Programiranje industrijskog računala

Šaponja, Tara

SVEUČILIŠTE U RIJECI

Sveučilišni preddiplomski studij politehnike

Završni rad

PROGRAMIRANJE INDUSTRIJSKOG RAČUNALA

Rijeka, rujan 2020. Tara Šaponja

SVEUČILIŠTE U RIJECI

Sveučilišni preddiplomski studij politehnike

Završni rad

PROGRAMIRANJE INDUSTRIJSKOG RAČUNALA

Mentor: prof. dr. sc. Saša Sladić

Rijeka, rujan 2020.

Tara Šaponja

SVEUČILIŠTE U RIJECI
STUDIJ POLITEHNIKE
Rijeka, Sveučilišna avenija 4
Povjerenstvo za završne i diplomske radove

U Rijeci, 24. lipnja 2020. godine

ZADATAK ZAVRŠNOG RADA

Pristupnik: **Tara Šaponja**

Zadatak: **Programiranje industrijskih računala**

Rješenjem zadatka potrebno je obuhvatiti sljedeće:

Analizirati funkcijski, povijesno i tehnološki značenje ljestvičastih dijagrama u programiranju programabilnih logičkih kontrolera (PLC) odnosno industrijskih računala. Analizirati značenje K-tablica u pristupu izradi ljestvičastih dijagrama. Provjeriti ispravnost pristupa primjenom komercijalno dostupnih PLC simulatora.

U završnom se radu obavezno treba pridržavati **Uputa za izradu završnog rada.**

Zadatak uručen pristupniku: 29. lipnja 2020.

Rok predaje diplomskog rada: 16. srpnja 2020.

Datum predaje diplomskog rada: _____

Predsjednik povjerenstva za završne i diplomske radove:

Doc. dr. sc. Damir Purković

Zadatak zadao:

Izv. prof. dr. sc. Saša Sladić

IZJAVA O IZVORNOSTI

Izjavljujem da je moj završni rad izvorni rezultat mojega rada te da se u njegovoj izradi nisam koristila drugim izvorima od onih navedenih u radu.

Rad je temeljen na znanjima stečenim na studiju politehnike uz vodstvo i mentorstvo prof. dr. sc. Saše Sladića.

SAŽETAK

Rad pokriva osnovne informacije o programiranju PLC uređaja pomoću ljestvičastih dijagrama. Također spominje se i nekoliko tehnika izrade logičkih sklopova i konstruiranja samog programa. Od Booleove algebre i K- tablica koriste se osnovna načela i pravila koja bi mogla koristiti programeru Ljestvičastog dijagrama u dobivanju završnog programa za upravljanje. Spominju se osnovne funkcije i mogućnosti ljestvičastih dijagrama kao i primjera računalnog programa koje pruža integrirano razvojno okruženje u kojem se kod može napisati i preuzeti na stvarni PLC stroj. Kao alat koristi se i PLC simulator.

Ključne riječi: INDUSTRIJSKA RAČUNALA, PROGRAMABILNI LOGIČKI KONTROLERI, PLC, AUTOMATIZACIJA, UPRAVLJANJE PROCESOM, K-TABLICE,DIGITALNI SUSTAVI, LJESTVIČASTI DIJAGRAM, PLC SIMULATOR

SUMMARY

This paper covers basic information about the programming of PLC devices using ladder diagrams. Several techniques for making logical circuits and constructing the program itself are also mentioned. Boolean algebra and K-maps basic principles and rules are used as a tool to aid a ladder logic programmer to achieve the final control program. The basic functions and possibilities of ladder diagrams are mentioned as well as examples of a computer program that provides an integrated development environment in which the code can be written and downloaded to the actual PLC. A PLC simulator is used as a tool.

Keywords: INDUSTRIAL COMPUTERS, PROGRAMMABLE LOGIC CONTROLLERS, PLC, AUTOMATION, PROCESS CONTROL, K-MAPS, DIGITAL SYSTEMS, LADDER DIAGRAM, PLC SIMULATOR

SADRŽAJ

SAŽETAK I

SUMMARY II

SADRŽAJ III

POPIS SLIKA V

POPIS TABLICA VII

1.UVOD 1

2.PROGRAMABILNI LOGIČKI KONTROLER 3

2.1 Arhitektura programabilnog logičkog kontrolera 5

2.2 Način djelovanja 6

3. PROGRAMIRANJE PLC UREĐAJA 7

3.1 Relejna logika 9

4. LJESTVIČASTI DIJAGRAMI 11

4.1 Ulazi i izlazi ljestvičastog dijagrama 12

4.2 Osnovne funkcije simulatora 13

4.2.1 Sklop za pokretanje i zaustavljanje Start-Stop 15

4.2.1 Svjetiljka s tri različita prekidača 18

5.BOOLEOVA ALGEBRA 19

5.1 Primjer industrijske peći 22

6. KARNAUGHOVE ILI K TABLICE 24

6.1 Protuprovalni alarm 24

7. VREMENSKI REGISTRI, BROJAČI I ZAKLJUČAVANJE 29

7.1 Zaključavanje statusa 30

7.2 Vremenski registri 30

7.2.1 Transportna traka 31

7.3 Brojači 35

8. FUNKCIJE LJESTVIČASTOG DIJAGRAMA 37

8.1 Funkcije upravljanja podacima 37
8.2 Logičke funkcije 38
9. PROCES PROGRAMIRANJA SUSTAVA UPRAVLJANJA 40
9.1 Program dizala s dva kata 40
10. ZAKLJUČAK 45
LITERATURA 46

POPIS SLIKA

Slika 2.1 Allen Bradley ControlLogix 5580 [12] 4
Slika 2.2 Dijelovi PLC uređaja 5
Slika 2.3 Ciklus rada PLC uređaja 6
Slika 3.1 Sličnost između relejnih (a) i ljestvičastih dijagrama (b) [17] 9
Slika 3.2 Elektromagnetski relej [15] 10
Slika 4.1 Izgled ljestvičastog dijagrama 11
Slika 4.2 Tipovi sklopki Normally open NO i Normally closed NC 12
Slika 4.3 Simboli izlaza ljestvičastog dijagrama [8] 13
Slika 4.4 Računalni program 13
Slika 4.5 PLC Simulator 14
Slika 4.6 Primjer analognog ulaza 15
Slika 4.7 Start-Stop dijagram 16
Slika 4.8 Prikaz Start –Stop dijagrama u PLC simulatoru 17
Slika 4.9 Upravljanje svjetiljkom 18
Slika 4.10 Programski kod primjera sa slike 4.9. 18
Slika 5.1 Tablice istinitosti operatora Booleove algebre 19
Slika 5.2 Logika START-STOP sklopa 21
Slika 5.3 Kompleksni logički operatori 21
Slika 5.4 Dijagram industrijske peći [8] 23
Slika 5.5 Dijagram industrijske peći na računalnom programu 23
Slika 6.1 K-tablica 26
Slika 6.2 Uzorci K-tablice 27
Slika 6.3 Dijagram alarma [8] 27
Slika 6.4 Dijagram alarmnog sustava u računalnom programu 28
Slika 7.1 Odziv uređaja [8] 29
Slika 7.2 SET i RESET izlazi ljestvičastog dijagrama 30
Slika 7.3 Primjer vremenskog registra[8] 31
Slika 7.4 Dijagram pokretne trake [8] 32
Slika 7.5 Pokretanje sustava 33
Slika 7.6 Ljestvičasti program transportne trake 34
Slika 7.8 Brojač uzlazno/silazni 36
Slika 8.1 MOVE funkcija 38
Slika 8.2 Primjer funkcija za zbrajanje 38

Slika 8.3 Primjer funkcije za usporedbu brojeva 39
Slika 8.4 Primjer funkcije logičkog operatora I 39
Slika 9.1 Program dizala u računalnom programu 44

POPIS TABLICA

Tablica 1 Tablica istinitosti protuprovalnog alarma 25
Tablica 2 Ulazi i izlazi programa dizala.....40
Tablica 3 Tablica istinitosti za ulaze i izlaze dizala.....41

1.UVOD

U današnje vrijeme proizvodni procesi bili bi nezamislivi bez automatizacije. Razvoj upravljanja proizvodnim procesima neprestano napreduje. Središtem proizvodnog procesa smatra se industrijsko računalo [1]. Ono je odgovorno za upravljanje svim ulaznim i izlaznim signalima, prema zadanom programskom kôdu. Kada se govori o industrijskim računalima, uglavnom se misli na programabilne logičke kontrolere (*eng. Programmable logic controllers)*. Njihov razvoj započinje 60-ih godina kako bi zamijenili relejne sklopove [2].

Načelno, PLC je industrijsko računalo premda se u praksi taj termin koristi za uređaj koji ima instaliran operacijski sustav, a svojom performansom i obimom zadataka koje rješava nadmašuje klasični PLC. Prema tome, termin industrijsko računalo se može koristiti za obje skupine uređaja.

PLC uređaji koriste se za automatiziranje raznih procesa kao što su primjerice različiti industrijski pogoni: strojarnica za pripremu tople vode, nadzor elektromotornih pogona, tvornice za valjanje čelika [3,4,5.]. PLC uređaji najčešće koriste jezik ljestvičastih dijagrama [6]. Taj pristup se pokazao djelotvornim u programiranju PLC uređaja. Kroz godine korištenja navedene izvedbe ljestvičastog programiranja se mijenjaju i inženjeri osmišljavaju nove, sve pristupačnije programske jezike, koji se najčešće temelje upravo na ljestvičastim dijagramima, odnosno, na pristupu bliskom relejnoj logici [7].Također, adekvatno su početno gradivo za upoznavanjem s procesom osmišljavanja i programiranja sustava upravljanja. Kao pomoćno sredstvo za bolji uvid u proces sastavljanja ljestvičastog dijagrama koristi se Do-more PLC simulator tvrtke AutomationDirect verzije 2.7.3. Softver se može besplatno preuzeti s interneta i jedan je od simulacijskih i komercijalnih programa na internetu.

Osmišljavanje softvera za sustave upravljanja može biti izazovno. Iskusni inženjeri razvili su mnoštvo različitih pristupa koji im omogućavaju rješavanje problema. Cijeli proces od koncepta za sustav upravljanja do ispravnog programskog kôda koji se pokreće na PLC uređaju može biti kompleksan i često se sastoji od određenog broja različitih koraka [8,9]. Savladavanje tih vještina u simulacijskom razvojnom okruženju prikladan je pristup . Takvim pristupom u potpunosti se mogu izbjeći štete uzrokovane ljudskim pogreškama koji si proizvođači ne mogu priuštiti kada je u pitanju PLC uređaj zbog njegove cjenovne nepristupačnosti .

Simulator poput spomenutog zasnovan je tako da doslovno simulira rad PLC uređaja. Korisnik se može, nakon što je savladao osnove operacija ljestvičastih dijagrama, prebaciti direktno na fizički PLC uređaj.

2.PROGRAMABILNI LOGIČKI KONTROLER

U automatiziranom sustavu središtem upravljanja obično se smatra programabilni logički kontroler (*eng. Programmable Logic Controller*) [1]. PLC je specijalizirano industrijsko računalo koje se koristi za upravljanje strojeva i procesa [10]. Koristi programabilnu memoriju za pohranjivanje instrukcija i specifičnih funkcija koje uključuju periodične funkcije, brojanje, aritmetiku i upravljanje podacima [8,2]. Neprekidno nadgleda stanje sustava preko signala povratne veze i upravlja sustavom po izvedbenom kodu koji je pohranjen u memoriji uređaja. S obzirom na isti kod PLC zatim daje izlazne signale koji su spojeni na vanjske uređaje. PLC je pogodan za jednostavne i repetitivne zadatke, kao i za upravljanje kompleksnim sustavom od nekoliko procesa međusobno povezanih drugim računalima [16]. Neke od prednosti logičkog kontrolera su fleksibilnost (dodavanje ulaza i izlaza), brzo vrijeme odaziva, nema pokretnih dijelova, modularni koncept, jednostavno dijagnosticiranje i lociranje pogrešaka, sposobnost za upravljanje kompliciranih sustava itd.

PLC ima mikroprocesor i memoriju te se očito smatra računalom. Osmišljen je kao zamjena za relejni panel koji je bio kompliciran i vremenski zahtijevan za korištenje, tražio je učestalije održavanje, bio je cjenovno nepristupačniji, kao i masivniji od tada novog PLC uređaja [8].

PLC je jednostavne građe (Slika 1.).Prikazan je Allen Bradley modularni programabilni logički kontroler. Tvrtke Allen-Bradley i Siemens najveći su svjetski proizvođači PLC uređaja.

S obzirom na to da je računalo, nema potreba da ga se ograničava na digitalne ulaze i izlaze. Sastoji se od modula centralne procesne jedinice(*eng. Central Processing Unit*) i modula za ulaz i izlaz(*eng. Input/Output*). U nekim slučajevima ovi su moduli sastavni dio samog PLC uređaja dok se nekada nalaze fizički na sasvim različitim mjestima i povezani su podatkovnim kabelima. CPU komunicira s modulom za izlaz i ulaz. PLC se tijekom godina prilagodio za upravljanje analognih i drugih numeričkih signala. Kako bi se ovi numerički uređaji učinili korisnim, uključuju mogućnost kalkuliranja u samom programu, tako da se mogu izračunati npr. statistički procesi upravljanja ili specifične vrijednosti.

Automatizirani sustav ovisi o sposobnosti PLC uređaja da čita ulazne signale od raznih tipova senzora i manualnih uređaja za ulazne signale [1,9]. Tipkala, tipkovnice i prekidači tipovi su manualnih uređaja- komponenti. S druge strane, u svrhu detekcije izlaznog signala tlaka i tekućine ili drugih ulaznih signala, PLC će dobiti signal od automatskog senzornog uređaja kao što je blizinski prekidač, granična sklopka, fotoelektrični senzor, senzor za razinu tekućine ili slično. Tipovi ulaznih signala mogu biti digitalni ili analogni.

Neki od najčešćih aktuatora su elektromotor, sklopnice i releji. Aktivacijom istih PLC uređaj može kontrolirati jednostavan sustav za hvatanje i stavljanje, kao i kompleksnije sustave za pozicioniranje. Aktuatori su mehanizam automatiziranog sustava i imaju izravan učinak na izvođenje sustava.

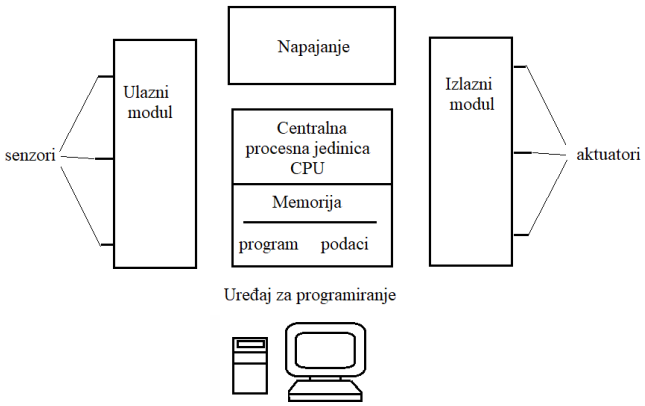
Ostali izlazni uređaji kao što su LED diode i alarmi služe za signalizaciju.

Građa otvorenog topa PLC uređaja omogućava spajanje s programima i uređajima drugih proizvođača.

Slika 2.1 Allen Bradley ControlLogix 5580 [12]

2.1 Arhitektura programabilnog logičkog kontrolera

Glavni dijelove programabilnog logičkog kontrolera čine centralna procesna jedinica, napajanje, uređaj za programiranje, i moduli za ulaz i izlaz (Slika 2.2). CPU je središnji dio uređaja i izvodi isprogramirane operacije. Te operacije ili izlazni signali izvode se s obzirom na ulazne signale i podatke sa spojenih ulaza. Moduli za ulaz spajaju se na razne vanjske uređaje kao što su senzori, prekidači, tipkala [1,2]. S druge strane spojeni su na PLC gdje se ti razni digitalni ili analogni podaci obrađuju. Moduli za izlaz pretvaraju signale iz centralne procesne jedinice u analogne ili digitalne vrijednosti kako bi komunicirali s vanjskim uređajima. Napajanje PLC uređaja predstavlja uređaj koji ga opskrbljuje uređaj električnom energijom pretvarajući dostupnu izmjeničnu struju u istosmjernu koja je potreba za rad centralne procesne jedinice i modula za ulaz i izlaz.



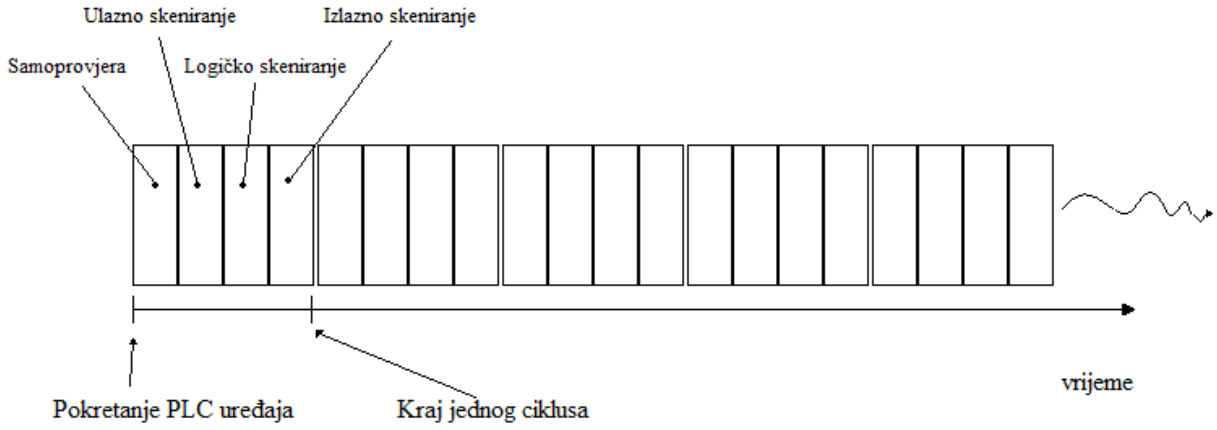
Slika 2.2 Dijelovi PLC uređaja

Postoji nekoliko različitih vrsta memorijskih jedinica. To je područje u kojem je pohranjena memorija sustava i korisnika. Na tim lokacijama spremaju se vrijednosti brojača, vremenskog registra i sadržaj programa.

2.2 Način djelovanja

PLC obavlja zadatke jedan po jedan, kao i većina računala, iako se čini kao da obavljaju više zadataka od jednom (više upravljanih strojeva može raditi istodobno). PLC obrađuje signale poput osobnog računala osim što su prisutni izlazi i ulazi u proizvodnim pogonima, čeličanama... pa je uređaj koncipiran tako da bude pouzdaniji zbog grubog industrijskog okruženja.

PLC ima četiri glavne razine operacija koje se ponavljaju određen broj puta u sekundi [2,8,9]. Pri pokretanju uređaja PLC provjerava hardver i softver, kako bi ustanovio postoji li negdje greška. Ako nije pronađena nepravilnost, kopirat će sve ulazne i izlazne signale i njihove vrijednosti pohraniti u memoriju, to se zove ulazno skeniranje. Koristeći samo kopiju ulaznih vrijednosti, napisani program PLC će skenirati jednom, to se zove logičko skeniranje. Dok se program izvršava ulazni signali mijenjaju se samo u privremenoj memoriji. Kada se izvrši čitanje programa, izlazni signali ažurirat će se koristeći vrijednosti iz privremene memorije, a to se zove izlazno skeniranje. Proces nakon toga ponovno počinje tražeći pogreške. Navedeni proces obično se ponavlja 10 do 100 puta u sekundi (Slika 2.3.).



Slika 2.3 Ciklus rada PLC uređaja

3. PROGRAMIRANJE PLC UREĐAJA

S obzirom na sam njihov naziv, da bi radili PLC uređaji moraju se isprogramirati. PLC uređaji najčešće se programiraju putem aplikacija na osobnom računalu (eng. *Personal Computer*). Računalo komunicira s PLC uređajem putem mrežnog kabela ili komunikacijske sabirnice, ovisno o proizvođaču. Inženjeri se ne mogu dogovoriti oko jedinstvenog načina programiranja PLC uređaja. Premda je ljestvičasti dijagram (*eng. Ladder Diagram*)najčešće priznat kao najdjelotvornija metoda te se koristi u približno 90 % slučajeva, pojedinosti tog jezika drugačije su kod svakog proizvođača. Razlike su u izvedbi, načinima za pisanje naredbi i funkcija te razlike u redoslijedu naredbi [1].

Osim samih elektroničkih sklopova, sama struktura programa za upravljanje PLC uređajima pokazala se kao važna za djelotvornost proizvodnje[13].

Zbog navedenih nesuglasica oko softvera uvedene je standard IEC 61131 američkog nacionalnog instituta na standarde (eng. *American National Standards Institute*) [14]. On definira nekoliko programskih jezika s različitim prednostima i nedostacima. Lokalni standardi za programiranje obično će odrediti programski jezik. Svaki proizvođač s kontrolerom isporučuje i pripadajući alat za programiranje.

Osim osobnih računala, kod manjih PLC uređaja koriste se i ručni uređaji za programiranje. Oni su kompaktniji, povoljniji i laki za korištenje, međutim, mogu biti nepregledni zato što mogu prikazati samo manji dio ljestvičastog koda [2].

Metode programiranja koje se najviše koriste, neovisno proizvođaču uključuju Ljestvičaste dijagrame, Blokove funkcija (*eng. function blocks*) i strukturirani tekst (*eng. structured text*) [8].

Strukturirani tekst je tekstualni PLC programski jezik sličan jezicima kao što su Python, Visual Basic ili C. Jedna od prednosti ove metode jest ta da sam kôd ne zauzima toliko memorijskog prostora s obzirom na to da se sastoji od pisanog teksta. Osim toga može se kombinirati s drugim metodama kao što je funkcijski blok. Metoda funkcijskih blokova je grafička i tako se prikazuju blokovi. Signali i podaci iz ulaza registriraju se u funkcijskom bloku. Kada događaj potakne program funkcijskog bloka, PLC na izlazu vrati jedan ili više

signala. Funkcijski blokovi imaju standardne funkcije kao što su vremenski registri, brojači, algebarski izrazi itd.

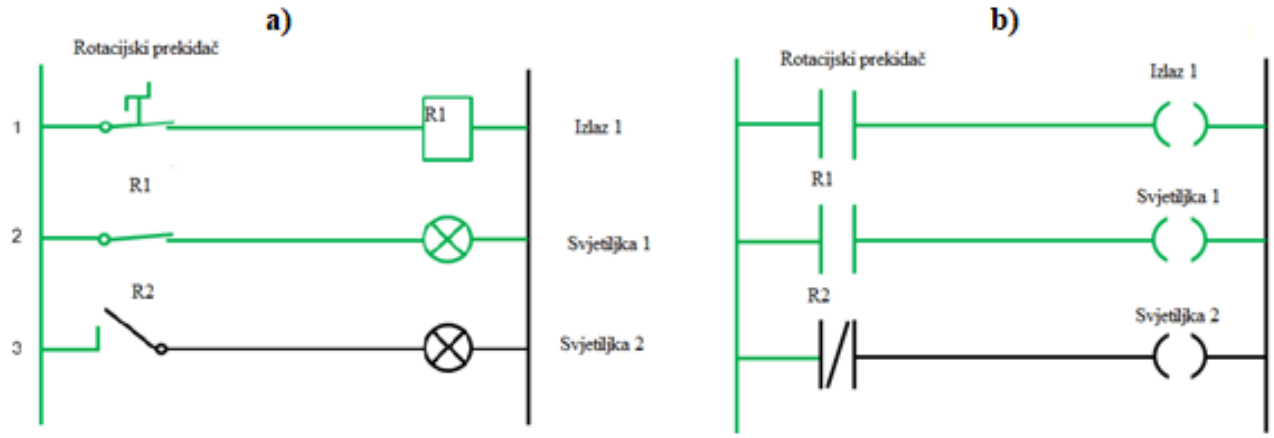
3.1 Relejna logika

Programabilni logički kontroleri zamijenili su elektromagnetske releje. Programski jezik koji se koristi za programiranje još uvijek koristi simbole koji su se koristili za prikazivanje relejnih sklopova. Relej (Slika 3.2) je izumljen 1800-tih godina. Koristio se prvenstveno za telegrafski sustav, međutim nije prošlo dugo dok se nije počeo primjenjivati u svrhu upravljanja procesa.

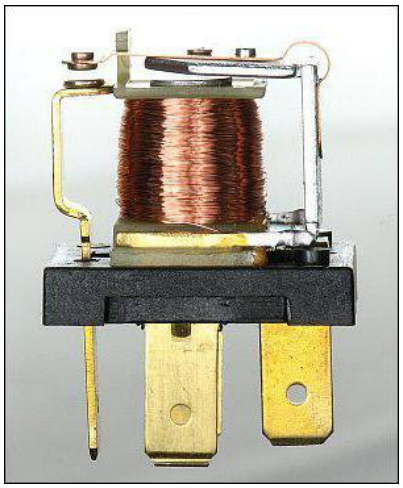
Kada je struja sklapanja do 20 mA govori se o releju, kod većih iznosa govori se o sklopniku koji sadrži komoru za gašenje. Kod relejne logike je riječ o povijesnim raritetima nepovezanim s PLC-ima tako da termin ljestvičasti dijagram predstavlja tehnologiju u koju je uključeno i povijesno nasljeđe na razini simbolike.

Ljestvičasti dijagram zbog jednostavnosti i prikladnosti gotovo nije promijenjen od kontaktnog dijagrama (ili relejne logike) kojima su se prikazivali relejni sustavi. Tako je prelazak s relejne logike na ljestvičaste dijagrame i PLC-e ostvaren bez simboličkog raskoraka [8].

Prikazana su dva dijagram od kojih je jedan relejni , a drugi ljestvičasti dijagram (Slika 3.1). Oba sklopa prikazuju istu logiku.



Slika 3.1 Sličnost između relejnih (a) i ljestvičastih dijagrama (b) [17]



Slika 3.2 Elektromagnetski relej [15]

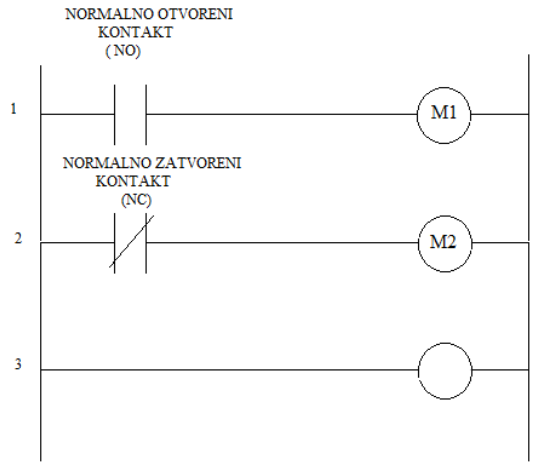
4. LJESTVIČASTI DIJAGRAMI

Ljestvičasti dijagrami (*eng. Ladder diagrams*) grafički je programski jezik i najčešća je metoda PLC programiranja. Može se upotrijebiti za zadatke kao što je brojanje, brojanje vremena, upravljanje podacima itd. Ljestvičasti dijagrami konstruirani su tako da su slični relejnim dijagramima, fizički prekidači i zavojnice koji se koriste u relejnim sklopovima zamijenjeni seu memorijskim lokacijama PLC uređaja kao i izlazima/ulazima..

Ljestvičasti dijagram čita se s lijeva na desno i od gore prema dolje (Slika 4.1). Sa svake strane nalazi se jedna vertikalna linija. One simboliziraju glavni dovod električne energije. Njih spajaju horizontalne linije (*eng. Rungs)* koje se mogu nazivati linijama koda. S lijeve strane pišu se ulazi dok su s desne izlazi. S lijeve strane dakle nalaze se razni uvjeti odnosno naredbe koje zatim utječu na izlazni signal koji je označen na desnom kraju linije koda. Ljestvičasti dijagram može se sastojati od velikog broja linija koda, ovisno o složenosti programa [6].

U ulaznom dijelu programa kontroliraju se ulazni uređaji (senzori) kao što su detektor svjetlosti, tipkala prekidači, a u izlaznom dijelu kontroliraju se izlazni uređaji (aktuatori) kao što su motor, alarmi i svjetiljke, ventili itd.

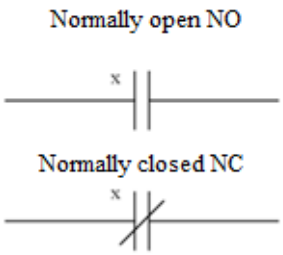
Procesor PLC uređaja također skenira kod ljestvičastog dijagrama s lijeva na desno i od gore prema dolje. To se može promijeniti ako se koriste instrukcije za preskakanje ili potprogrami.



Slika 4.1 Izgled ljestvičastog dijagrama

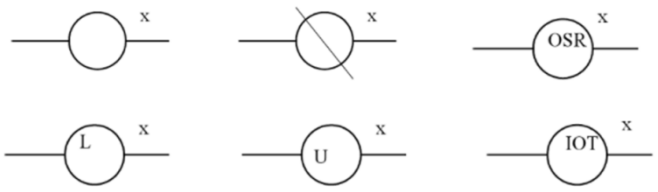
4.1 Ulazi i izlazi ljestvičastog dijagrama

Dva su tipična ulazna kontakta, normalno otvoreni (*eng. Normally open) i* normalno zatvoreni (*eng. Normally closed)* (Slika 4.2). U normalno otvorenom tipu nema protoka struje dok je ulaz neistinit. Kada status postane istinit, kontakt se zatvori. Kod normalno zatvorenog tipa kontakta uz ulaz u stanju logičke nule kontakt je zatvoren odnosno ako je ulaz neistinit, u suprotnom kontakt se otvara i prekida se protok struje.



Slika 4.2 Tipovi sklopki Normally open NO i Normally closed NC

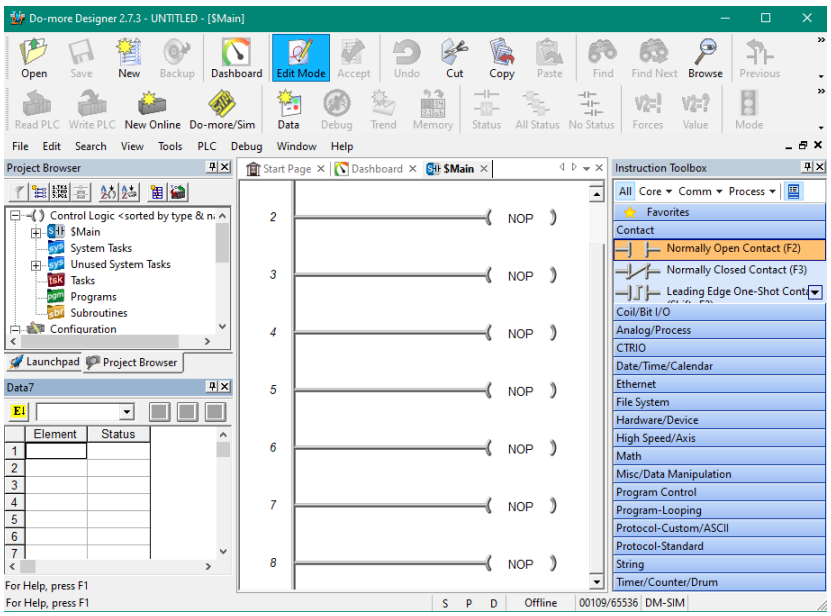
U ljestvičastim dijagramima postoji više vrsta izlaza, ali oni nisu dostupni na svim PLC-ima. Pojedini izlazi spojeni su na uređaj izvana, ali za ostvarenje logičke funkcije moguće je koristiti unutarnje memorijske lokacije unutar PLC uređaja. Na prikazu (Slika 4.3) je šest vrsta izlaza. Prvi simbol prikazuje normalni izlaz, kada se aktivira izlaz postaje istinit. Krug s dijagonalnom linijom prikazuje normalno uključen izlaz koji će, kada se aktivira, postati neistinit, međutim, taj izlaz nije dostupan na svim PLC uređajima. OSR izlaz (*eng. One Shot Relay*) će, kada se aktivira, postati pozitivan samo za prvo skeniranje programa, zatim će se isključiti. L izlaz (*eng. Latch*) i U izlaz (*eng. Unlatch*) koriste se za blokiranje izlaz, odnosno, kada je potrebno da izlaz ostane u jednom stanju, čak i ako se prethodni uvjet ponovno promijeni. Kada se L aktivira jednom, izlaz ostaje uključen besprekidno. Izlaz se može ponovno promijeniti samo koristeći naredbu U. U nekim programima ti se izlazi označuju kao SET i RESET, njihov način djelovanja je isti. Posljednji simbol prikazuje IOT (*eng. Immediate OutpuT*) izlaz koji omogućuje izlazu ažuriranje bez čekanja da skeniranje završi [8].



Slika 4.3 Simboli izlaza ljestvičastog dijagrama [8]

4.2 Osnovne funkcije simulatora

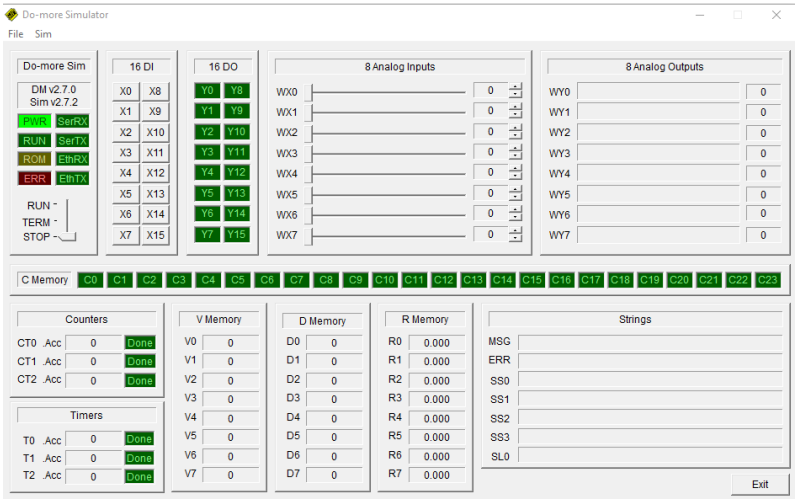
U osnovi je simulatorom moguće vrlo intuitivno rukovati. U razvojnom okruženju nalaze se prazne linije ljestvičastog dijagrama (Slika 4.4). S desne strane se nalazi alatna traka s ulazima, izlazima i funkcijama. Prikazani računalni program može se spojiti i na pravi PLC uređaj. U ovom slučaju koristi se simulator.



Slika 4.4 Računalni program

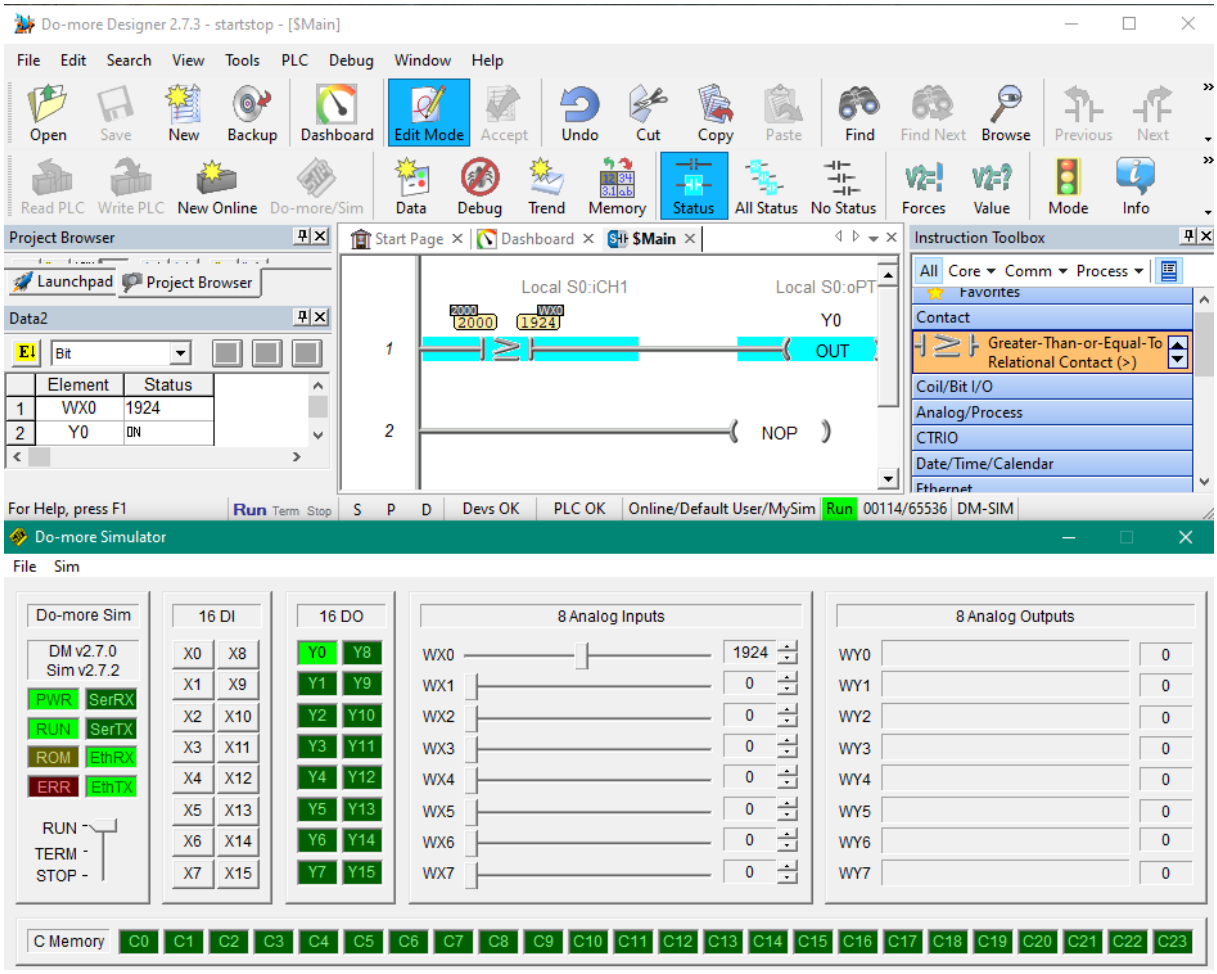
Nakon što se unesu ulazi i izlazi sa zadanim adresama, klikom na *Accept* i zatim *Save* (preuzmi i pohrani) , ako nije otkrivena niti jedna greška, ljestvičasti dijagram je spreman za PLC. Odabirom *PLC>Connect* otvara se izbornik koji omogućuje odabir uređaja za spajanje. Kada je ostvarena veza sa simulatorom (prema riječima proizvođača moguće je koristiti i različite fizičke PLC-e), ljestvičasti kôd može se pohraniti na simulator klikom na *WritePLC.*

Ako je potrebno program s uređaja prenijeti u simulator odaberemo *Read PLC*. U samom simulatoru (Slika 4.5) na gornjem lijevom kutu prikazan je status uređaja koji je zaustavljen - *STOP*. Kako bi se proces pokrenuo status uređaja mora biti *RUN*. Na simulatoru se nalazi šesnaest Digitalnih ulaza *X0-15* kojima se može upravljati direktno iz simulatora klikom. Prikazan je status šesnaest izlaza Y0-15, 8 analognih ulaza i izlaza označenih s WX i WY. Prikazan je i status c, v, i d memorije, kao i status brojača i vremenskog registra.



Slika 4.5 PLC Simulator

Na primjeru (Slika 4.6) prikazan je aktivni izlaz Y0 uvjetovan analognim ulazom koji je uključen za sve vrijednosti ulaza WX0 koje su manje od 2000.



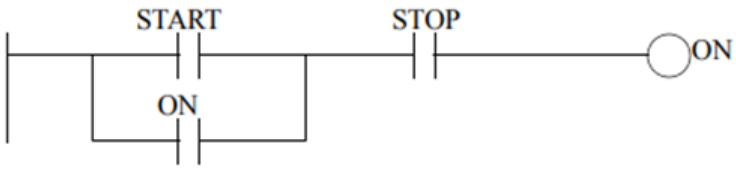
Slika 4.6 Primjer analognog ulaza

Očito je da se priloženi pristup može koristiti primjerice za regulaciju temperature ili razine tekućine. Grafičko sučelje programa ujedno pruža podatke s ulaznih senzora, njihove granične vrijednosti ili vrijednosti prorade i status izlaznih signala (primjerice napajanje grijača ili pumpi).

4.2.1 Sklop za pokretanje i zaustavljanje Start-Stop

Programi za sustave upravljanja procesa najčešće imaju manualno tipkalo za pokretanje i zaustavljanje pogona [1]. START tipkalo normalno otvorenog tipa (NO), u trenutku kada se pritisne, krug je zatvoren i iznos na izlazu npr. električnog stroja je istinit. Stroj je pod naponom i primjerice električna traka u supermarketu prenosi namjernice. Međutim, kada se tipkalo otpusti, strujni krug je ponovno otvoren i na izlazu (načelno) nema

signala. Da bi se riješio taj problem potrebno je dodati pomoćni kontakt kao na slici (Slika 4.7).

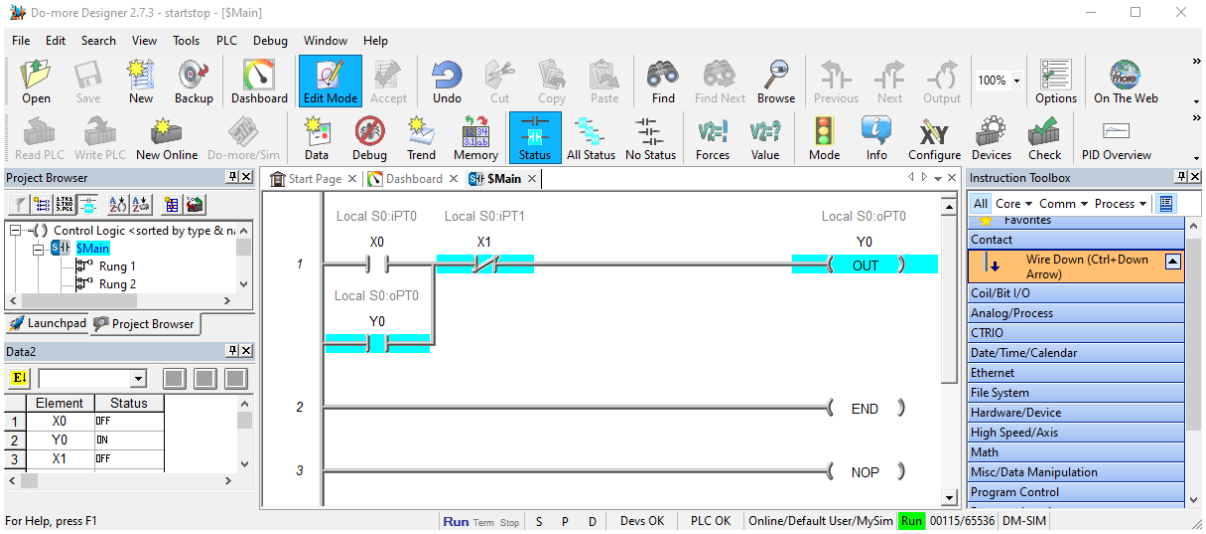


Slika 4.7 Start-Stop dijagram

Pomoćni kontakt služi kako bi strujni krug ostao zatvoren nakon što se otpusti START tipkalo. U primjeru (Slika 13) ulaz je označen s x0 predstavlja START tipkalo, ulaz x1 stop tipkalo. Izlaz y0 može biti električni motor. Ulaz x0 je normalno otvorenog, a x1 normalno zatvorenog tipa. Pomoćni kontakt normalno otvorenog tipa označen je s y0. To je zato što je uvjet uključivanja ovog kontakta pozitivan signal na izlazu y0. Tako da, dokle god je izlaz y0 istinit pomoćni kontakt ostaje zatvoren. U relejnoj tehnici, to bi značilo, da čim električni stroj dođe pod napon, aktivira se upravljački krug sklopnika koji predstavlja ulaz y0. Dok je stroj pod naponom sklopka y0 jest zatvorena.

To znači, kada se pritisne start tipkalo motor se uključi i pomoćni kontakt (y0) se zatvori. Kada se tipkalo otpusti motor ostaje uključen jer je pomoćni kontakt zatvoren.

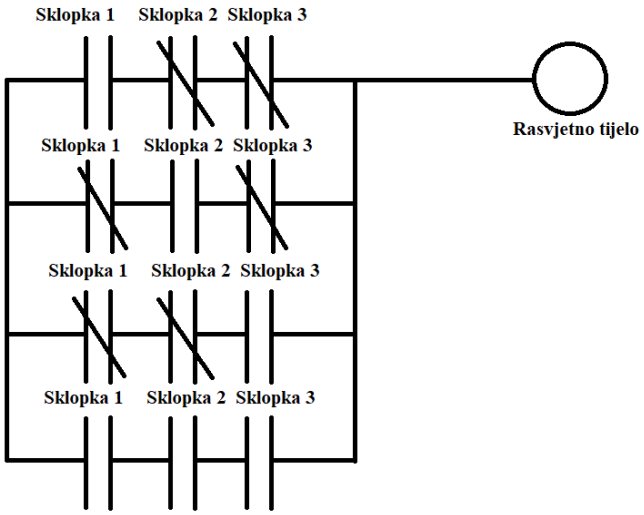
Plavo je označen logički tok, odnosno tok signala. U primjeru na slici u donjem lijevom uglu nalazi se status izlaza i ulaza. Iako je ulaz x0 isključen (stanje 0), status izlaza ostaje istinit (stanje logičke jedinice).



Slika 4.8 Prikaz Start –Stop dijagrama u PLC simulatoru

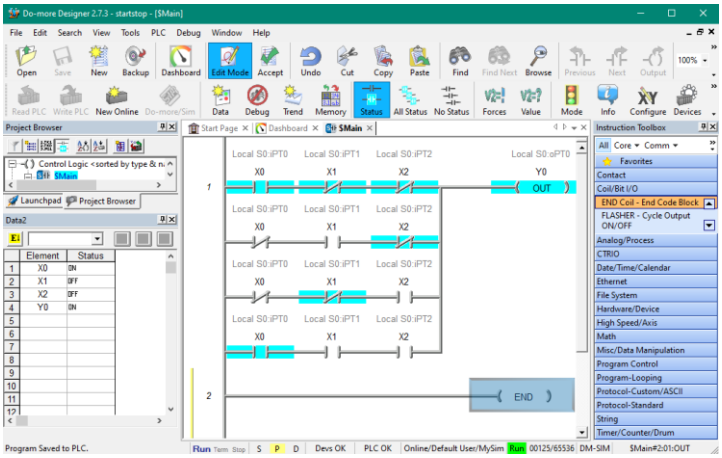
4.2.1 Svjetiljka s tri različita prekidača

Problem iz primjera [8] opisuje tri različita prekidača sa sklopkama se kontrolira jedna svjetiljka. Ako se spoje paralelno normalno otvoreni kontakti, kada jedan kontakt postane istinit svjetiljka će biti uključena, ali za isključivanje svjetiljke potrebno je da sva tri ulaza budu neistinita. Za konkretnije rješenje potrebno je više paralelno spojenih kontakata (Slika 4.9). Na slici je prikazan 'Isključivo ILI' sklop. Tako je nadopunjeno upravljanje svjetiljkom.



Slika 4.9 Upravljanje svjetiljkom

U primjeru (Slika 4.10), uključena je prva sklopka sklopka 1 ili ulaz 1 i status Svjetiljke je istinit (logička jedinica).

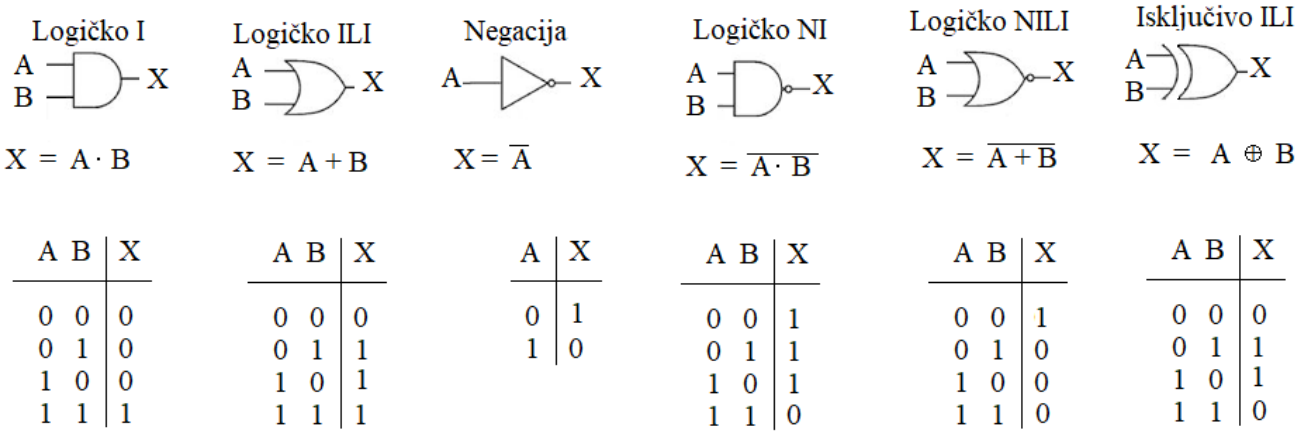


Slika 4.10 Programski kod primjera sa slike 4.9

5.BOOLEOVA ALGEBRA

Proces pretvorbe pretvaranja ciljeva kontrolnog sustava u ljestvičastu logiku zahtjeva strukturiranu misao. Booleova algebra razvijena je 1800-ih. Razvio ju je Irski matematičar James Boole. Otkrilo se da je krajnje korisna za konstruiranje digitalnih sklopova, i još uvijek koristi u elektrotehnici i računalnim znanostima. Ova tehnika može opisati logični sustav jednom jednadžbom. Tu jednadžbu se može zatim pojednostavljivati ili ju pretvoriti u druge oblike. Ista tehnika koja je razvijena za projektiranje sklopova dobro je usvojena za programiranje ljestvičastim dijagramom [8].

Booleove jednadžbe sastoje se od varijabli i operacija i izgleda vrlo slično kao normalne algebarske jednadžbe. Tri osnovna operatora su I (konjunkcija), ILI (disjunkcija) i NE (negacija). Kompleksnije operacije su 'isključivo ILI' , NILI i NI .Svaki operator u tablicama istinitosti (Slika 5.1) prikazan je jednostavnom jednadžbom s varijablama A i B s kojima se računa vrijednost X. Tablice istinitosti su jednostavna metoda za prikaz svih mogućih kombinacija za koje će izlaz biti istinit ili neistinit. Neistinito stanje se zove isključeno *OFF* ili 0, a istinito uključeno *ON* ili 1.



Slika 5.1 Tablice istinitosti operatora Booleove algebre

U Booleovoj algebri operatori su postavljeni u kompleksnijem obliku .Varijable ovih jednadžbi prate zakonitosti slične klasičnoj algebri. Dijelovi jednadžbe unutar zagrade rješavaju se prvo. Operacije se rješavaju po redoslijedu NE, I, ILI. U jednadžbi (1) funkcija NE za varijablu C rješava se prva, ali funkcija NE koja je iznad prve zagrade rješava se nakon što se riješi dio jednadžbe unutar iste zagrade. Kada postoji izbor, I operacije rješavaju se prije ILI operacija. Za zadani primjer rezultat jednadžbe je neistinit, odnosno 0.

Jednadžba:

$X=\overline{(A + B \cdot C)} + A \cdot (B+\overline{C})$ (1)

Proizvoljne vrijednosti A=1, B=0, C=1

$X= \overline{(1 + 0 \cdot 1)} + 1 \cdot (0+\overline{1})$

$X=\overline{(1 + 0)} +1 \cdot (0+0)$

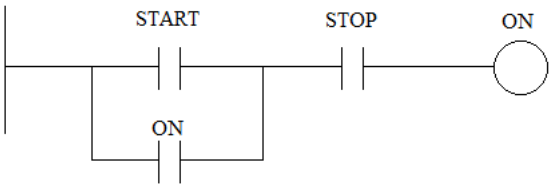
$X= \overline{(1)}+1 \cdot (0)$

X=0+0

X=0

Koncepti za procesni sustav mogu se prevesti direktno u Booleovu jednadžbu. Oblik Booleove jednadžbe se može zatim pojednostaviti ili presložiti i zatim prenijeti u program ljestvičastog dijagrama ili sklop.

Logika za SART- STOP sklop može se analizirati putem Booleove algebre (Slika 5.2)

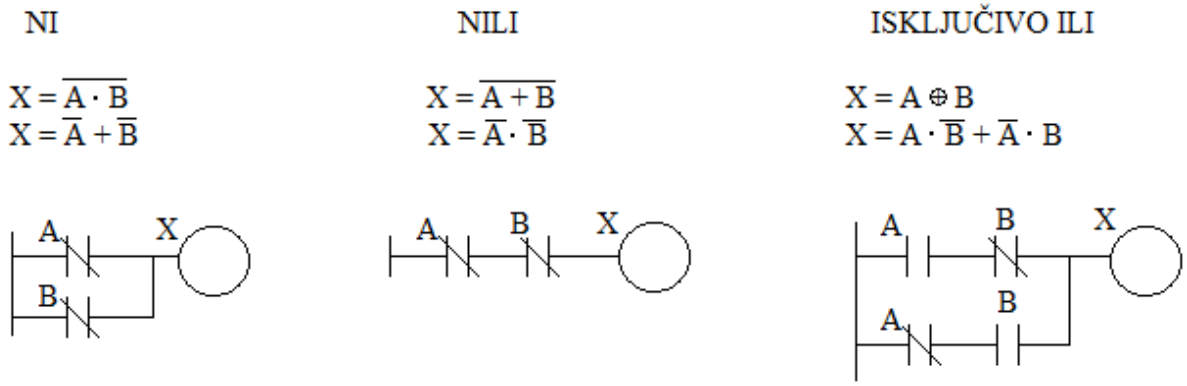


ON' = (START + ON) · STOP

ON	STOP	START	ON'
0	0	0	0
0	0	1	0
0	1	0	0
0	1	1	1
1	0	0	0
1	0	1	1
1	1	0	1
1	1	1	1

Slika 5.2 Logika START-STOP sklopa

Za dva ulaza postoji 16 različitih tipova sklopova osnovnih operatora. Najjednostavniji su I i ILI. Tri popularna kompleksna sklopa su NI, NILI i isključivo ili. Sve od navedenih može se reducirati na jednostavnije oblike koristeći I i ILI sklopove koji su adekvatni za ljestvičasti dijagram (Slika 5.3) [2].



Slika 5.3 Kompleksni logički operatori

5.1 Primjer industrijske peći

Ako se rad PLC-a može opisati riječima, često se može i prevesti u Booleovu jednadžbu. U primjeru iz literature [8] zadan je opis procesa.

Zadana je grijača peć s dvije trake koja može grijati po jednu metalnu palicu ili ingot na svakoj traci. Kada je grijač uključen on dovodi dovoljno topline za dva ingota. Ako se u peći nalazi samo jedan ingot, peć se može pregrijati, stoga je potrebno uključiti ventilator za hlađenje kada peć dostigne određenu temperaturu.

U stvarnom svijetu te informacije mogu se dobiti od konstruktora mehaničkog dijela sustava. Sljedeći korak je odrediti na koji način bi PLC trebao raditi. Zadatak se postavi rečenicom koja se prevodi u Booleovu jednadžbu.

Logički opis upravljanja glasio bi da ako je temperatura previsoka i u peći se nalazi jedan ingot, uključi ventilator. Izlazi i ulazi mogu se definirati kao B1 za ingot na jednoj traci i B2 za ingot na drugoj traci, V za ventilator i T za senzor temperature koji se uključi ako ona nadmaši određenu vrijednost.

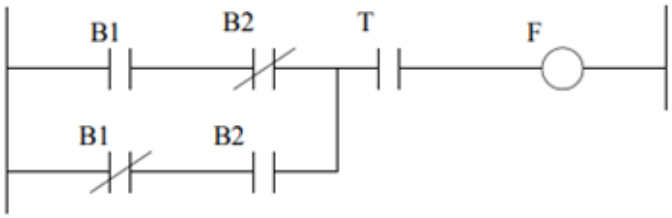
Jednadžba:

$F=T\cdot (B1 \oplus B2)$ (3)

$F= T\cdot(B1\cdot \overline{B2}+\overline{B1}\cdot B2)$

Iz jednadžbe (3) dalje se može dobiti željeni oblik logičkog sklopa. Jednadžba (2) opisana je operatorom 'isključivo ILI', a taj izraz nije prikladan za ljestvičasti dijagram. Jednadžba se prevodi u željeni oblik koristeći osnovne logičke operatore.

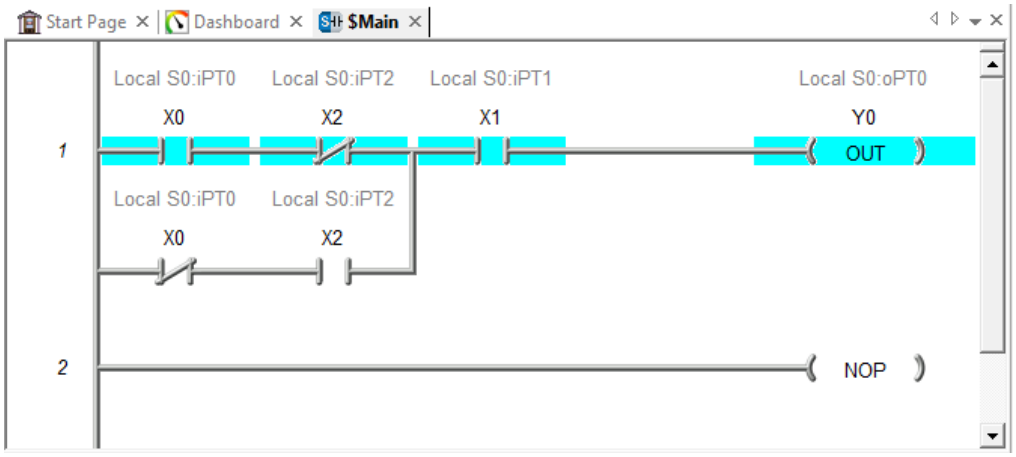
Kako bi tu jednadžbu pretvorili u ljestvičasti dijagram, izrazi koje spaja operator I, spojeni su u seriji. Izrazi s operatorom ILI spojeni su paralelno, a izrazi pod negacijom koriste normalno zatvoreni kontakt. Iz krajnje jednadžbe izraz je preveden u ljestvičasti dijagram. Time se dokazuje da se Booleova jednadžba može lako prevesti u drugačiji, ali i ekvivalentni ljestvičasti dijagram (Slika 5.4).



Slika 5.4 Dijagram industrijske peći [8]

U stvarnom programu navedeni dijagram bio bi samo jedan dio cijelog koda.

Na slici je prikazan isti ljestvičasti dijagram u računalnom programu (Slika 5.5). Stanje ulaza X0 i X1 je istinito, što znači da je jedan ignot na jednoj traci. X2 koji označava ignot na drugoj traci nije istinit. X1 koji označava senzor za pre visoku temperaturu je upaljen. Tamnim markerom označen je logički tok istinitosti programa. Ventilator je uključen



Slika 5.5 Dijagram industrijske peći na računalnom programu

Simulacijski program, dakle, otvara mogućnost provjere djelovanja automatiziranih industrijskih postrojenja ili preciznije rješavanje logičkih jednadžbi u suvremenom programskom okružju (Slika 20).

6. KARNAUGHOVE ILI K TABLICE

Karnaughove ili K tablice omogućuju pretvorbu tablice istinitosti u pojednostavljenu Booleovu jednadžbu bez korištenja Booleove algebre. Najlakše ih je opisati pomoću primjera [8]. K-tablice alternativne su metode pojednostavljivanja jednadžbi Booleovom algebrom. Ova metoda prikladan je način za provjeru Booleovih jednadžbi i adekvatna je za vizualni stil učenja. Primjer koji slijedi sadrži četiri varijable, dvije za stupce i dvije za redove. Mogu se koristiti i s više varijabli. Kod pet varijabli Mogu se koristiti tri varijable za red ili stupac, a niz bi slijedio 000,001,011,010,111,101,100. Kod više od jednog izlaza potrebna je po jedna K-tablica za svaki.

6.1 Protuprovalni alarm

Kada se uključi alarm, mora se uključiti i svjetiljka. Alarm se aktivira kada nepoželjna prisutnost potakne senzor na prozoru i senzor pokreta. Senzor na prozoru ponaša se kao normalno zatvoreni prekidač. Senzor pokreta je konstruiran tako da kada je osoba detektirana izlaz će se uključiti. Potreban je, također jedan prekidač za aktivaciju i deaktivaciju.

Ulazni i izlazni signali sastoje se od prekidača za alarme i svijetla A, senzor na prozoru W, senzor pokreta M, prekidač za uključivanje alarmnog sustava S i status bez alarma Q (Alarm Quiet).

Način djelovanja alarma : ako je alarmni sustav uključen, čitaj senzore, ako je senzor na prozoru aktiviran (ako se otvori prekidač) uključi alarm i svijetla.

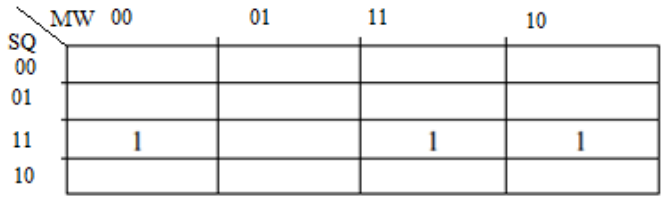
Sljedeći korak je definirati logičku jednadžbu. U ovom slučaju ona sadrži tri različita ulaza i jedan izlaz, tako da je prikladno koristiti se tablicom istinitosti. Booleova jednadžba se može zatim napisati koristeći tablicu istinitosti (Slika 6.1). Od osam mogućih kombinacija ulaza, samo tri ostvaruju uvijete za aktiviranje izlaza.

Tablica 1 Tablica istinitosti protuprovalnog alarma

S	M	W	Q	A
0	0	0	0	0
0	0	0	1	0
0	0	1	0	0
0	0	1	1	0
0	1	0	0	0
0	1	0	1	0
0	1	1	0	0
0	1	1	1	0
1	0	0	0	0
1	0	0	1	1
1	0	1	0	0
1	0	1	1	0
1	1	0	0	0
1	1	0	1	1
1	1	1	0	0
1	1	1	1	1

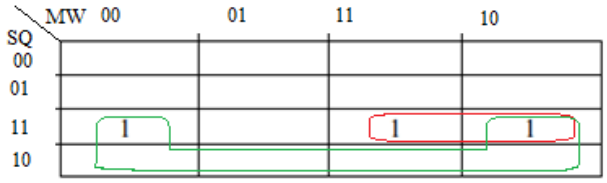
U recima gdje je varijabla S=0, alarm je isključen. Kada je varijabla M istinita, senzor pokreta je aktiviran, što znači da je osoba detektirana. Ako je varijabla W neistinita (0) znači da je kontakt na prozoru prekinut, odnosno, uljez je detektiran.

Umjesto pretvaranja direktno u Booleovu jednadžbu, stanje se postavlja u tablicu (Slika 6.2). Retke i stupce tvore ulazne varijable. Odluka o tome koje varijable koristiti može biti proizvoljna. U tom slučaju tablica će imati drugačiji izgled, no i dalje se može dobiti slično rješenje. Za retke i stupce niz nije binaran, ali je organiziran tako da se samo po jedan bit promijeni, tako da je niz bitova: 00,01,11,10. Taj korak je ključan. Zatim se vrijednosti iz tablice istinitosti za koje je izlaz istinit unose u Karnaughovu tablicu. Nule se također mogu unijeti, ali nisu nužne. U ovom primjeru tri vrijednosti iz tablice istinitosti su unesene u K-tablicu. Sljedeći korak je podijeliti varijable na dva dijela. U ovom slučaju su SQ i MW. Zatim se nacrta K-tablica prema tim varijablama.



Slika 6.1 K-tablica

Nakon što se unesu podaci u K-tablicu trebali bi se pojaviti očiti uzorci. Ovi uzorci u pravilu pokazuju neku vrstu simetrije. Na primjeru tablice (Slika 23) dva uzorka su zaokružena. Desni je zaokružen jer se dva bita nalaze jedan pokraj drugog. Drugi je uzorak teže je uočiti jer, iako tablica ima desnu i lijevu krajnju stranu, stupci se, kao i redci nadovezuju. Lijevi krajnji stupac s desnim krajnjim, kao i gornji stupac s donjim. Tako da je drugi uzorak zaokružen jer se smatra da su dva bita jedan pokraj drugog. Neki bitovi se koriste više od jedanput, ovo će dovesti do ponavljanja u krajnjoj jednadžbi, no dobiveni izraz bit će jednostavniji

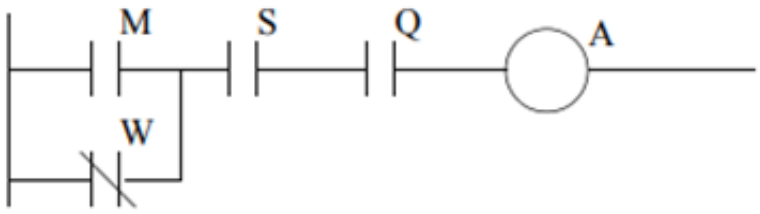


Slika 6.2 Uzorci K-tablice

Ovi uzorci se zatim mogu prevesti u Booleovu jednadžbu (4). Svi se uzorci nalaze u trećem retku, stoga će jednadžba biti pomnožena (Logički operator I) s varijablama SQ. U trećem retku nalaze se dva uzorka, jedan ima M kao zajednički izraz, a drugi ima W kao zajednički izraz. Oni se zatim mogu uvrstiti u jednadžbu. Na posljetku, jednadžba se prenese u ljestvičasti dijagram (Slika 6.4).

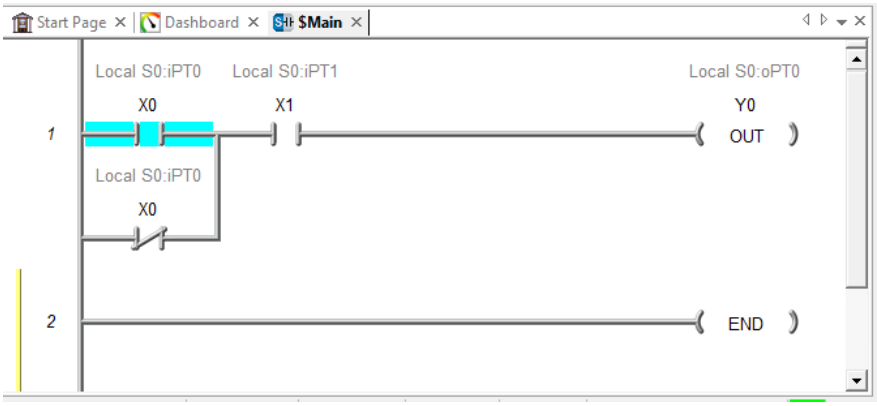
Jednadžba:

$$A= S\cdot Q\cdot(M+\overline{W}) \qquad (4)$$



Slika 6.3 Dijagram alarma [8]

Na primjeru iz računalnog programa (Slika 6.5) je ulaz X0 koji označava senzor pokreta uključen. Ulaz X1, koji označava tipku S je isključen, što znači da alarm nije aktiviran, iako je okinut senzor pokreta, jer nije uključen.



Slika 6.4 Dijagram alarmnog sustava u računalnom programu

7. VREMENSKI REGISTRI, BROJAČI I ZAKLJUČAVANJE

Kompleksniji sustavi ne mogu se upravljati koristeći samo kombinatornu logiku. Glavni razlog je taj što se uglavnom ne mogu postaviti detektori za sva stanja. U takvim slučajevima moguće je koristiti događaje kako bi pretpostavili stanje sustava. Uobičajeni događaji koji se koriste u PLC programiranju su:

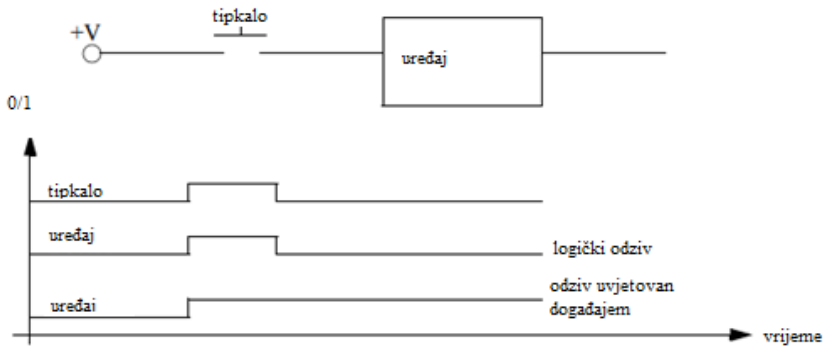
Prvo skeniranje računala- ukazuje da se je uređaj u tom trenutku uključio,

Vrijeme koje je prošlo od kada se je neki ulaz uključio,

Brojač događaja- naredba za čekanje dok se nije pojavio određen broj događaja,

Naredba za zaključavanje nekog seta naredbi u uključeno stanje ili isključiti ih (SET i RESET).

Zajednička pitanja oko kojih se vrte ovi događaji su „Koliko čega?“ i „Koliko dugo vremena?“. Na primjeru (Slika 7.1) ulaz uređaja je tipkalo [8]. Kada se pritisne tipkalo aktivira se ulaz. Kada se tipkalo zatim pusti uređaj se isključuje, to je logički uređaj. Kada bi uređaj ostao uključen nakon što se tipkalo otpusti, to bi bio uređaj uvjetovan događajem. Prema navedenom, uređaj je uvjetovan događajima ako može reagirati na jedan ili više događaja koji su se dogodili. Ako uređaj reagira samo na jedan način u trenutku promijene ulaza, onda je taj uređaj logički.



Slika 7.1 Odziv uređaja [8]

7.1 Zaključavanje statusa

Autor izvora [8] H. Jack opisuje naredbu zaključavanja statusa metaforom ljepljivih tipki (*eng. Sticky keys*). Blokiranje (*eng. latching*)je kao ljepljiva tipka - kada se pritisne uključi se, ali mora se povući kako bi se otpustila i isključila. Blokada ili zaključavanje u ljestvičastom dijagramu koristi jednu naredbu za zaključavanje (*SET*)i drugu za otključavanje (*RESET*)(Slika 7.2). Izlaz C uključit će se kada ulaz A postane istinit . C će ostati uključen čak i kada ulaz A postane neistinit. Isključit će se ako ulaz B postane istinit. Ako izlaz postane zablokiran u uključeno stanje, zadržat će vrijednost, čak i kada se prekine dovod struje.

Slika 7.2 SET i RESET izlazi ljestvičastog dijagrama

7.2 Vremenski registri

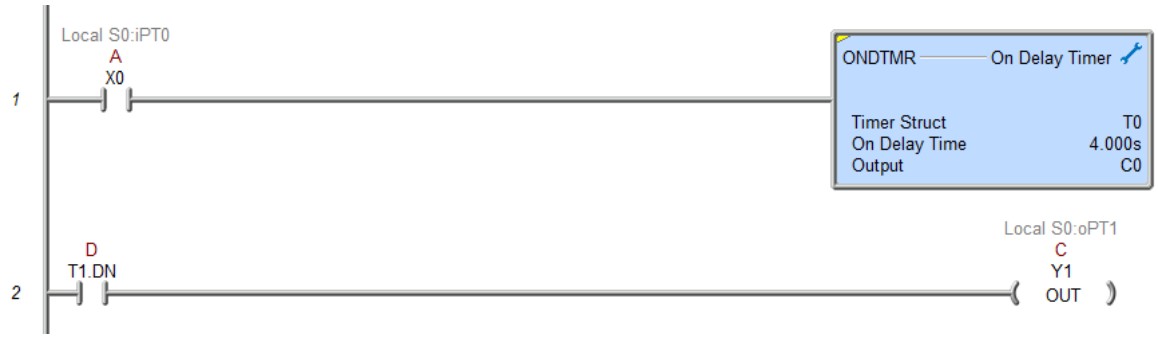
Četiri su osnovna tipa vremenskih registara (Slika 7.3). Odgađanje uključivanja (*eng. ON-delay)* vremenski registar čeka određeno vrijeme nakon što je linija ljestvičastog dijagrama istinita prije nego što će se uključiti, ali će se istog trenutka isključiti .Odgađanje isključivanja (*eng. OFF-delay)* mjerač uključit će se istog trena kada linija ljestvičastog dijagrama postane istinita, ali će odgoditi vrijeme isključivanja [4]. Za shvaćanje rada navedenog primjera slijedi metafora automobila. Ako se okrene ključ za paljenje motora, automobil se ne pali trenutno, to je (*ON-Delay)*. Ako okrenete ključ kako biste ugasili motor ali on se ne ugasi par sekundi, to je (*OFF-delay)* [8]. *On-delay* vremenski registar može se koristiti kako bi dopustio peći da dostigne temperaturu prije početka proizvodnje. *Off-delay* registar može ostaviti ventilatore uključenima nakon što se peć ugasi.

Retentivni vremenski registar sumirat će svo vrijeme uključenosti ili isključenosti nekog vremenskog registra, čak i ako nije završio brojati. Neretentivni vremenski registar

započeti će brojanje zaostajanja od nule svaki put. Uobičajene primjene za retentivni vremenski registar je praćenje vremena do sljedeće potrebe za održavanjem [8].

Neretentivni vremenski registar može se koristiti kada je potrebno vrijeme odgode nakon što se pritisne START tipkalo prije nego što se transportna traka počne kretati.

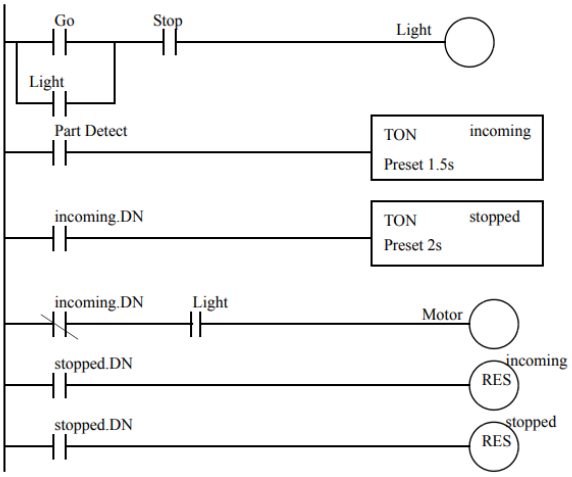
Primjer ONDTMR vremenski registar prikazan je na slici (Slika 7.4). Program ima jedan ulaz A i funkcijski blok za ONDTMR funkciju. Informacija unutar vremenskog registra opisuje parametre brojanja (mjerenja) vremena. Prvi redak *Timer Struct* je lokacija u memoriji PLC-a u kojoj će se pohraniti informacije registra. *On Delay Time* je odgoda registra u milisekundama, u ovom slučaju 4000 ms, odnosno 4 s. Dok vremenski registar radi, u registru *Accumulator* se povećava vrijednost dok ne postigne zadanu vrijednost. *EN* i *DN* izlazi ne mogu se promijeniti prilikom programiranja, ali su bitni kod traženja pogreški na programima ljestvičastom dijagramu. Drugi redak ljestvičastog dijagrama koristi *DN* izlaz vremenskog registra za upravljanje izlaza *B*.



Slika 7.3 Primjer vremenskog registra[8]

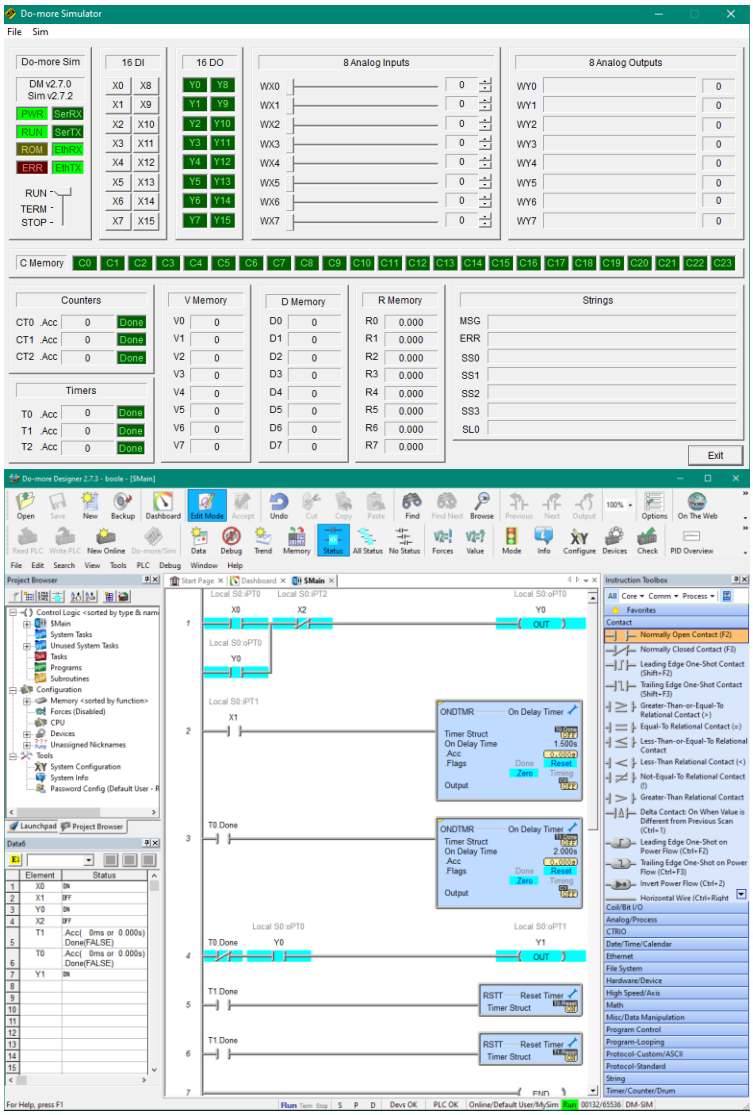
7.2.1 Transportna traka

Transportna se traka pokreće uključivanjem i isključivanjem motora. Dijelovi se pozicioniraju na traku pomoću optičkog detektora. Kada se optički senzor uključi, čeka se 1.5 sekundi i zatim zaustavlja traka. Nakon odgode od 2 sekunde traka ne ponovno pokreće. Svijetlo bi trebalo biti upaljeno kada je sustav aktivan.



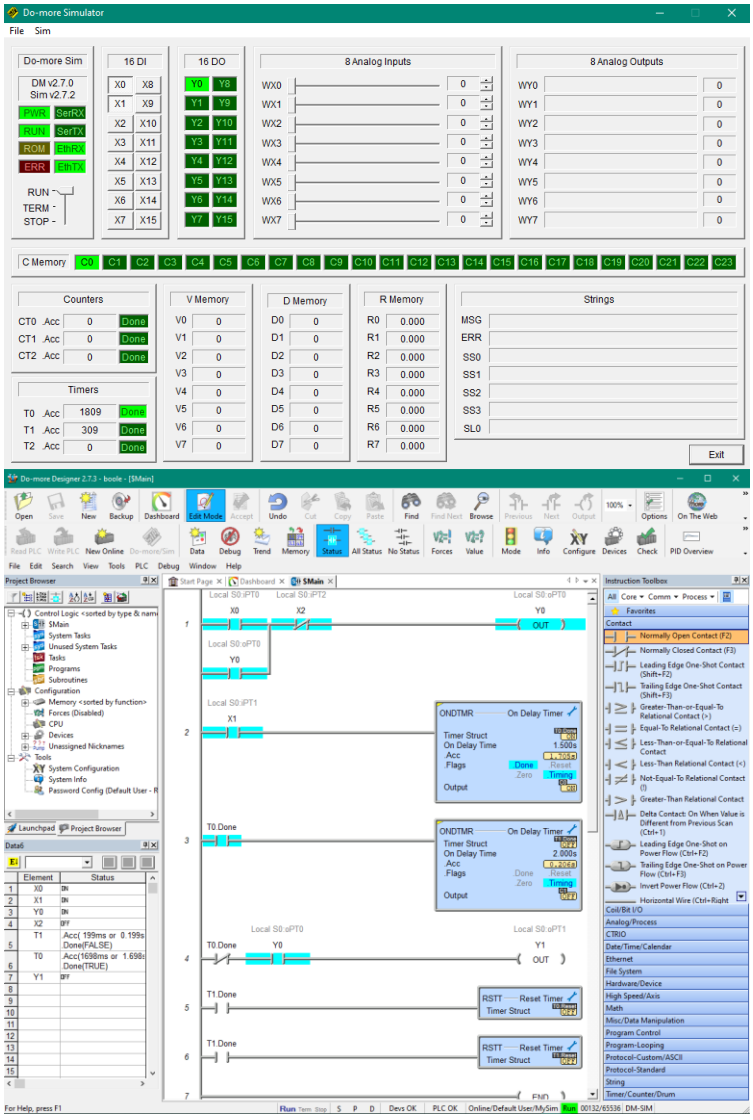
Slika 7.4 Dijagram pokretne trake [8]

U primjeru dolje (Slika 7.6) sustav je pokrenut i svjetlo (Y0) i motor (Y1) su uključeni. Senzor komada (X1) nije se uključio stoga vremenski registar nije pokrenut. Na simulatoru su prikazani izlazi Y0 i Y1 kao osvijetljeni.



Slika 7.5 Pokretanje sustava

Na sljedećem prikazu (Slika 7.7) senzor X1 se uključio. Vremenski registar T0 je odbrojao 1,5 sekundi i motor (Y1) se isključio. Vremenski registar T2 odbrojava 2 sekunde prije nego što ponovno pokrene motor. Na simulatoru može se pratiti stanje vremenskog registra u svakom trenutku. Nakon toga oba vremenska registra će se resetirati pomoću funkcije RSTT.

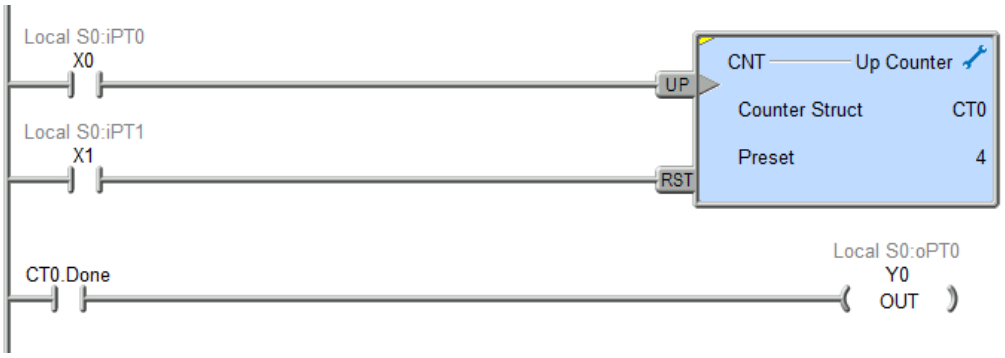


Slika 7.6 Ljestvičasti program transportne trake

7.3 Brojači

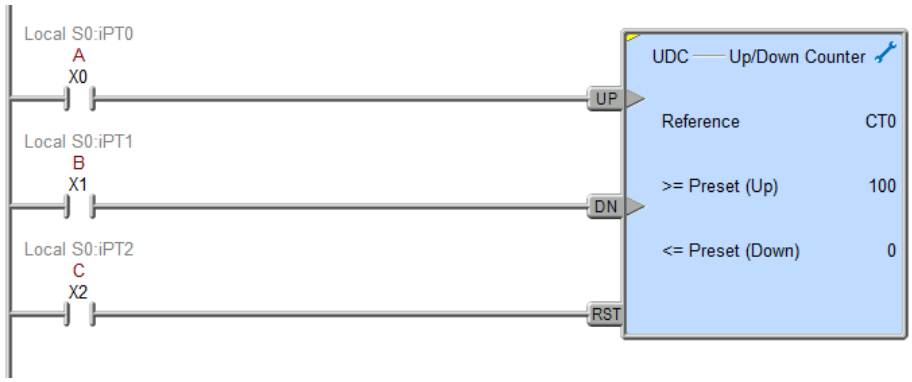
Dva osnovna tipa brojača su uzlazni i silazni. Kada ulaz na uzlaznom brojaču bude istinit vrijednost registra *Accumulator* podiči će se za jedan , i tako sve dok ne dostigne zadanu vrijednost. Silazni brojač smanjivat će vrijednost registra *Accumulator* dok se ne dostigne zadana vrijednost.

Uzlazni brojač CNT naredba (Slika 7.8) zahtijeva prostor u memoriji za pohranu vrijednosti i statusa. Zadana vrijednost u ovom slučaju je 4. Ako ulaz X0 iz neistinitog promijeni status u istinit vrijednost na registru *Accumulator* postat će 3. Ako bi se ulaz X0 isključio i zatim ponovno uključio, vrijednost registra *Accumulator* porasla bi na 4, i *DN* izlaz bi se uključio. Brojanje se može nastaviti i dalje od određene vrijednosti. Ako ulaz X1 postane istinit, iznos registra *Accumulator* postat će nula.



Slika 7.7 Prikaz ljestvičastog dijagrama s brojačem [8]

Uzlazni brojači vrlo su slični silaznima i oba se mogu koristiti na istoj memorijskoj lokaciji za brojače. Primjer ulaza A (Slika 7.9) pokreće naredbu brojanja na *UDC* brojaču CT0. Ulaz B pokreće naredbu silaznog brojanja na istoj lokaciji brojača. Određena vrijednost za brojač je pohranjena za svaku krajnju vrijednost. 100 za uzlazno i 0 za silazno brojanje. Ulaz C reset resetirat će oba brojača.



Slika 7.8 Brojač uzlazno/silazni

8. FUNKCIJE LJESTVIČASTOG DIJAGRAMA

Ulazni i izlazni kontakti ljestvičastog dijagrama dozvoljavaju jednostavne logičke odluke. Funkcije proširuju osnovu ljestvičastog dijagrama tako da dozvoljavaju druge tipove upravljanja. Dodatak brojača i vremenskih registara dozvoljava upravljanje uvjetovano događajima [3,4].

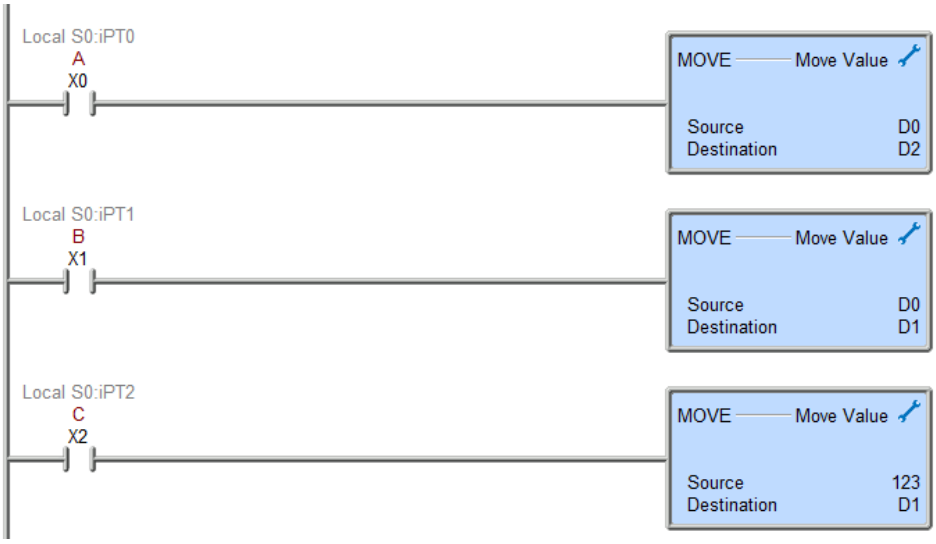
Tipovi funkcija u izvoru autora Hugha Jacka [8] dijeli se na kombinatornu logiku, događaje, upravljanje podataka, Brojevna logika, liste (popisi), upravljanje programom i funkcije ulaza i izlaza.

Većina funkcija koristit će memoriju PLC-a za dohvaćanje vrijednosti, pohranu podataka i praćenje statusa. Neke funkcije kao što su TOF, odnosno OFDTMR vremenski registri, mogu ostati aktivni kada je ulaz isključen. Pojedine funkcije radit će samo kada ulaz s neistinitog postane istinit (*eng. negative edge triggered*), a neke kada se promijeni s pozitivnog na negativan (*eng. pozitive edge triggered*).

8.1 Funkcije upravljanja podacima

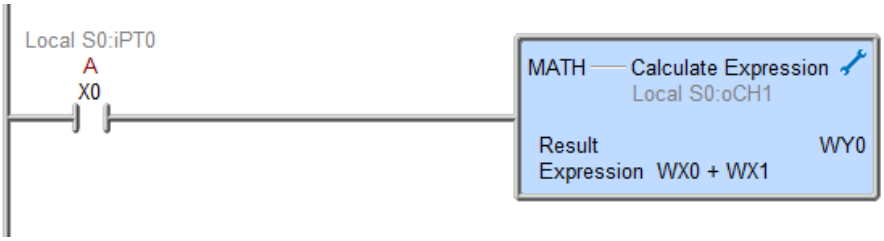
Jedna od funkcija za upravljanje podacima je funkcija za premještanje (*eng. move)*. Dva su osnovna tipa ove funkcije, MOVE (*value, destination*)- premješta vrijednost na memorijsku lokaciju i MVM(*value,mask,destination*)- premješta vrijednost na memorijsku lokaciju, ali s maskom za odabir specifičnih bitova.

Jednostavna MOVE funkcija uzeti će vrijednost iz jedne lokacije u memoriji i pohraniti je na drugoj lokaciji. Na primjeru funkcije (Slika 8.1) kada je A istinito MOVE funkcija pomiče broj s izvora *Source* D0 na destinacijsku adresu *Destination* D2. Podatak u izvornoj adresi ostao je nepromijenjen. Kada je B istinit broj će biti pohranjen na destinacijsku adresu u memoriji. Kada je C istinit broj 123 smjestit će se na adresu D1.



Slika 8.1 MOVE funkcija

Matematičke funkcije vraćaju jednu ili više vrijednosti, izvode operacije i pohranjuju rezultate u memoriji. Funkcija za zbrajanje (Slika 8.2) će dohvatiti vrijednosti *WX0 i WX1*, pretvoriti ih u tip koji je na destinacijskoj adresi, zbrojiti decimalne brojeve i pohraniti rezultat u *WY0*.

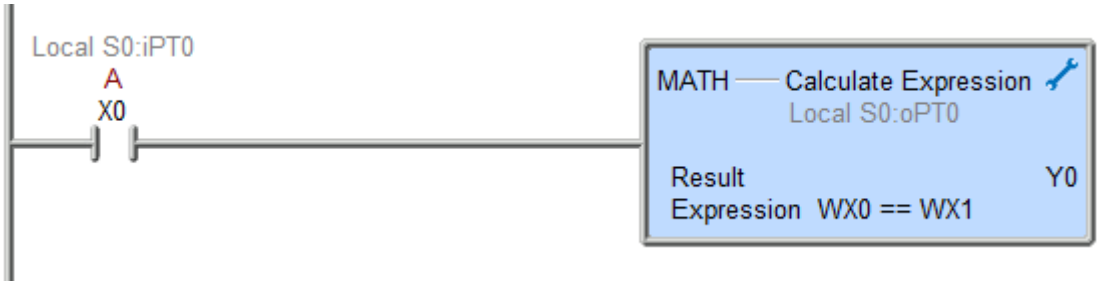


Slika 8.2 Primjer funkcija za zbrajanje

8.2 Logičke funkcije

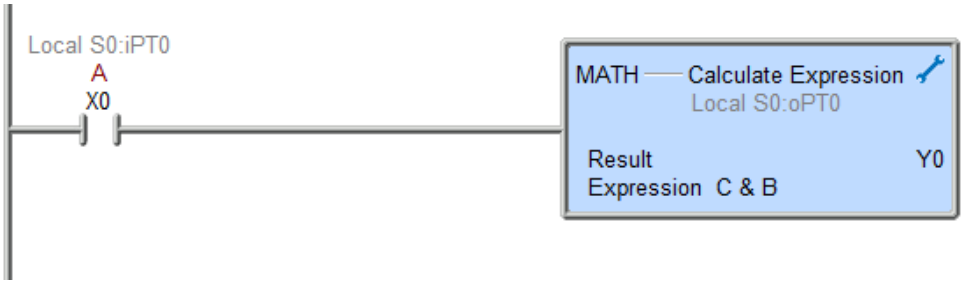
Logičke funkcije mogu biti integrirane u jedinstvenoj funkciji matematičkih operacija kao na primjeru (Slika 8.3).Prikazana je funkcija *MATH* u kojoj se odabere željena operacija.

Primjer pokazuje funkciju za jednakost. Funkcija uspoređuje dva broja. Ako su jednaka izlaz Y0 bit će istinit, u suprotnom će biti neistinit.



Slika 8.3 Primjer funkcije za usporedbu brojeva

Funkcije isto mogu biti operatori Booleove algebre kao što su konjunkcija disjunkcija i negacija.Funkcija AND (Slika 8.4) prihvaća dvije vrijednosti, izvrši operaciju i pohranjuje rezultat na novoj lokaciji.



Slika 8.4 Primjer funkcije logičkog operatora I

9. PROCES PROGRAMIRANJA SUSTAVA UPRAVLJANJA

Kroz prethodna poglavlja analizirani su pristupi osmišljavanja i realiziranja programa upravljanja. Problem u pravilu započinje verbalnim opisom željenog rada sustava. Logika se zatim, pomoću tablica istinitosti ili K- tablica pretvori u Booleovu jednadžbu. Kada se od nje dobije prikladni oblik, može se napisati ljestvičasti dijagram.

9.1 Program dizala s dva kata

Na primjeru dizala s dva kata (prizemlje, prvi kat i drugi kat) nalaze se dvije korisničke ploče. Jedna se nalazi s vanjske, a druga s unutarnje strane dizala. Na svakom katu nalazi se jedno dugme koje, ako se pritisne, šalje dizalu signal. Unutar dizala nalaze se tipke za svaki od navedenih katova. Dizalo sadrži i senzore za položaj i senzore za status vrata. Senzori Položaja koriste se kako bi se dizalo zaustavilo na svakom katu. Zbog jednostavnosti, potrebno je problem podijeliti na ulazne i izlazne jedinice (Tablica 2).

Tablica 2 Ulazi i izlazi programa dizala

Br.	Adresa	Naziv	Ulaz/Izlaz
1	**X0**	start	Ulaz
2	**X1**	stop	Ulaz
3	**Y0**	Start latch	Izlaz
4	**X5**	Tipkalo 0	Ulaz
5	**X6**	Tipkalo 1	Ulaz
6	**X7**	Tipkalo 2	Ulaz
7	**X2**	Limit 0	Ulaz
8	**X3**	Limit1	Ulaz
9	**X4**	Limit2	Ulaz
10	**X8**	Otvorena vrata	Ulaz
11	**X9**	Zatvorena vrata	Ulaz
12	**Y3**	Motor gore	Izlaz
13	**Y4**	Motor dolje	Izlaz
14	**Y1**	Motor otvori vrata	Izlaz
15	**Y2**	Motor zatvori vrata	Izlaz
16	**Y5**	Tipkalo 0 latch	Izlaz
17	**Y6**	Tipkalo 2 latch	Izlaz
18	**Y7**	Tipkalo 3 latch	Izlaz

S obzirom na tablicu, potrebno je napraviti tablicu istinitosti (Tablica 2).S obzirom na velik broj ulaza, kada bi bila potpuna, tablica bi imala prevelik broj redaka. S obzirom na to

da nisu potrebne sve kombinacije ulaza, može se po potrebi skratiti. To se može učiniti pomoću dodatnih logičkih pretpostavki.

Samo jedan od motora gore/dolje može biti istinit u svakom trenutku. Dizalo može biti samo na jednom katu u svakom trenutku, i pretpostavit ćemo, zbog jednostavnosti, da je samo jedno tipkalo pritisnuto u svakom trenutku. Ako je ulaz *stop* 0, svi izlazi su neistiniti. Ulaz za otvorena vrata može biti istinit samo ako je jedan od senzora položaja istinit. Ulazi X0,X5, X6 i X7 označeni su s x jer njihovo trenutno stanje ne utječe direktno na izlaze. Oni predstavljaju tipkala koja , kada se jednom uključe, uključuju izlaze Y5, Y6 i Y7 koji predstavljaju status pritisnutog tipkala. Y0 predstavlja status uključenosti sustava. Problem statusa tipkala riješen je u programskom kodu. Djeluje kao i Start/Stop sklopka.Prema tome, moguće kombinacije ulaza su sljedeće (Tablica 3).

Tablica 3 Tablica istinitosti za ulaze i izlaze dizala

STANJE	X0	X1	X2	X3	X4	X5	X6	X7	X8	X9	Y0	Y1	Y2	Y3	Y4	Y5	Y6	Y7
1.	**x**	**0**	**1**	0	0	x	x	x	0	0	**1**	**1**	0	0	0	0	0	0
2.	x	0	0	**1**	0	x	x	x	0	0	**1**	**1**	0	0	0	0	0	0
3.	x	0	0	0	**1**	x	x	x	0	0	**1**	**1**	0	0	0	0	0	0
4.	x	0	0	**1**	0	x	x	x	0	0	**1**	0	**1**	0	0	**1**	0	0
5.	x	0	0	**1**	0	x	x	x	0	0	**1**	0	**1**	0	0	**1**	0	0
6.	**x**	**0**	**1**	0	0	x	x	x	0	0	**1**	0	**1**	0	0	0	**1**	0
7.	x	0	0	0	**1**	x	x	x	0	0	**1**	0	**1**	0	0	0	**1**	0
8.	**x**	**0**	**1**	0	0	x	x	x	0	0	**1**	0	**1**	0	0	0	0	**1**
9.	x	0	0	**1**	0	x	x	x	0	0	**1**	0	**1**	0	0	0	0	**1**
10.	x	0	0	0	**1**	x	x	x	0	1	**1**	0	0	**1**	0	0	0	0
11.	**x**	**0**	**1**	0	0	x	x	x	0	1	**1**	0	0	**1**	0	0	0	0
12.	**x**	**0**	**1**	0	0	x	x	x	0	1	**1**	0	0	**1**	0	0	0	0
13.	**x**	**0**	**1**	0	0	x	x	x	0	1	**1**	0	0	0	**1**	0	0	0
14.	x	0	0	**1**	0	x	x	x	0	1	**1**	0	0	0	**1**	0	0	0
15.	x	0	0	**1**	0	x	x	x	0	1	**1**	0	0	0	**1**	0	0	0

S obzirom na to da postoji više ulaza, pisanje K-tablica moglo bi biti kompliciranije nego prevođenje Booleovih jednadžbi direktno iz tablice istinitosti. Za svaki izlaz potrebno je napisati izraz koji ih uključuje.

$$Y1= Y0\cdot\overline{Y3}\cdot\overline{Y4}\cdot\overline{X8}\cdot (x2+x3+x4) \quad (4)$$

$$Y2=Y0\cdot\overline{Y3}\cdot\overline{Y4}\cdot\overline{X9}\cdot (Y5\cdot (X3+X4)+Y6\cdot (X2+X4)+Y7\cdot (X2+X3)) \quad (5)$$

$$Y3=Y0\cdot X9\cdot (Y6\cdot\overline{X3}\cdot X4+Y7\cdot\overline{X4}\cdot (X2+X3)) \quad (6)$$

$$Y4=Y0\cdot X9\cdot (Y5\cdot\overline{X2}\cdot (X3+X4)+Y6\cdot\overline{X3}\cdot X2) \quad (7)$$

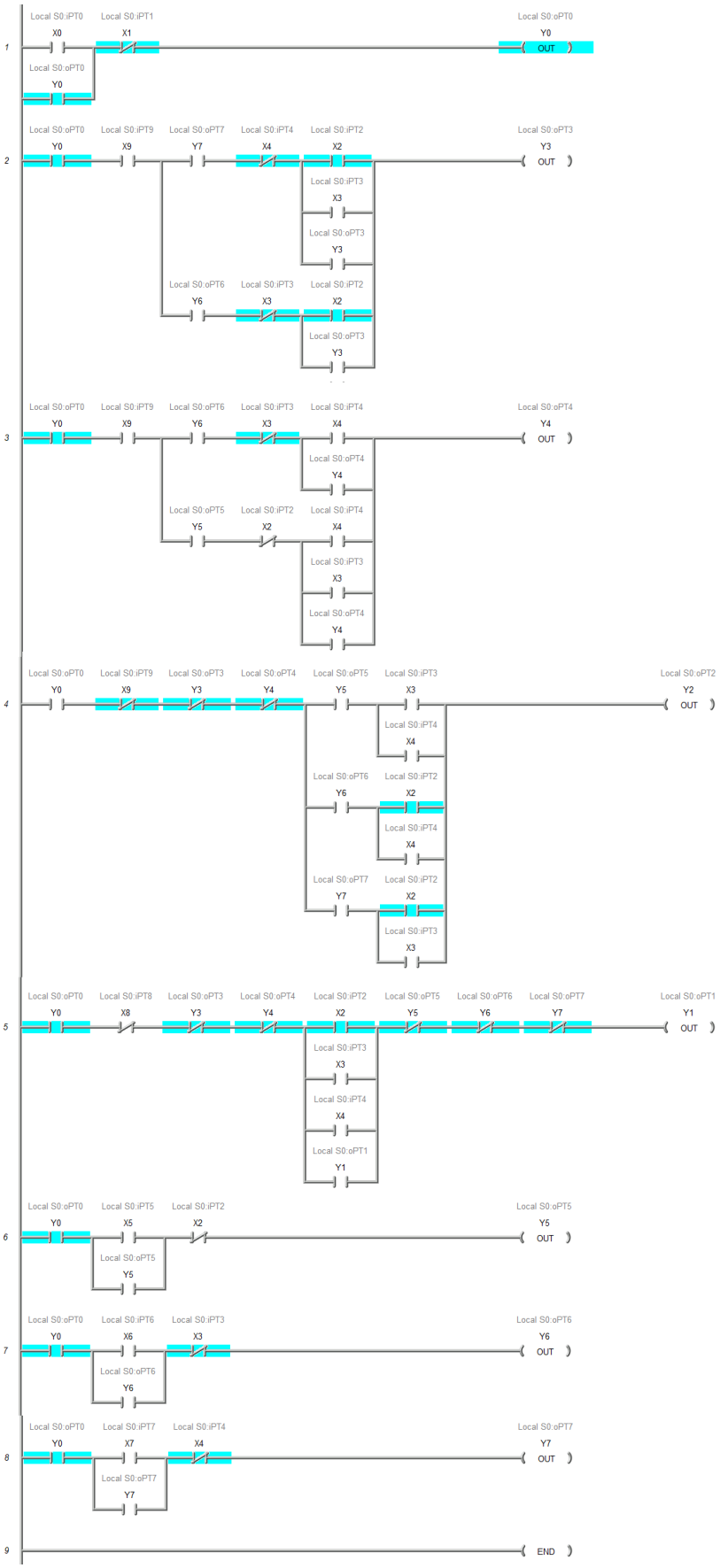
Motor otvori vrata Y1 uključit će se samo ako su *motor gore Y4* i *motor dolje Y3* isključeni i ako je ulaz *vrata otvorena X8* neistinit. Jednadžba (4) to opisuje, kao i to da ako je senzor za poziciju *X2*, *X3* ili *X4* pozitivan. Jednadžba (5) opisuje djelovanje izlaza *motor zatvori vrata Y2*. On će također, biti uključen samo ako su vrata otvorena i *motor gore Y4* i *motor dolje Y3* su isključeni. Jednadžba (6) opisuje *motor gore*. On će se uključiti samo ako su vrata zatvorena, i ovisno o položaju lifta i pritisnutom tipkalu. *Motor dolje* (7) uključit će se samo ako su vrata zatvorena, i ovisno o položaju lifta i pritisnutom tipkalu.Ove jednadžbe samo su teoretski opis logike jednog dizala. U stvarnosti sadrže više varijabli i rad dizala je puno precizniji. Program za upravljanje liftom služi samo kao primjer simuliranja logike simulacijskim alatom.

Iz navedenog se dobivaju izrazi ljestvičastog dijagrama za PLC koji kontrolira dizalom. Dodavanjem pomoćnih kontakata status pritisnutog tipkala postat će pohranjeno kada se otpusti kao i status istinitosti motora dok se ne ostvari predodređeni uvjet.

Opis linija ljestvičastog dijagrama (Slika 9.1):

1. Start/Stop za cijeli sustav. U svakoj daljnjoj liniji status uključenosti bit će istinit samo ako je status Y0 istinit, to jest, ako je dizalo pokrenuto.
2. Djelovanje izlaza *motor gore Y3*. Ako je tipkalo za drugi kat Y7, ili prvi kat Y6 pritisnuto, ovisno o položaju dizala, X2 ili X3, Motor gore Y3 izlaz postat će istinit sve dok pozicija lifta ne bude jednaka katu za kojeg je pritisnuto tipkalo.
3. Djelovanje izlaza *motor dolje Y4*. Isti način kao i izlaz *motor gore* , samo su drugačiji uvjeti.
4. Motor za zatvaranje vrata Y2 uključit će se samo ako vrata nisu zatvorena *X9* , ako su oba motora za pokretanje lifta isključena, i ako je pritisnuto tipkalo za kat na kojem se u tom trenutku lift ne nalazi.

5. Motor za zatvaranje vrata Y1 uključit će se samo ako vrata nisu zatvorena *X8*, ako nije pritisnuto niti jedno tipkalo za kat i ako je lift dosegnuo neki kad. Ako je neki od ulaza X2, X3 ili X4 istinit.
6. Sklop za zaključavanje statusa tipkala. Ako ulaz X5 postane pozitivan i odmah se isključi, Status izlaza Y5 postat će i ostati pozitivan, sve dok dizalo ne dosegne kat X2
7. Sklop za zaključavanje tipkala za prvi kat. Jednak način djelovanja kao u liniji 6.
8. Sklop za zaključavanje tipkala za drugi kat. Jednak način djelovanja kao u linijama 6 i 7.
9. Linija koja označava kraj programa.



Slika 4.1 Program dizala u računalnom programu

10. ZAKLJUČAK

Programiranje PLC-a ili kontrolera procesnih sustava rasprostranjeno je, ali i zahtjevno. Svaki programer, kao i proizvođač koristi svoju metodu, međutim s istim ciljem. U ovom radu opisan je samo teorijski i simulacijski dio tog procesa. Nabrojene su neke od tehnika dobivanja završnog programa. Alati koji su opisani prikladni su za početno saznanje o programiranju PLC strojeva.

Većina upravljačkih procesa koristi programabilne logičke kontrolere kao automatizacijske centre procesa. PLC sadrži programirane funkcije kao što su funkcije vremenskih registara (eng. timers) i brojače, što ga čini sofisticiranim, ali i lakšim za korištenje. Također pruža fleksibilnost upravljanja ovisno o programima i može izvršavati jednostavne kao i kompleksne logičke naredbe koje se koriste u ljestvičastom dijagramu.

Kakva god metoda se koristi, zajednički jezik uvijek je logika. Ljestvičasti dijagrami za to su adekvatan alat i uspješna metoda prenošenja kôda iz logičkih tvrdnji u proizvodni proces (programiranje PLC-a), a simulacijski programi omogućavaju provjeru većeg dijela procesa i bez njihove primjene.

LITERATURA

[1] Wright A. J.: „Ladder Logic Programming Fundamentals“, samostalno objavljeno. 9. rujna 2019.

[2] Bolton W.: „Programmable Logic Controllers“,Newnes, 6. Izdanje, 17. ožujka 2015

[3] Čerina O. „Automatizacija strojarnice za pripremu potrošne tople vode i podnog grijanja s bazenom“, diplomski rad, Tehnički fakultet u rijeci, rujna 2020.

[4] Butković v. „Nadzor elektromotornog pogona“, diplomski rad, Tehnički fakultet u rijeci, rujna 2020.

[5] Krpan D. „Sustav automatizacije vaganja i ukrcavanja čeličnih greda u peć za dogrijavanje u tvornici za valjanje čelika“,diplomski rad, Tehnički fakultet u rijeci, rujna 2020.

[6] Hackworth J. R., Hackworth F.D., Jr: „Programming Methods and Applications“, Prentice Hall, 21. travnja 2003

[7.] Naebi A., Hassanpour Aghdam M., Ghalehban Zanjanab A., Nourani E., Hassanpour Aghdam S. „A New Programming Language for PLC Using Bond Graph“ s Interneta, https://link.springer.com/chapter/10.1007/978-3-642-25553-3_10, 16. rujna 2020.

[8] Jack H.:“Automating Manufacturing Systems with PLCs“, Hugh Jack; 5.2. Izdanje, Lulu.com , travanj 20010.

[9] Petruzella F. D.:“Programmable Logic Controllers“, McGraw-Hill Education; 5. Izdanje, 13. siječanj. 2016.

[10] Langmann R. „The PLC as a Smart Service in Industry 4.0 Production Systems“ s Interneta, https://www.mdpi.com/2076-3417/9/18/3815, 11. rujna 2020.

[11] Llano A., Angulo I., de la Vega D., Marron L. „Virtual PLC Lab Enabled Physical Layer Improvement Proposals for PRIME and G3-PLC Standards“ s Interneta, https://www.mdpi.com/2076-3417/10/5/1777, 11. rujna 2020.

[12.] Allen Bradley ControlLogix 5580 Controllers, s Interneta, https://www.rockwellautomation.com/search/ra-en-

US;keyword=controllogix%25205580;startIndex=0;activeTab=Literature;spellingCorrect=true;facets=;languages=en;locales=en_GLOBAL,en-US;sort=bma;isPLS=false;sessionID=502577fd-0338-d7cc-8197-3a972d9240d1;deepLinking=false , 15. rujna 2020.

[13] Walters E. G., Bryla E. J. „Software Architecture and Framework for Programmable Logic Controllers: A Case Study and Suggestions for Research“ s Interneta, https://www.mdpi.com/2075-1702/4/2/13/htm, 16. rujna 2020.

[14] IEC 61131, American National Standards Institute ANSI, s Interneta, https://ibr.ansi.org/Standards/iec.aspx, 11. rujna 2020.

[15.] Elektromagnetski relej: zređaj, princip rada, s Interneta, https://hr.puntomarinero.com/electromagnetic-relay-device-principle-of/, 15.rujna 2020.

[16] Mytskowski A., Kiederdelewicz A. „ Fractal.Order Water Level Control Basec on PLC: Hardware_In_The_Loop Simulation and Experimental Validation“, s Interneta, https://www.mdpi.com/1996-1073/11/11/2928/htm, 11. rujna 2020.

[17] „Relay logic vs ladder logic“, s interneta, https://ladderlogicworld.com/relay-logic-vs-ladder-logic/, 10. rujna 2020.