportna traka 1 dostavlja pakete u skladište. Fotoelektrični senzor na kraju transportne trake 1 u blizini skladišnog prostora određuje koliko se paketa isporučuje u skladište.

Transportna traka 2 transportuje pakete od privremenog skladišta do utovarnog pristaništa gde kamioni odvoze pakete za isporuku kupcima. Fotoelektrični senzor na kraju transportne trake 2 u blizini prostora za skladištenje određuje koliko paketa napusti prostor za skladištenje i ide do utovarnog pristaništa.

DISPLAY sa pet lampica pokazuje nivo napunjenosti prostora za privremeno skladištenje.



Slika 4.12: Prikaz sistema skladišta.

Rješenje navedenog zadatka je realizirano pomoću funkcionalnih blok dijagrama. Pritiskom na START pokreće se obje transportne trake (TRAKA-1 i TRAKA-2) i počinje dolazak paketa. Sve dok senzor 1 ne detektuje prisustvo paketa svijetli lampica za prazno skladište (PRAZNO). Kada senzor 1 (PAKET+) detektuje prisustvo paketa ugasi se lampica praznog skladišta, te se pali lampica za početak punjenja (POČETAK).

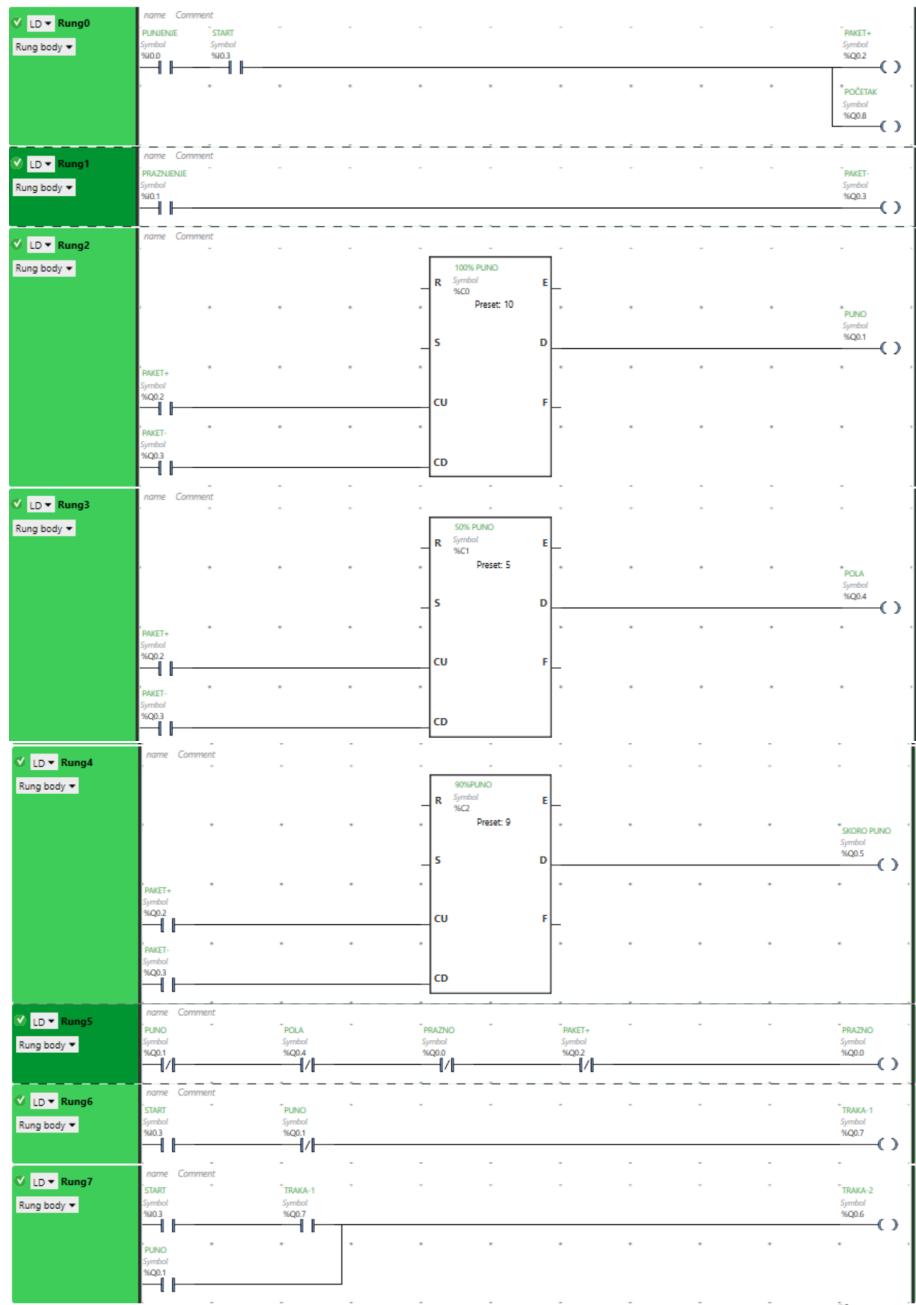
Povećavanje i smanjivanje broja paketa vrši se pomoću registrovanja senzora 1 i senzora 2 (PUNJENJE i PRAŽNJENJE).

Rješenje je dalje realizirano sa tri brojača iz razloga što ne postoji brojač koji prikazuje trenutnu vrijednost kao varijablu. Dakle, poteban je brojač do 50 zbog signalizacije 50% popunjenoosti (50% PUNO), brojač do 90 zbog signalizacije 90% popunjenoosti (90% PUNO) i konačno brojač do 100 zbog signalizacije 100% popunjenoosti (100% PUNO).

Ukoliko je u skladištu 50 paketa upaljena je signalizacija za 50% popunjenoosti (POLA). Ako se u privremenom skladištu nalazi 90 paketa upaljena je signalizacija za 90% popunjenoosti (SKORO PUNO), dok ukoliko je u skladištu 100 paketa upaljena je signalizacija za 100% popunjenoosti (PUNO).

Kada je skladište puno, traka 1 za dolazak paketa je isključena i onemogućen dolazak paketa dok se skladište ne počne prazniti, dok traka 2 konstantno radi zbog potrebe za odlaskom paketa iz privremenog skladišta.

Na slici ispod prikazana je realizacija navedenog primjera pomoću funkcionalnih blok dijagrama. Vrijednost maksimalnog broja paketa u privremenom skladištu je 10 zbog jednostavnijeg testiranja rada, dok promjena vrijednosti na 100 ne mijenja funkcionalnost zadatka.



Slika 4.13: Rješenje drugog primjera pomoću funkcionalnih blok dijagrama.

4.6.3 Primjer 3.

Automat za naplaćivanje

Potrebno je implementirati automat za naplaćivanje iz dva dijela. Prvi dio se odnosi na detekciju kupca, signalizaciju koraka prilikom kupovine i puštanje kupaca kroz rampu nakon obavljene naplate. Drugi dio odnosi se na brojanje novca, te ukoliko je kupac premašio očekivani dio vremena za naplatu, automat vraća novac i vraća se u početno stanje. Dva dijela se implementiraju odvojeno:

1. Prvi dio treba da na osnovu podatka sa senzora implementira detekciju kupca (pretpostaviti da senzorski sistem daje "1" kada je kupac prisutan i "0" ukoliko nije). Nakon uspješne detekcije kupca signalizirati odgovarajućom lamicom da je započelo stanje u kojem kupac može ubacivati novac. Nakon detekcije potrebno je aktivirati Timer koji dozvoljava ubacivanje novca određeno vrijeme. Nakon 30 sekundi, automat treba signalizirati kupcu da je prošlo pola vremena i da bi kupac trebao uskoro finalizirati proces naplate. Nakon narednih 30 sekundi, automat treba signalizirati kupcu da naplata nije uspješno izvršena, te nakon toga automat se vraća u početno stanje detekcije kupca. Ukoliko unutar 60 sekundi od paljenja Timer-a automat dobije podatak da je kupac uspješno uplatio, automat treba to posebno signalizirati, te nakon toga, vratiti se u početno stanje.
2. U početnom stanju ovog automata, treba signalizirati kupcu da trenutno automat ne prima novac. U ovom stanju automat ostaje sve dok ne dobije ulazni signal (od prvog automata) koji označava početak prihvatanja i brojanja uplate. Radi jednostavnosti pretpostavlja se da svaki kupac mora ubaciti ukupno 3 KM, a da automat prima samo kovanice od 1 KM. Ukoliko kupac izvrši uplatu potrebno je to signalizirati odgovarajućom lamicom. Ukoliko u bilo kojem trenutku automat dobije ulazni signal (od prvog automata) da je vrijeme za naplatu isteklo, automat treba to signalizirati i vratiti se u početno stanje.

Prvo će se pružiti uvid u ulaze i izlaze iz sistema (slika 4.14). Postoje samo dva ulaza u sistem, senzor koji detektuje prisustvo kupca i unos kovanice od 1KM. Izlazi se generalno sastoje od svjetlosne signalizacije: stanje kada se smije i ne smije ubacivati novac, naznaka da je naplata uspješno izvršena, naznaka da je pola vremena za uplatu isteklo, i da je svo vrijeme isteklo.

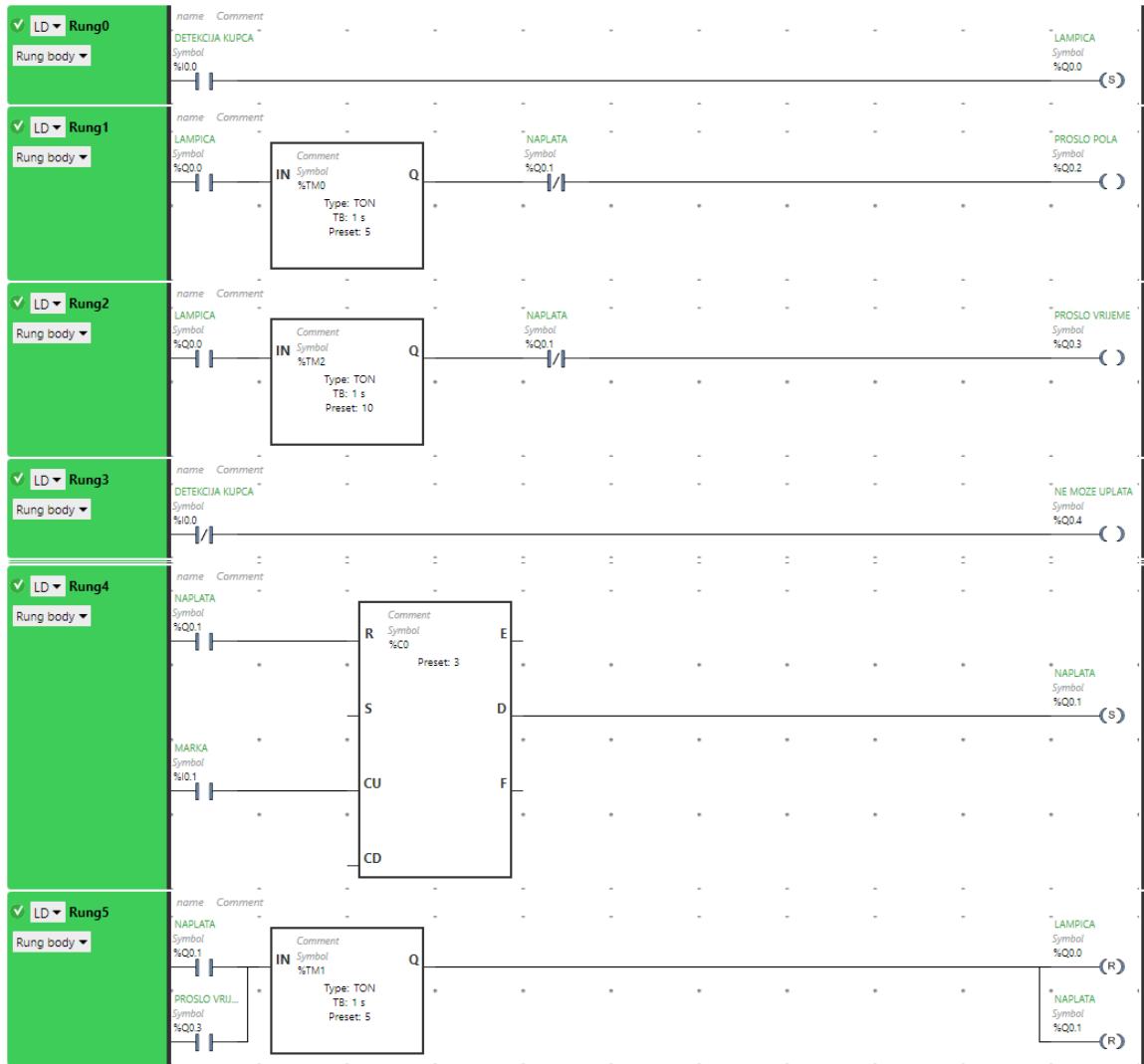
Used	Address	Comment
<input checked="" type="checkbox"/>	%Q0.0	LAMPICA
<input checked="" type="checkbox"/>	%Q0.1	NAPLATA
<input checked="" type="checkbox"/>	%Q0.2	PROSLO POLA
<input checked="" type="checkbox"/>	%Q0.3	PROSLO VRIJEME
<input checked="" type="checkbox"/>	%Q0.4	NE MOZE UPLATA

Slika 4.14: Korišteni ulazi i izlazi.

Predloženo softversko rješenje zadatka se može vidjeti na slici 4.15. Detekcijom kupca se aktivira signalizacija za ubacivanje novca (red 1), čime se aktiviraju tajmeri kojim se prati koliko je vremena prošlo od prilaska kupca. Ako je prošlo vrijeme a naplata se i dalje nije izvršila aktiviraju se odgovarajući indikatori (redovi 2 i 3). Dok kupac nije detektovan, aktivirana je

signalizacija koja označava da nije još moguća naplata (red 4). Dalje se koristi brojač koji se inkrementira na svaki novi unos kovanice u automat. Kada se dostigne postavljeni limit od 3, varijabla koja označava uspješnu naplatu poprima vrijednost logičke jedinice, čime ujedno i resetuje brojač (red 5). Završetkom naplate ili preostalog vremena pokreće se još jedan tajmer nakon kojeg se sistem vraća u početno stanje (red 6).

Može se primjetiti da su korišteni set i reset ulazi za varijable koje označavaju početak i kraj naplate. To je rađeno kako bi se neometano mogle desiti tranzicije između tih stanja.



Slika 4.15: Rješenje trećeg primjera pomoću FBD programiranja.

4.7 GRAFCET Lista programiranje

Grafcet (Lista) programiranje u EcoStruxure Machine Expert - Basic nudi jednostavnu metodu prevođenja kontrolne sekvence u korake. Moguće je prevesti kontrolne sekvence u Grafcet korake, a zatim koristiti ove korake u programu uz Grafcet uputstva. Maksimalan broj Grafcet koraka zavisi od kontrolera. Broj aktivnih koraka u bilo kojem trenutku ograničen je samo ukupnim brojem koraka [13].

Program EcoStruxure Machine Expert - Basic kreiran kao Grafcet (Lista) ima sljedeće dijelove:

- Pred-obrada
- Sekvencijalna obrada
- Post-obrada

Pred-obrada

Pred-obrada se sastoji od sljedećeg:

- Povratak napajanja
- Upravljanje greškama
- Promjena načina rada
- Prepozicioniranje Grafcet koraka
- logika unosa

U primjeru koji je prikazan na slici 4.16, sistemski bit %S21 je postavljen na 1 sa rastućom ivicom ulaza %I0.1 (Rung1). Ovo deaktivira aktivne korake i aktivira početne korake.



Slika 4.16: Primjer programiranja GRAFCET (Lista).

Pred-obrada počinje s prvim redom programa i završava se prvim pojavljivanjem ($= * =$) ili ($- * -$) instrukcije. Sistemski biti %S21, %S22 i %S23 su posvećeni Grafcet kontroli. Svaki

od ovih sistemskih bitova aplikacija postavlja na 1 (ako je potrebno), obično u predobradi. Povezanu funkciju izvodi sistem na kraju pred-obrade i sistemski bit se tada resetuje na 0. U tabeli 4.8 prikazani su sistemski biti korišteni za Grafset kontrolu.

Tabela 4.8: Sistemski biti za Grafset kontrolu.

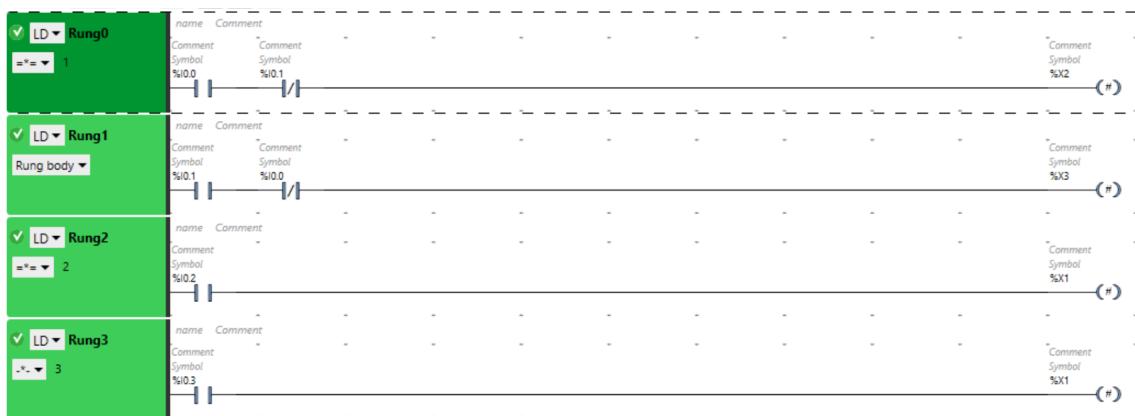
Sistemski bit	Naziv	Opis
%S21	Inicijalizacija grafceta	Svi aktivni koraci su deaktivirani, a početni koraci su aktivirani.
%S22	Ponovna inicijalizacija grafceta	Svi koraci su deaktivirani.
%S23	Grafset pre-pozicioniranje	Ovaj bit mora biti postavljen na 1 ako su $\%Xi$ objekti eksplicitno napisani od strane aplikacije u pred-obradi. Ako se ovaj bit održava na 1 preprocesiranjem bez eksplicitne promjene $\%Xi$ objekata, Grafset je zamrznut (nikakva ažuriranja se ne uzimaju u obzir).

Sekvencijalna obrada

Sekvencijalna obrada se odvija u grafikonu (instrukcije koje predstavljaju grafikon) [13]:

- Koraci
- Radnje povezane sa koracima
- Tranzicije
- Uslovi tranzicije

Sekvencijalna obrada završava se izvršavanjem POST instrukcije ili završetkom programa. Primjer sekvencijalnog programiranja je prikazan na slici 4.18, dok je primjer sekvencijalnog programiranja pomoću Ladder dijagrama prikazan na slici 4.17.



Slika 4.17: Primjer sekvencijalnog programiranja - Ladder dijagrami.



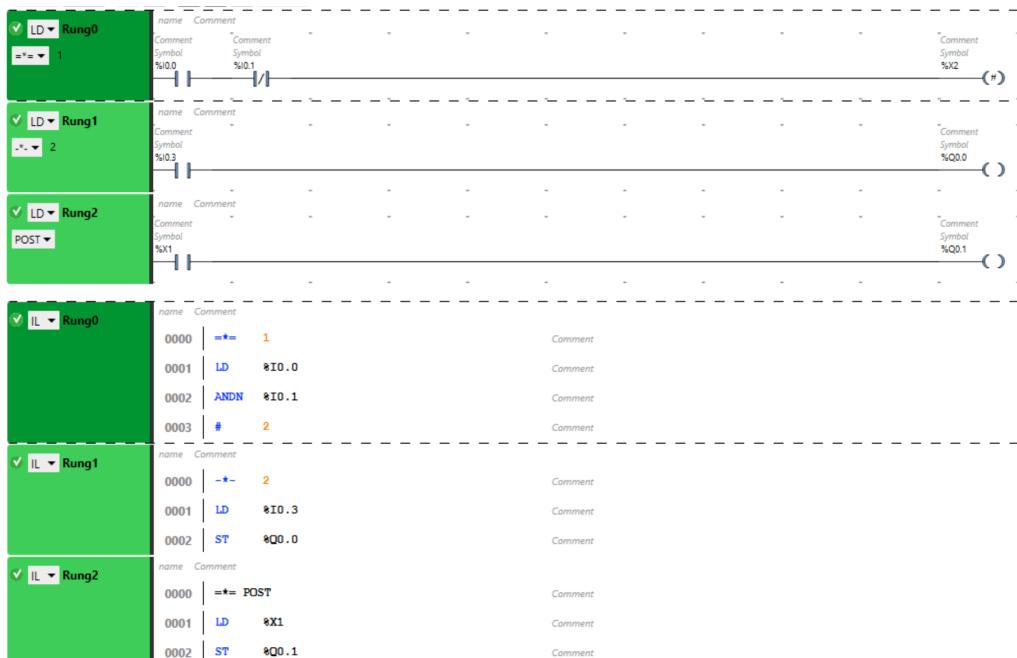
Slika 4.18: Primjer sekvencijalnog programiranja - GRAFCET (Lista).

Post-obrada

Post-obrada se sastoji od:

- Naredbe iz sekvencijalne obrade za kontrolu izlaza
- Blokade specifične za izlaze

Primjer post-obrade prikazan je ljestivčastim načinom programiranja i listom instrukcija na slici 4.19.



Slika 4.19: Primjer post-obrade.

4.8 GRAFCET (SFC) progamiranje

Grafcet (SFC) je grafički programski jezik koji opisuje vremenski redoslijed izvršavanja diskretnih zadataka, poznatih kao koraci. Redoslijed kojim se koraci izvršavaju određen je prijelazima/tranzicijama koji povezuju korake. Grafcet (SFC) POU ima sljedeće komponente:

- **Korak:** Korak izvršava skup akcija definisanih u jednoj ili više naredbi napisanih u Ladder/IL programskim jezicima. Koraci mogu biti:
 - Početni korak: Izvršava se na početku programa ili nakon ponovnog pokretanja kontrolera. Predstavljen je čelijom sa dvostrukim rubom.
 - Obični korak: Koraci koji se uslovno izvršavaju nakon što se početni korak izvrši.
- **Prijelaz/Tranzicija:** Logički izraz koji se nalazi između koraka i predstavlja vezu između dva ili više koraka. Logički izraz je definiran u jednoj prijelaznoj naredbi napisanoj u Ladder/IL programskom jeziku. [13]

Grafcet (SFC) obrada

Logički kontroler primjenjuje sljedeća pravila prilikom izvršavanja Grafceta (SFC):

- Počinje ciklus glavnog zadatka
- Svaki POU koji prethodi prvom koraku Grafceta (SFC) se izvršavaju uzastopno
- Prvi Grafcet (SFC) korak pokreće Grafcet monitor
- Kada se Grafcet monitor završi, poziva se prvi POU koji slijedi posljednji pozvani Grafcet (SFC) korak

Ponašanje monitora grafceta:

1. Logički kontroler obrađuje pridružene Grafcet (SFC) sistemske bite %S21, %S22 i - %S23.
2. Logički kontroler ažurira stanja aktivacije svakog Grafcet (SFC) koraka
 - Koraci označeni za deaktiviranje su deaktivirani
 - Koraci označeni za aktiviranje su aktivirani
 - Koraci označeni da se istovremeno aktiviraju i deaktiviraju će biti deaktivirani ili će ostati aktivirani
 - Liste za aktiviranje i deaktiviranje su resetirane
3. Logički kontroler skenira korake (petlja od najmanjeg definiranog broja koraka do najvećeg definiranog broja koraka). Kada je skenirani korak aktiviran, poziva se pripadajući kod koraka.
4. Kada prijelazni kod aktivira ili deaktivira korak, ova radnja se stavlja na listu za aktivaciju ili deaktivaciju za sljedeći ciklus zadataka
5. Kada se izvrši posljednji aktivni kod koraka, Grafcet monitor se završava

4.8.1 Principi Grafset (SFC)

EcoStruxure Machine Expert - Basic Grafset POU je multi-token koji nije usklađen sa standardom IEC 61131-3.

Principi korištenja Grafset (SFC):

1. Početna situacija kontrolira se koracima definiranim kao početni koraci.
2. Više koraka može biti aktivno istovremeno u Grafset POU.
3. Procesi statusa aktivnog signala odvijaju se duž usmjerenih veza, trigerovani prebacivanjem jednog ili više prijelaza. Smjer procesa prati smjerne veze i kreće se od donje strane prethodnog koraka do gornje strane uzastopnog koraka.
4. Prijelaz se evaluira ako su koraci koji mu neposredno prethode aktivni. Prijelazi se ne evaluiraju ako koraci koji im neposredno prethode nisu aktivni.
5. Prijelaz se pokreće kada su pridruženi uslovi prijelaza zadovoljeni.
6. Pokretanje prijelaza označava deaktivaciju prethodnih koraka koji su povezani s prijelazom, i aktivaciju odmah sljedećih koraka.
7. Pravo aktiviranje ili deaktiviranje koraka izvodi se na početku svakog ciklusa glavnog zadatka.
8. Ako je zadovoljeno više od jednog uvjeta prijelaza u nizu uzastopnih koraka, tada se obrađuje jedan po jedan korak po ciklusu.
9. Ako je korak istovremeno aktiviran i deaktiviran, tada će korak biti detaktiviran ili će ostati aktiviran.
10. Više od jedne grane može biti aktivno s alternativnim granama.
11. Grane koje će se pokrenuti određene su rezultatom prijelaznih uslova prijelaza koji slijede alternativnu granu. Prijelazi grana se obrađuju paralelno.
12. Pokreću se grane sa zadovoljenim prijelazima.
13. Pozivi potprograma se mogu koristiti u koracima. [13]

Tabela 4.9 prikazuje kontrolne bite u GRAFCET (SFC) načinu programiranja.

Tabela 4.9: Kontrolni biti u Grafset (SFC).

Kontrolni bit	Naziv	Opis
%S21	Inicijalizacija Grafseta	Ako je postavljeno na 1, evaluiraju se početni koraci u Grafset POU.
%S22	Grafset reset	Ako je postavljeno na 1, koraci se deaktiviraju i izvršenje se ponovo pokreće.
%S23	Preset i zamrzavanje Grafset-a	Ako je postavljeno na 1, izvršenje Grafset POU se zauzavlja dok se bit ne postavi na 0.
%Xi	Grafset koraci	Biti od %X1 do %Xi su povezani sa koracima grafseta. Bit koraka %Xi je postavljen na 1 kada je odgovarajući korak aktivan, a postavljen na 0 kada je korak deaktiviran. Bit se ne može pisati kada se koristi Grafset (SFC).

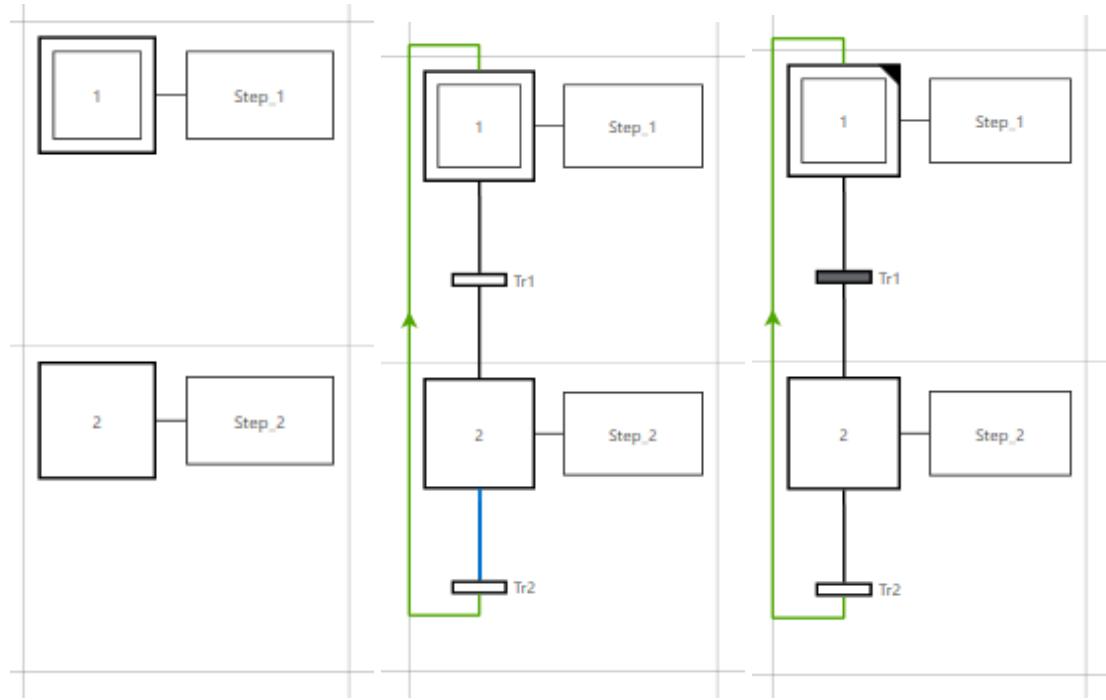
4.8.2 SFC način programiranja

EcoStruxure Machine Expert - Basic nudi mogućnost programiranja sa sekvencijalnim funkcijskim grafikonima (SFC). Desnim klikom na *Master Task* se otvara prozor za SFC programiranje, koji se može vidjeti na slici 4.20.



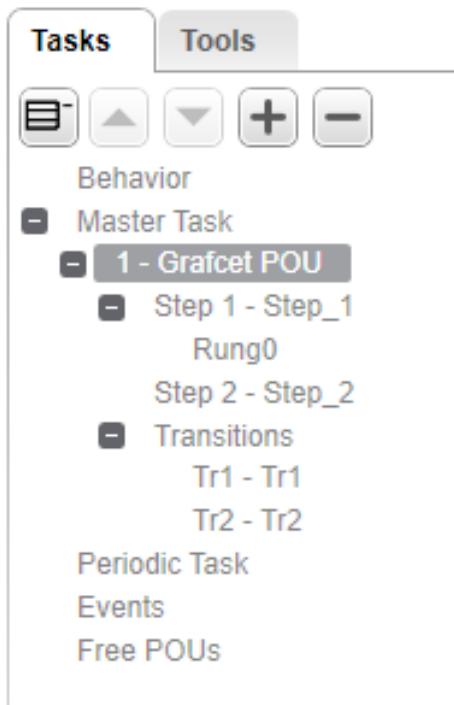
Slika 4.20: Otvaranje prozora za SFC programiranje.

Blokovi se dodaju dvoklikom na željeno polje. Prvi blok predstavlja početnu tačku izvršavanja programa, i on je posebno označen sa dodatnim kvadratom (kao što se može vidjeti na slici 4.21). Povlačenjem sa jednog na drugi step blok se povezuju blokovi, i kreiraju se blokovi tranzicije. Za svaki od ovih blokova se može/treba kreirati odgovarajući kod. Na zadnjem dijelu slike 4.22 se može vidjeti vizuelna razlika između step i tranzicionih blokova za koji jest i za koji nije vezan kod.



Slika 4.21: Kreiranje i povezivanje step blokova.

Prozorima za pisanje koda se može pristupiti iz *Master Task* menija (slika 4.20). Step blokovi se programiraju na klasičan način (ljestvičasti i/ili FBD), dok blokovi tranzicije imaju jednu restrikciju - ne mogu se postavljati izlazni svici (slika 4.23).



Slika 4.22: Prikaz svih step blokova i tranzicija.



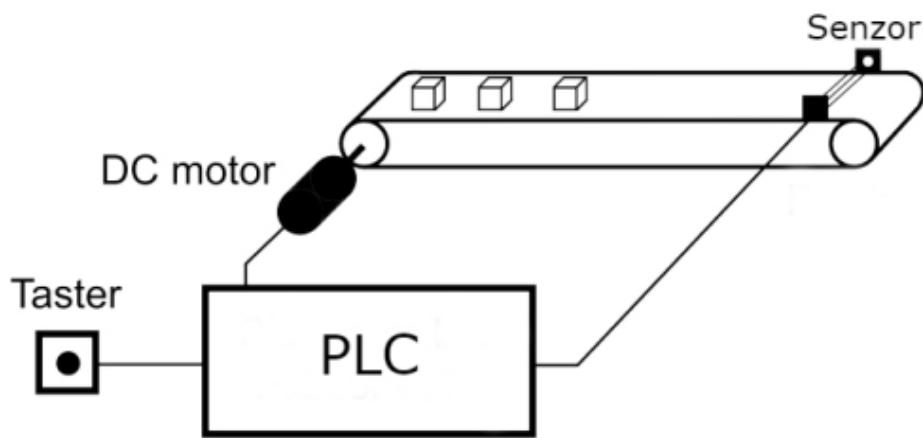
Slika 4.23: Prozor za programiranje tranzicija.

4.9 Primjeri SFC progamiranja

4.9.1 Primjer 1.

Transportna traka

Sistem na bazi PLC-a se koristi za upravljanje transportnom trakom za prenos proizvoda zapakiranih u kutije (Slika 4.24). Traka se pogoni DC motorom, koji se uključuje pomoću releja. Operator pokreće traku pritiskom na taster, a traka se zaustavlja nakon što se detektira prolazak 20 kutija. Za detekciju prolaska kutija se koristi fotoelektrični senzor PNP tipa. Sistem posjeduje prekidač za hitan stop te ukoliko je aktivran, traka se istog momenta zaustavlja do deaktiviranja prekidača.



Slika 4.24: Ilustrativni prikaz sistema.

Tabela 4.10: Tablica signala za prvi primjer.

Naziv	Tip signala
START	digitalni ulaz
HITAN STOP	digitalni ulaz
senzor	digitalni ulaz
motor	digitalni izlaz

Prilikom pokretanja programa, aktivan je Step 1 korak (početak) te traka miruje. Nakon pritiska na taster START, prelazi se na korak Step 2 (brojanje) gdje se vrši pokretanje motora te brojanje proizvoda korištenjem brojača. Resetovanje brojača se obavlja njegovom izlaznom vrijednošću koja je jednaka logičkoj vrijednosti "TRUE" nakon što se izbroji dvadeset kutija. Kada se izbroji potreban broj kutija, preko naredbe skoka se prelazi u početno stanje.

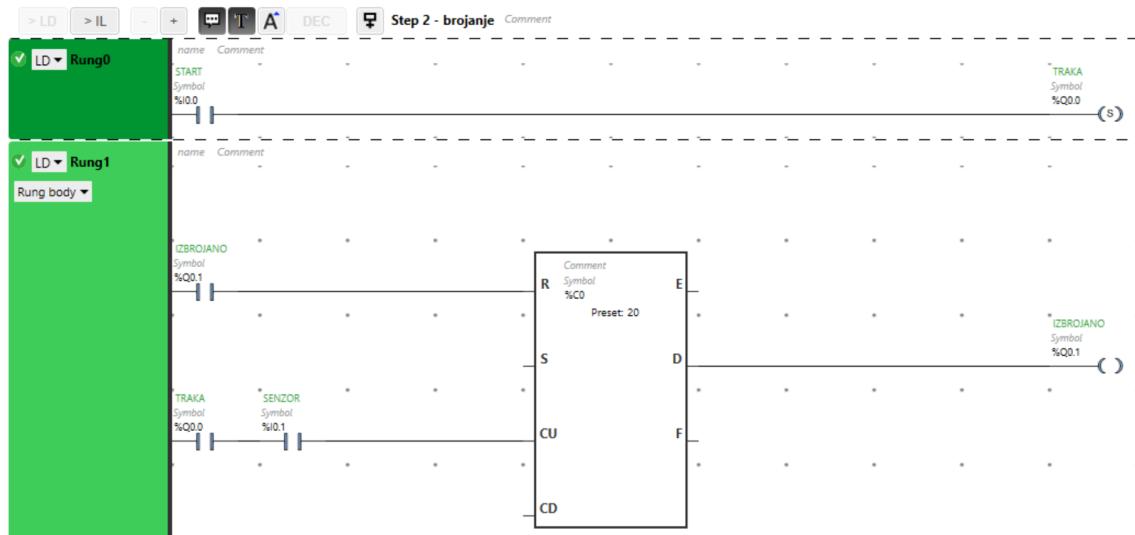
Ukoliko se u bilo kojem trenutku za vrijeme brojanja kutija aktivira prekidač HITAN STOP, izvršava se prelazak na korak Step 3 (hitani stop) gdje se pri ulasku zaustavlja traka. Nakon deaktiviranja prekidača, postaje aktivno stanje Step 2 te se nastavlja brojanje. U nastavku će biti prikazani kodovi za svaki step, te tranzicije.

Step 1 prikazuje početak sistema i prikazan je na slići 4.25.



Slika 4.25: Prikaz koda za početni step 1.

Step 2 prikazuje pokretanje trake i brojanje kutija, te reset brojača ukoliko se izbroji zadani broj kutija. Način na koji je navedena funkcionalnost realizirana dat je na slići 4.26.



Slika 4.26: Prikaz koda za step 2.

Step 3, prikazan na slići 4.27, prikazuje kod za hitan stop.



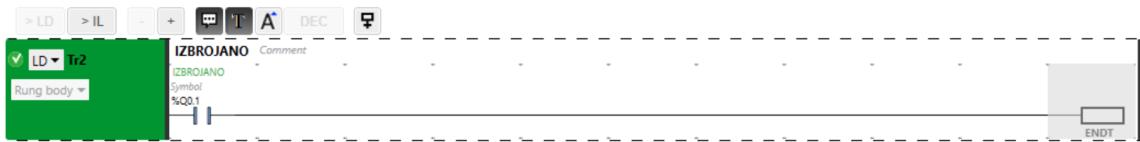
Slika 4.27: Prikaz koda za step 3.

Tranzicija iz Stepa 1 u Step 2 ostvarena je pritiskom tastera START i prikazana je na slići 4.28.



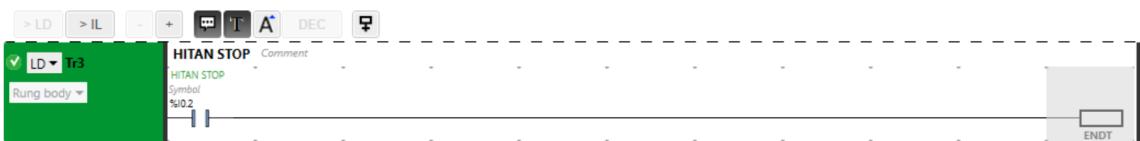
Slika 4.28: Prikaz koda za prvu tranziciju.

Tranzicija iz Step 2 u Step 1 ostvarena je nakon što je izbrojan zadani broj proizvoda. Navedena tranzicija data je na slici 4.29.



Slika 4.29: Prikaz koda za drugu tranziciju.

Tranzicija iz Step 2 u Step 3 ostvarena je pritiskom tastera HITAN STOP. Način na koji je navedena tranzicija realizirana prikazan je na slici 4.30.



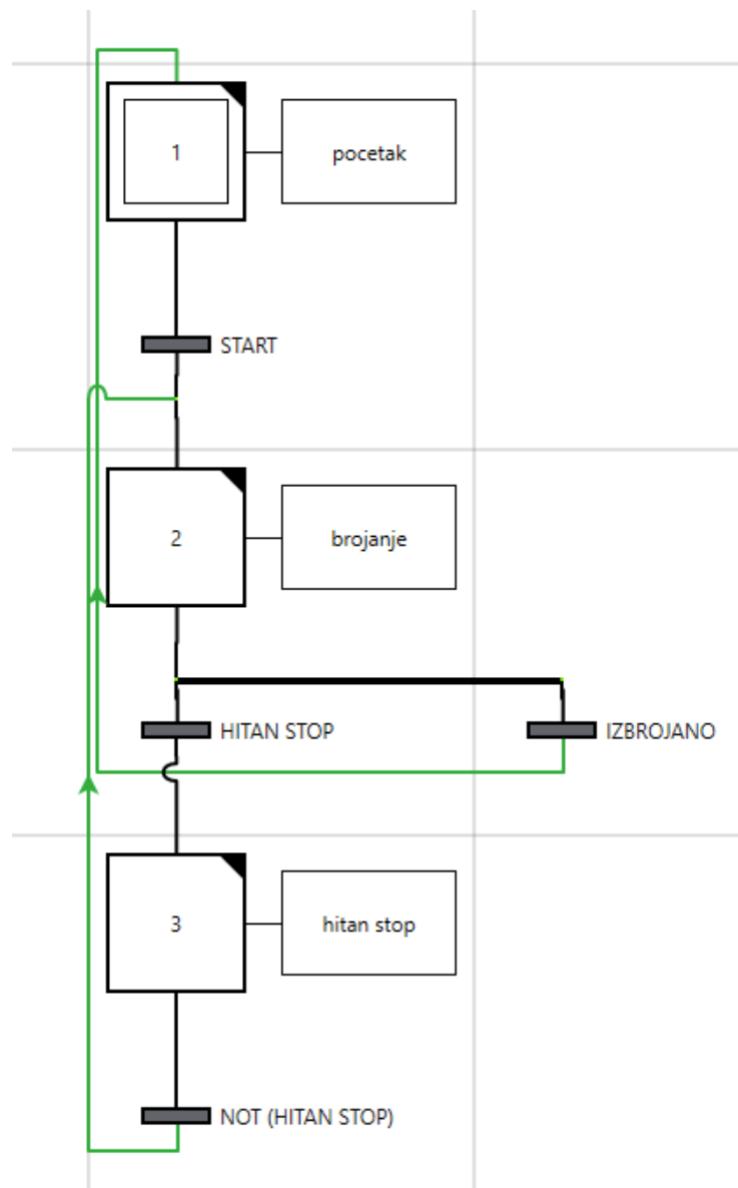
Slika 4.30: Prikaz koda za treću tranziciju.

Tranzicija iz Step 3 u Step 1 ostvarena je ukoliko nije došlo do pritiska tastera HITAN STOP i data je na slici 4.31.



Slika 4.31: Prikaz koda za četvrту tranziciju.

Realizacija transportne trake pomoću sekvencijalnih funkcijskih grafikona prikazana je na slici 4.32.

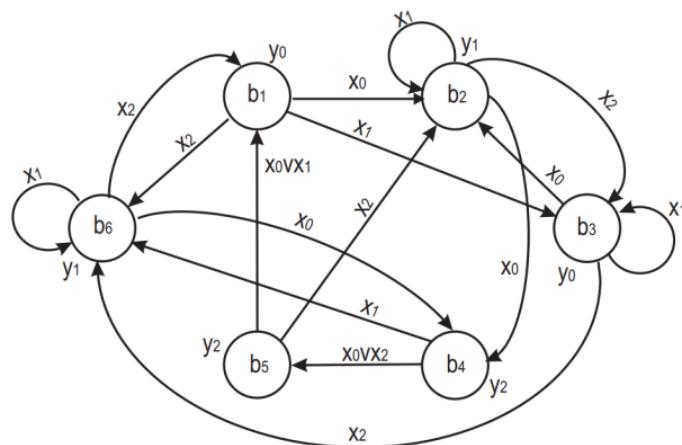


Slika 4.32: Prikaz realizacije transportne trake.

4.9.2 Primjer 2.

Mooreov automat

Funkcionalnost Mooreovog automata zadana je grafom na slici 4.33. Kreirati program koji pored osnovnih izlaza (y_0 , y_1 , y_2) ima još 6 izlaza (q_1 , q_2 , q_3 , q_4 , q_5 , q_6) koji će indicirati u kojem se stanju automat trenutno nalazi. Ukoliko je izlaz q_0 aktivan, automat se nalazi u stanju b_0 . Početno stanje je b_1 .



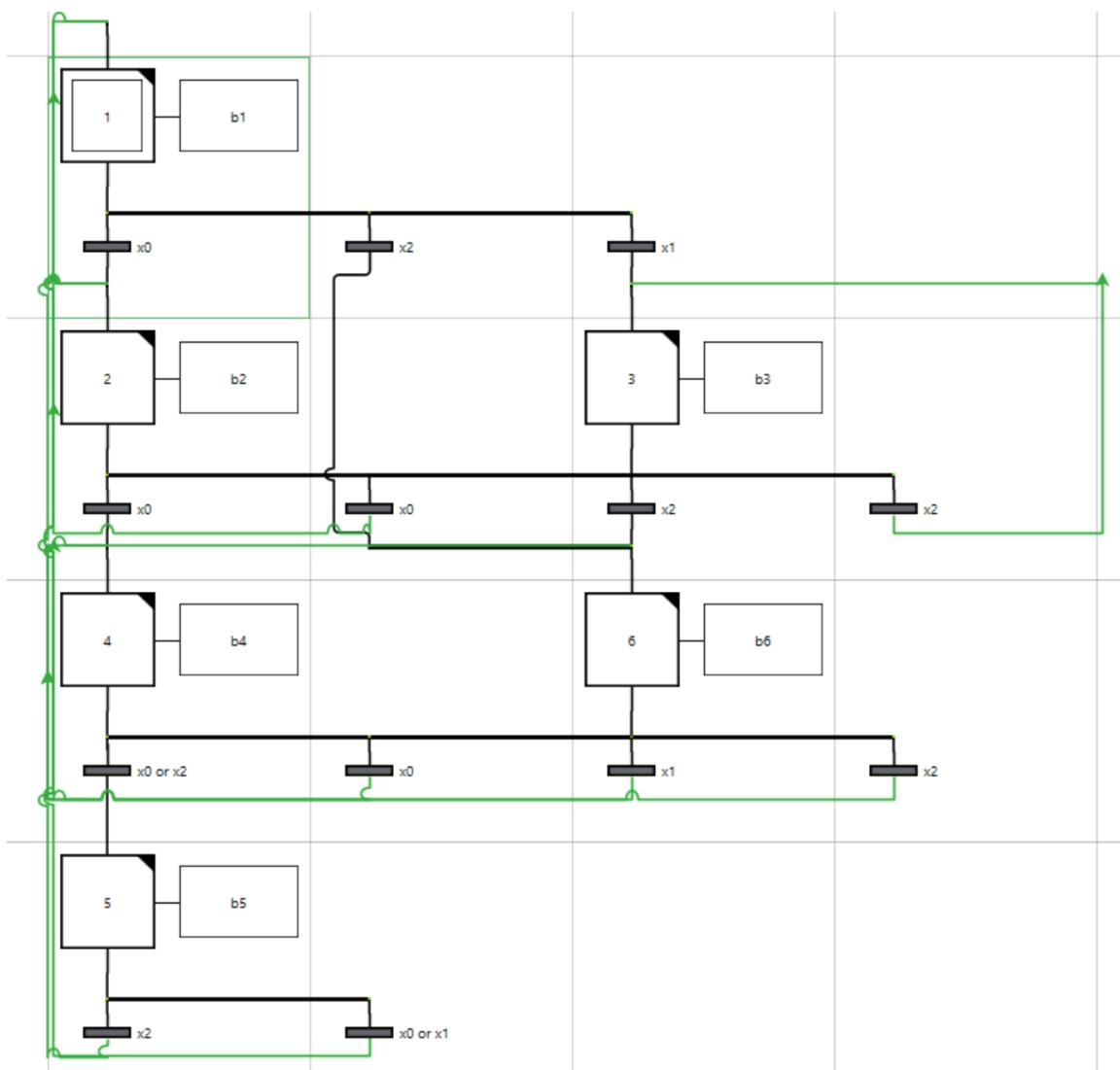
Slika 4.33: Graf automata.

Kod Mooreovog automata svako stanje ima pridružen odgovarajući izlaz (y_0 , y_1 , y_2). U ovom slučaju, prelazi su varijable x_0 , x_1 , x_2 . Kako automat posjeduje šest stanja, kreirano je isto toliko koraka ($b_1, b_2, b_3, b_4, b_5, b_6$). Za svaki korak je izvršeno je ažuriranje izlazna varijable te varijable vezana za stanje u kojem se program nalazi. Prilikom prelaska sa određenog koraka na drugi potrebno je ispuniti uslov koji je naveden na grafu.

Tabela 4.11: Lista signala za Mooreov automat.

Naziv	Tip signala
x_0	digitalni ulaz
x_1	digitalni ulaz
x_2	digitalni ulaz
y_0	digitalni izlaz
y_1	digitalni izlaz
y_2	digitalni izlaz
q_1	digitalni izlaz
q_2	digitalni izlaz
q_3	digitalni izlaz
q_4	digitalni izlaz
q_5	digitalni izlaz
q_6	digitalni izlaz

Na slici 4.34 prikazana je realizacija Mooreov automata pomoću SFC.



Slika 4.34: Realizacija Mooreovog automata pomoću SFC.

4.9.3 Primjer 3.

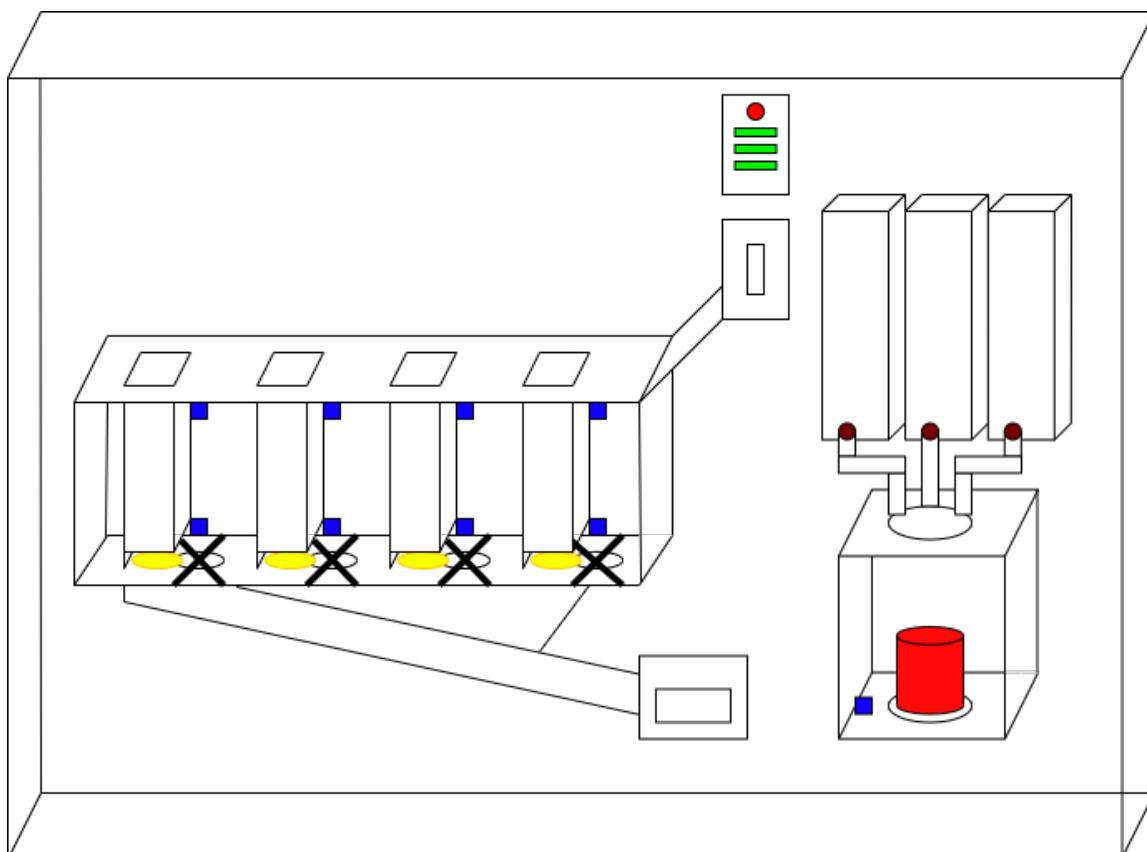
Kafe aparat

Potrebno je realizirati kafe aparat korištenjem SFC blokova. Korištenjem 3 tastera može se naručiti kafa (košta 0,8 KM), kapučino (košta 1,3 KM) i latte (košta 0,9 KM). Aparat prima kovanice od 0.1KM, 0.2KM, 0.5KM i 1KM. Proces spravljanja napitaka se može zanemariti, ali aparat treba da vraća kusur kovanici po kovanici. Ako se naruči napitak sa nedovoljnom količinom novca u aparatu, blokira se aparat na 5 sekundi nakon čega se ponovno može naručiti napitak.

Predloženo rješenje zadatka se može vidjeti na slici 4.35. Prilikom unosa novca, kovanica će upasti u jedan od zazora koji odgovara njenom poluprečniku (nisu svi zazori iste širine). Zazor je povezan sa posudicama u kojim će se skladištiti kovanice. Na vrhu posude je postavljen senzor koji će detektovati kovanicu (označen plavom bojom).

Ubacuje se čaša na specificirano mjesto, te se tasterima (označeni zelenom bojom) bira željeni napitak, koji se počne ispušтati iz rezervoara puštanjem ventila (označen smeđom bojom). Ako se nije unijelo dovoljno novca, crvena signalizacija iznad tastera će to označiti, a ako se nije detektovalo prisustvo čaše aparat će čekati dok se ne postavi.

Nakon kreiranja napitka vraća se kusur na sljedeći način: ako se detektuje potrebna kovanica na dnu posude pokreće se motor za čiji je rotor zakačen razdjelnik. Pomjeranjem rotora se izbacuje kovanica na lim koji je naget tako da svaki tip kovanice može ispasti u posudicu za preuzimanje.



Slika 4.35: Prikaz predloženog rješenja.

Prvo će se pružiti uvid u ulaze i izlaze iz sistema (slika 4.36). Unesene kovanice, mogući napici i očitanje senzora da li je napitak preuzet predstavljaju ulaze u sistem, dok su izlazi kusur, napici i signalizacija neuspjelog naručivanja. Što se tiče drugih korištenih varijabli, korišteni su sljedeći tipovi:

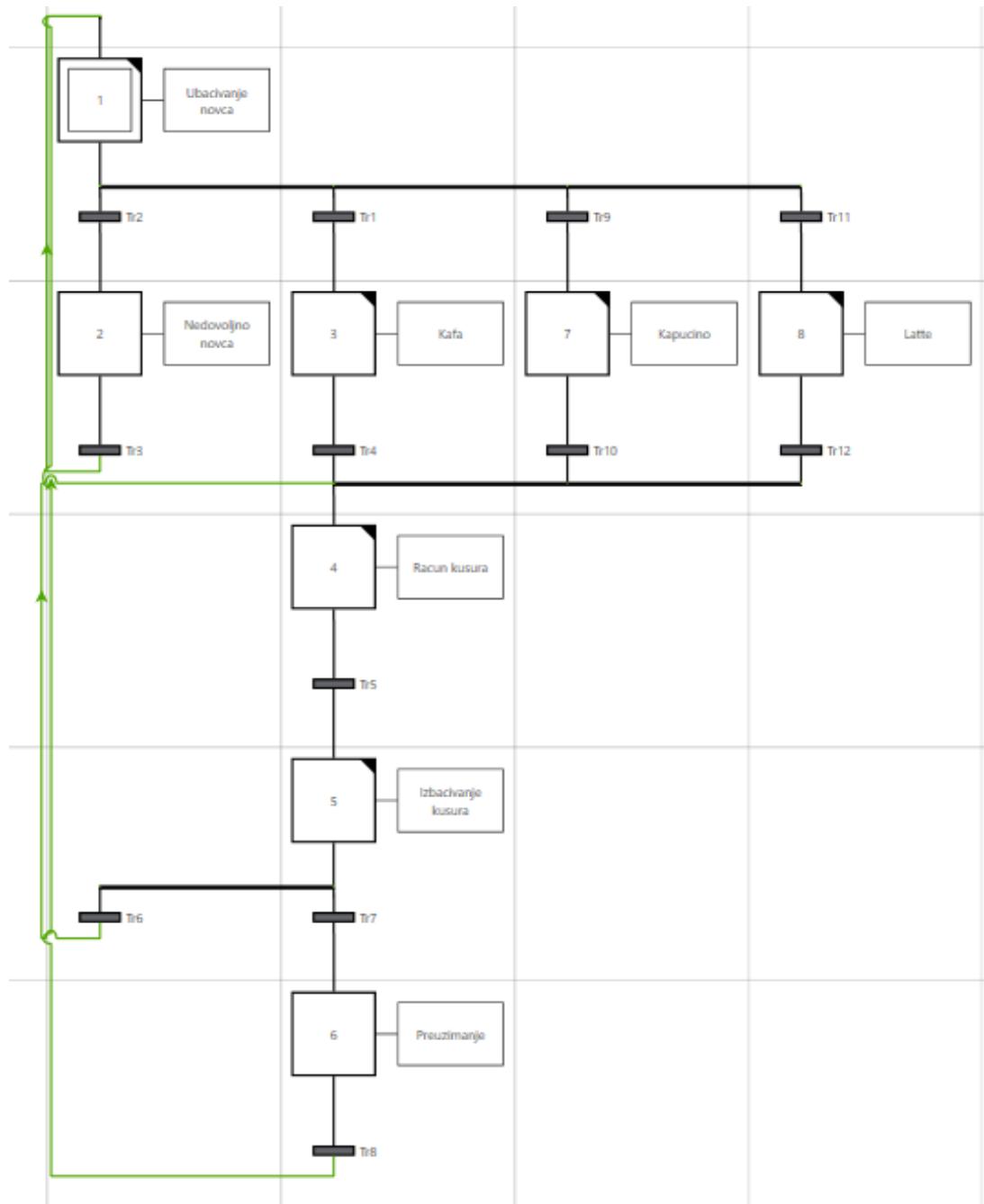
- logička varijabla $%M$ - korištena kao pomoćna varijabla,
- memoriske riječi $%MW$ - korištene za pamćenje trenutne vrijednosti novca u aparatu (što je fleksibilnije od brojača za dati zadatak),
- varijabla koja označava SFC step blok $%X$ ($%Xi$ označava i-ti step).

Digital input properties		
Used	Address	Comment
<input checked="" type="checkbox"/>	%I0.0	10F
<input checked="" type="checkbox"/>	%I0.1	20F
<input checked="" type="checkbox"/>	%I0.2	50F
<input checked="" type="checkbox"/>	%I0.3	1KM
<input checked="" type="checkbox"/>	%I0.4	PRTISUSTVO CASE
<input checked="" type="checkbox"/>	%I0.5	NARUCI KAFU
<input checked="" type="checkbox"/>	%I0.6	NARUCI KAPUCINO
<input checked="" type="checkbox"/>	%I0.7	NARUCI LATTE
<input type="checkbox"/>	%I0.8	
<input checked="" type="checkbox"/>	%I0.9	imaKM
<input checked="" type="checkbox"/>	%I0.10	ima50f
<input checked="" type="checkbox"/>	%I0.11	ima20f
<input checked="" type="checkbox"/>	%I0.12	ima10f

Digital output properties		
Used	Address	Comment
<input checked="" type="checkbox"/>	%Q0.0	GRESKA
<input type="checkbox"/>	%Q0.1	
<input checked="" type="checkbox"/>	%Q0.2	IZBACIVANJE 1KM
<input checked="" type="checkbox"/>	%Q0.3	IZBACIVANJE 50f
<input checked="" type="checkbox"/>	%Q0.4	IZBACIVANJE 20f
<input checked="" type="checkbox"/>	%Q0.5	IZBACIVANJE 10f
<input checked="" type="checkbox"/>	%Q0.6	KAFA
<input checked="" type="checkbox"/>	%Q0.7	KAPUCINO
<input checked="" type="checkbox"/>	%Q0.8	LATTE

Slika 4.36: Korišteni ulazi i izlazi.

Predloženo softversko rješenje zadatka se može vidjeti na slici 4.37.



Slika 4.37: Prikaz realizacije kafe aparata.

Prvi step (unos novca):

- svakim unosom kovanice memorijska riječ `%MW1` se povećava za odgovarajući iznos;
- pomoćne varijable za napitke se resetuju (što će trebati u sljedećem koraku).

Na slici 4.38 je prikazan po primjerak ovih redova.



Slika 4.38: Prvi step (unos novca).

Stepovi 3, 7 i 8 (pravljenje napitka):

- pomoćne varijable za napitke se setuju kako bi se u sljedećem redu `%MW1` mogao smanjiti na pozitivnu ivicu ovog signala (u suprotnom bi se konstantno izvršavao blok na kraju reda),
- `%MW1` se smanjuje za cijenu napitka.

Realizacija ovih stepova se može vidjeti na slici 4.39.

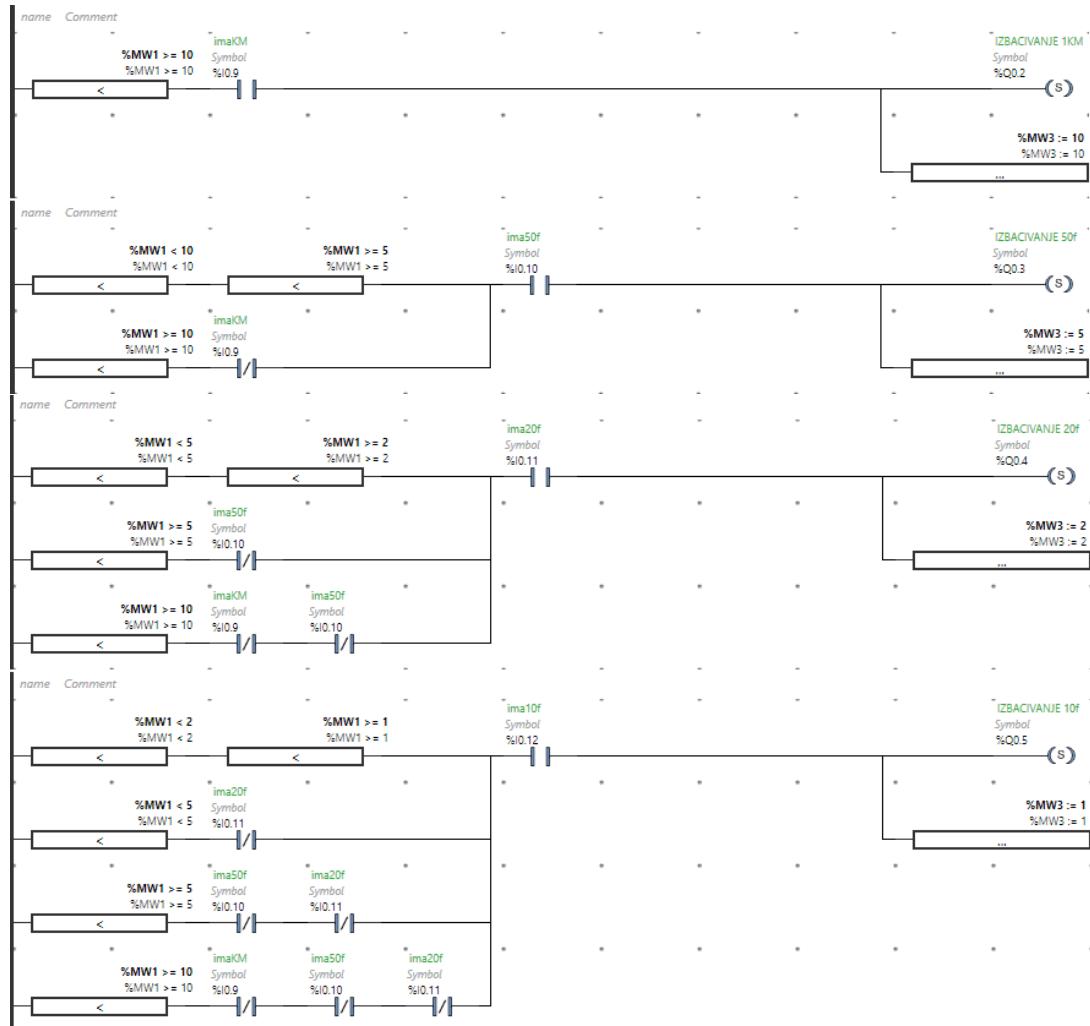


Slika 4.39: Stepovi 3, 7 i 8 (pravljenje napitka).

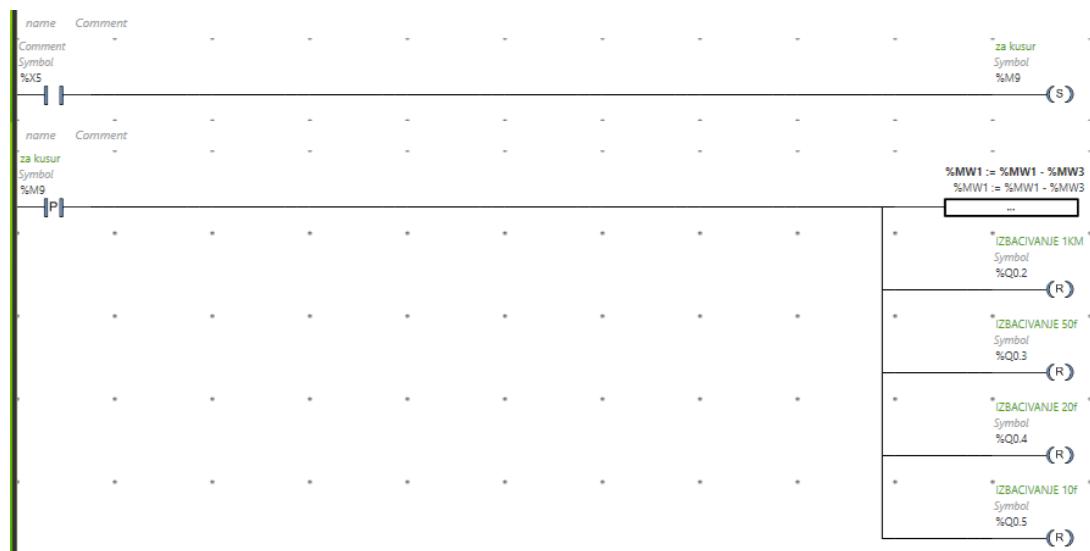
Step 4 i 5 (kusur):

- pronalazi se najveća kovanica koja se može izbaciti,
- aktiviraju se izlazi kako bi označili ispadanje kovanice,
- ponovno se izvršava procedura sa resetovanjem i setovanjem pomoćne varijable za račun,
- deaktiviraju se izlazi kako bi označili da je kovanica ispala.

U ovim stepovima se oduzima vrijednost kovanice koja se izbacuje iz aparata. Za praćenje prisustva kovanica u aparatu dodati su ulazi senzora kao dodatni uslov za proračun, i to tako da se za kovanice manje vrijednosti provjerava stanje svih kovanica više vrijednosti. Realizacija ovih stepova se može vidjeti na slikama 4.40 i 4.41.



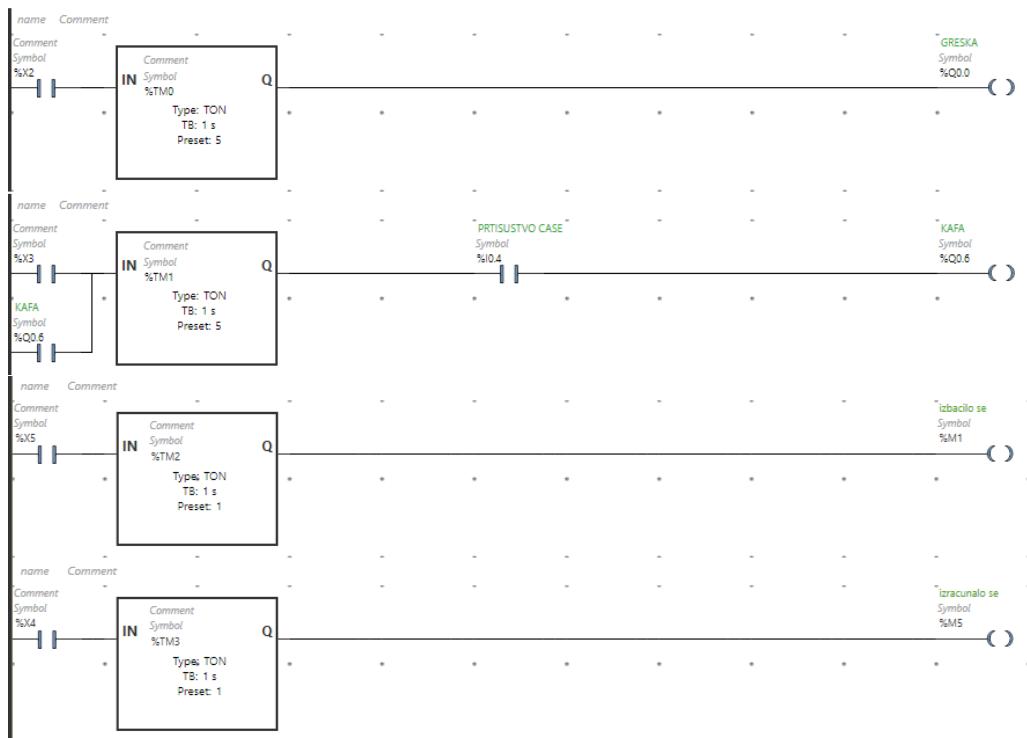
Slika 4.40: Step 4 (proračun kusura).



Slika 4.41: Step 4 i 5 (vraćanje kusura).

Dalje, napravljen je pomoći POU koji se izvršava paralelno sa prikazanim kodom. Kao što se može vidjeti na slici 4.42, tu su smješteni tajmeri za različita stanja:

- tajmer prilikom greške pri unosu,
- tajmeri tokom kreiranja napitka, i
- tajmeri za izbacivanja kusura (sistem treba ispuštati jednu po jednu kovanicu, za šta je potrebno određeno vrijeme).



Slika 4.42: Pomoći POU.

Sada se može preći na tranzicije.

- Početno stanje - kreiranje napitka: provjerava se da li ima dovoljno novca, da li je postavljena čaša i da li je taj napitak zapravo naručen (slika 4.43).
- Početno stanje - stanje greške: disjunkcija svih "neuspjelih" uslova iz prethodne tačke (slika 4.43).
- Kreiranje napitka - račun kusura: signal s tajmera koji označava da je tajmer kreiran.
- Račun kusura - vraćanje kusura: signal s tajmera koji omogućava fizičko vrijeme da kusur ispadne. Sistem će se vrtiti između ova dva bloka dok se ne vrati sav kusur.
- Vraćanje kusura - preuzeto: kada je iznos novca u opticaju jednak 0, ili ako nema kovanica (slika 4.43).
- Preuzeto - početno stanje: provjerava da li je čaša idalje prisutna na platformi.
- Stanje greške - početno stanje: signal s tajmera.



Slika 4.43: Prikaz nekih tranzicija.

* * *

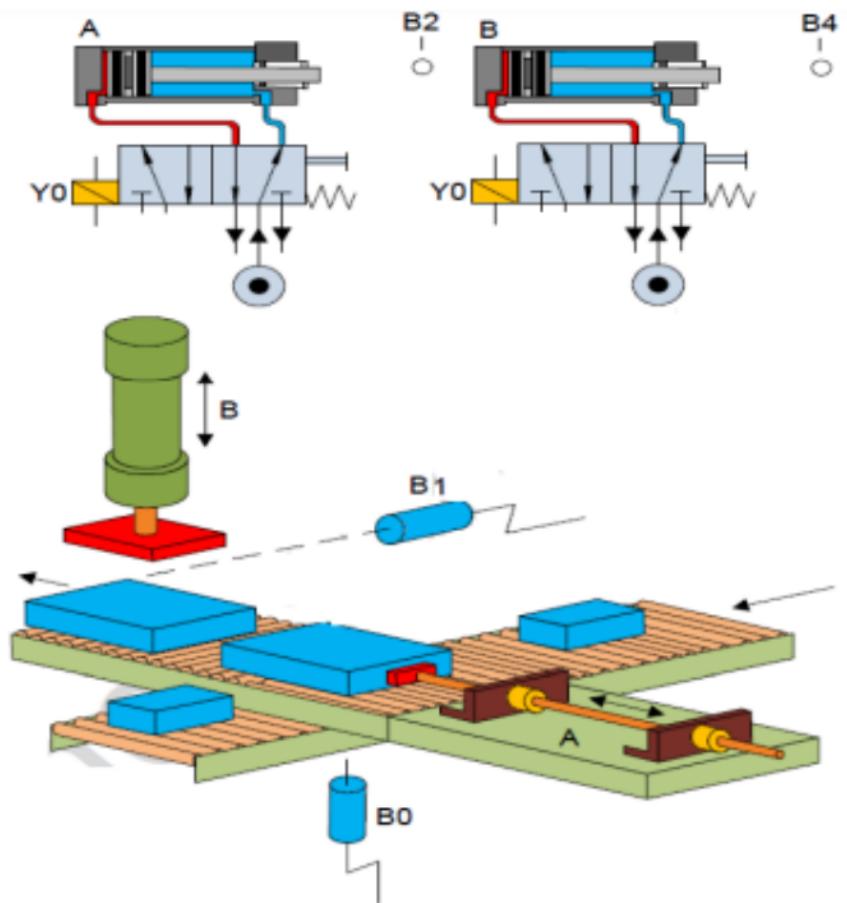
U ovom poglavlju prikazani su različiti načini programiranja programabilnog logičkog kontrolera TM200CE24T: Ladder dijagrami (LD), instrukcijske liste (IL), funkcijski blok dijagrami (FBD), Grafket (List) i Grafket (SFC). Svaki od ovih pristupa pruža određene mogućnosti i omogućava programiranje aplikacije na PLC-u prema specifičnim zahtjevima industrijskih procesa. Razumijevanje korištenja ovih programske jezike ključno je za uspješnu realizaciju kompleksnih automatizacijskih rješenja, što povećava efikasnost i pouzdanost proizvodnih sistema. U nastavku je realiziran konkretan projekt linije za sortiranje, koji će demonstrirati praktičnu primjenu programabilnog logičkog kontrolera TM200CE24T u procesima upravljanja.

Poglavlje 5

Primjer realizacije linije za sortiranje

5.1 Tehnički opis

Na slici 5.1 prikazana je linija za sortiranje proizvoda.



Slika 5.1: Linija za sortiranje.

Potrebno je upravljati linijom za sortiranje dijelova na mjestu dopremanja. Svi dijelovi se nalaze na pokretnoj traci. Nakon detekcije odgovarajućeg dijela, aktivira se pneumatski klip sa kašnjenjem od 100 [ms] koji ga premješta na drugu pokretnu traku. Kada drugi optički senzor detektuje dio ispod prese, ponovo se izvlači pneumatski klip i presa tada otisne pečat.

5.2 Prijedlog rješenja

Pokretne trake postavljene su pod uglom od 90° . Trake se pokreću sa dva odvojena trofazna asinhrona motora. Prva traka doprema sve dijelove, dok se drugom trakom premještaju samo dijelovi koji su zadovoljili kriterij odabira.

Nakon što optički senzor B0 prepozna dio, izvlači se klip cilindra A sa kašnjenjem od 100 [ms] zbog njegovog premještanja i gura ga do prese za pečatiranje. Ovo kašnjenje je neophodno da bi se dio postavio na ispravno mjesto na pokretnoj traci. Nakon što optički senzor B1 prepozna dio ispod prese, izvlači se klip cilindra B i na njega utisne pečat. Oba cilindra se uvlače čim dostignu svoje krajne položaje što se detektuje kontaktima B2 i B4. Oba cilindra A i B kontrolišu monostabilni razvodnici.

U rješenju zadatka realiziran je taster za hitan stop, kao i stanje u kojem je klip zaglavljen. Tačnije, ukoliko klip za neko određeno vrijeme ne dodje do svoje krajne pozicije, dolazi do greške i zaustavljanja procesa. Hitani stop zaustavlja motore i klipove, te onemogućava ponovno pokretanje dok se taster hitnog stopa ne vrati na vrijednost FALSE.

5.3 Lista signala

U tabeli 5.1 prikazani su korišteni signalni potrebni za realizaciju funkcionalnosti linije za sortiranje.

Tabela 5.1: Lista signala linije za sortiranje.

Adresa	Tip	Opis
%I0.0	Digitalni ulaz	Taster START
%I0.1	Digitalni ulaz	Senzor B0
%I0.3	Digitalni ulaz	Senzor B1
%I0.4	Digitalni ulaz	Senzor B2
%I0.5	Digitalni ulaz	Senzor B4
%I0.6	Digitalni ulaz	Taster HITAN STOP
%Q0.0	Digitalni izlaz	MOTOR
%Q0.1	Digitalni izlaz	A KLIP
%Q0.3	Digitalni izlaz	PRESA i klip B

5.4 Izbor komponenti i opis opreme

Oprema koja je potrebna za liniju za sortiranje proizvoda je:

- Razvodni ormar IL177212US 4-redni proizvođača Schrack, limeni - služi za organizaciju svih neophodnih komponenti na jednom mjestu.
- Odvodnik prenapona COMBTEC kl.B/C TNC 275/12,5, proizvođača Schrack je sigurnosna komponenta koja štiti elemente od prenapona i podnapona. Dakle, to je komponenta koja reguliše trofazni napon.
- Redna stezaljka VS, URTK/S, $6mm^2$ korištena za međusobno povezivanje komponenti.

- FID sklopka proizvođača Schrack osigurava sigurnost radnika koji se nalaze u blizini machine. Registruje curenje struje i sprječava situacije koje bi mogле ugroziti ljudske živote.
- Kabelski kanal/kanalica proizvođača Schrack - služi za vizuelno ljepši izgled ormara, sakrivanje kablova i raspored istih.
- Hitan stop proizvođača GQELE omogućava automatski prekid rada u iznimnim situacijama. Taster je crvene boje, nakon što se pritisne, da bi se omogućio dalji rad potrebno je isti vratiti u prvobitno stanje.
- Taster Start XB7NA3133 proizvođača Schneider Electric služi za pokretanje sistema.
- PLC TM200CE24T, proizvođača Schneider Electric - programabilni logički kontroler koji čita ulaze, te u skladu sa programom i logikom mijenja stanja na izlazima.
- AC Motor 0,75kW 1500rpm-Trofazni proizvođača GAMAK - potreban za pokretne trake kojima se vrši dostava proizvoda.
- Mrežni ispravljač, 3f, taktni, 400[V] AC/24[V] DC, 22[A] pri 50[°C], proizvođača Schrack - pretvara trofazni naizmjjenični napon u istosmjerni koji je potreban za ostale komponente.
- Kapacitivni senzori IFM Efector KI0203 služe za detekciju pozicije pneumatskih klipova.
- Pneumatski razvodnik 4A 330C-08, proizvođača XCPC služi za dovod zraka pneumatskim klipovima koji se izvlače i odvode zrak da se klipovi mogu vratiti u prvobitni položaj.
- Pneumatski klip SMC CDJ2B10-50-B, proizvođača SMC - potrebnii za prebacivanje izabranog proivoda s jedne trake na drugu, te za utiskivanje pečata.
- Optički senzor detekcije udaljenosti DM7/0P-1H služi za detekciju objekata koje je potrebno prebaciti na narednu traku i ustanoviti da li se predmeti nalaze na željenom mjestu kako bi se na njih mogao utisnuti pečat.
- Relej RSL1AB4BD proizvođača Schneider Electric - služe kao zaštita opreme u slučaju da dođe do kvara ostalih komponenti, te za pokretanje aktuatora kojima je potrebna veća struja za rad.

5.5 Predračun opreme

Predračun korištene opreme dat je u tabeli 5.2. Ukupna cijena opreme potrebne za liniju za sortiranje je 4919.76 KM.

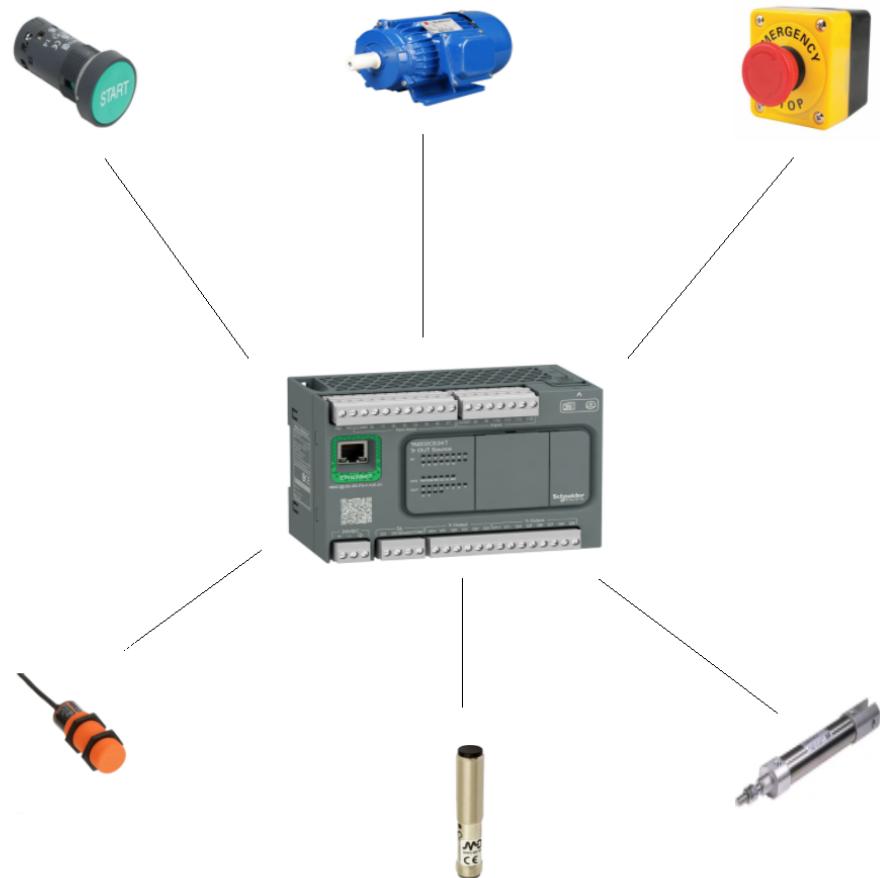
Tabela 5.2: Predračun opreme.

Br.	Naziv komponente	Vrsta	Količina	Iznos (KM)
1	IL177212US	Razvodni ormar	1	1026.60
2	IS211230-A	Odvodnik prenapona	1	372.20
3	VS, URTK/S, 6mm ² ,	Redna stezaljka	59	186.50
4	BCP36603	FID sklopka	1	560.80
6	RH732849	Kanalica	2 [m]	32.80
7	LA115-B2-11	Taster Hitan stop	1	16.60
8	XB7NA3133	Taster Start	1	12.70
9	TM200CE24T	PLC	1	686.56
10	AGM3E 80H B34-00-B	AC motor	2	536.00
11	LP432422	Mrežni ispravljač	1	590.00
12	KI0203	Kapacitivni senzor	2	441.90
13	4A 330C-08	Pneumatski razvodnik	2	87.40
14	CDJ2B10-50-B	Pneumatski klip	2	103.16
15	DM7/0P-1H	Optički senzor	2	320.00
16	RSL1AB4BD	Relej	7	16.54
Ukupno (KM)				4919.76

5.6 Grafički dio

5.6.1 Principijelna shema

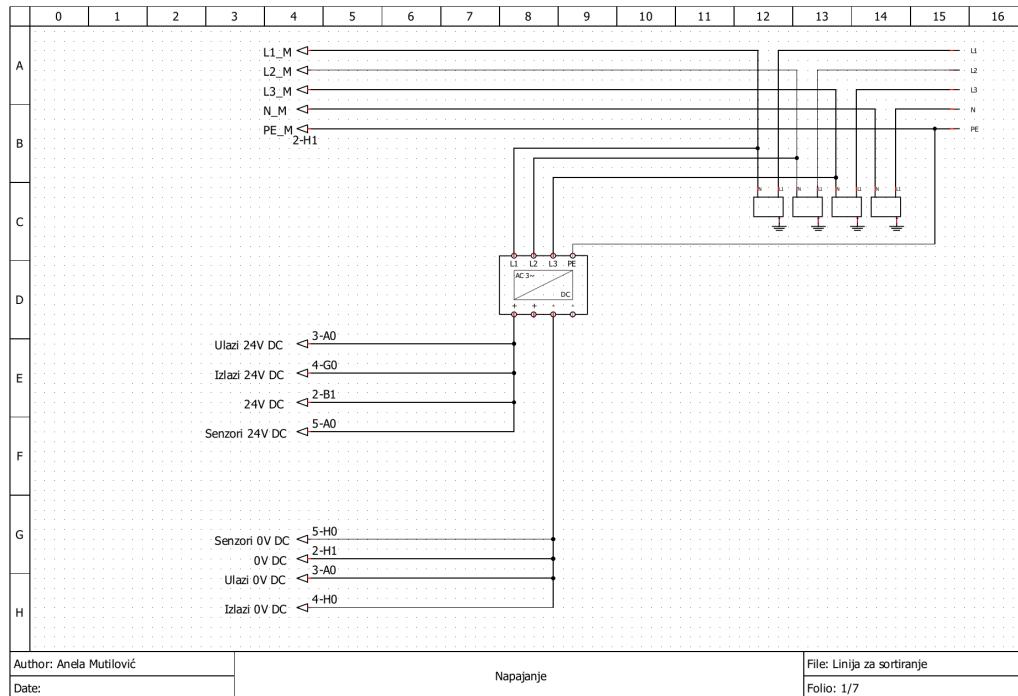
Principijelna shema sistema je prikazana na slici 5.2.



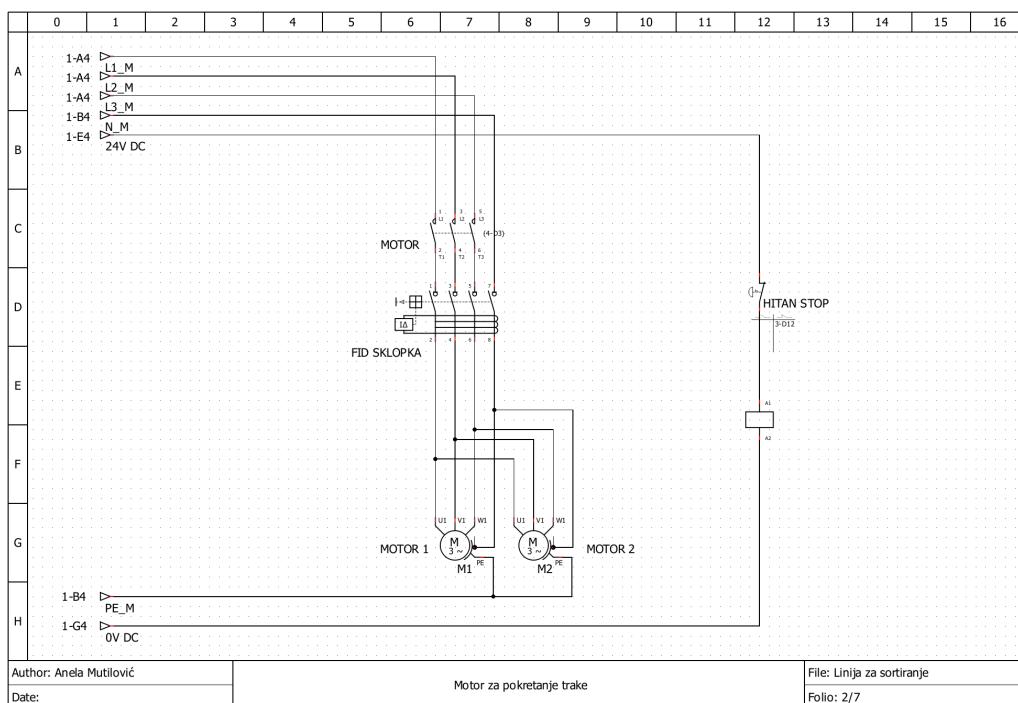
Slika 5.2: Principijelna shema povezivanja.

5.6.2 Sheme djelovanja

Sheme djelovanja linije za sortiranje prikazane su na slikama u nastavku.

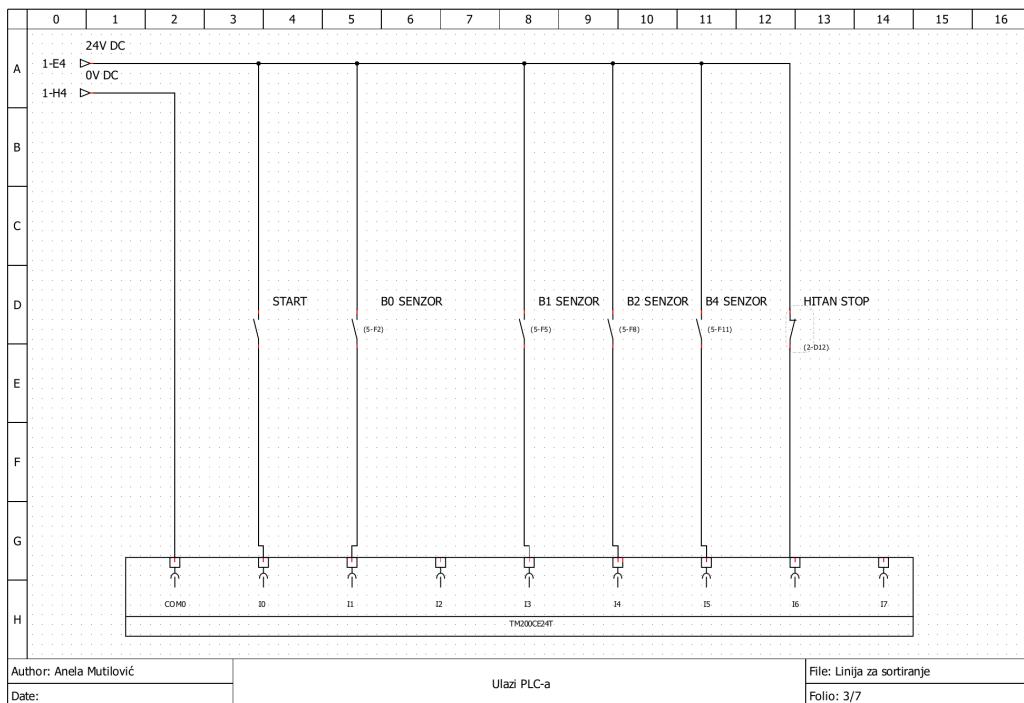


Slika 5.3: Shema povezivanja napajanja.

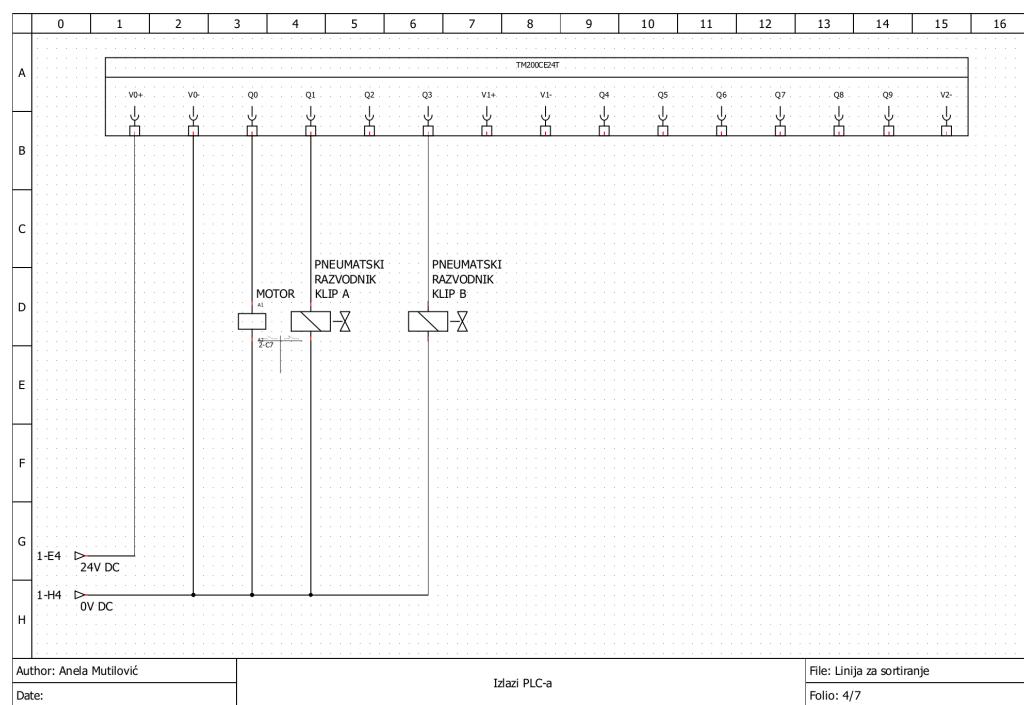


Slika 5.4: Shema povezivanja motora.

Primjer realizacije linije za sortiranje

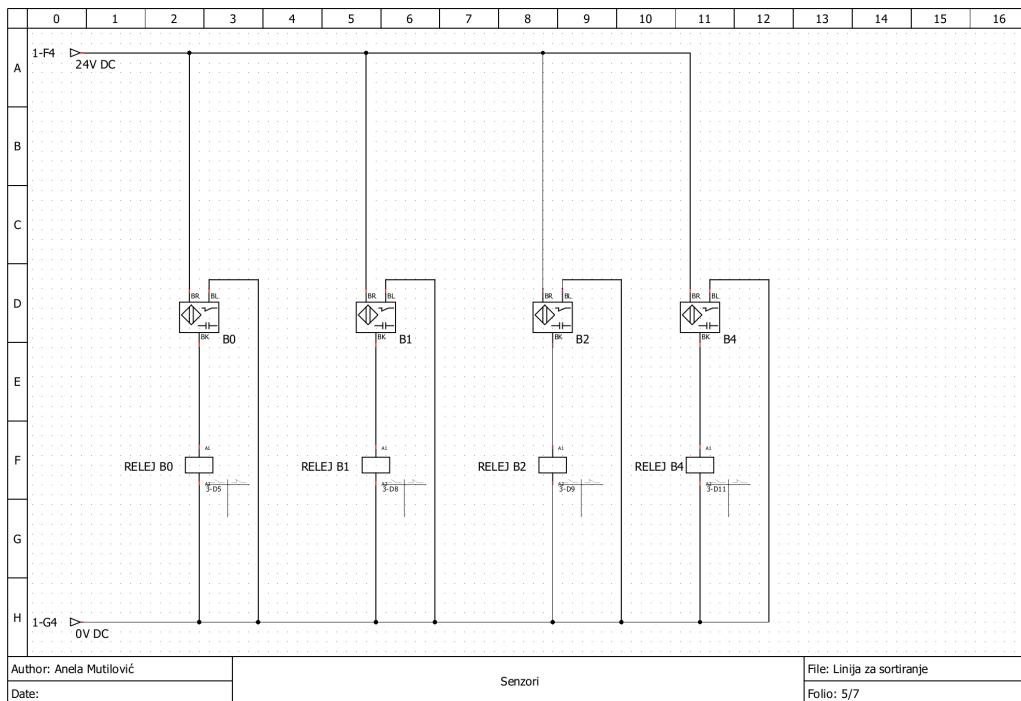


Slika 5.5: Shema povezivanja ulaza PLC-a.

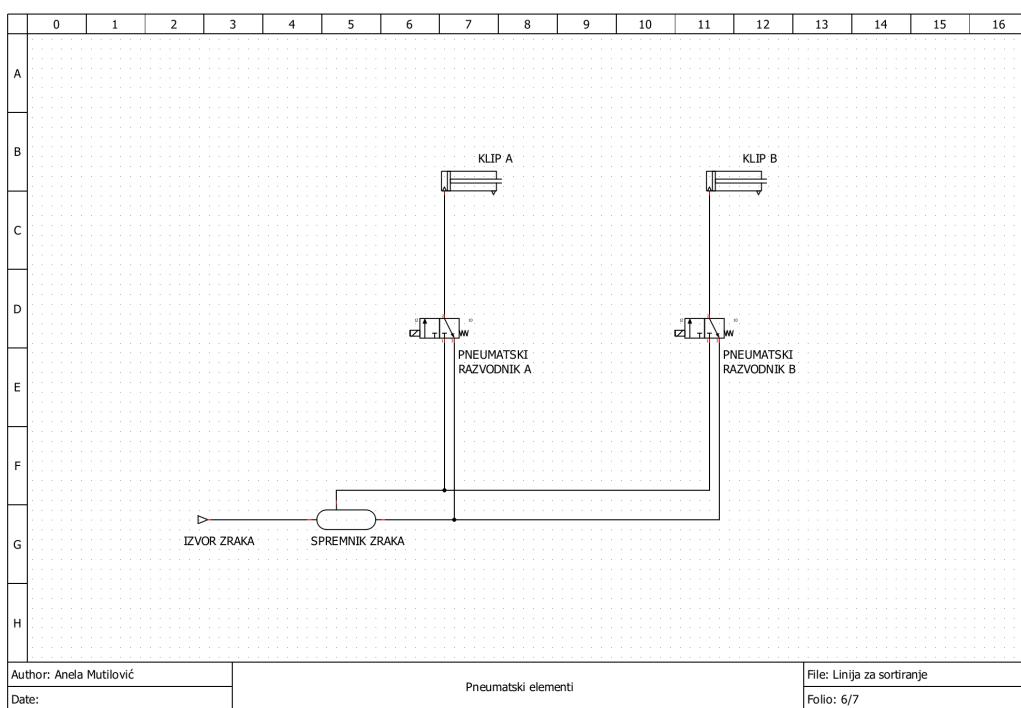


Slika 5.6: Shema povezivanja izlaza PLC-a.

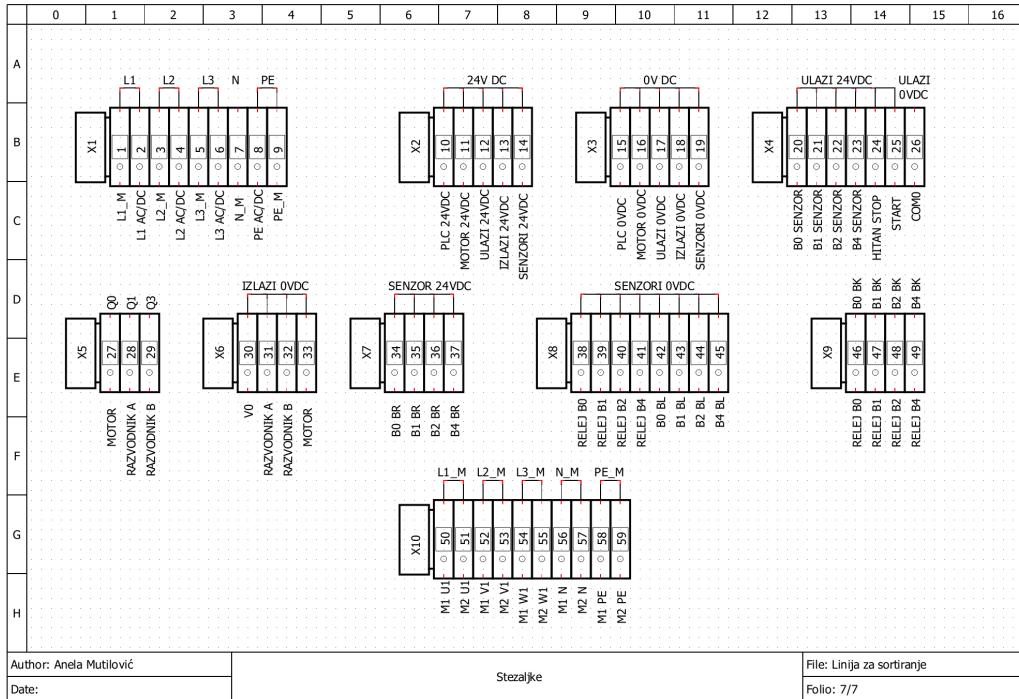
Primjer realizacije linije za sortiranje



Slika 5.7: Shema povezivanja senzora.



Slika 5.8: Shema povezivanja pneumatskih elemenata.



Slika 5.9: Shema povezivanja stezaljki.

5.7 Priključni plan

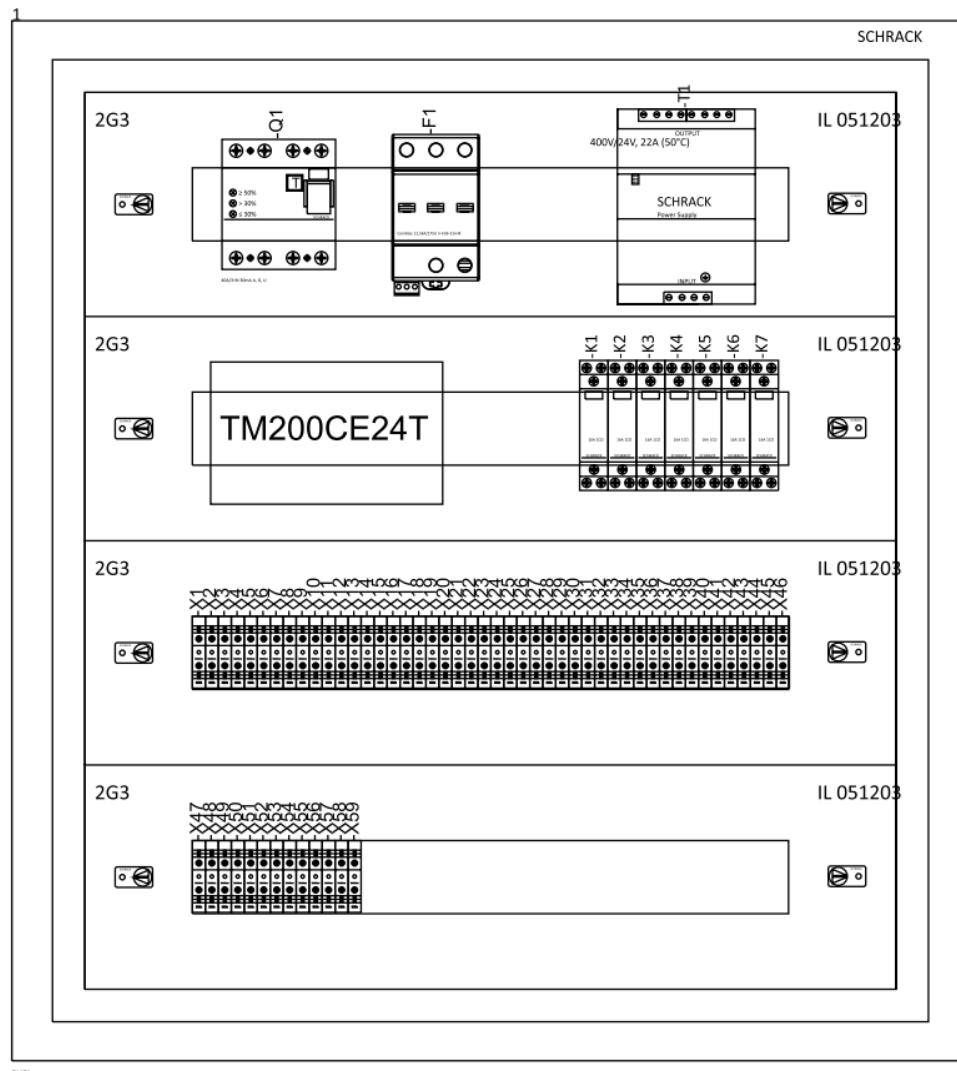
Redne stezaljke prikazane na shemi 5.9 olakšavaju otkljanjanje kvarova i lakše snalaženje u razvodnom ormaru. U nastavku će biti objašnjenje skupine stezaljki, te komponenti koje su na njih povezane. Na navedenoj shemi lako je uočiti da postoji deset skupina stezaljki označene od X1 do X10.

- **Skupina X1** predstavlja prvu skupinu stezaljki na koju su dovedene tri faze (označene redom sa: L1, L2 i L3), neutralni vod i uzemljenje. Potrebne su za trofazne motore i mrežni ispravljач.
- **Skupina X2** je skupina gdje se stezaljke povezuju na 24 [V] DC napona koji je preuzet sa mrežnog ispravljaja. Taj napon potreban je za napajanje ostalih elemenata.
- **Skupina X3** je skupina gdje se stezaljke povezuju na 0 [V] DC napona koji je preuzet sa mrežnog ispravljaja. Taj napon potreban je za rad ostalih elemenata.
- **Skupina X4** predstavlja stezaljke na koje su dovedeni signali koji će se proslijediti na ulaze PLC-a.
- **Skupina X5** predstavlja poveznicu digitalnih izlaza PLC-a sa reljima i ventilima koji su potrebni za pokretanje klipova i motora.
- **Skupina X6** je skupina stezaljki pomoću kojih se komponente koje se nalaze na izlazima kontrolera povezuju na 0 [V] DC.
- **Skupina X7** je sačinjena od stezaljki koje služe za povezivanje senzora na 24[V] DC, to je napon koji je potreban za rad senzora.
- **Skupina X8** slično kao i prethodna služi za dovođenje napona od 0[V] DC do senzora.

- **Skupina X9** predstavlja stezaljke koje su potrebne za povezivanje senzora sa relejima.
- **Skupina X10** je posljednja skupina stezaljki koje spajaju dva trofazna motora u paralelnu vezu.

5.8 Nacrt ormara

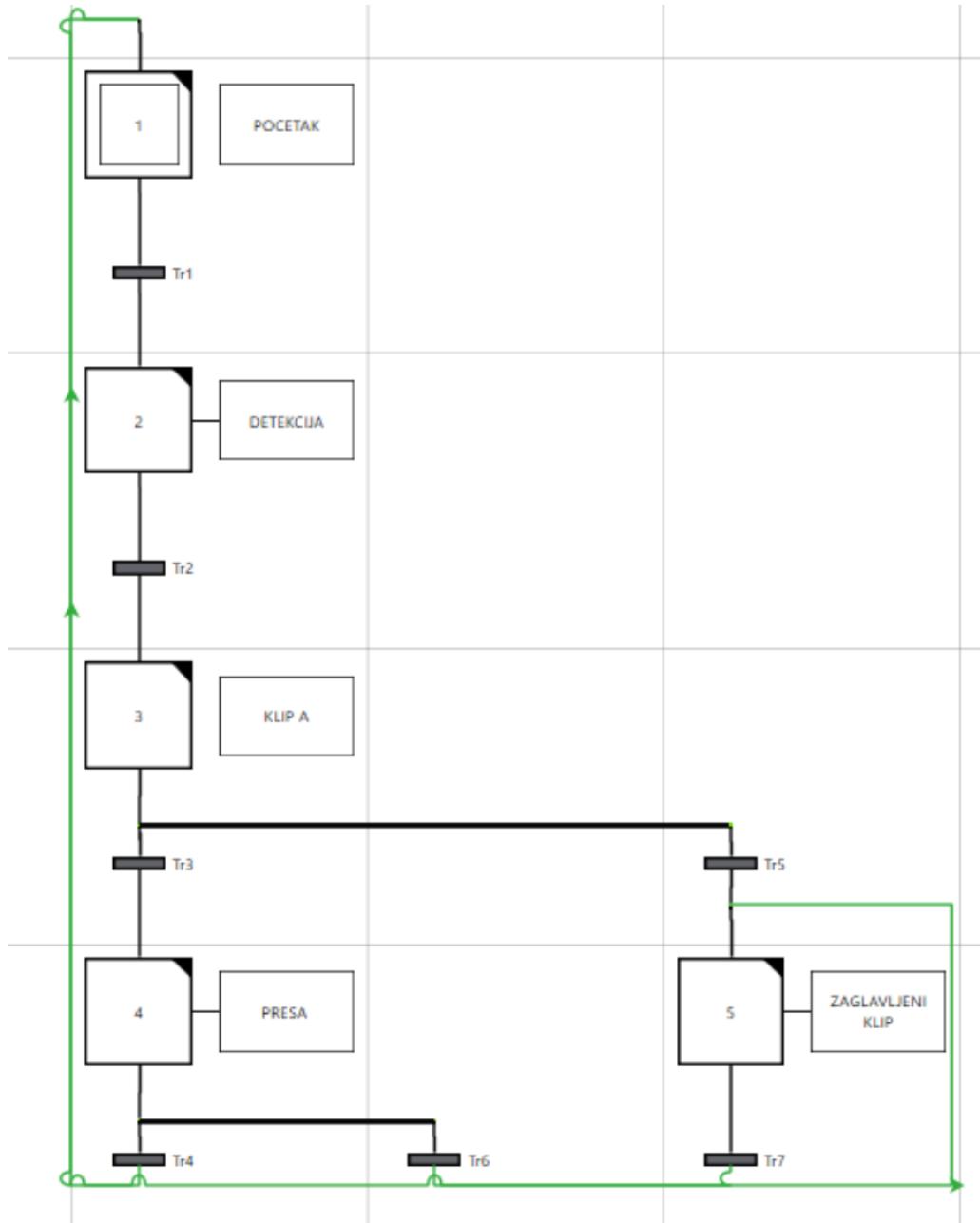
Izgled razvodnog ormara prikazan je na slici 5.10.



Slika 5.10: Prikaz razvodnog ormara.

5.9 Softversko rješenje

Na slici 5.11 prikazano je softversko rješenje linije za sortiranje urađeno u programu EcoStuxure Machine Expert - Basic pomoću SFC načina programiranja.



Slika 5.11: Realizacija linije za sortiranje pomoću SFC.

Funkcionalnost programa će biti pojašnjena kroz pet stanja i tranzicija između njih.

Stanje POCETAK je početno stanje u kojem se čeka na pritisak tastera START i pokretanje motora pokretnih traka. Nakon pokretanja motora, dolazi se do prve tranzicije Tr1 u kojoj se čeka detekcija proizvoda senzorom B0.

Stanje DETEKCIJA je stanje u kojem se nakon aktivacije senzora B0 pokreće tajmer koji predstavlja kašnjenje pokretanja klipa A od 100 [ms]. U Tranziciji Tr2 se nakon isteka tajmera prelazi u stanje KLIP A.

Primjer realizacije linije za sortiranje

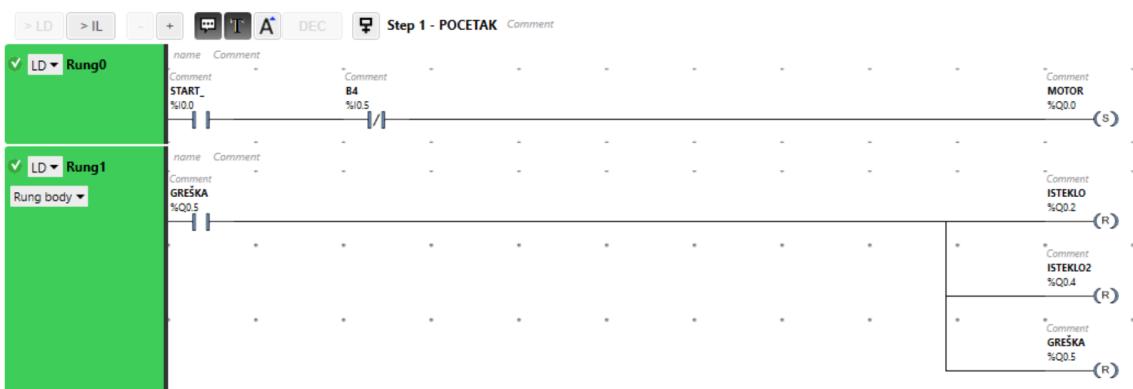
Stanje KLIP A pokreće klip sve do detekcije krajne pozicije B2, ili dok ne istekne 10 [s] što uzrokuje grešku i zaustavljanje procesa. Tranzicije Tr3 i Tr5 su vezane za stanje KLIP A. Tranzicija Tr3 prelazi u stanje aktivacije klipa B i prese, što znači da se dostigla krajnja pozicija klipa A. Ukoliko se pri isteku tajmera ne dostigne krajnja pozicija, tranzicijom Tr5 prelazi se u stanje ZAGLAVLJENI KLIP koje predstavlja grešku.

Stanje PRESA pokreće klip B i presu sve do detekcije krajne pozicije B4, ili dok ne istekne 10 [s] što uzrokuje grešku i zaustavljanje procesa. Tranzicije Tr4 i Tr6 su vezane za stanje PRESA. Tranzicija Tr4 prelazi u početno stanje gdje se čeka ponovna detekcija proizvoda senzorom B0, što znači da se dostigla krajnja pozicija klipa B. Ukoliko se pri isteku tajmera ne dostigne krajnja pozicija, tranzicijom Tr6 prelazi se u stanje ZAGLAVLJENI KLIP koje predstavlja grešku.

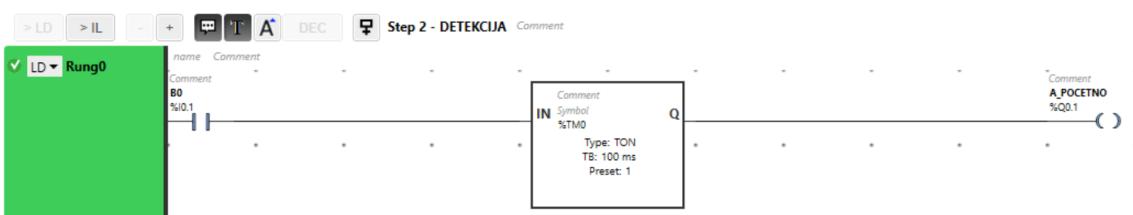
Stanje ZAGLAVLJENI KLIP je stanje u kojem se zaustavljaju motori i klipovi jer je prouzrokovana greška dostizanja krajnje pozicije klipa A ili B. Tranzicija Tr7 je izlaz ovog stanja koja je povezana sa početnim stanjem u kojem se čega ponovna aktivacija motora pokretnih traka.

Za realizaciju hitnog stopa kreiran je novi POU koji se izvršava konstantno kao i GRAFCET POU sa stanjima i tranzicijama. Ukoliko se aktivira HITAN STOP tasterom obustavlja se rad svih motora i klipova, sve dok se isti ne deaktivira. Nakon deaktivacije moguće je pokrenuti liniju za sortiranje pritiskom na START.

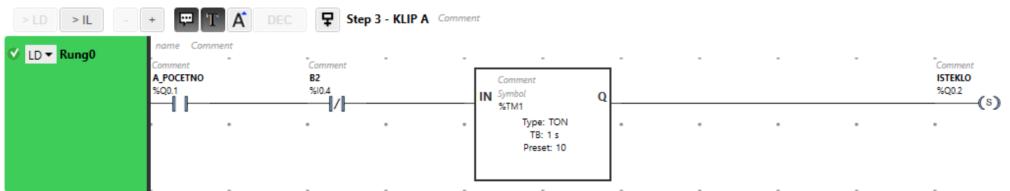
U nastavku će biti prikazana sva ranije navedena stanja, na slikama 5.12, 5.13, 5.14, 5.15 i 5.16.



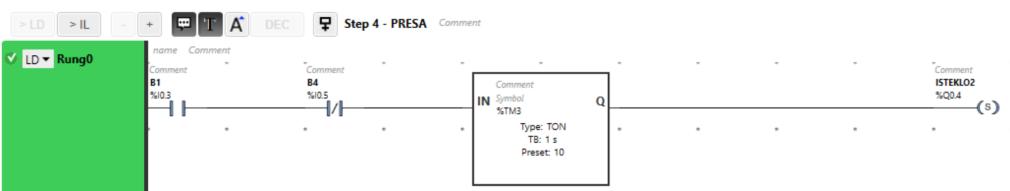
Slika 5.12: Prikaz početnog stanja.



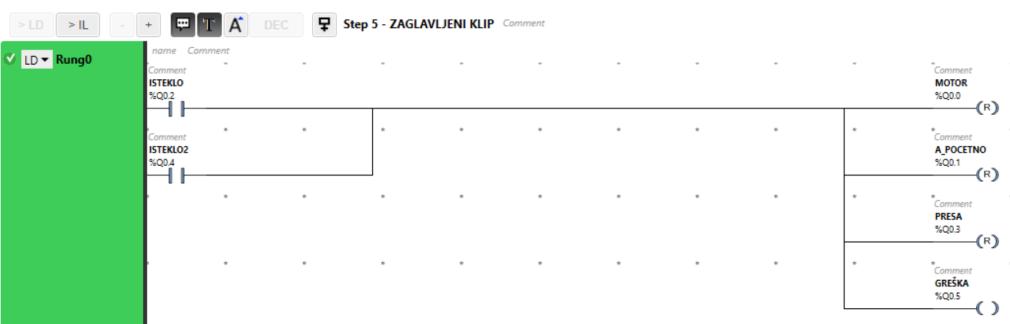
Slika 5.13: Prikaz stanja detekcije.



Slika 5.14: Prikaz stanja klipa A.



Slika 5.15: Prikaz stanja aktiviranja prese i klipa B.



Slika 5.16: Prikaz stanja zaglavljenog klipa.

Primjer realizacije linije za sortiranje

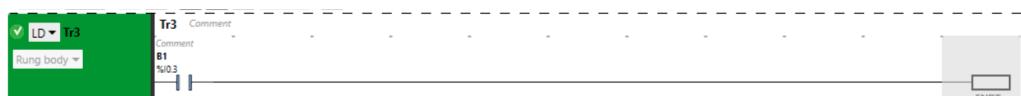
Tranzicije između navedenih stanja prikazane su na slikama 5.17, 5.18, 5.19, 5.20, 5.21, 5.22 i 5.23, te posebni POU koji predstavlja hitan stop 5.24.



Slika 5.17: Prikaz prve tranzicije.



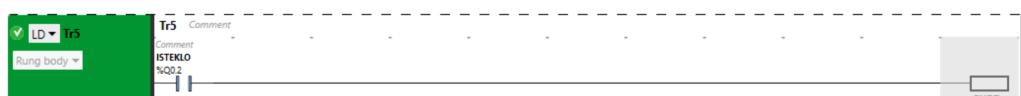
Slika 5.18: Prikaz druge tranzicije.



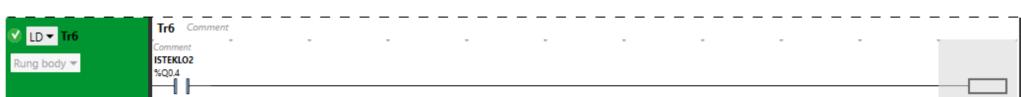
Slika 5.19: Prikaz treće tranzicije.



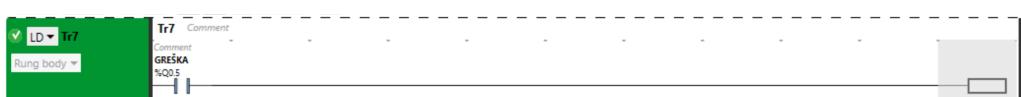
Slika 5.20: Prikaz četvrte tranzicije.



Slika 5.21: Prikaz pете tranzicije.



Slika 5.22: Prikaz šeste tranzicije.



Slika 5.23: Prikaz sedme tranzicije.

Primjer realizacije linije za sortiranje



Slika 5.24: POU - Hitan stop.

* * *

U ovom poglavlju detaljno je prikazana realizacija linije za sortiranje, koristeći programabilni logički kontroler TM200CE24T i softver EcoStruxure Machine Expert-Basic. Tehnički opis linije obuhvatio je listu signala, izbor komponenti te prikaz opreme potrebne za efikasno funkcionisanje sistema. Analizirani su troškovi opreme i principijelna shema rada, te shema djelovanja i priključni plan. Također, prikazan je i nacrt ormara za instalaciju opreme. Softversko rješenje bazirano je na Grafset (SFC) načinu programiranja, pružajući pouzdanu kontrolu nad procesima sortiranja na liniji.

Realizacija linije za sortiranje naglašava integrисани pristup automatizaciji u industrijskom okruženju, demonstrirajući kako PLC može biti ključan za optimizaciju proizvodnih procesa i poboljšanja efikasnosti cijelog sistema.

Poglavlje 6

Zaključak

Kroz završni rad pojašnjene su osnovne karakteristike programabilnih logičkih kontrolera, njihove prednosti i mane, kao i mogućnosti primjene. Detaljno su prikazani podržani načini programiranja uz praktične primjere upotrebe kontrolera TM200CE24T kako bi se adekvatno nagnasio značaj upotrebe programabilnih logičkih kontrolera u unapređenju automatizacije procesa i sistema upravljanja. Iako svestran, rad na ovom PLC-u unutar opisanog programskog okruženja može imati određene izazove. Određene mogućnosti su ograničene, poput praćenja trenutnog stanja brojača ili vizuelnog interfejsa za interakciju sa programom (HMI). Isto tako i sami PLC nema velik broj ulaza i izlaza, što također ograničava njegov opseg upotrebe. Uprkos svemu rečenom, primjena ovog PLC-a je obimna, i uz pažljivu analizu problema može se smatrati kao idealan izbor za automatizaciju procesa. Upravljanje navedenim PLC pomoću softvera EcoStruxure Machine Expert - Basic predstavlja najpogodnije rješenje za automatske sisteme koji u svom radu zahtijevaju jednostavno kretanje i veliku fleksibilnost. Softver je jako intuitivan i jednostavan za korištenje. Automatizacija procesa pomoću softvera se vrši kroz svega nekoliko koraka: i) konfiguracije hardvera, ii) programiranje i iii) puštanje u rad. Kao primjer realizacije sistema, kreirana je projektna dokumentacija za sistem upravljanja linijom za sortiranje.

Ostvareni ciljevi završnog rada - Programabilni logički kontroler TM200CE24T posjeduje veoma dobar omjer performansi sa cijenom, što je velika prednost pri izboru korištenog kontrolera. Prilikom rada sa softverom EcoStruxure Machine Expert - Basic nisu uočene greške niti problemi prilikom pokretanja simulacije ili programiranja rješenja. Velika prednost softvera je jednostavnost korištenja. Moguće je u vrlo kratkom vremenskom periodu realizirati projekt bez obzira na izbor načina programiranja. Detaljnog analizom i razumijevanjem ovog rada moguće je realizirati upravljanje kako jednostavnim, tako i zahtjevnim procesima, što je i bio primarni cilj ovog rada.

Literatura

- [1] HDIN, Programmable Logic Controller (PLC) Producers and PLC Market Trend, Research, 20.10., 2022, dostupno na: <https://www.hdinresearch.com/news/276>
- [2] Eaton, “Ec4p compact programmable logic controller”, dostupno na: <https://www.eaton.com/gb/en-gb/catalog/industrial-control--drives--automation---sensors/ec4p-compact-plc.html> -.
- [3] Eaton, “Xc100/200 modular programmable logic controllers”, dostupno na: <https://www.eaton.com/tw/en-us/catalog/industrial-control--drives--automation---sensors/xc100-200-modular-plcs.html> -.
- [4] PHÁT, M., “Board plc mitsubishi fx1n-40mr”, dostupno na: <https://minhphat65.com/board-plc-mitsubishi-fx1n-40mr-747.html> -.
- [5] Electric, S., “Plc tm200ce24t”, dostupno na: <https://www.se.com/eg/en/product/TM200CE24T/controller-m200-24-io-transistor-source+-ethernet/> datasheet.
- [6] Goran Malčić, Programirljivi logički kontroleri, Tehničko veleučilište u Zagrebu, Elektrotehnički odjel, dostupno na: https://nastava.tvz.hr/gmalcic/PLC_skripta_TVZ.pdf
- [7] Community, A., “Advantages and disadvantages of plc”, dostupno na: <https://automationcommunity.com/advantages-and-disadvantages-of-plc/> 12/2023.
- [8] Solutions, W. A., “Advantages of plc systems in industrial automation”, dostupno na: https://in.linkedin.com/in/wiztech-automation-solutions-64b873265?trk=article-ssr-frontend-pulse_publisher-author-card&original_referer=https%3A%2F%2Fwww.linkedin.com%2F 10/2023.
- [9] Kamel, K., Kamel, E., (ur.), Programmable logic controllers. New York: USA:McGraw-Hill Education, 2013.
- [10] John G., Programmable Logic Controllers, The PLC/PAC Tutorial, November 11, 2009, dostupno na: <https://theplctutor.com/history.html>
- [11] Kreso, S., Huseinbegovic, S., (ur.), Programabilno logički kontroleri. Projektovanje mikroprocesorskih sistema, pp. 157–220: Sarajevo: Elektrotehnički fakultet u Sarajevu, 2012.
- [12] N.Osmic, “Elektrotehnički fakultet u Sarajevu”, in Industrijska automatizacija, i laboratorijske vježbe na predmetu, P., (ur.), 2024.
- [13] Electric, S., “Catalog”, dostupno na: <https://media.distributordatasolutions.com/schneider2/2020q3/documents/0c8dc5684ff9f3c41cecdd3dab0dc388d10bafef.pdf> 2020.