

UNIVERZITET U KRAGUJEVCU
MAŠINSKI FAKULTET KRAGUJEVAC

Programabilni logički kontroler SIEMENS SIMATIC S7-200

Seminarski rad iz predmeta
PROJEKTOVANJE SISTEMA AUTOMATSKOG UPRAVLJANJA

Mentor
Dr Milan Matijević, docent

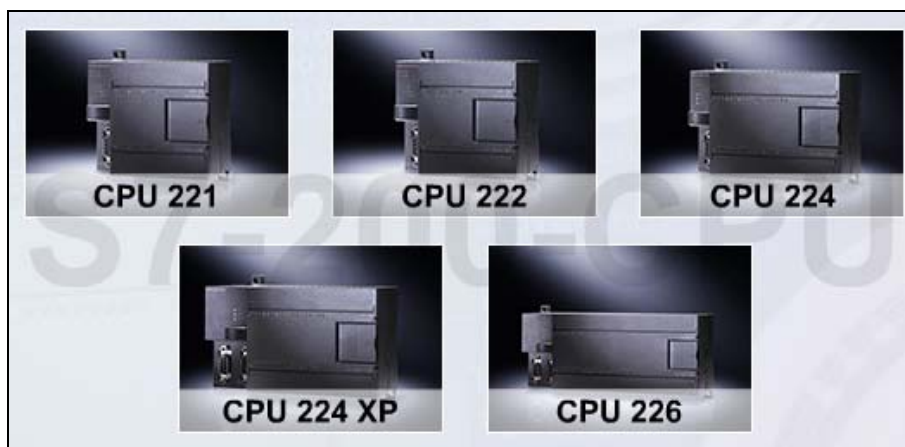
Student
Ivan Dačić, 76/96

SADRŽAJ

| | |
|--|----|
| UVOD | 3 |
| Sistemi upravljanja i PLC | 5 |
| 1.1 Razvoj sistema upravljanja i PLC-a | 5 |
| 1.2 Glavne osobine i prednosti PLC-a | 6 |
| 1.3 Osnovni delovi PLC-a | 9 |
| 1.4 Boole-ova algebra – ‘jezik’ PLC-a | 12 |
| 1.5 Princip rada PLC-a | 13 |
| 1.6 Vreme odziva PLC-a | 15 |
| 1.7 Povezivanje PLC-a sa perifernim uređajima..... | 18 |
| Programiranje PLC-a | 22 |
| 2.1 Sistemski prilaz projektovanju – opšte smernice | 22 |
| 2.2 Vrste programskih editora | 23 |
| 2.3 Razlike između SIMATIC i IEC 1131-3 seta instrukcija | 25 |
| 2.4 Osnovni delovi programa | 26 |
| 2.5 Osnove programskog paketa STEP 7-MicroWIN | 28 |
| 2.6 Set instrukcija S7-200 | 28 |
| 2.6.1 Bit logičke instrukcije | 30 |
| 2.6.2 Satne instrukcije – <i>Clock Instructions</i> | 33 |
| 2.6.3 Instrukcije poređenja | 33 |
| 2.6.4 Brojačke instrukcije | 34 |
| 2.6.5 Matematičke instrukcije | 36 |
| 2.6.6 Instrukcije za upravljanje programom | 37 |
| 2.6.7 Tajmer instrukcije | 38 |
| 2.6.8 Podprogramske instrukcije | 40 |
| REŠENI ZADACI | 41 |
| LITERATURA | 72 |

UVOD

Razvoj novih tehnologija u izradi visoko integrisanih elektronskih sklopova omogućio je da se na tržištu pojavi nova generacija programabilnih automata. Naziv "mikro PLC" dovoljno govori za sebe: minijaturne dimenzije i niska cena. Poslednjih godina većina proizvođača ove vrste opreme tržištu je ponudila svoju verziju mikro-PLC-a, ukazujući na minijaturne dimenzije, niske cene kao i neverovatne brzine izvršavanja instrukcija. U toj trci pomalo je zanemarena činjenica da PLC nije prosta zamena za relejnu tehniku; on može pružiti mnogo više od toga. Vođen takvom idejom, SIEMENS je svojim korisnicima ponudio novu generaciju mikro-PLC-a čime je još jednom postavio nove standarde u oblasti automatizacije. Reč je o familiji programabilnih automata SIMATIC S7-200.



Slika 1. SIEMENS-ova SIMATIC S7-200 familija.

Glavni aduti ove familije, osim izuzetno malih dimenzija i niske cene, jesu lakoća rukovanja, širok spektar integrisanih funkcija za rad u realnom vremenu i gotovo neograničene mogućnosti za komunikaciju. Zahvaljujući ovim osobinama SIMATIC S7-200 veoma lako nalazi primenu u najrazličitijim oblastima automatizacije, bilo kao samostalan kompaktan uređaj, bilo kao grupa umreženih uređaja, ili u sistemima sa decentralizovanom periferijom.

Revolucionarno nov koncept koji forsira SIEMENS, tzv. filozofija TOTALNO INTEGRISANE AUTOMATIZACIJE, termin je koji je SIEMENS prvi put prezentirao u Roterdamu, u novembru 1996. godine. Ideja je da se kompletna automatizacija celokupnog postrojenja izvede integralno sa hardverskim i softverskim modularnim komponentama jednog sistema, u ovom slučaju SIMATIC-a. Da bi se istakla kompatibilnost, tj. jedinstvenost u konfiguraciji, obradi podataka i komunikaciji, pojmom SIMATIC, koji je do sada bio sinonim za u svetu najviše zastupljen industrijski programabilni logički kontroler - PLC (28% učešća u svetu), obuhvaćene su i ostale tipične komponente jednog sistema za upravljanje industrijskim procesom. U TOTALNO INTEGRISANU AUTOMATIZACIJU spadaju tako:

- industrijski PLC-ovi (SIMATIC Controller),
- industrijske računarske mreže (SIMATIC NET),
- industrijski personalni računari (SIMATIC PC),
- uređaji za vizuelizaciju (SIMATIC HMI),
- decentralne periferije (SIMATIC DP),
- softver (SIMATIC Industrial Software),
- sistem za vođenje procesa (SIMATIC PCS7).

Jezgro TOTALNO INTEGRISANE AUTOMATIZACIJE je softver. Kompletan savremen industrijski softver pre svega mora da bude u skladu sa svetskim standardima (SIEMENS-ov industrijski softver zadovoljava npr. IEC 1131 i DIN EN 6.1131-3, a SIEMENS je i član udruženja PLCopen). Na raspolaganju su najpre standardni programski alati STEP7, STEP7 mini i STEP7 micro. Oni omogućavaju kompletno programiranje, pre svega u vidu ladder dijagrama – koji najviše odgovaraju načinu funkcionisanja programabilnih kontrolera, ili pak u vidu statement listi (nivo blizak assembleru) itd. Ali, ono što štedi inženjersko vreme i napor, i omogućava korisniku da se koncentriše zaista na projektni zadatak, jesu inženjerski alati. Oni "govore" inženjerskim jezikom, pomoću njih se program smanjuje i čini preglednijim, olakšava se testiranje i puštanje u rad.

Glavni cilj ovog rada je upoznavanje sa osnovnim karakteristikama i funkcijama SIEMENS-ovog PLC-a iz familije SIMATIC S7-200, i programom za njegovu kontrolu i upravljanje STEP 7-MicroWIN. Mada se mnoge karakteristike odnose na PLC kontrolere uopšte, dalji tekst će se uglavnom odnositi na SIEMENS-ove kontrolere iz familije SIMATIC S7-200. Na primeru rešenih zadataka biće pokazano programiranje kontrolera uz pomoć *ladder* dijagrama.

Sistemi upravljanja i PLC

1.1 Razvoj sistema upravljanja i PLC-a

Skup međusobno povezanih elemenata, komponenti, i/ili podsistema, među kojima postoji razmena informacija i koji se nalaze u interakciji čineći novu funkcionalnu celinu, jeste **sistem**. Sistem predstavlja više od skupa komponenti koje ga čine, i neke od osobina njegovih komponenti neće biti manifestovane u okviru njega.

Koncept sistema je, iako sam po sebi neprecizan jer se kompletan sistem ili delovi sistema mogu posmatrati kao podsistemi drugog sistema, glavno polje interesovanja i delovanja u inženjeringu. Sposobnosti inženjera da osmisli i konstruiše uređaje, opremu i sredstva za proizvodnju, i poveže ih u efikasnu celinu – sistem, dolaze do izražaja u disciplini sistemskog inženjeringa.

Istorijski razvoj sistema upravljanja podrazumeva evoluciju samog koncepta, razvoj mernih uređaja i računarskog hardvera, evoluciju alata matematičke analize, i tehnologija koje dopuštaju bržu konstrukciju sve preciznijih i složenijih sistema.

Razvoj sistema upravljanja do neposredno posle II svetskog rata dao je najviše rezultata u teorijskom domenu i u analizi i sintezi SISO sistema (Single Input Single Output), a danas su te teorije opšte poznate pod terminom *klasična teorija upravljanja*. Teorijski razvoj 50-tih i 60-tih godina XX veka obeležila je pojava *moderne teorije upravljanja*, koja analizu i sintezu sistema, iz prethodnog perioda, pretežno baziranu na grafoanalitičkim metodama u kompleksnom i frekvencijskom domenu, prenosi u prostor stanja. Ovakav pristup omogućava jednak tretman multivarijabilnih (MIMO) i SISO sistema upravljanja, definisanje novih osobina sistema (kontrolabilnost, opservabilnost), i razvoj novih upravljačkih metoda, koje se nisu mogle postići klasičnom teorijom upravljanja.

Uporedo sa razvojem moderne teorije upravljanja počinje i razvoj *teorije digitalnih sistema upravljanja*. Tako se javlja i tendencija primene digitalnih računara u upravljanju procesa. Prvi računarski sistemi namenski proizvedeni za upravljanje procesima, implementacija u industriji i upravljanje industrijskim postrojenjima vezani su za 50-te godine prošlog veka.

Međutim, tadašnji računarski sistemi su se bazirali na elektronskim cevima, imali su veliko vreme za obavljanje osnovnih operacija i kratko srednje vreme između dva otkaza centralnog procesora. Već tada su mogle da se naslute glavne prednosti digitalne tehnologije i računarskih sistema:

- visoka cena početnih ulaganja, u slučaju proširenja proizvodnih kapaciteta se višestruko smanjivala,
- fleksibilnost prilikom prebacivanja jednostavnim reprogramiranjem bila je neuporediva sa analognim sistemima.

Razvoj računara povukao je za sobom i razvoj software-skih rešenja i razvoj specijalizovanih računarskih sistema za procesnu industriju i konsekvntno pojavu industrijskih *miniračunara*, sredinom 60-tih godina, i *mikroprocesora*, početkom 70-tih godina. Zahtevi za sve većom pouzdanošću, većom brzinom rada i niskom cenom doveli su do brzog razvoja ovog segmenta industrije i velikim promenama u sistemima za upravljanje.

Dalji tehnički razvoj računarskih sredstava automatike bio je usmeren ka razvoju PLC-ova (programabilnih logičkih kontrolera) prvobitno namenjenih za zamenu logičkih kola i sekvencijalnih elemenata koji su bili realizovani pomoću banke releja, tajmera, brojača i drugih hardverskih digitalnih komponenti.

1.2 Glavne osobine i prednosti PLC-a

PLC je projektovan za rad u nepovoljno klimo-tehničkim uslovima koji vladaju u industrijskim postrojenjima. Nekada ograničen na ON-OFF upravljanje, intenzivnim razvojem omogućeno je upravljanje veoma složenih aplikacija. Prema tipu se mogu izdvojiti neke od sledećih vrsta PLC kontrolera:

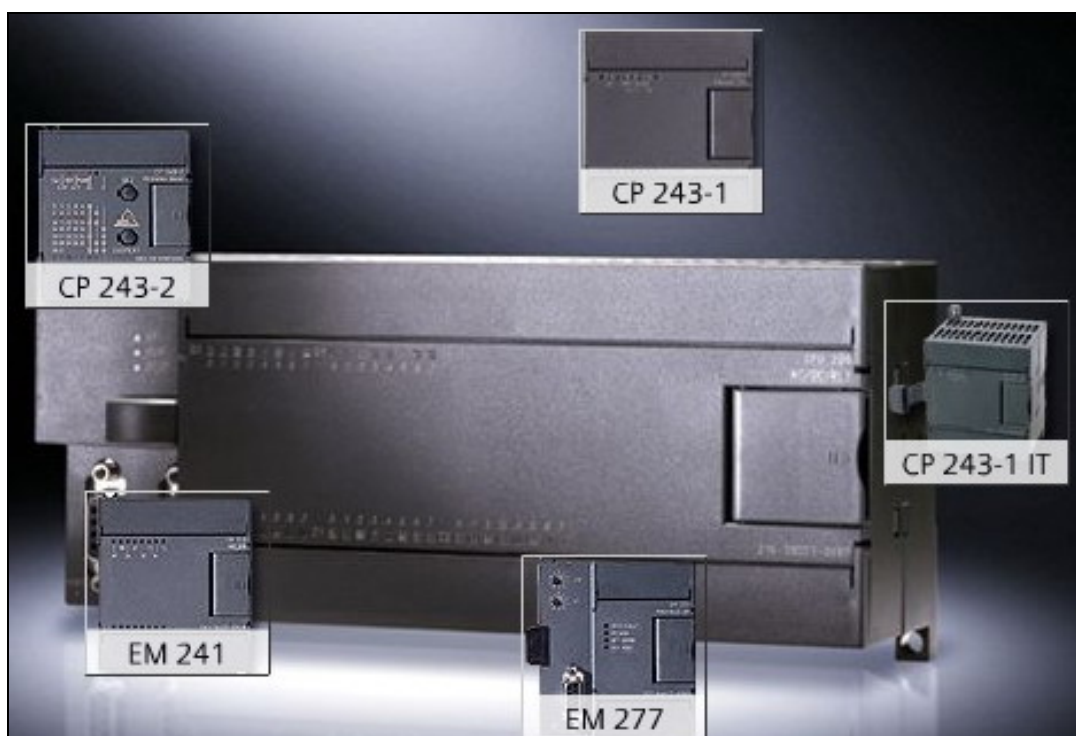
- kontinualni,
- pulsni i
- step kontroleri.

PLC nije zamišljen kao računar opšte namene, već kao sistem čiji operativni sistem omogućava da se jednostavno i u realnom vremenu obavi akvizicija velikog broja podataka, izvesna, ne preterano složena obrada tih podataka i prenošenje rezultata obrade na aktuatore. Neke od glavnih prednosti PLC-ova nad klasičnim relejnim sistemima upravljanja su:

- ožičenje sistema se smanjuje za 80% upoređujući sa konvencionalnim relejnim sistemom upravljanja,
- potrošnja struje koja je potrebna za upravljanje se uveliko smanjuje,
- funkcije PLC za samodijagnosticiranje omogućavaju brzo otklanjanje grešaka na sistemu,
- ispravljanje upravljačke sekvence ili aplikacije se lako radi ponovnim programiranjem na konzoli ili kompjuterskom softveru bez promena ožičenja,
- upotrebom PLC-a smanjuje se potreba za relejima i hardverskim tajmerima,
- vreme ciklusa na mašinama se znatno smanjilo zbog brzine PLC-a, znači, povećava se produktivnost,
- košta mnogo manje u poređenju sa konvencionalnim sistemima u situacijama gde je broj ulaza/izlaza veoma veliki i gde su upravljačke funkcije složene,

- pouzdanost PLC-a je veća nego kod mehaničkih releja i tajmera.

PLC je zamišljen kao modularan sistem, na koji se, prema potrebi, mogu priključiti raznovrsni ulazno/izlazni moduli. Vremenom se gama U/I modula širila tako da su formirani specijalizovani merno-pretvarački moduli u kojima se merni signali obrađuju na izuzetno složen način, kao i izlazni moduli koji sadrže regulatore pojedinih aktuatora.



Slika 2. Različiti komunikacioni moduli.

Pored jako velike osnovne funkcionalnosti pet različitih CPU-a (karakterišu ih različit broj integrisanih ulazno-izlaznih linija), modularan dizajn SIMATIC S7-200 nudi širok spektar, kao opštih, tako i usko specijalizovanih modula za proširenje (preko 20 različitih digitalnih, analognih i komunikacionih modula) za obavljanje najrazličitijih funkcija. Dodatna fleksibilnost ogleda se i u mogućnostima priključenja na različite napone napajanja.

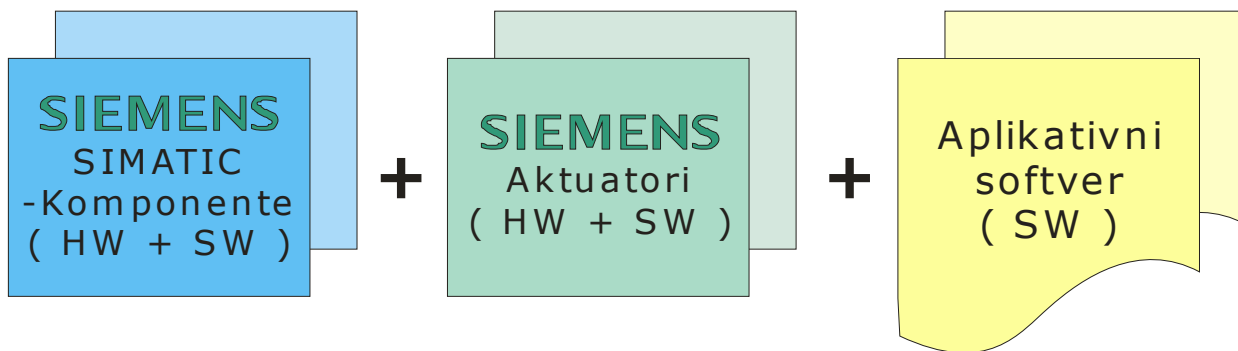


Slika 3. Mogućnosti proširenja PLC-a.

Programiranje je takođe krajnje pojednostavljeno: softverski paket STEP7 Micro/WIN koristi sve pogodnosti koje pruža operativni sistem Windows i zbog svoje jednostavnosti nije izgubio sveobuhvatnost i široku primenu i kod usko specijalizovanih zahteva.

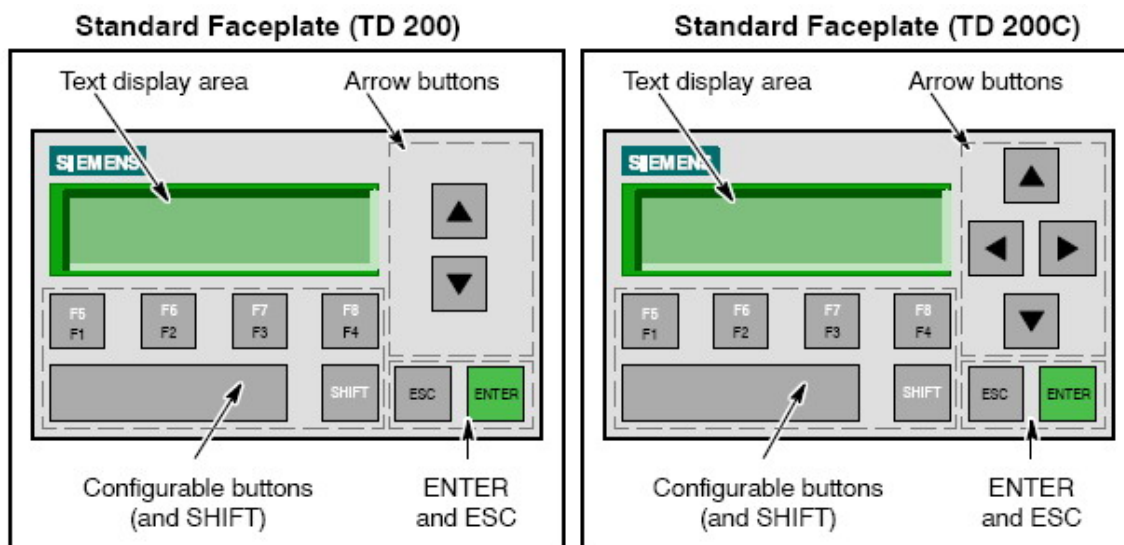
Relativno sužen skup instrukcija biran je tako da bude pregledan i lak za učenje, ali da se istovremeno sa lakoćom mogu realizovati kompleksne funkcije kao što su rad sa tabelama, komunikacija i razmena podataka preko mreže, aritmetika sa tekućim zarezom ili PID-algoritam. Ovakvim pristupom SIEMENS se približio krajnjem korisniku, kako početnicima, tako i onim naprednijim.

Celokupno rešenje koje pruža SIEMENS ogleda se u hardveru, i softveru koji je implementiran u taj hardver, kao i u softveru koji sami razvijamo, a koji 'ubacujemo' u taj hardver da bi ceo sistem odgovorio specifičnim zahtevima koje postavljamo pred proizvodni proces.



Slika 4. Osnovni princip rešavanja aplikacije

Projektovanje tekst-displeja TD 200 je takode integrisano u softverski paket STEP7 Micro/WIN. Pomoću ovog kompaktnog operator panela mogu se ispisivati tekstualne poruke ili podešavati razni parametri sistema.



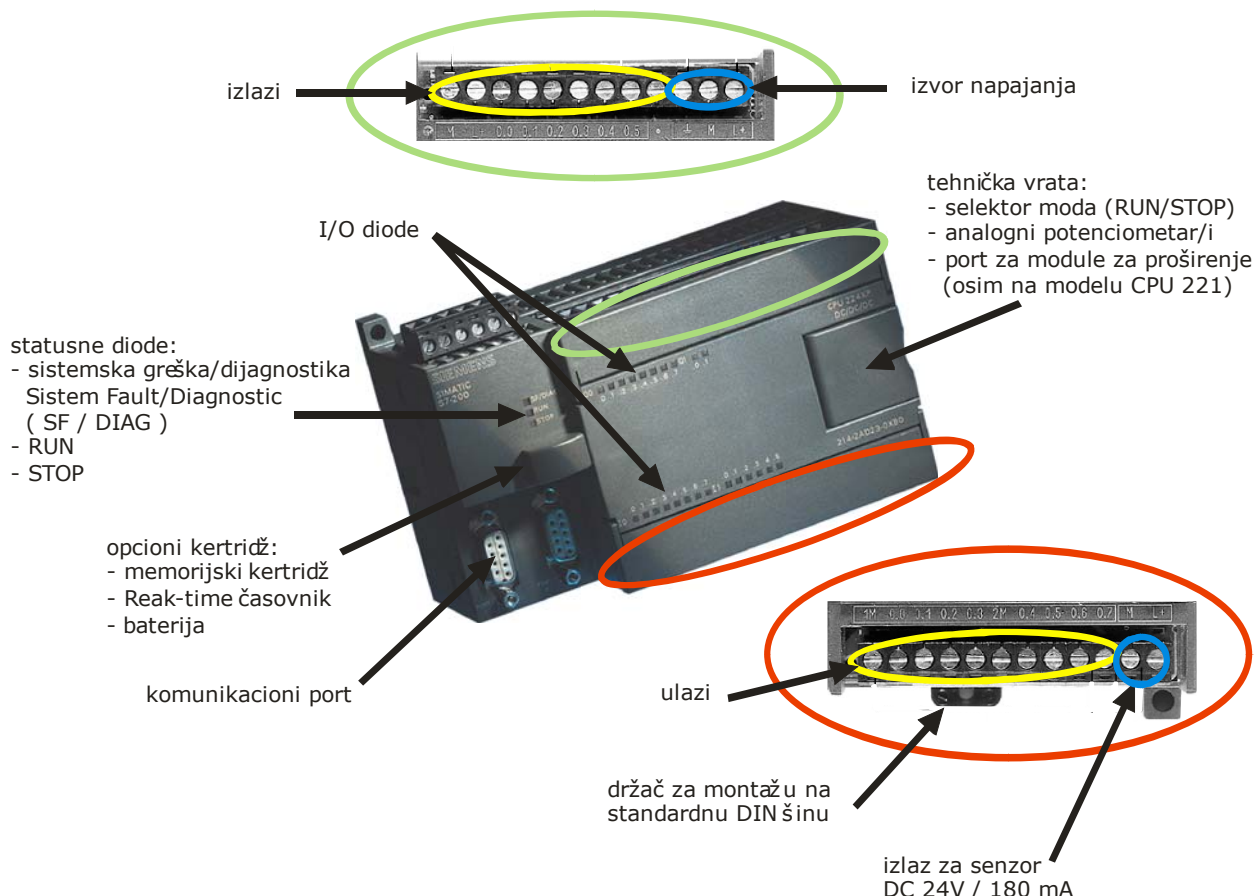
Slika 5. Tekst displej TD 200 i TD 200C

1.3 Osnivni delovi PLC-a

Glavni delovi S7-200, i mikro PLC-a uopšte, su:

- mikroprocesor - CPU,
- integrisani izvor napajanja,
- ulazna kola i
- izlazna kola

spakovani u kompaktnom kućištu malih dimenzija.



Slika 6. Delovi mikro PLC-a

CPU je mikroprocesor koji koordiniše aktivnosti PLC sistema. On izvršava program, obrađuje ulazno/izlazne signale i komunicira sa spoljnim uređajima. S7-200 smešta informacije u različite memorijske lokacije koje imaju jedinstvene adrese.

Postoje različiti tipovi memorijskih jedinica:

- ROM - Read Only Memory – memorija koja se može programirati samo jedanput,
- RAM – Random Access Memory – u nju se smešta program i može biti izbrisana po isključenju struje,
- EPROM – Erasable Programable ROM, slično kao ROM, samo što se njen sadržaj može isprazniti podvrgavanjem UV svetlosti,

- EEPROM – Electrically EPROM - kombinuje fleksibilnost RAM memorije i nemogućnost brisanja EPROM-a a njen sadržaj može biti izbrisan i reprogramiran elektronski, ograničen broj puta (minimum 100.000 puta, najčešće 10^6 puta) pa treba voditi računa da se samo bitne vrednosti upisuju.

Navođenjem memorijske adrese omogućen je direktan pristup podacima. Da bi se pristupilo bitu u delu memorije, navodi se adresa, koja sadrži identifikator memorijskog mesta, adresu bajta i broj bita. Pristupanje bitu memorije naziva se i 'byte.bit' adresiranje. U sledećem primeru memorijsko mesto i adresu bajta (I=input, 3=byte 3) od adrese bita (bit 4) odvaja tačka ('.').



Slika 7. Adresiranje 'byte.bit'

Svakom ulaznom/izlaznom uređaju se dodeljuje adresa. Na ovaj način se rezerviše mesto gde će biti pohranjena logička vrednost koja definiše stanje tog uređaja (1 kada je uređaj uključen, a 0 kada je isključen).

Na početku svakog ciklusa skeniranja, o kome će kasnije biti reči, S7-200 uzorkuje fizičko mesto ulaza i zapisuje ove vrednosti u ulazni registar. Ulaznom registru se pripisuje slovna oznaka 'I' i pristupa sa: *I[adresa bajta].[adresa bita]*, npr. I0.4.

Na kraju svakog ciklusa skeniranja, S7-200 kopira vrednosti smeštene u izlaznom registru na fizičko mesto izlaza. Izlaznom registru se pripisuje slovna oznaka 'Q' i pristupa sa: *Q[adresa bajta].[adresa bita]*, npr. Q1.1.

Slovne oznake za ostale memorijske lokacije su:

- **V** za promenljivu memoriju; skladišti posredne podatke operacija samog programa na kontroleru,
- **M** za bit memoriju; koristi se kao kontrolni relej za skladištenje posrednog stanja operacija ili drugih kontrolnih informacija,
- **T** za memoriju tajmera; S7-200 obezbeđuje tajmere koji broje u inkrementima vremena u rezolucijama od 1 ms, 10 ms ili 100 ms. Sa tajmerom su povezane dve promenljive: trenutna vrednost (16-to bitna vrednost koja čuva odbrojano vreme) i bit tajmera (ovaj se bit setuje ili čisti kao rezultat poređenja trenutne vrednosti i unapred zadate vrednosti),

- **C** za memoriju brojača; S7-200 obezbeđuje tri vrste brojača koji broje svaki niže-na-više (low-to-high) prelazni događaj na ulazu brojača: jedan broji samo na više, drugi na niže, i treći broji i na više i na niže. Postoje dve promenljive, a analogija je ista kao i kod tajmera,
- **HC** za brzi brojač; broji brze događaje nesavisno od ciklusa skeniranja,
- **AC** za akumulatore; akumulatori su čitaj/piši uređaji koji se mogu koristiti kao memorija,
- **L** za lokalnu memoriju; lokalna memorija je slična V memoriji sa razlikom da je V memorija globalna (može joj se pristupiti iz bilo kog dela programa: glavnog programa, podprograma ili rutine prekida),
- **AI** za analogne ulaze; analogne ulaze (temperatura, napon...) S7-200 pretvara u 16-to bitne digitalne reči;
- **AQ** za analogne izlaze; S7-200 pretvara 16-to bitne digitalne vrednosti u struju ili napon, proporcionalan digitalnoj vrednosti;
- **S** za memoriju sekvencijalnog kontrolnog releja (SCR); SCR ili S bitovi se koriste da organizuju operacije mašine ili korake u jednake programske segmente.

U sledećoj tabeli je dat uporedni prikaz SIEMENS-ovih PLC-a iz familije S7-200 sa brojem ulaza/izlaza i veličinom memorije.

| KARAKTERISTIKE | CPU 221 | CPU 222 | CPU 224 | CPU 224XP | CPU 226 |
|--------------------------------------|---------|---------|---------|-----------|---------|
| Digitalni ulazi | 6 | 8 | 14 | 14 | 24 |
| Digitalni izlazi | 4 | 6 | 10 | 10 | 16 |
| Maks. broj digitalnih ulaza i izlaza | 10 | 78 | 168 | 168 | 248 |
| Analogni izlazi | 0 | 8 | 28 | 30 | 28 |
| Analogni ulazi | 0 | 4 | 14 | 15 | 14 |
| Maks. broj analognih ulaza i izlaza | 0 | 10 | 35 | 38 | 35 |
| Programska memorija | 4 | 4 | 8 / 12 | 12 / 16 | 16 / 24 |
| Memorija podataka | 2 | 2 | 8 | 10 | 10 |

Tabela 1. SIEMENS-ova familija S7-200

1.4 Boole-ova algebra - 'jezik' PLC-a

Matematička osnova za sekvencijalno upravljanje je **Booleova algebra**. Ako logički zahtevi koji se postavljaju mogu biti specificirani u formi Booleovih jednačina tada programabilni logički kontroler (PLC) postaje računar koji rešava Booleove jednačine. Prema IEC (International Electrotechnical Commission) PLC se definiše kao: "... digitalni elektronski sistem, projektovan da se koristi u industrijskom okruženju koji upotrebljava programabilnu memoriju za interno memorisanje, uskladištenje korisnički orijentisanih instrukcija za implementiranje specifičnih funkcija takvih kao što su: logičke, sekvencijalne, brojačke, tajming i aritmetičke do upravljačkih, kroz digitalne ili analogne ulaze za upravljanje raznih tipova mašina i procesa. PLC i njemu pridruženi periferni uređaji projektovani su tako da budu lako ugradivi u industrijski sistem upravljanja i da omogućavaju laku upotrebu svojih funkcija."

Najrasprostranjenije programiranje PLC-a je pomoću **ladder** dijagrama, koji se potom prevode u odgovarajuće Boole-ove jednačine, koje služe za izvršavanje zadatih logičkih funkcija. Zbog toga je bitno podsetiti se osnovnih teorema i logike same Booleove algebre.

Osnovne teoreme Booleove algebre:

$$A + B = B + A$$

$$AB = BA$$

$$A + 0 = A$$

$$A \cdot 0 = 0$$

$$A + 1 = 1$$

$$A \cdot 1 = A$$

$$AA = A$$

$$A + \bar{A} = 1$$

$$A\bar{A} = 0$$

$$\overline{\overline{A}} = A$$

$$(A + B) + C = A + (B + C)$$

$$(AB)C = A(BC)$$

$$(A + B)(A + C) = A + BC$$

$$A + AB = A$$

$$A + \bar{A}B = A + B$$

$$A(A + B) = A$$

$$\overline{AB} = \bar{A} + \bar{B}$$

$$\overline{A + B} = \bar{A} \cdot \bar{B}$$

Promenljive u Booleovoj algebri mogu imati samo vrednosti '0' i '1'. Logičke funkcije koje sadrže operacije AND, OR, EXCLUSIVE OR, NOT, NAND, NOR, itd. lako se izražavaju korišćenjem Booleove algebre i pomoću njih mogu biti formirane jednačine koje sadrže Booleove promenljive i operacije. Ako prekidačko-logičke funkcije treba da budu ostvarivane PLC-om, one mogu biti izražene u formi Booleovih jednačina, a digitalni računar unutar upravljačke jedinice PLC može biti upotrebljen da reši ove jednačine u većini slučajeva na isti način kao i digitalni računar opšte namene kada rešava algebarske jednačine. Vrednosti nepoznatih promenljivih se mogu odrediti iz vrednosti poznatih.

Razmotrimo primer dva prekidača PR1 i PR2, čije su promenljive stanja određene stanjem (ON=1, OFF=0) prekidača koji su povezani na PLC. LAMP je promenljiva čija će vrednost biti upotrebljena da odredi stanje

indikatora sijalice koja je spojena na PLC. Ako je odnos između ovih promenljivih određen jednačinom:

$$LAMP = PR1 + PR2$$

onda se vrednost nepoznate promenljive LAMP u bilo kom trenutku tačno određuje preko nepoznatih promenljivih PR1 i PR2 saglasno OR (ILI) funkciji. Prethodna jednačina uspostavlja neprekidan odnos između promenljivih LAMP, PR1 i PR2. Neka sada bude kompleksniji slučaj:

$$LAMP = (LAMP + PR1) \cdot \overline{PR2}$$

Stanje izraza LAMP koje je određeno prethodnom relacijom dato je u tabeli 2.

| PR1 | PR2 | LAMP |
|-----|-----|------|
| 0 | 0 | LAMP |
| 0 | 1 | 0 |
| 1 | 0 | 1 |
| 1 | 1 | 0 |

Tabela 2. Stanje izlaza LAMP

Prva kombinacija je najinteresantnija zbog toga što ona pokazuje da kada su oba ulaza 0 (OFF), sijalica ostaje u svom prethodnom stanju koje može biti 0 ili 1.

1.5 Princip rada PLC-a

Osnovni princip rada S7-200 je veoma jednostavan:

- S7-200 čita stanje ulaza,
- program smešten u memoriji S7-200 koristi ove ulaze da obradi kontrolnu logiku. Dok program radi, S7-200 ažurira podatke,
- S7-200 upisuje podatke na izlaze.

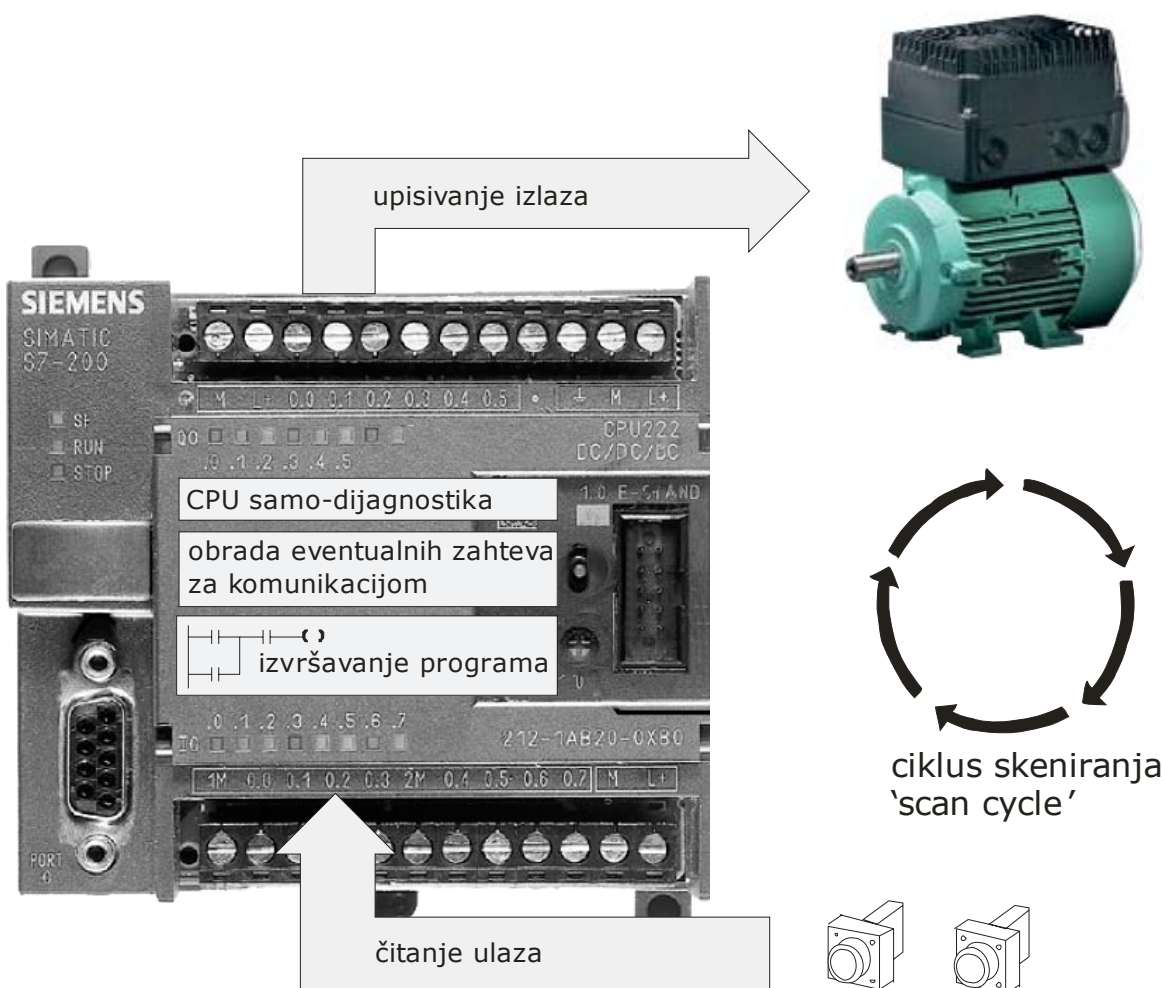
S7-200 ponavlja izvršenje niza zadataka. Ovo ciklično izvršavanje zadataka zove se *ciklus skeniranja* (eng. scan cycle). Kako je pokazano na slici 8, S7-200 obavlja sledeće zadatke u ciklusu skeniranja:

- čitanje ulaza: S7-200 kopira stanje ulaza na ulazni registar,
- izvršavanje programa: S7-200 izvršava instrukcije u programu i smešta vrednosti u različite memorijske lokacije,
- obrada eventualnih zahteva za komunikacijom: S7-200 obavlja zahtevane zadatke za komunikacijom, ako ih ima,
- izvršavanje CPU samo-dijagnostike: S7-200 proverava da 'firmware', programska memorija i moduli za proširenje rade ispravno,

- upisivanje izlaza: vrednosti sačuvane u izlaznom registru upisuju se na izlaze.

Po završetku upisivanja izlaza, PLC se vraća na prvi korak (čitanje ulaza) itd. **Vreme skeniranja** (eng. scan time) se definiše kao vreme potrebno za izvršavanje ovih koraka. Ono je promenljivo, zavisi od broja ulaza i izlaza, dužine programa. Važno je imati na umu da vreme skeniranja nije isto što i perioda odabiranja!!!

Izvršavanje korisničkog programa zavisi od toga da li je S7-200 u 'STOP' ili 'RUN' modu. U 'RUN' modu, program se izvršava, dok se u 'STOP' modu ne izvršava.



Slika 8. Ciklus skeniranja – 'scan cycle'.

Svaki ciklus skeniranja počinje čitanjem trenutnih vrednosti digitalnih ulaza i zatim se nastavlja upisivanjem vrednosti u ulazni registar. Analogni ulazi sa modula za proširenje se, međutim, ne čitaju u normalnom ciklusu skeniranja, osim ako je filtriranje analognih ulaza uključeno. Analogni filter se uvodi da bi se dobili stabilniji signali. Kada je filtriranje analognih ulaza uključeno, S7-200 ažurira taj analogni ulaz jedanput u cokusu skeniranja,

vrši filtriranje, i skladišti filtrirane vrednosti interno. Filtrirane vrednosti onda snabdevaju program svaki put kada on pristupa analognom ulazu.

Tokom izvršavanja programa u ciklusu skeniranja, S7-200 izvršava program, počev od prve instrukcije i kreće se dalje ka krajnjoj instrukciji. Direktne I/O instrukcije (Input/Output) daju trenutni pristup ulazima i izlazima u toku izvršavanja bilo programa, bilo rutine prekida. Rutine prekida, povezane sa događajima prekida, čuvaju se kao deo programa, i ne izvršavaju se kao deo normalnog ciklusa skeniranja, već onda kada nastupi događaj prekida (može se desiti bilo kad u toku ciklusa skeniranja).

Neke od naprednih funkcija i mogućnosti SIEMENS-ovog S7-200 koje neće biti dublje objašnjene, biće samo navedene. S7-200 omogućava korisniku ili korisnikovom programu:

- momentalan prekid čitanja/upisivanja ulaza/izlaza,
- prekid normalnog ciklusa skeniranja sa pojavom događaja prekida,
- određivanje vremena obrade za editovanje 'RUN' moda i status izvršavanja,
- zadavanje stanja digitalnih izlaza za 'STOP' mod,
- zadavanje vrednosti analognih izlaza posle prelaska RUN-u-STOP ili čuvanje izlaznih vrednosti koje su postojale pre prelaska u STOP mod,
- definisanje memorije koja će biti sačuvana u slučaju gubitka napajanja,
- filtriranje analognih/digitalnih ulaza,
- 'hvatanje' kratkotrajnih pulseva,
- LED diodu kontrolisanu od strane korisnika,
- čuvanje istorije važnijih CPU događaja,
- proširenje korisniku dostupne memorije,
- zaštitu funkcija i memorije šifrom,
- podešavanje vrednosti sačuvanih u specijalnoj memoriji analognim potencijometrom,
- brze ulaze/izlaze (brojače i pulsne izlaze).

1.6 Vreme odziva PLC-a

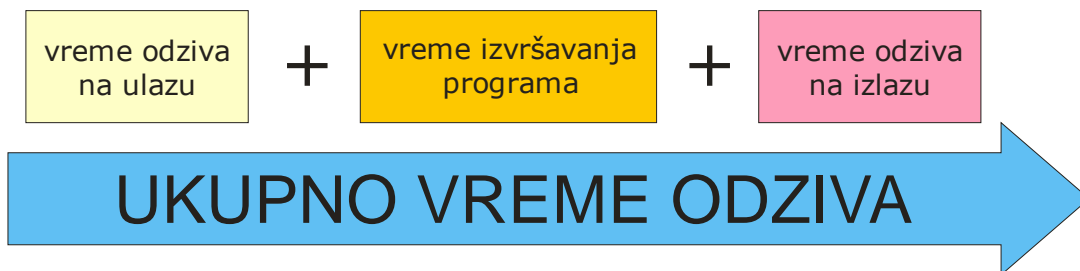
Ukupno vreme odziva je bitan faktor pri izboru PLC-a. PLC-u je potrebno određeno vreme za reagovanje na promene. U velikom broju slučajeva brzina nije bitna, ali postoje i polja primene PLC-a gde je brzina vrlo bitan faktor.

Vreme odziva na ulazu, vreme potrebno PLC-u da izvrši program i vreme odziva na izlazu daju ukupno vreme odziva PLC-a.

PLC može da "vidi" promene (uključivanje/isključivanje) na ulazu samo kada proverava status na ulazu (prvi korak u skeniranju). Izuzeci od ovog pravila su navedeni u prethodnom poglavlju.

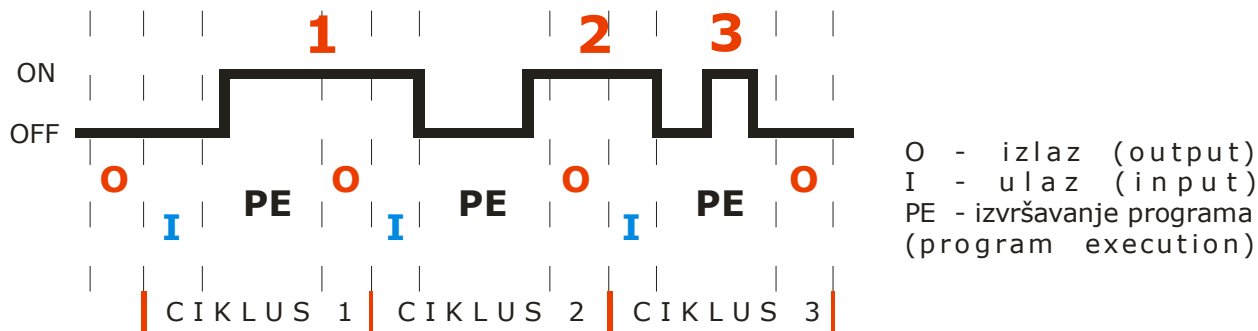
Na slici 10, dat je dijagram koji vremenski prikazuje tri ulaza koji se pojavljuju u različitim segmentima ciklusa skeniranja i traju različito vreme.

Ulaz_1 se ne vidi do drugog ciklusa skeniranja jer kada je ulaz_1 uključen, prvi ciklus skeniranja je završio sa skeniranjem ulaza.



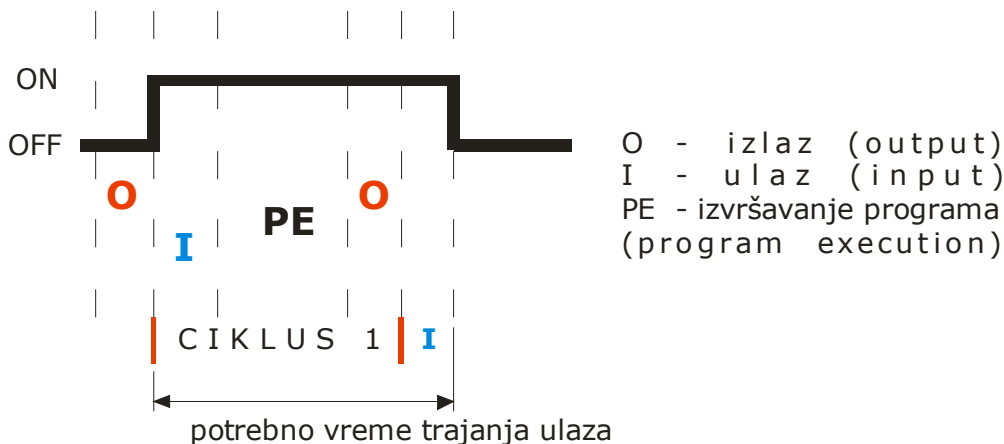
Slika 9. Ukupno vreme odziva.

Ulaz_2 se ne vidi do trećeg ciklusa skeniranja iz istog razloga. Ulaz_3 se uopšte neće videti, jer kada su u trećem ciklusu skeniranja skenirani ulazi, ulaz_3 još nije bio uključen (nije postojao). On se pali u trenutku kada se izvršava program, i završava pre četvrtog ciklusa skeniranja ulaza, tako da ga ni četvrti ciklus čitanja ulaza neće registrovati.



Slika 10. Vremenski dijagram trajanja ulaza i skeniranja.

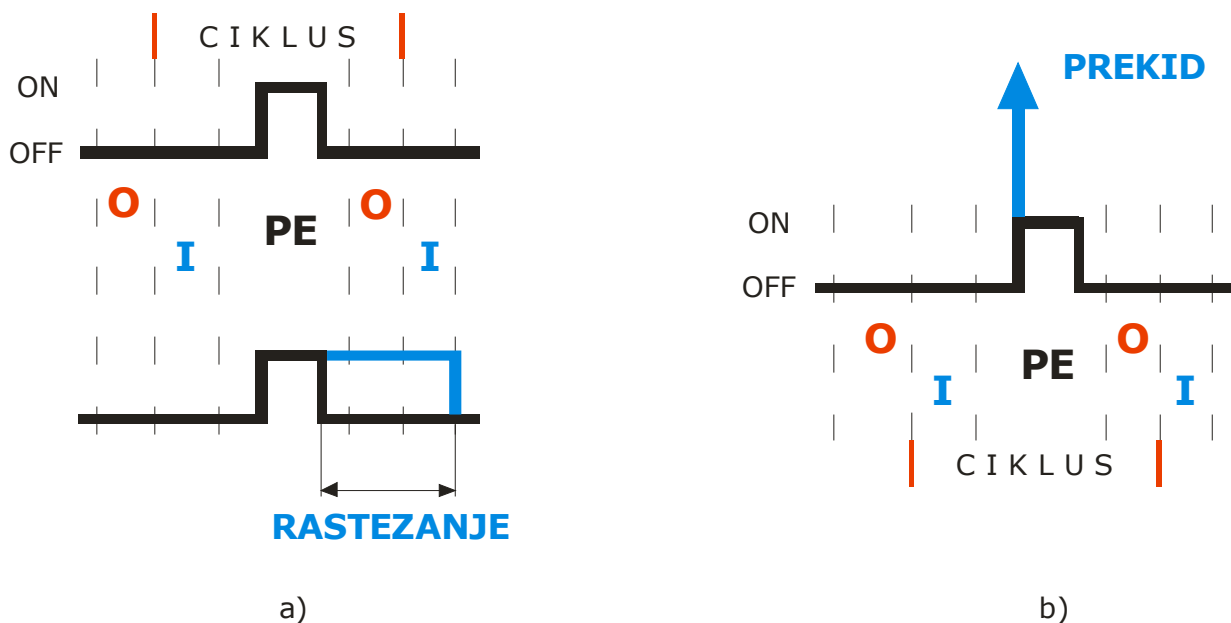
Da bi se ovo izbeglo ulaz mora da bude uključen (da postoji) za vreme koje je jednako zbiru jednog ciklusa skeniranja i jednog vremena kašnjenja ulaza, kao što je prikazano na slici 11.



Slika 11. Potrebno vreme trajanja ulaza

Međutim, u nekim situacijama nije moguće ostvariti da ulaz traje toliko dugo. Postoje dva načina da se ovaj problem reši.

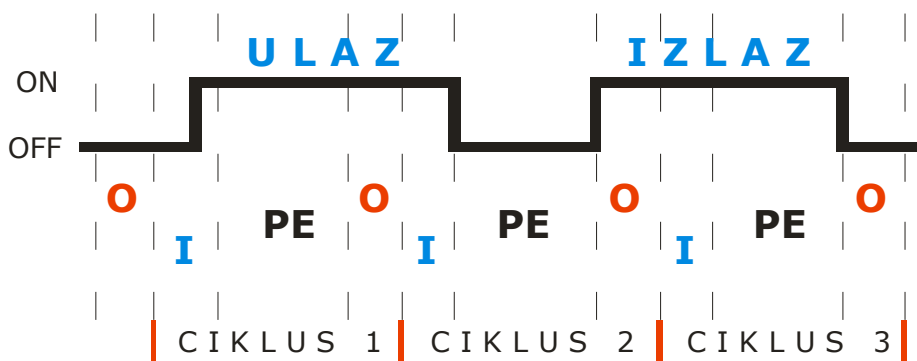
Prvi je **'rastezanje' ulaza**. Ova funkcija produžava trjanje ulaznog signala dok PLC ne izvrši skeniranje ulaza u sledećem ciklusu skeniranja (Slika 12a.).



Slika 12. Rastezanje i prekidna funkcija.

Drugi način je **prekidna (interrupt) funkcija** (slika 12b.). Ova funkcija prekida proces skeniranja i izvršava, od strane korisnika, specijalno napisanu rutinu. Funkcija prekida se može posmatrati kao mali program izvan glavnog programa. Odmah po uključenju ulaza, bez obzira na trenutni status procesa skeniranja, PLC staje i izvršava funkciju prekida. Kada završi izvršavanje funkcije prekida PLC se vraća na mesto gde je prekinuo i nastavlja sa uobičajenim procesom skeniranja.

Na slici 13 prikazano je najduže moguće kašnjenje paljenja izlaza, nakon što se uključio ulaz. Ovo je najgori mogući slučaj, jer se ulaz ne vidi do drugog ciklusa skeniranja.



Slika 13. Najgori mogući slučaj.

Dakle, najduže moguće kašnjenje izlaza koje može da se dogodi je jednako razlici vremena dva ciklusa skeniranja i vremena kašnjenja ulaza.

1.7 Povezivanje PLC-a sa perifernim uređajima

Da bi se zadovoljili najrazličitiji zahtevi, pored toga što SIMATIC S7-200 poseduje 5 različitih CPU-a, postoji i široka paleta ekspanzionih modula koji dodatno proširuju mogućnosti SIEMENS-ovih PLC-a.

Najslabiji model iz SIMATIC familije, CPU 221 nije moguće nadograditi, dok je model CPU 222 moguće nadograditi sa 2 modula, a sve ostale modele (CPU 224, CPU 224XP i CPU 226) moguće je proširiti sa čak 7 modula. Bitno je prilikom nadogradnje voditi računa o ukupnim potrebama za napajanjem (power requirements).

Postoje četiri grupe modula za proširenje:

- digitalni,
- analogni,
- moduli specijalne namene i
- komunikacioni moduli.

Digitalne module sa oznakom EM, koji postoje u tri serije, EM 221, EM 222 i EM 223, odlikuju digitalni ulazi/izlazi u rasponu od 4/4 do 16/16. Seriju EM 221 čine tri modela sa 8 i 16 ulaza, sa različitim naponskim nivoima (24V DC, 120/230V i 24V DC). Seriju EM 222 čini pet modela sa 4 i 8 digitalnih izlaza, i naponima 24V DC, 120/230V, kao i relejnim izlazima. Seriju EM 223 čini šest modela koji kombinuju ulaze i izlaze. Modele koji u svojoj oznaci nose prefiks 'SIPLUS' (SIPLUS EM 221...) karakteriše proširen temperaturni opseg i namenjeni su radu u znatno težim uslovima. Više o ovim modulima može se videti na zvaničnom sajtu SIEMENS-a na adresi: <https://mall.automation.siemens.com/WW/quest/index.asp?aktprim=0&nodeID=5000156&lang=en&display=r&foldersopen=-1111-1092-1091-1026-1025-1024-1-1104-1105-1526-1513-1512-1511-1510-1509-1515-&jumpto=1526>.

Sistem SIMATIC S7-200 daje i izbor različitih analognih modula sa 4/0, 4/1 ili 0/2 ulaza/izlaza, kao i sa analognim ulazima za očitavanje temperature. Oynake su EM 231, EM 232, EM 235, kao i SIPLUS EM 231, SIPLUS EM 232 i SIPLUS EM 235, z amodule sa proširenim temperaturnim opsegom (od -25⁰C do +70⁰C).

U grupu modula za specijalne namene spadaju:

- RDM modul EM 231 – služi za merenje temperature korišćenjem termoparova velike preciznosti (podržana je 31 vrsta standardnih otporničkih temperaturnih senzora),
- modul za pozicioniranje EM 253 – koristi se za jednostavno pozicioniranje (1 osa), za step motore i servo motore. Poseduje punu podršku starovanja i zadavanja parametara preko STEP 7-Micro/Win-a.

Komunikacionu grupu modula čine:

- PROFIBUS DS modul sa oznakom EM 277 (i ekvivalentni moduli sa prefiksom SIPLUS),

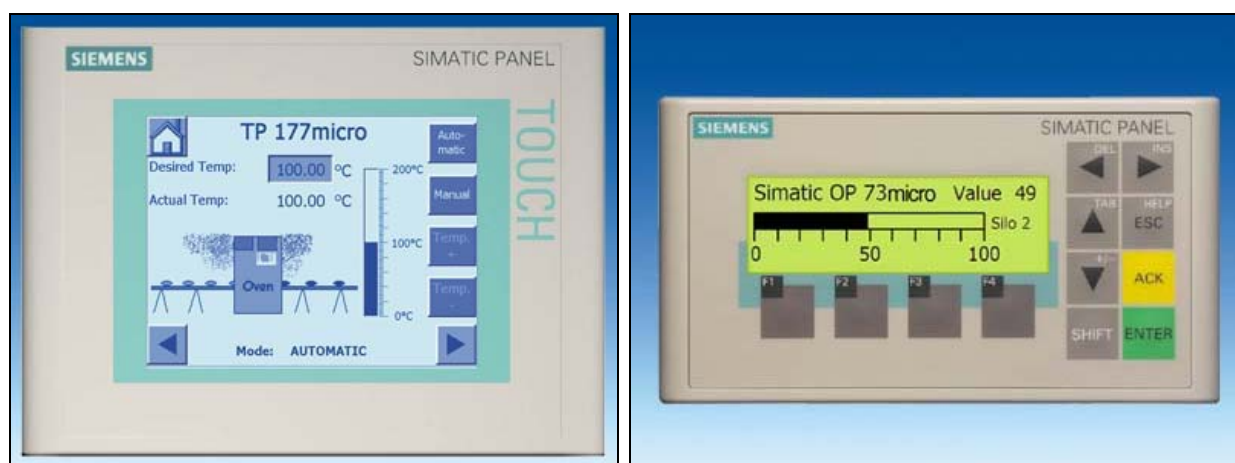
- moduli sa AS interfejsom i oznakom CP 243-2,
- modemski modul sa oznakom EM 241 i
- 'ethernet' moduli sa oznakama CP 243-1 i CP 243-1 IT.



Slika 14. Komunikacioni moduli za proširivanje SIEMENS-ove familije SIMATIC

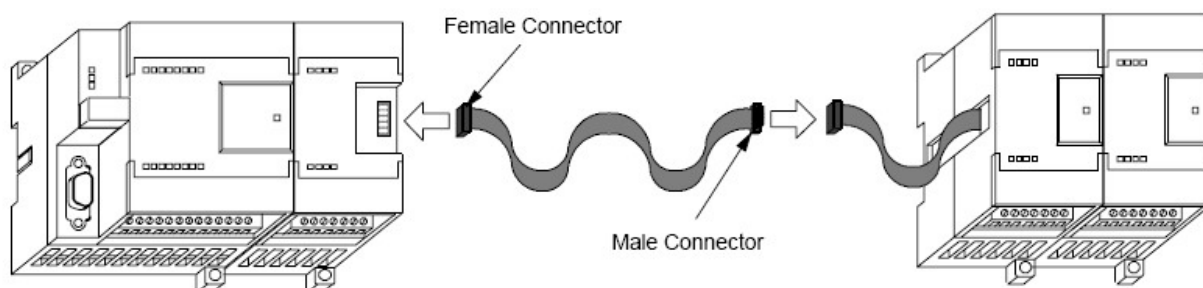
Pored ovih modula postoji još širok izbor dodatka, memorijskih kertridža, dodatnih izvora napajanja, tekst panela, panela osetljivih na dodir (Touch Panels) i operatorskih panela (Operator Panel), GSM/GPRS modema, antena, itd.

Svi ovi i mnogi drugi modeli mogu se lako naći preko prethodno navedenog linka proizvođača, sa detaljnim tehničkim karakteristikama.



Slika 15. Touch Panel SIMATIC TP 177micro i SIMATIC OP 73micro Operator Panel.

Povezivanje PLC-a i modula, kao i njihov montaža je jako bitan faktor. Prilikom horizontalnog postavljanja, lanac kreće sa PLC-om sa leve strane i nastavlja na desno sa modulima, dok je u vertikalnom položaju bitno da PLC bude na najnižoj tački. Svi elementi se pričvršćuju na panelnu ploču ili na DIN šin. Treba voditi računa da oko PLC-a i komponenti ima dovoljno prostora za normalno odvođenje toplote. Na glavni CPU PLC-a, moduli za proširenje se povezuju pomoću ulazno/izlaznog kabla (slika 16.).



Slika 16. Način povezivanja PLC-a i ekspanzionih modula.

Jednim kablom dozvoljeno je povezati samo jedan par CPU-modul u lancu.

Pored povezivanja sa modulima, najznačajnija karakteristika PLC-a je mogućnost povezivanja sa računarom. U kombinaciji sa softverom koji, uz PLC, isporučuje proizvođač dobija se moćan i za korišćenje jednostavan visoko pouzdan alat.

SIEMENS je korisniku stavio na raspolaganje dva načina za povezivanje računara na S7-200:

- direktna konekcija uz pomoć 'PPI Multi-Master' kabla ili
- karica sa komunikacionim procesorom (CP) sa 'MPI' kablom.

'PPI Multi-Master' kabal za programiranje je najčešće i najekonomičnije

rešenje za povezivanje. Kabal se sa jedne strane vezuje za komunikacioni port na S7-200, a sa druge na serijski port računara (RS232). 'PPI Multi-Master' kabal se takođe može iskoristiti za povezivanje ostalih komunikacionih uređaja na S7-200. Sada je korisniku na raspolaganju i varijanta kabla sa USB priključkom na računar.



Slika 17. Povezivanje sa računarom 'PPI Multi-Master' kablom sa RS-232 ili USB priključkom.

Prilikom formiranja lanca, bitno je znati kako se adresiraju dodatni moduli, tj. njihovi ulazi/izlazi. Lokalni ulazi/izlazi koje obezbeđuje određeni model CPU-a, obezbeđuju i fiksni set ulazno/izlaznih adresa. Dodavanjem ulazno/izlaznih modula desno od CPU-a, formira se ulazno/izlazni lanac. Adrese priključnih mesta modula su određena tipom ulaza/izlaza, i pozicijom modula u lancu, uzimajući u obzir ulaze ili izlaze modula istog tipa. Npr.: modul sa izlazima ne utiče na adrese priključnih mesta modula sa ulazima, i obrnuto. Isto tako, analogni moduli ne utiču na adresiranje digitalnih modula, i obrnuto.

Prostor registra za digitalne ulaze/izlaze uvek je rezervisan u priraštajima od po osam bita (jedan bajt). Ako modul ne obezbedi (iskoristi) mesto za svaki bit od svakog rezervisanog bajta, ovi neiskorišćeni bitovi ne mogu se dodeliti narednim modulima u ulazno/izlaznom lancu. Za ulazne module, neiskorišćeni bitovi dobijaju vrednost logičke nule '0' sa svakim ciklusom ažuriranja ulaza.

Analogni ulazi/izlazi se uvek alociraju u priraštajima od dva mesta. Ako modul ne obezbedi ulaz/izlaz za svako od ovih mesta, ova mesta ulaza/izlaza su izgubljena i nisu na raspolaganju narednim modulima u ulazno/izlaznom lancu.

Programiranje PLC-a

2.1 Sistemski prilaz projektovanju – opšte smernice

Postoje različite metode projektovanja PLC sistema. Bez obzira na naizgled neograničen broj različitih aplikativnih situacija projektovanja i implementacije sistema, pri analizi svake od njih, moguće je zapaziti da postoji ograničen broj tipskih situacija koje se javljaju na nivou projektovanja izvesnih podsistema i sistema merenja i/ili upravljanja. Individualno projektovanje sistema, naročito manjih, podsistemskih celina u današnjim tehničko-tehnološkim uslovima nema puno smisla, jer je najčešće ekonomski neisplativo. U tom smislu, već su sistematizovani prilazi (principi formalnog projektovanja) koji treba da skrate vreme projektovanja sistema, definišući načine i procedure korišćenja standardnih ili tipičnih projektnih situacija i elemenata. Naravno, svaka kompanija koja se bavi projektovanjem razvija sebi svojstvene principe.

Prilikom projektovanja sistema upravljanja korišćenjem PLC-a, osnovni sistematski prilaz podrazumeva sledeću proceduru:

- određivanje i analiza objekta upravljanja; treba odrediti kojim objektom se želi upravljati, a zatim izvršiti analizu tog objekta. Ovaj objekat može biti neka mašina ili proces i uopšteno se naziva *upravljani sistem*.
- dodela ulaza i izlaza; treba odrediti sve ulazne i izlazne uređaje koji se vezuju na PLC, a zatim po identifikovanju ulaznih/izlaznih uređaja, svakom od njih treba dodeliti adresu,
 - pisanje programa,
 - pokretanje sistema.

Jako bitan i nezaobilazan korak, koji se često zaboravlja, je i projektovanje sigurnosnih petlji. Uređaji kojima se upravlja mogu u stanju otkaza prouzrokovati pokretanja neželjenih procesa ili promene u radu drugih elemenata sistema koji mogu rezultirati mehničkim kvarom mašine ili, u najgorem slučaju, povredom radnika. Zadatak projektanta je da i ovakve situacije ima na umu i predvidi, kao i da preduzme sve mere da spreči da so ovakvih situacija dođe. Pored toga što sam PLC treba projektovati za ovakve situacije, na kritičnim mestima, na kojima može doći do ovakvih pojava, treba projektovati i ugraditi elektro-mehanička i/ili ručna premošćenja čije je funkcionisanje nezavisno od rada PLC-a.

Pre pokretanja sistema, treba proveriti da li su svi ulazno/izlazni uređaji dobro povezani sa PLC-om. Posle pokretanja, predstoji fino podešavanje upravljačkog sistema. U ovu svrhu mogu se iskoristiti i statistički alati. Sistem je potrebno podešavati sve dok se ne utvrdi da je rukovanje njime apsolutno bezbedno. Takođe treba proveriti, namernom inicijalizacijom neželjenih stanja, funkcionisanje sigurnosnih sistema koji su nezavisni od PLC-a.

2.2 Vrste programskih editora

Programski paket koji je razvio SIEMENS, STEP 7-Micro/Win, omogućava korisnicima tri vrste programskih editora za pisanje programa:

- leder dijagram (Ladder Logic) – LAD,
- iskaznu listu (Statement List) – STL, i
- funkcijski blok dijagram (Function Block Diagram) – FBD.

Sa određenim ograničenjima, programi pisani u bilo kom od ova tri editora mogu se pregledati i modifikovati sa ostalim editorima.

STL editor program prikazuje kao tekstualni jezik. Program se kreira zadavanjem simboličkih instrukcija. Pomoću njega se mogu napisati programi koji se inače ne bi mogli napisati pomoću leder dijagrama. To je zato što se programiranje vrši u prirodnom jeziku S7-200, umesto u grafičkom editoru gde su morala biti primenjena neka ograničenja da bi dijagrami mogli da se nacrtaju korektno. Ovaj jezik je jako sličan assembleru. S7-200 izvršava svaku instrukciju u redosledu koji je naznačen u programu, od vrha ka dnu, a zatim ponovo kreće od početka.

Primer STL programa:

```
LD   IO.0      //Čitanje jednog ulaza
A    IO.1      //sabiranje sa drugim ulazom
=    Q1.0      //Upisivanje vrednosti na izlaz 1
```

Glavne karakteristike STL editora, koje su bitne prilikom opredeljivanja za ovaj pristup, su:

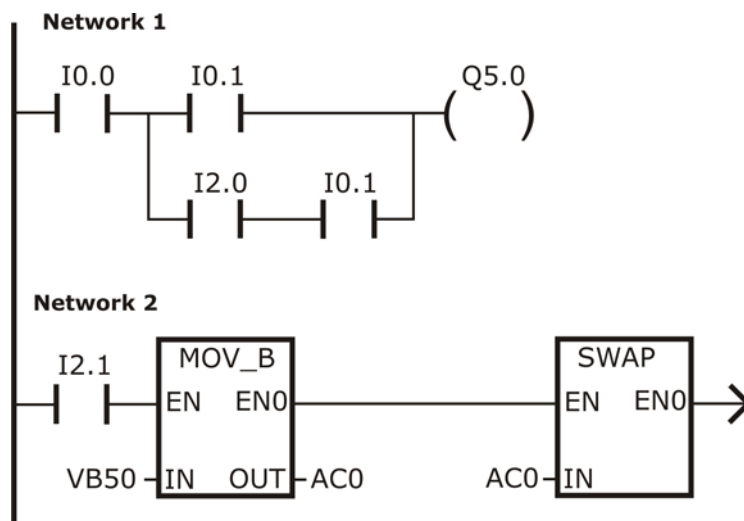
- STL je najpogodniji iskusnim programerima,
- STL ponekad omogućava rešavanje problema koji se ne mogu lako rešiti uz pomoć LAD ili FBD editora,
- STL se može koristiti samo sa SIMATIC setom instrukcija,
- STL editorom mogu se pregledati i modifikovati programi pisani u LAD ili FBD editoru, ali obrnuti proces nije uvek moguć. Ne mogu se uvek iskoristiti LAD i FBD editori da bi se prikazao program pisan u STL editoru.

LAD editor predstavlja program kao grafički prikaz sličan električnim šemama. Leder program emulira tok električne struje od izvora napajanja, kroz niz ulaznih logičkih uslova, koji na kraju rezultiraju stanjima izlaza. Leva strana LAD dijagrama se napaja, zatvoreni kontakti omogućavaju prolaz energije, a otvoreni kontakti prekidaju tok.

Logika je podeljena u nivoe – *network*, i program se izvršava jedan po jedan nivo, se leva na desno, i od vrha ka dnu. Takav izgled podseća na lestvice, i po tome je ovaj prikaz i dobio ime.

Različite instrukcije su predstavljene grafičkim simbolima i uključuju tri osnovne forme. *Kontakti* predstavljaju stanja logičkih ulaza kao što su prekidači, tasteri, ili unutrašnji uslovi. *Kalem* obično predstavlja logičke izlaze kao što su lampe, motori, starteri, releji, ili stanja logičkih izlaza. *Blok*

predstavlja dodatne instrukcije, kao što su tajmeri, brojači, ili matematičke funkcije.



Slika 18. Primer LAD programa.

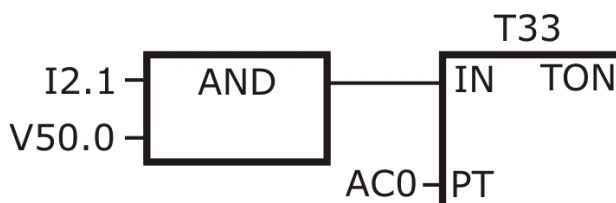
Glavne karakteristike LAD editora, koje su bitne prilikom opredeljivanja za ovaj pristup, su:

- jednostavan je za korišćenje za početnike,
- može se koristiti i sa SIMATIC i sa IEC 1131-3 instrukcijama,
- grafičko predstavljanje je jednostavno za razumevanje i rasprostranjeno je u svetu,
- uvek se može iskoristiti STL editor za prikazivanje programa napisanog uz pomoć SIMATIC LAD editora.

FBD editor prikazuje program koji grafički podseća na blok dijagram. Ne postoje kontakti i kalemovi ako u LAD editoru, ali postoje ekvivalentne instrukcije koje se predstavljaju blok instrukcijama.

Koncept 'toka struje' se koristi da se izrazi analogija sa LAD dijagramom, mada FBD ne koristi koncept levog ili desnog izvora napajanja. Tokom struje smatra se put logičke jedinice kroz FBD elemente.

Programska logika se dobija iz veza između blok instrukcija, odnosno, izlaz iz jedne instrukcije (kao što je AND blok) može se iskoristiti da se omogući druga instrukcija (kao što je tajmer) da bi se napravila odgovarajuća kontrolna logika. Ovakva logika povezivanja omogućava korisniku da reši širok skup problema.



Slika 19. Priemer FBD programa.

Prilikom izbora FBD editora, treba uzeti u obzir sledeće njegove karakteristike:

- grafički prikaz blok dijagrama je dobar za praćenje toka programa,
- FBD editor se može koristiti i sa SIMATIC i sa IEC 1131-3 instrukcijama,
- uvek se može iskoristiti STL editor za prikazivanje programa napisanog uz pomoć SIMATIC FBD editora.

U daljem tekstu biće objašnjavani i korišćeni samo LAD dijagrami i njihova sintaksa.

2.3 Razlike između SIMATIC i IEC 1131-3 seta instrukcija

Najveći broj PLC-a nudi slične osnovne instrukcije, ali obično postoje male razlike od proizvođača do proizvođača u izgledu, funkcionisanju itd. Tokom poslednjih godina, Internacionalni Elektrotehnička Komisija – International Electrotechnical Commission (IEC), razvila je globalni standard koji se posebno odnosi na mnoge aspekte PLC programiranja. Ovaj standard ohrabrio je mnoge proizvođače PLC-ova da ponude instrukcije koje su istog izgleda i funkcionišu na isti način.

SIEMENS-ov S7-200 nudi dva seta instrukcija koji omogućavaju da se reši širok spektar zadataka automatizacije. IEC set instrukcija je saglasan sa IEC 1131-3 standardom, a SIMATIC set je posebno projektovan za S7-200.

Kada je STEP7-Micro/Win prebačen na IEC mod, crveni dijament je nacrtan, na stablu instrukcija, pored instrukcija koje nisu definisane IEC 1131-3 standardom.

Glavne razlike između SIMATIC i IEC seta instrukcija su:

- IEC set je ograničen na one instrukcije koje su standardne među proizvođačima PLC-ova. Neke instrukcije koje su uključene u SIMATIC set nisu standardne u IEC 1131-3 specifikaciji. One su i dalje dostupne za korišćenje kao ne-standardne instrukcije, ali ako se upotrebe, program više nije strogo IEC 1131-3 kompatibilan.

- neke IEC blok instrukcije prihvataju višestruke formate podataka. Ovo se često zove i preopterećivanje (overloading), Umeso upotrebe dva različita bloka, npr. ADD_I (Add Integer) i ADD_R (Add Real), IEC ADD instrukcija proverava format podatka koji se dodaje i automatski bira pravilnu instrukciju. Ovakav pristup znatno skraćuje vreme programiranja.

- IEC instrukcije automatski proveravaju ispravnost formata podatka parametara instrukcije smanjujući tako sintaksne greške u programiranju.

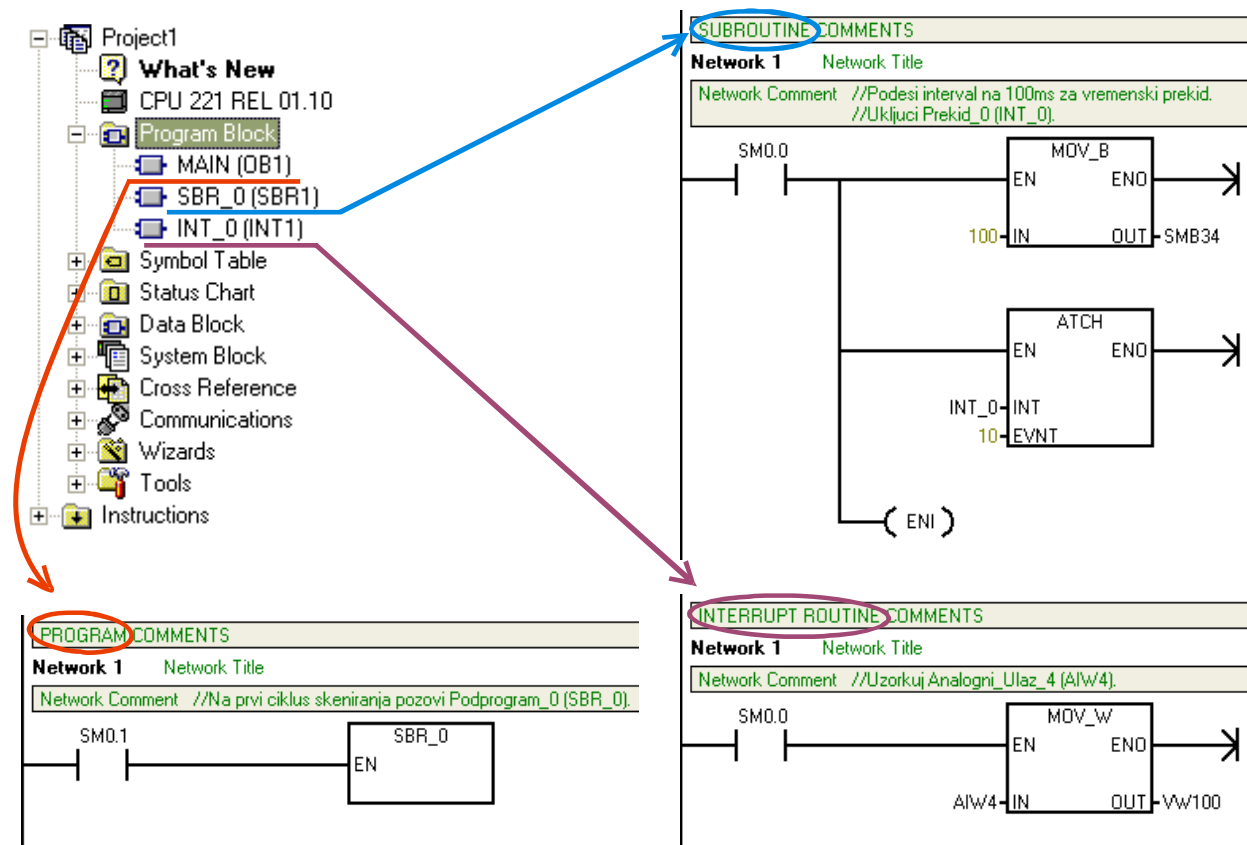
Prilikom odlučivanja između SIMATIC i IEC seta instrukcija treba se rukovoditi sledećim:

- SIMATIC instrukcije često imaju najkraće vreme izvršavanja,

- sva tri programska editora (STL, LAD i FBD) mogu se koristiti sa SIMATIC instrukcijama; za IEC instrukcije na raspolaganju su samo LAD i FBD,
- neke IEC instrukcije, kao tajmeri, brojači, množioci, delioci, funkcionišu drugačije od svojih SIMATIC ekvivalenata,
- način rada IEC instrukcija su standardne za različite marke PLC-ova,
- iako IEC standard poseduje manje instrukcija nego SIMATIC, u IEC program se uvek mogu uključiti SIMATIC naredbe,
- IEC 1131-3 zahteva da tip promenljive bude deklarisan, i podržava sistemsku proveru tipa podatka.

2.4 Osnovni delovi programa

Program se sastoji od izvršnog koda i komentara. Izvršni kod sačinjavaju glavni program i podprogrami i/ili rutine prekida. Izvršni kod se, za rauliku od komentara, kompajlira (prevodi) i učitava (download) na S7-200. Glavni program, podprogrami, i rutine prekida, koriste se kao organizacioni elementi programa.



Slika 20. Delovi programa.

Glavno telo programa sastoji se od instrukcija koje kontrolišu korisničku aplikaciju. S7-200 izvršava ove instrukcije sekvencijalno, jedanput u toku ciklusa skeniranja. Glavni program se još naziva i OB1.

Podprogrami su opcioni elementi programa, i izvršavaju se samo po pozivanju od strane: glavnog programa, rutine prekida, ili drugog podprograma. Pogodni su u situacijama kada se neka funkcija, ili skup funkcija često izvršavaju. Umesto da se kopira kod na svakom mestu u programu gde treba da se izvrši odgovarajuća funkcija, ili skup funkcija, kod se upisuje samo jednom u podprogram, a podprogram se zatim poziva onoliko puta koliko je to potrebno iz glavnog programa. Na taj način podprogrami pružaju niz pogodnosti:

- smanjuje se ukupna veličina programa,
- smanjuje se vreme skeniranja jer se kod izmešta iz glavnog programa. S7-200 izračunava kod u glavnom programu u svakom ciklusu, bez obzira da li se kod izvršava ili ne, ali u podprogramu S7-200 izračunava kod samo kada je podprogram pozvan,
- kod je lako prenosiv. Može se lako izdvojiti kod za funkciju u podprogramu, i onda kopirati u druge programe.

Korišćenje V memorijskih adresa (promenljiva memorija) može smanjiti prenosivost podprograma jer se može desiti da dođe do sukoba između dodeljivanja V memorijskih adresa iz jednog programa sa dodeljivanjima u drugom programu. Podprogrami koji koriste lokalnu memoriju (memorija L) za sva memorijska dodeljivanja adresa, sa druge strane, visoko su prenosivi jer nema bojazni od konflikta adresa između podprograma i drugog dela programa kada se koriste lokalne promenljive.

Rutine prekida su, kao i podprogrami, opcioni delovi programa, a reaguju na određene događaje prekida. Korisnik projektuje rutine prekida koje obrađuju unapred definisane događaje prekida. Rutine prekida se ne pozivaju od strane glavnog programa, već se povezuju sa događajima prekida, a S7-200 izvršava instrukcije u rutini prekida svaki put kada se događaj prekida desi.

Zato što nije moguće predvideti kada bi S7-200 mogao da generiše prekid, poželjno je da se ograniči broj promenljivih koje se koriste kako od strane rutine prekida, tako i na drugim mestima u programu. Za rutine prekida treba koristiti lokalne promenljive da bi se obezbedilo da rutina prekida koristi samo privremenu memoriju i ne menja podatke koji se koriste na drugim mestima u programu.

Sistemske blokove i blokove podataka su takođe elementi programa, i korisnik prilikom učitavanja programa bira da li će učitati ili ne i ove delove programa. Sistemski blok daje korisniku mogućnost da konfigurira različite delove hardvera. Blok podataka čuva vrednosti različitih promenljivih (V memorija) koje koristi korisnički program. Blok podataka se može koristiti da se unesu početne vrednosti podataka.

2.5 Osnove programskog paketa STEP 7-Micro/Win

Programe pisane leđer dijagramima PLC ne razume, a STL programski editor je za početnika dosta složen i nerazumljiv. Zato proizvođači uz svoj harver razvijaju i softver koji pored uloge alata za pisanje programa služi i za prevođenje korisničkog programa na jezik razumljiv PLC-u.

Da bi izašao u susret onima koji se prvi put susreću sa njihovim PLC-om, a i PLC-om uopšte, SIEMENS-ov tim stučnjaka razvio je veoma intuitivan, jednostavan, i krajnjem korisniku razumljiv softverski paket, koji pored osnovne funkcije, pisanja programa za PLC-e, ima mnogo širi krug mogućnosti. Program se lako instalira, prilagođavanje menia i prečica je takođe dostupno. Pomoću STEP 7-Micro/Win programskog paketa, programi se mogu pisati u LAD, FBD ili STL editoru. Sa leve strane u 'Navigation bar'-u nalazi se grupa ikonica za pristup različitim programskim odlikama (Program Block, Simbol Table, Status Chart, Data Block, Cross Reference itd.). Ispod nje nalazi se široka grupa alata u vidu čarobnjaka, od čarobnjaka za povezivanje sa raznim komunikacionim modulima, preko pozicioniranja i PID kontrole, do dizajnera tastature i 'touch' panelom.

Stablo sa instrukcijama prikazuje sve objekte u trenutnom projektu i instrukcije za kreiranje programa. Pojedinačne instrukcije se u program ubacuju prevlačenjem ili dvoklikom, na trenutnu poziciju kursora u programskom editoru. Programski editor sadrži program i tabelu lokalnih promenljivih, gde korisnik može zadati simbolička imena za privremene lokalne promenljive. Podprogrami i rutine prekida pojavljuju se u polju na dnu prozora programskog editora.

2.6 Set instrukcija S7-200

Set instrukcija razvijen u SIEMENS-u za programiranje PLC-ova iz familije S7-200 je jako širok. Kao što je već rečeno instrukcije su predstavljene grafičkim simbolima i uključuju tri osnovne forme: kontakte, kalemove, i blokove. Prema svojoj funkciji mogu se podeliti na nekoliko grupa: logičke bit instrukcije, vremenske (satne) instrukcije, komunikacijske instrukcije, upoređivačke instrukcije, pretvaračke instrukcije, brojačke instrukcije, matematičke instrukcije, instrukcije logičkih operacija, podprogramske instrukcije, itd.

U ovom radu će biti naznačene samo one najbitnije instrukcije. Korisno i jednostavno za praćenje uputstvo isporučuje se u vidu 'Help' fajla u okviru programskog paketa STEP 7-Micro/WIN.

U tabeli 3, prikazane su veličine i adresni raspored nekih memorijskih lokacija za familiju S7-200. Tabela je data, ne kao pun opis, već kao orjentir i smernica u smislu opšteg sagledavanja resursa kojima raspolažu SIEMENS-ovi PLC-ovi, od najslabijeg modela CPU-a 221, pa do najmoćnijeg CPU-a 226. Detaljniji opis karakteristika za svaki CPU ponaosob može se naći na sajtu SIEMENS-a, na ranije datom linku.

| OPIS \ CPU | CPU 221/222** | CPU 224 | CPU 224XP | CPU 226 |
|---|--------------------------|---------------------------|----------------------------|----------------------------|
| Veličina kor. prog. bez/sa edit. u RUN modu | 4096 bytes 4096 bytes | 8192 bytes 12288 bytes | 12288 bytes 16384 bytes | 16384 bytes 24576 bytes |
| Vel. kor. podataka | 2048 bytes | 8192 bytes | 10240 bytes | 10240 bytes |
| Ulazni registar | I0.0 do I15.7 | I0.0 do I15.7 | I0.0 do I15.7 | I0.0 do I15.7 |
| Izlazni registar | Q0.0 do Q15.7 | Q0.0 do Q15.7 | Q0.0 do Q15.7 | Q0.0 do Q15.7 |
| Analogni ulazi | AIW0 do AIW30 | AIW0 do AIW62 | AIW0 do AIW62 | AIW0 do AIW62 |
| Analogni izlazi | AQW0 - AQW30 | AQW0 - AQW62 | AQW0 - AQW62 | AQW0 - AQW62 |
| Prom. mem. V | VB0 do VB2047 | VB0 do VB8191 | VB0 - VB10239 | VB0 - VB10239 |
| Lokalna mem. L* | LB0 do LB63 | LB0 do LB63 | LB0 do LB63 | LB0 do LB63 |
| Bit memorija M | M0.0 do M31.7 | M0.0 do M31.7 | M0.0 do M31.7 | M0.0 do M31.7 |
| Tajmeri | T0 do T255 | T0 do T255 | T0 do T255 | T0 do T255 |
| Brojači | C0 do C255 | C0 do C255 | C0 do C255 | C0 do C255 |
| Brzi brojači | HC do HC5 | HC do HC5 | HC do HC5 | HC do HC5 |
| Poziv/podprogram | 0 do 63 | 0 do 63 | 0 do 63 | 0 do 127 |
| Prekidne rutine | 0 do 127 | 0 do 127 | 0 do 127 | 0 do 127 |
| PID petlje | 0 do 7 | 0 do 7 | 0 do 7 | 0 do 7 |

* LB60 do LB63 su rezervisane za STEP 7-Micro/WIN, od verzije 3.0.

** Karakteristike date u tabeli su identične kod ova dva modela.

Tabela 3. Memorijski prostor kod SIEMENS-ove familije S7-200.

Pre nego što se pređe na konkretne instrukcije, biće iznete osnovne konvencije vezane za programiranje S7-200.

EN (Enable IN) – uključen ulaz – predstavlja Bulov ulaz za blokove u LAD (i FBD) dijagramima. Da bi blok instrukcija bila izvršena potrebno je da na ulazu postoji tok struje.

ENO (Enable OUT) – uključen izlaz – predstavlja Bulov izlaz za blokove u LAD (i FBD) dijagramima. Ako blok ima tok struje na EN ulazu, i blok instrukcija bude izvršena bez greške, ENO izlaz prosleđuje tok struje sledećem elementu. Ako se otkrije greška prilikom izvršavanja blok instrukcije, tok struje je prekinut u bloku koji je generisao grešku.

Uslovni/bezuvlovni ulazi – u LAD (i FBD) dijagramima, blok, ili kalem koji je zavistan od toka struje predstavljen je sa vezom na neki element na levoj strani. Blok, ili kalem koji je nezavistan od toka struje predstavljen je direktnim vezivanjem na levu šinu izvora napajanja.

Instrukcija koja zavisi od toka struje – USLOVNA $\text{---}(\text{ JMP })$.
 Instrukcija koja NE zavisi od toka struje – BEZUSLOVNA $\text{---}(\text{ NEXT })$.

Instrukcije poređenja - izvršavaju se bez obzira na stanje toka struje. Ako je nema toka struje (*false*) i izlaza nema (*output is false*). Ako postoji tok struje, izlaz zavisi od rezultata poređenja. Instrukcije poređenja se predstavljaju kao blokovi, iako se operacije obavljaju kako kontakt.

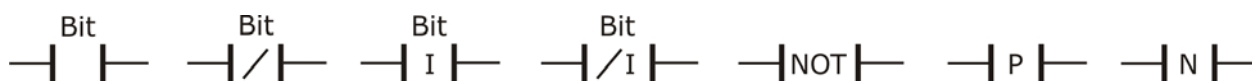
2.6.1 Bit logičke instrukcije

Kontakti

Standardni kontakti mogu biti *normalno otvoreni* (LD, A i O) i *normalno zatvoreni* (LDN, AN i ON), a vrednosti dobijaju iz memorije, ili registra ako je tip podatka I ili Q. Normalno otvoreni kontakt je zatvoren (uključen – *on*) kada je bit jednak jedinici, a normalno zatvoreni kontakt je zatvoren (uključen – *on*) kada je bit jednak nuli.

Direktni kontakti (eng. *Normally Open/Closed Immediate*) se ne oslanjaju na ciklus skeniranja S7-200 da bi se ažurirali. Instrukcije *normalno otvorenog direktnog kontakta* (LDI, AI i OI) i *normalno zatvorenog kontakta* (LDNI, ANI i ONI) dobijaju vrednosti ulaza odmah kada se instrukcija izvrši, ali registar nije ažuriran. Logika je ista kao i kod standardnih kontakata.

Instrukcije negacije (*NOT*) menjaju status toka struje ulaza, tj. menja vrednosti iz 1 u 0, ili iz 0 u 1.



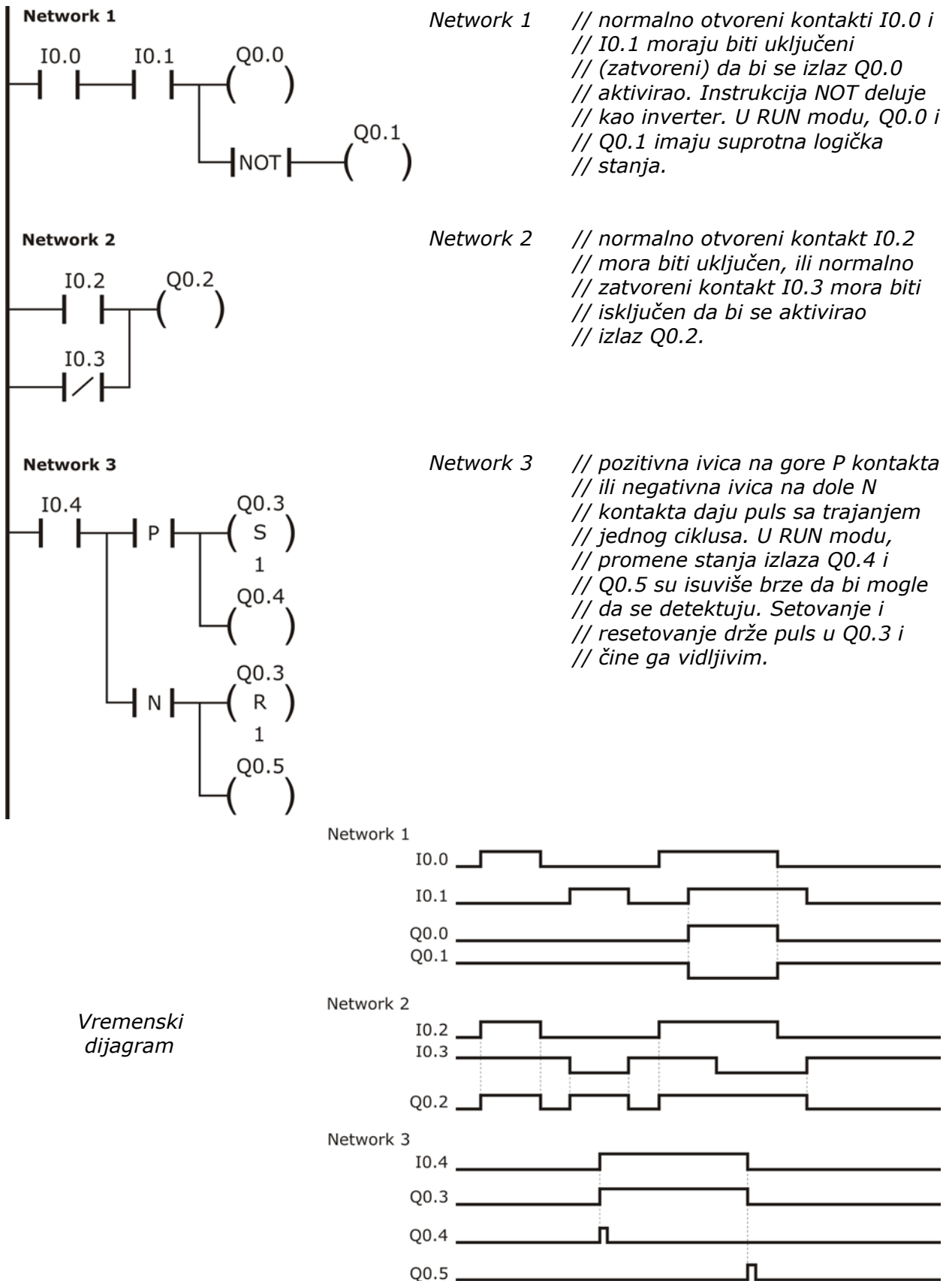
Slika 21. Simboli kontakata u LAD dijagramima.

Instrukcije pozitivnog i negativnog prelaza – omogućavaju tok struje za jedno vreme ciklusa skeniranja za svaki ISKLJUČENO-U-UKLJUČENO i UKLJUČENO-U-ISKLJUČENO prelaz. Zbog toga što zahtevaju 0-u-1 ili 1-u-0 prelaz, u prvom ciklusu skeniranja ove instrukcije ne mogu da detektuju uzlazne ili silazne ivice. U toku prvog ciklusa skeniranja, S7-200 postavlja stanje bitova određene od strane ovih instrukcija. U sledećim ciklusima skeniranja ove instrukcije mogu otkriti prelaze za definisane bitove.

Kalemovi

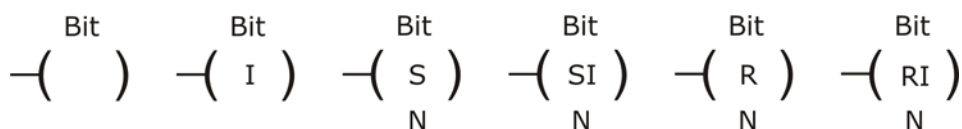
Izlazne instrukcije (=) upisuju novu vrednost izlaznog bita u registar. Kada se izvrši izlazna instrukcija, S7-200 uključuje ili isključuje izlazni bit u registru.

Instrukcija *direktnog izlaza* (=I) upisuje novu vrednost i na izlaz i u odgovarajući registar, kada je izvršena. Izlaz se, po izvršenju instrukcije, odmah izjednačava sa stanjem toka struje.



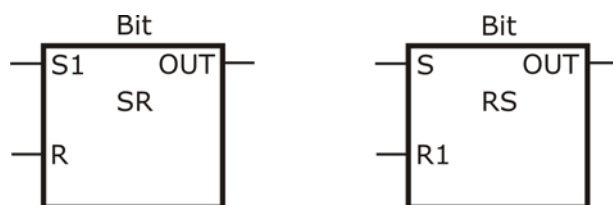
Slika 22. Kontakti i kalemovi u LAD programu.

Instrukcije *setovanja* (S) i *resetovanja* (R) setuju (postavljaju, uključuju) ili resetuju (isključuju) određeni broj tačaka (N), počevši od naznačene adrese. Može se setovati ili resetovati od 1 do 255 tačaka. Ako se reset instrukcija odnosi na bit tajmera (T) ili bit brojača (C), instrukcija resetuje bit tajmer ili brojač i briše trenutnu vrednost tajmera ili brojača.



Slika 23. Simboli kalemova u LAD dijagramima.

Instrukcije *direktnog setovanja* (SI) i *direktnog resetovanja* (RI) funkcionišu kao i prethodne, sa razlikom da se nova vrednost odmah po izvršenju instrukcije upisuje i na izlaz i u odgovarajući registar, dok prethodne instrukcije upisuju novu vrednost samo u registar. Može se setovati ili resetovati od 1 do 128 tačaka trenutno.



Slika 24. Simboli instrukcija dominantnog setovanja i resetovanja.

Instrukcije *dominantnog setovanja* (SR) i *resetovanja* (RS) su prekidačka kola sa pamćenjem gde je setovanje, odnosno resetovanje dominantno. Ako su i signal setovanja (S1) i resetovanja (R) prisutni (tačni), onda je i izlaz (OUT) tačan, odnosno netačan, respektivno.

| Instrukcija | S1 | R | IZLAZ (Bit) |
|----------------------------------|----|---|------------------|
| Set dominantna bistabilna (SR) | 0 | 0 | prethodno stanje |
| | 0 | 1 | 0 |
| | 1 | 0 | 1 |
| | 1 | 1 | 1 |
| Instrukcija | S1 | R | IZLAZ (Bit) |
| Reset dominantna bistabilna (RS) | 0 | 0 | prethodno stanje |
| | 0 | 1 | 0 |
| | 1 | 0 | 1 |
| | 1 | 1 | 0 |

Tabela 4. Tabela iskaza istinitosti za dominantnu set i reset instrukciju.

2.6.2 Satne instrukcije – Clock Instructions

Čitanje i setovanje sata realnog vremena

Instrukcija čitanja sata realnog vremena (TODR) čita trenutno vreme i datum iz hardverskog sata i upisuje ih u vremenski bafer dužine 8 bajta, počevši od adrese T. Setovanje (postavljanje) sata realnog vremena (TODW) upisuje trenutno vreme i datum u hardverski sat, počevši od osmog bajta vremenskog bafera sa adresom T.

Treba voditi računa o ispravnosti datuma, jer S7-200 ne vrši ovu proveru, tako da datum, kao npr. 30. Februar, može biti prihvaćen.

Ne treba koristiti TODR/TODW instrukcije u isto vreme i u glavnom programu i u rutini prekida jer će doći do greške (non-fatal error 0007).

Takođe treba, ako se u korisničkom programu operiše sa godinama, imati na umu da S7-200 uzima u ozir samo poslednje dve cifre iz godine.

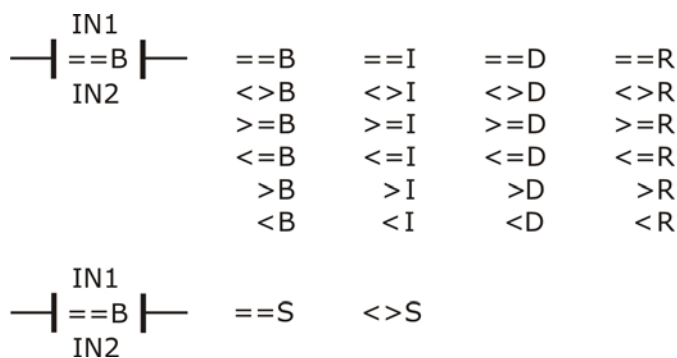
2.6.3 Instrukcije poređenja

Poređenje brojnih vrednosti

Instrukcije poređenja se koriste za poređenje dve vrednosti i to sa sledećim relacijama:

| | | |
|-----------|------------|------------|
| IN1 = IN2 | IN1 >= IN2 | IN1 <= IN2 |
| IN1 > IN2 | IN1 < IN2 | IN1 >< IN2 |

Kada je poređenje istinito, instrukcija poređenja uključuje kontakt.



Slika 25. Simboli instrukcija poređenja.

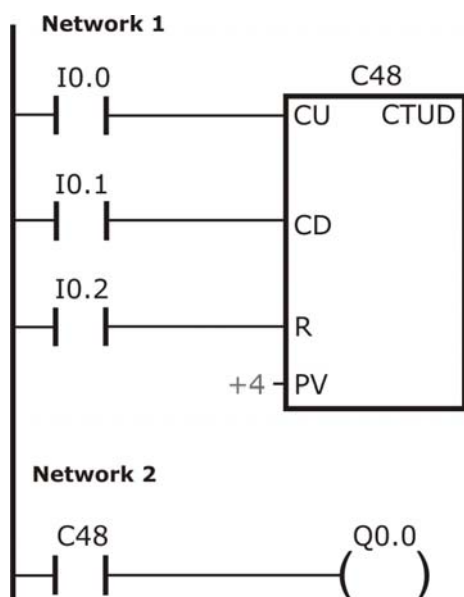
Mora se voditi računa kod pravilnog inicijalizovanja pointera i vrednosti koje sadrže realne brojeve pre izvršavanja instrukcija poređenja koje koriste ove vrednosti, jer u slučaju greške S7-200 odmah prekida danje izvršavanje korisničkog programa.

Instrukcije poređenja se izvršavaju nezavisno od stanja toka struje.

promeni stanje iz isključenog u uključeno. Kada je trenutna vrednost Cxx jednaka nuli, bit brojača Cxx se uključi. Kada se uključi LD ulaz, brojač resetuje bit brojača Cxx i učitava prepodešenu vrednost PV. Kada dostigne nultu vrednost, brojač se zaustavlja i bit brojača se uključuje.

Brojač na više/na niže

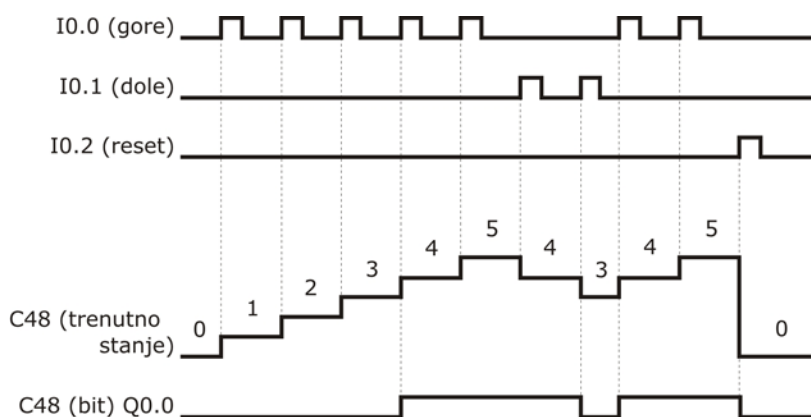
Instrukcija ove vrsta brojača (CTUD) broji na više svaki put kada ulaz brojanja na više (CU) promeni stanje iz isključenog u uključeno, i broji na niže, od trenutne vrednosti brojača, svaki put kada ulaz brojanja na niže (CD) promeni stanje iz isključenog u uključeno. Trenutna vrednost Cxx je trenutna vrednost brojača, a PV je prepodešena vrednost koja se poredi sa trenutnom svaki put kada se izvrši instrukcija brojača.



Network 1 // ulaz I0.0 broji na više
 // ulaz I0.1 broji na niže
 // ulaz I0.3 resetuje trenutnu vrednost na 0

Network 2 // brojač na više/na niže C48
 // uključuje C48 bit kada
 // trenutna vrednost brojača
 // bude >=4

Vremenski dijagram



Slika 27. Primer primene brojačkih instrukcija.

Po dostizanju maksimalne vrednosti (32767), sledeća uzlazna ivica na ulazu brojača na gore uslovljava da trenutna vrednost pređe u minimalnu (-32768). Isto važi i za obrnuti slučaj.

Kada je trenutna vrednost Cxx veća ili jednaka prepodešenoj vrednosti PV, bit brojača se uključuje. U suprotnom, bit brojača se isključuje. Brojač se resetuje kada se ulaz za resetovanje (R) uključi, ili kada se izvrši instrukcija resetovanja. Kada se dostigne vredniost PV, brojač CTUD prestaje sa brojanjem.

2.6.5 Matematičke instrukcije

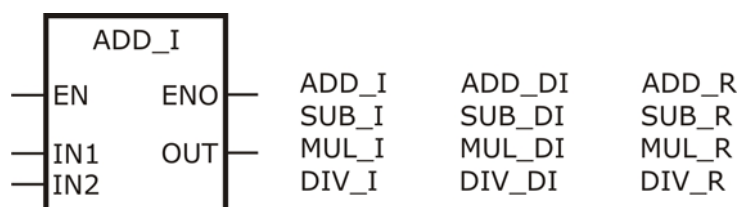
Instrukcije sabiranja, oduzimanja, množenja i deljenja

$$IN1 + IN2 = OUT$$

$$IN1 * IN2 = OUT$$

$$IN1 - IN2 = OUT$$

$$IN1 / IN2 = OUT$$



Slika 28. Simboli matematičkih instrukcija.

Ove matematičke instrukcije mogu da operišu sa 16-to bitnim celim brojevima i sa 32-o bitnim celim i realnim brojevima, poštujući matematičke operacije, pri čemu se kao rezultat dobijaju odgovarajući 16-to, odnosno 32-o bitni brojevi, respektivno.

Pored ovih, osnovnih matematičkih operacija, postoje i druge operacije, kao što su: *sin*, *cos*, *tan*, *prirodni* i *eksponencijalni logaritam*, i *kvadratni koren*.

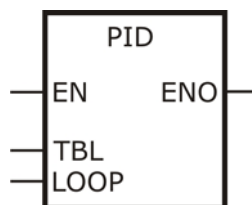
Instrukcija PID dejstva

Instrukcija proporcionalno-integralno-diferencijalnog ili PID dejstva, takođe je deo seta instrukcija podržanih od strane SIEMENS-ovih kontrolera iz serije S7-200.

Da bi se izvršila ova instrukcija, ulaz na njenom bloku u leder dijagramu mora da postoji (mora biti uključen). Instrukcija koristi dva operanda: *adresu tabele*, koja je početna adresa tabele PID petlje, i broj petlje, koji je konstanta od 0 do 7. Dakle, korisnik raspolaže sa osam PID instrukcija koje može iskoristiti u programu.

Ako se koriste dve, ili više, PID instrukcija sa istim brojem petlje (čak i ako imaju različite adrese tabele), PID izračunavanja će doći u sukob među sobom, i rezultat je nepredvidiv.

Tabela petlje čuva devet parametara koji se koriste za kontrolu i nadzor izvršenja petlje, i sadrži trenutnu i prethodnu vrednost promenljivih koje se obrađuju, zadatu vrednost, izlaznu vrednost, pojačanje, vreme semplovanja, integralno vreme (reset), derivatno vreme, i integralno vreme (odstupanje).



Slika 29. Simboli PID instrukcija.

Da bi se ostvarilo PID izračunavanje sa željenim vremenom semplovanja, PID instrukcija se mora izvršiti ili iz vremenske rutine prekida, ili iz glavnog programa brzinom koja je kontrolisana od strane tajmera. Vreme semplovanja mora se obezbediti kao ulaz PID instrukciji kroz tabelu petlje.

STEP 7-Micro/Win nudi PID čarobnjak koji korisnika u nekoliko koraka vodi kroz definisanje PID algoritma zatvorene petlje.

2.6.6 Instrukcije za upravljanje programom

Uslovni kraj

Instrukcija uslovnog kraja prekida trenutni ciklus skeniranja na osnovu stanja logike koja mu neposredno prethodi. Može se koristiti u glavnom programu, ali ne i u podprogramima i rutinama prekida.



Slika 30. Simboli instrukcija za upravljanje programom.

Stop

Ova instrukcija prekida izvršavanje programa prebacivanjem CPU-a S7-200 iz moda RUN u stanje STOP.

Ako se STOP instrukcija izvrši u rutini prekida, rutina prekida se odmah prekida, i svi prekidi koji su u toku se zanemaruju. Preostali zadaci, u okviru trenutnog ciklusa skeniranja, se izvršavaju i na kraju ciklusa skeniranja prelazi se iz moda RUN u STOP.

2.6.7 Tajmer instrukcije

Tajmer sa kašnjenjem paljenja (*On-Delay Timer - TON*)

TON tajmer instrukcija (i TONR – sa zadržkom) broji vreme kada je ulaz uključen (ON). Nakon isteka zadatog perioda vremena po paljenju ulaza, pali se izlaz. Broj tajmera Txx određuje rezoluciju tajmera, i ona je prikazana u bloku instrukcije.

TON i TONR instrukcije broje vreme od trenutka kada se upali ulaz. Kada je trenutna vrednost veća ili jednaka prepodešenoj vrednosti, bit tajmera se pali. Trenutna vrednost TON tajmera se briše kada se ulaz ugasi, dok se trenutna vrednost TONR tajmera zadržava (čuva). I TON i TONR tajmeri nastavljaju da broje i nakon što dostignu predefinisanu vrednost, i prestaju sa brojanjem kada dostignu maksimalnu vrednost brojača – 32767.



Slika 31. Simboli tajmer instrukcija.

Tajmer sa kašnjenjem gašenja (*Off-Delay Timer - TOF*)

TOF tajmer se koristi da se odloži gašenje izlaza za određeni period vremena nakon što se ulaz ugasi. Nakon što se ulaz upali, bit tajmera se pali, i trenutna vrednost se postavlja na nulu. Kada se ulaz ugasi, tajmer broji dok proteklo vreme ne dostigne zadatu vrednost. Kada dostigne zadatu vrednost, bit tajmera se isključuje, i trenutna vrednost prestaje da raste. Međutim, ako se ulaz opet upali pre nego što TOF dostigne zadatu vrednost, bit tajmera ostaje uključen. Ulaz, dakle, mora preći iz uključenog-u-isključeno stanje (*on-to-off*) da bi TOF krenuo sa brojanjem.

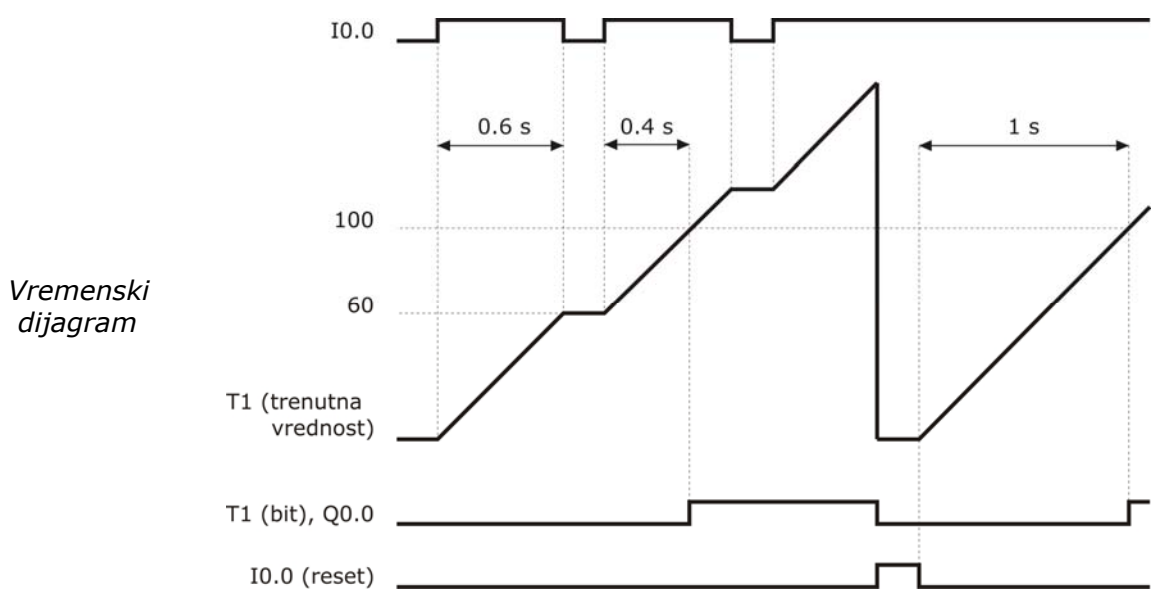
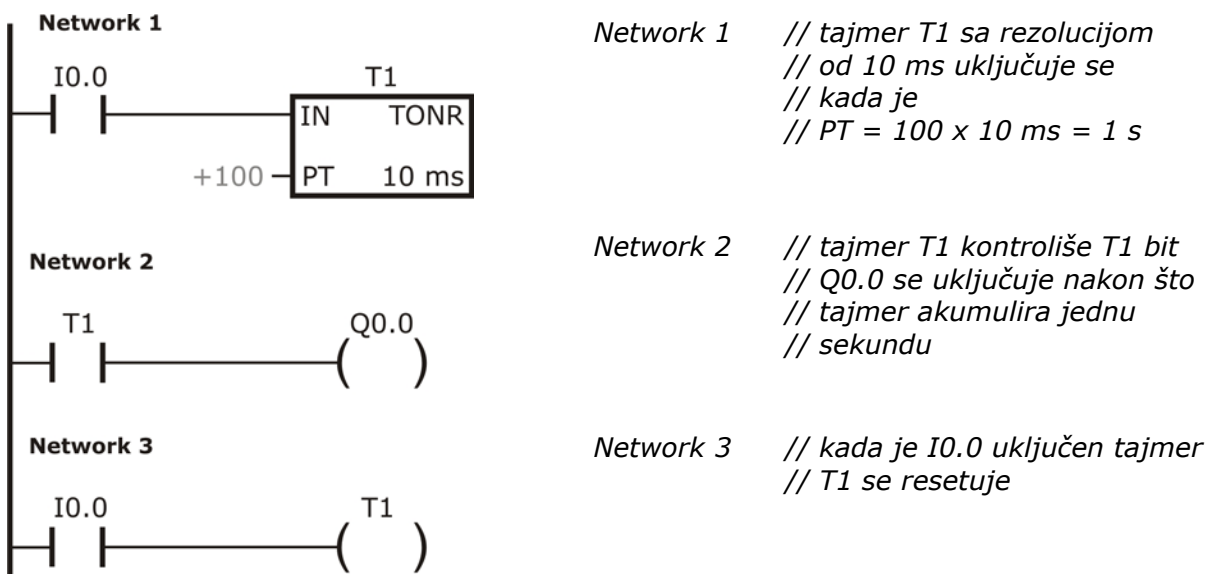
| Tip tajmera | Rezolucija | Maksimalna vrednost | Broj tajmera |
|----------------------------|------------|----------------------|--------------------------|
| TONR (sa pamćenjem) | 1 ms | 32.767 s (0.546 min) | T0, T64 |
| | 10 ms | 327.67 s (5.46 min) | T1 do T4, T65 do T68 |
| | 100 ms | 3276.7 s (54.6 min) | T5 do T31, T69 do T95 |
| TON, TOF (bez pamćenja) | 1 ms | 32.767 s (0.546 min) | T32, T96 |
| | 10 ms | 327.67 s (5.46 min) | T33 do T36, T97 do T100 |
| | 100 ms | 3276.7 s (54.6 min) | T37 do T63, T101 do T255 |

Tabela 5. Brojevi tajmera i njihove rezolucije.

Tajmeri se mogu resetovati korišćenjem reset instrukcije (R). Reset instrukcija isključuje bit tajmera a trenutnu vrednost postavlja na nulu. Nakon reseta, TOF tajmer zahteva prelazak ulaza iz uključeno-u-isključeno stanje da bi se tajmer restartovao.

Rezolucija tajmera određuje dužinu vremena u svakom intervalu. Npr. TON tajmer sa rezolucijom od 10 ms računa broj intervala od 10 ms koji prođu nakon što je uključen ulaz na TON-u. 50 intervala na tajmeru sa rezolucijom od 10 ms predstavlja 500ms. SIMATIC tajmeri su dostupni u tri rezolucije: 1 ms, 10 ms i 100 ms. Kao što se vidi u tabeli 5, broj tajmera određuje njegovu rezoluciju.

U istom programu ne mogu se imati dva ista broja tajmera Txx za TOF i TON tajmere (npr. TON T32 i TOF T32).



Slika 32. Primer primene tajmera.

2.6.8 Podprogramske instrukcije

Poziv podprograma

Instrukcija poziva podprograma (CALL - *Call Subroutine*) prenosi kontrolu na podprogram SBR_N. Može se koristiti sa i bez parametara. Nakon što podprogram završi sa izvršavanjem, upravljanje se vraća na instrukciju koja sledi posle instrukcije poziva podprograma.

Uslovni povratak iz instrukcije podprograma (CRET - *Conditional Return from Subroutine*) prekida podprogram na osnovu logike koja mu prethodi.



Slika 33. Simboli podprogramskih instrukcija.

Iz glavnog programa, podprogrami se mogu umetati do dubine od osam nivoa. Iz rutine prekida se ne mogu umetati podprogrami. Poziv podprograma se ne može smestiti u podprogram pozvan od strane rutine prekida. Rekurzija (podprogram koji poziva sam sebe) nije zabranjena, ali treba biti jako pažljiv u ovakvim situacijama.

STEP 7-Micro/WIN automatski dodaje безусловni povratak iz svakog podprograma.

Zadatak 1

Napisati LAD program za PLC, kojim se na pritisak tastera START startuje motor, a na pritisak tastera STOP isti motor zaustavlja.

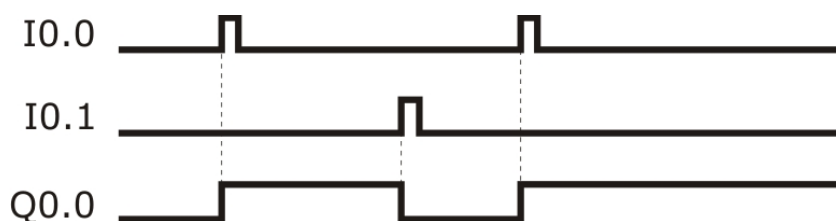
Rešenje

Pre nego što počnemo sa pisanjem programa treba napraviti tabelu promenljivih koje ćemo koristiti u programu. Ulazne promenljive su tasteri START_M i STOP_M (STOP je rezervisano ime i ne može se koristiti, jer program vraća grešku), a izlazna je MOTOR, tabela 6.

| PROMENLJIVA | Naziv | Adresa | Komentar |
|-------------|---------|--------|---------------------------------|
| ULAZNE | START_M | I0.0 | Taster za startovanje motora. |
| | STOP_M | I0.1 | Taster za zaustavljanje motora. |
| IZLAZNE | MOTOR | Q0.0 | Stanje motora. |

Tabela 6. Tabela promenljivih u zadatku 1.

Prilikom pisanja porograma od velike pomoći može biti i vremenski dijagram, slika 34.



Slika 34. Vremenski dijagram za zadatak 1.

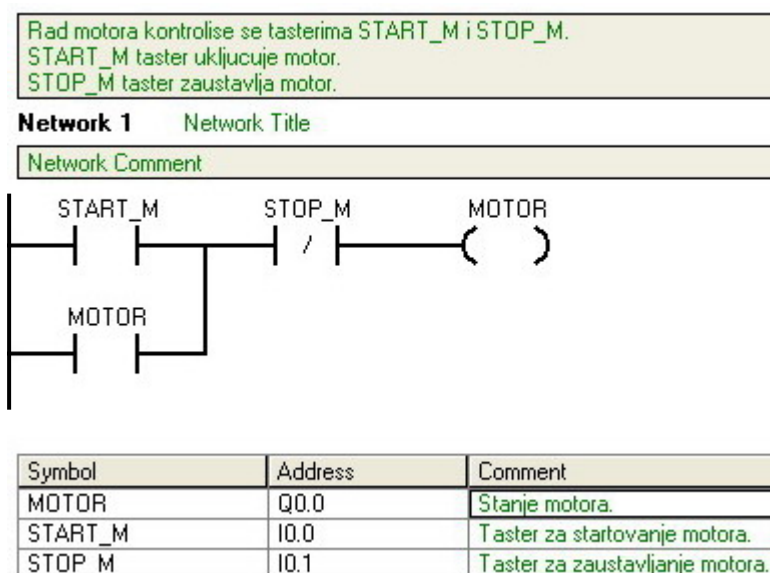
Sa vremenskog dijagrama se vidi da se motor uključuje pritiskom na taster START_M, sa adresom I0.0. U glavnom programu, u *Network*-u 1, dodajemo normalno otvoreni kontakt, iz bit logičkih instrukcija. Iznad njegovog simbola pojavljuju se znaci pitanja gde treba upisati naziv (simbol) elementa, u našem slučaju START_M. Element još uvek nije u potpunosti definisan, jer PLC elemente vezuje za njihove adrese a ne nazive. Desnim klikom na element, izborom opcije 'Define Szmbol...', možemo upisati adresu elementa i njegov komentar.

Da bi motor i nakon prestanka delovanja na taster START_M bio uključen dodajemo granu paralelnu grani u kojoj se nalazi taster za startovanje, i dodajemo kao ulaz stanje motora, tj. Q0.0. Izaz Q0.0 se u ovom slučaju ponaša kao ulaz i 'drži' stanje na izlazu i nakon prestanka delovanja tastera START_M.

Pincip dodavanja paralelnih grana predstavlja, zapravo, logičku ILI funkciju. Dakle, ako je uključen taster START_M, ili je motor radio, motor treba da ostane upaljen. Ako je motor bio ugašen, pritiskom na taster START_M motor se pali, i nakon isključenja tastera START_M, stanje motora (upaljen) `drži` motor i dalje upaljenim.

Nakon toga motor radi, sve do trenutka dok se ne pritisne taster STOP_M, na adresi IO.1. Dakle, u produžetku glavnog programa dodajemo na glavnoj liniji normalno zatvoreni kontakt STOP_M, sa adresom IO.1. Zapravo, ovde se radi o inverznoj logici, ako motor radi, i ako taster STOP_M NIJE uključen, motor treba da ostane upaljen. Ako motor radi, i pritisne se taster STOP_M, tok struje se prekida i motor se gasi. Motor nakon toga ostaje isključen.

Nakon što završimo pisanje programa u STEP 7-Micro/WIN-u, glavni deo programa treba da izgleda kao na slici 35.



Slika 35. Izgled glavnog dela programa u zadatku 1.

Zadatak 2

Napisati porogram za PLC, kojim se motor pokreće pritiskom na taster, i pritiskom na isti taster zaustavlja.

Rešenje

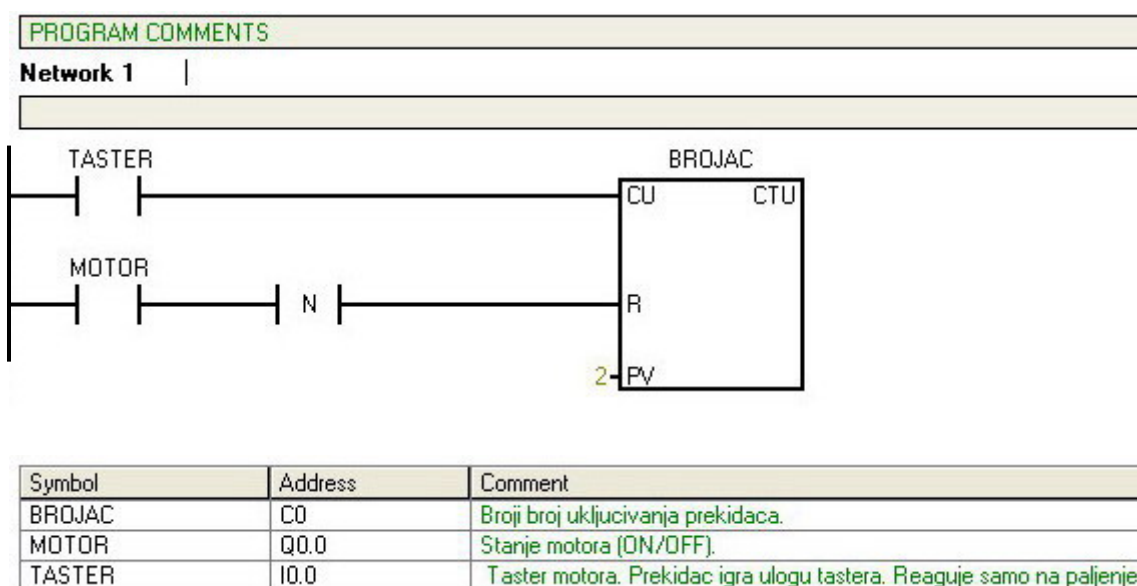
Za rešavanje ovog programa korišćićemo dve promenljive, jednu ulaznu i jednu izlaznu, i jednu pomoćnu promenljivu, u vidu brojača na gore. Promenljive su date u tabeli 7.

| PROMENLJIVA | Naziv | Adresa | Komentar |
|-------------|--------|--------|-------------------------------------|
| ULAZNE | TASTER | I0.0 | *Taster za paljenje/gašenje motora. |
| IZLAZNE | MOTOR | Q0.0 | Stanje motora. |
| BROJAČ | BROJAČ | C0 | Broji broj uključivanja prekidača. |

* Taster je za potrebe provere programa na simulatoru, koji ima samo klasične prekidačke ulaze, izveden kao prekidač (normalno otvoreni kontakt) koji reaguje samo na uzlaznu ivicu (OFF-ON prelaz). Način rada programa se ne bi promeno na realnom modelu sa tasterima.

Tabela 7. Tabela promenljivih u zadatku 2.

Po uključivanju PLC-a, motor je isključen i trenutna vrednost brojača je nula. Na pritisak tastera (ulaz CU brojača izvrši prelaz OFF-ON) uvećava se vrednost brojača za jedan (trenutna vrednost je 1). Sa prelaskom tastera iz OFF u ON stanje (Network 2), S1 ulaz instrukcije dominantnog setovanja SR – MOTOR, dobija signal i motor se pali.

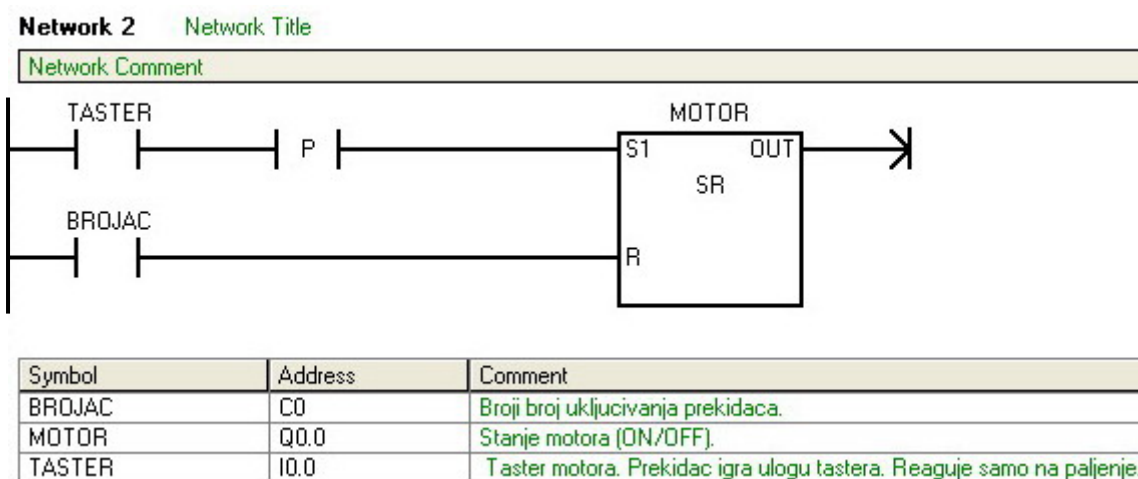


Slika 36. Zadatak 2_Network 1.

Kada prestanemo sa delovanjem na taster, što bi u slučaju običnog prekidača, kakav je korišćen na simulatoru radi provere ispravnosti programa, bilo ekvivalentno vraćenju prekidača u položaj ISKLJUČENO, ništa se ne menja. Motor i dalje radi, trenutna vrednost brojača je 1.

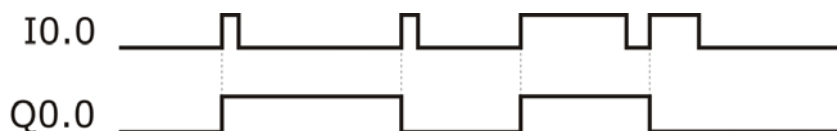
Kada ponovo pritisnemo taster (uključimo prekidač), brojač uvećava vrednost na dva, i propušta signal (Network 1), i time resetuje SR instrukciju, time gaseci i motor.

Gašenje motora u sledećem ciklusu skeniranja (*Network 1*), svojom silaznom ivicom (OFF-ON), resetuje brojač, i sve se vraća na početne uslove.



Slika 37. Zadatak 2_Network 2.

Na slici 38 dat je vremenski dijagram. Jasno se vidi da stanje izlaza, stanje motora (upaljen/isključen), ne zavisi od dužine trajanja ulaza (delovanja tastera).



Slika 38. Vremenski dijagram za zadatak 2.

Zadatak 3

Napisati PLC program kojim će se obezbediti kašnjenje ulaznog signala od 2.5 sekunde.

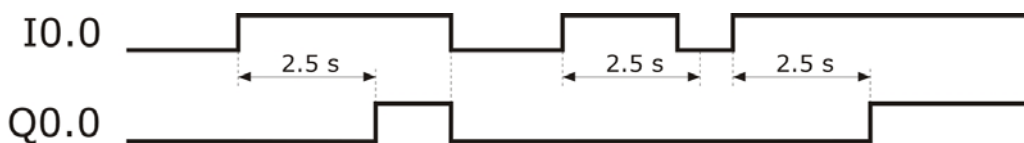
Rešenje

Za rešavanje ovog problema koristiće se jedna ulazna i jedna izlazna promenljiva, i tajmer. U tabeli 8, dati su nazivi i adrese korišćenih promenljivih.

| PROMENLJIVA | Naziv | Adresa | Komentar |
|-------------|--------|--------|---|
| ULAZNE | ULAZ | I0.0 | Ulazni element (prekidač, taster, senzor...). |
| IZLAZNE | IZLAZ | Q0.0 | Izlazni element (sijalica, motor, sirena...). |
| TAJMER | TAJMER | T37 | Tajmer rezolucije 100ms (25 x 100ms = 2.5s). |

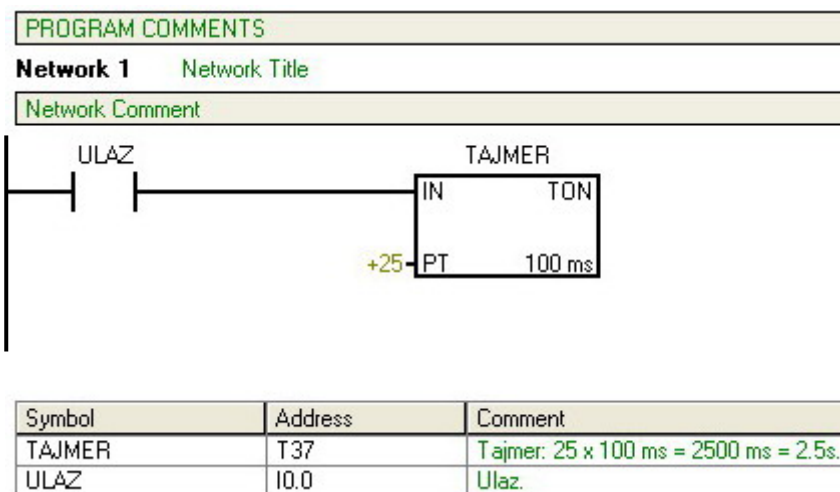
Tabela 8. Tabela promenljivih u zadatku 3.

Vremenski dijagram prikazan je na slici 39.



Slika 39. Vremenski dijagram za zadatak 3.

Nakon uključivanja ulaza (*Network 1*) tajmer počinje da odbrojava inkremente vremena od 100ms. Kada tajmer dostigne vrednost 25 x 100 ms = 2.5 s, uključuje se njegov izlaz TON.



Slika 40. Zadatak 3_Network_1.

Izlaz tajmera u sledećem koraku (*Network 2*) pali izlaz. Nakon gašenja ulaza, izlaz se odmah gasi.



Slika 41. Zadatak 3_Network_2.

Zadatak 4

Napisati LAD program kojim izlazni signal nastaje 5 sekundi posle početka ulaznog signala i koji će delovati još 7 sekundi po presktanku ulaznog signala.

Rešenje

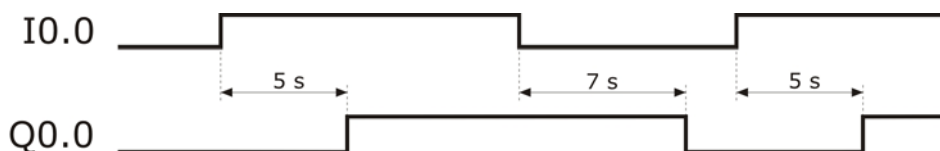
Za rešavanje ovog zadatka koristiće se jedan ulaz, jedan izlaz i dva tajmera. Jedan tajmer definišaće vreme kašnjenja izlaznog signala za ulaznim signalom, a drugi vreme zadržke izlaznog signala (vreme za koje će, nakon prestanka delovanja ulaznog signala, izlazni signal postojati).

U tabeli 9, dati su nazivi i adrese korišćenih promenljivih.

| PROMENLJIVA | Naziv | Adresa | Komentar |
|-------------|---------|--------|--|
| ULAZNE | ULAZ | I0.0 | Ulazni signal (prekidač, taster, senzor...). |
| IZLAZNE | IZLAZ | Q0.0 | Izlazni signal (sijalica, motor, sirena...). |
| TAJMER | T_START | T37 | Startovanje kasni: 50 x 100ms = 5s. |
| | T_STOP | T38 | Zaustavljanje kasni: 70 x 100ms = 7s. |

Tabela 9. Tabela promenljivih u zadatku 4.

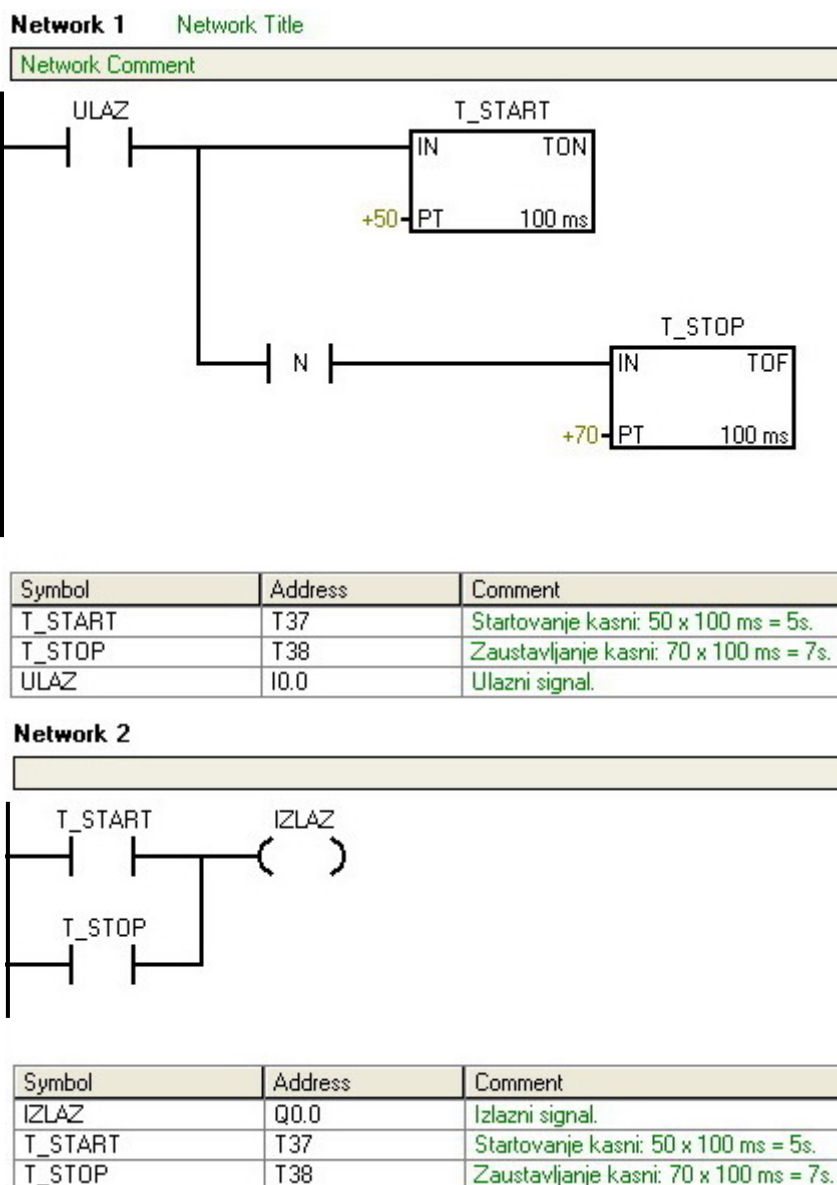
Vremenski dijagram je prikazan na slici 42.



Slika 42. Vremenski dijagram za zadatak 4.

Na pojavu ulaznog signala (*Network 1*), startuje se tajmer T_START, tajmer sa kašnjenjem paljenja (TON), koji definiše vreme kašnjenja izlaznog signala za ulaznim. Nakon isteka predefinisano vremena (50 x 100ms = 5s), T_START će na svom izlazu dati signal, koji će upaliti izlaz Q0.0 (*Network 2*).

U paralelnoj grani, nakon prestanka delovanja ulaznog signala, na negativnu ivicu ulaznog signala (prelaz ON-OFF) staruje se tajmer sa kašnjenjem gašenja (TOF) T_STOP, koji definiše vreme koje prođe od gašenja ulaznog signala do gašenja izlaznog signala. Nakon što istekne predefinisano vreme na ovom tajmeru (70 x 100ms = 7s), njegov izlaz će se ugasi. Kako T_START nema signal (zbog gašenja ulaza, T_START gasi svoj izlaz), i izlaz T_STOP tajmera se ugasio, izlaz se gasi.



Slika 43. Izgled glavnog dela programa u zadatku 4.

Ukoliko bi ulazni signal nestao za kraće vreme od 5 sekundi nakon svog pojavljivanja, izlazni signal se ne bi pojavio.

Zadatak 5

Napisati LAD program koji ponavlja izlazne signale u određenim vremenskim intervalima, kada se na ulaz dovodi signal.

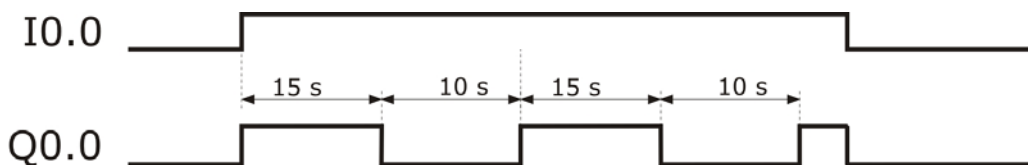
Rešenje

Za rešavanje ovog zadatka koristiće se jedan ulaz, jedan izlaz i dva tajmera (tabela 10). Jedan tajmer definišaće vreme za koje je izlazni signal uključen, a drugi, vreme za koje je izlaz isključen.

| PROMENLJIVA | Naziv | Adresa | Komentar |
|-------------|------------|--------|--|
| ULAZNE | ULAZ | I0.0 | Ulazni signal (prekidač, taster, senzor...). |
| IZLAZNE | IZLAZ | Q0.0 | Izlazni signal (sijalica, motor, sirena...). |
| TAJMER | TAJMER_ON | T37 | Izlaz je uključen (ON): 15 x 100ms = 15s. |
| | TAJMER_OFF | T38 | Izlaz je isključen (OFF): 10 x 100ms = 10s. |

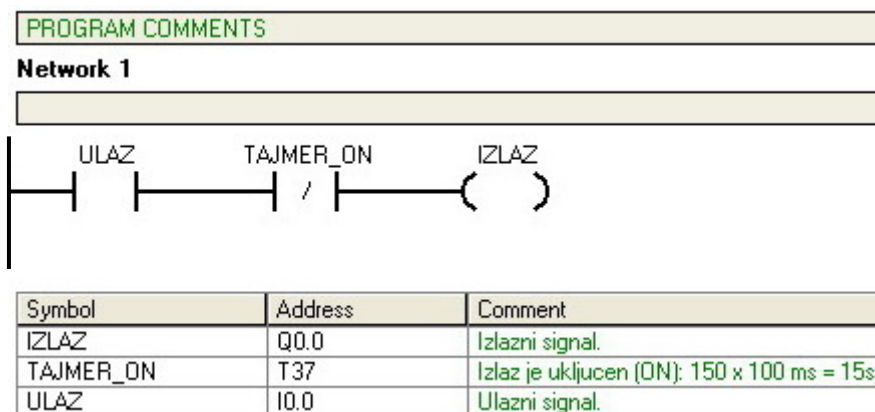
Tabela 10. Tabela promenljivih u zadatku 5.

Vremenski dijagram je prikazan na slici 44.



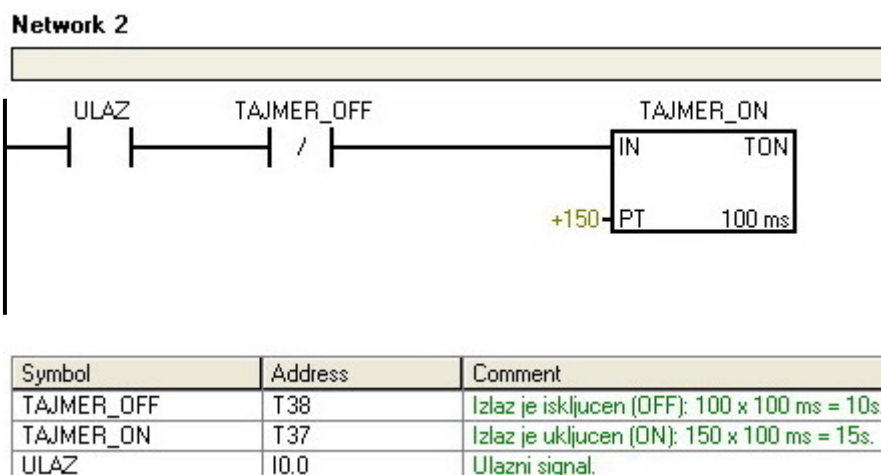
Slika 44. Vremenski dijagram za zadatak 5.

Nakon uključenja ulaznog signala, stanje TAJMER_ON tajmera je OFF, izlaz se pali (Network 1), stanje tajmera TAJMER_OFF je takođe OFF, pa se startuje TAJMER_ON (Network 2).



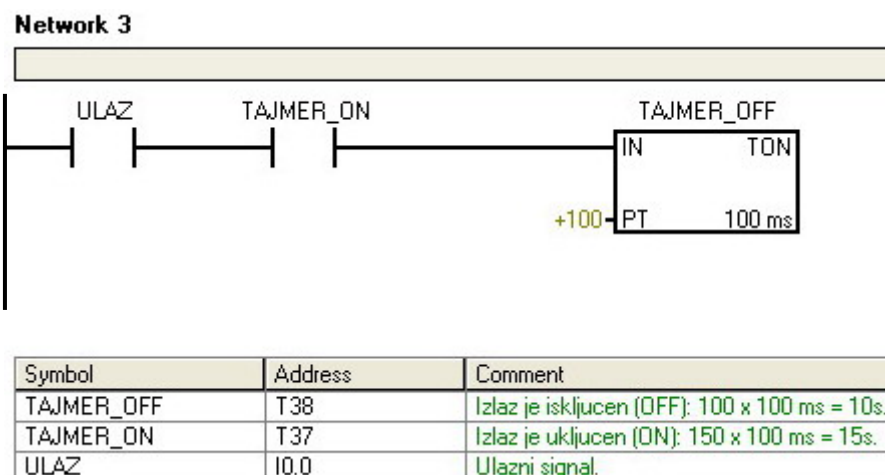
Slika 45. Zadatak 5_Network 1.

Kada TAJMER_ON, nakon 15 sekundi, uključi svoj izlaz, pali se i tajmer TAJMER_OFF (Network 3), i odbrojava 10 sekundi. Izlaz se gasi jer je ulaz uključen, a TAJMER_ON ima stanje logičke jedinice, pa normalno zatvoren kontakt prekida tok struje (Network 1).



Slika 46. Zadatak 5_Network 2.

Kada tajmer TAJMER_OFF, nakon deset sekundi, uključi svoj izlaz, TAJMER_ON se resetuje, prelazi u stanje logičke nule (OFF) i počinje sa odbrojavanjem novih 15 sekundi (Network 2). Isključeni tajmer TAJMER_ON ponovo pali izlaz (Network 1).



Slika 47. Zadatak 5_Network 3.

Signal na izlazu gasi se onog trenutka kada nestane ulazni signal, bez obzira u kom stanju su tajmeri, ili izlaz.

Zadatak 6

Napisati LAD program koji omogućava duže kašnjenje signala na izlazu od maksimalnog vremena kašnjenja jednog tajmera.

Rešenje

Razmotrimo najpre koja su ograničenja SIEMENS-ovih PLC-ova kada je reč o tajmerima i brojačima.

maksimalna vrednost brojača tajmera - 32767

maksimalna dužina jednog vremenskog odbirka tajmera (rezolucija) – 100ms

maksimalno vreme kašnjenja sa jednim tajmerom:

$$32767 \times 100 \text{ ms} = 3276.7 \text{ s} = 54 \text{ min } 36.7 \text{ s}$$

Ukoliko se ukaže potreba za ostvarivanjem dužeg kašnjenja izlaznog signala za ulaznim, kao u ovom zadatku, problem se može rešiti uvođenjem pomoćnog brojača, koji ima ulogu umnožavanja maksimalnog vremena kašnjenja jednog tajmera (vrednost brojača je množilac vremena kašnjenja dobijenog tajmerom).

Razmotrimo sada ograničenja PLC-a sa programom koji ima jedan tajmer i jedan brojač.

maksimalna vrednost brojaca – 32767

maksimalno vreme kašnjenja sa programom:

$$32767 \times 32767 \times 100 \text{ ms} = 107069449.2 \text{ s} = \text{oko } 3.5 \text{ godina}$$

Ukoliko je potrebno postići još veća kašnjenja (!) moguće je najjednostavnije dodati potreban broj brojača.

U tabeli 11, dati su nazivi i adrese korišćenih promenljivih.

| PROMENLJIVA | Naziv | Adresa | Komentar |
|-------------|--------|--------|--|
| ULAZNE | ULAZ | I0.0 | Ulazni signal (prekidač, taster, senzor...). |
| IZLAZNE | IZLAZ | Q0.0 | Izlazni signal (sijalica, motor, sirena...). |
| TAJMER | TAJMER | T37 | Period koji množimo (max 3276.7s). |
| BROJAČ | BROJAČ | C0 | Broj perioda tajmera (max 32767). |

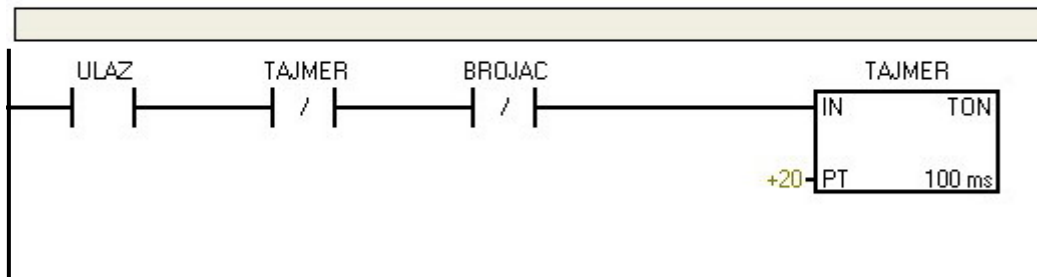
Tabela 11. Tabela promenljivih u zadatku 6.

Nakon uključivanja ulaza, TAJMER i BROJAČ su isključeni (OFF), pali se TAJMER (*Network 1*), i odbrojava predefinisano vreme kašnjenja koje množimo sa brojem koji zadajemo BROJAČU (u ovom primeru $20 \times 100\text{ms} =$

2s). Nakon isteka prvog perioda tajmera, pali se njegov izlaz i trenutna vrednost brojača se povećava za jedan (*Network 2*), ali se u isto vreme ulaz tajmera gasi, tajmer se resetuje i ponovo počinje odbrojavanje (*Network 1*).

Program omogućava kasnjenje izlaza u periodu dužem od maksimalnog vremena kasnjenja koje je moguće ostvariti sa jednim tajmerom, povećavanjem vrednosti brojača. Vremensko kasnjenje je jednako proizvodu vrednosti brojača i tajmera.
 32767 je maksimalna vrednost brojača tajmera
 100 ms je maksimalna dužina jednog vremenskog odbirka tajmera (tajmer ima rezolucije: 1ms, 10ms i 100ms)
 32767 je maksimalna vrednost brojača
 $32767 \times 100 \text{ ms} = 3276.7 \text{ s} = 54 \text{ min } 36.7 \text{ s}$ - maksimalno vreme kasnjenja sa jednim tajmerom
 $32767 \times 32767 \times 100 \text{ ms} = 107069449.2 \text{ s} = \text{oko } 3.5 \text{ godina}$ - maksimalno vreme kasnjenja sa programom.

Network 1

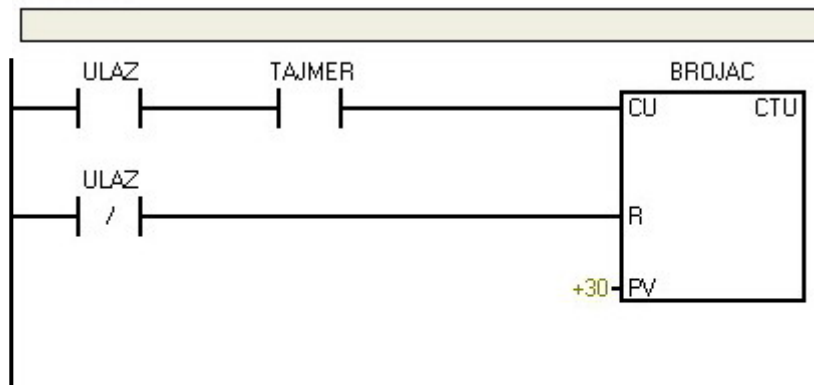


| Symbol | Address | Comment |
|--------|---------|-----------------------|
| BROJAC | C0 | Broj perioda tajmera. |
| TAJMER | T37 | Period koji mnozimo. |
| ULAZ | I0.0 | Ulaz. |

Slika 48. Zadatak 6_Network 1.

Ovaj proces se ponavlja sve dok trenutna vrednost brojača ne dostigne predefinisanu vrednost, kada se pali izlaz brojača (*Network 2*), i izlaz Q0.0. Od tog trenutka izlaz postoji sve dok se ne promeni stanje ulaza, tj. dok se ulaz ne ugasi.

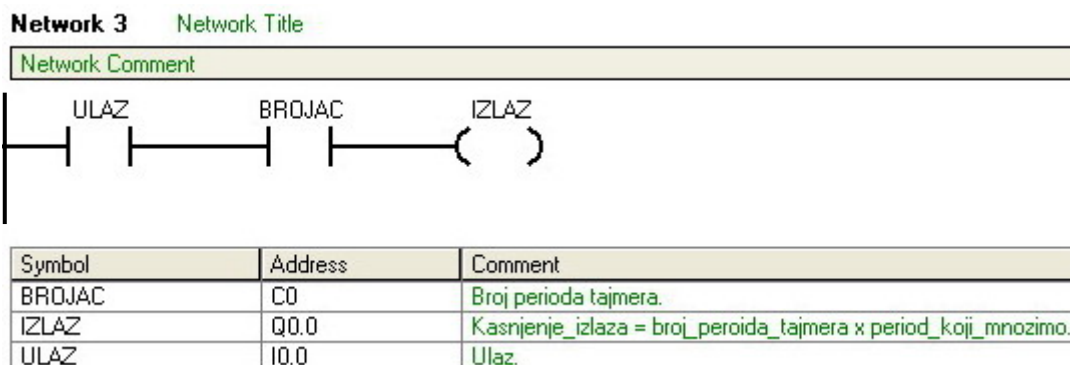
Network 2



| Symbol | Address | Comment |
|--------|---------|-----------------------|
| BROJAC | C0 | Broj perioda tajmera. |
| TAJMER | T37 | Period koji mnozimo. |
| ULAZ | I0.0 | Ulaz. |

Slika 49. Zadatak 6_Network 2.

Gašenjem ulaza resetuje se trenutna vrednost brojača (*Network 2*), i ceo proces, ponovnim paljenjem ulaza može da počne.

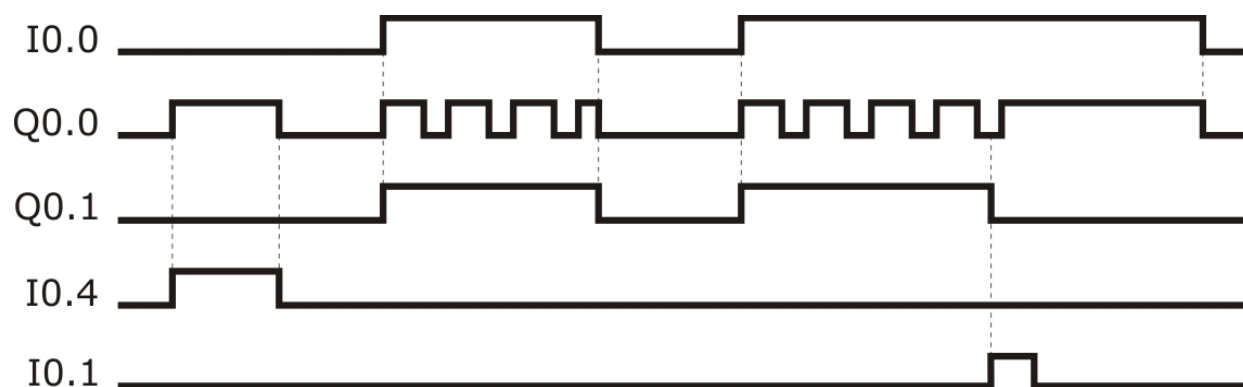


Slika 50. Zadatak 6_Network 3.

Zadatak 7

Kao posledica uključenja alarma, pali se alarmna sijalica (signal sa prekidima – treptanje) i alarmna sirena (kontinualan signal). Pritiskom na reset taster (konstatovano je da postoji kvar), alarmna sirena se isključuje, a alarmna sijalica radi kontinualno dok se ne otkloni uzrok alarma.

Napisati LAD program koji na način prikazan na vremenskom dijagramu pali i gasi alarmno svetlo i sirenu.



Slika 51. Vremenski dijagram za zadatak 7.

Potrebno je predvideti i postojanje tastera za testiranje alarmne sijalice.

Rešenje

U tabeli 12, dati su nazivi i adrese korišćenih promenljivih.

| PROMENLJIVA | Naziv | Adresa | Komentar |
|-------------|---------------|--------|---|
| ULAZNE | ALARM | I0.0 | Alarm za postojanje kvara. |
| | RESET | I0.1 | Taster za resetovanje sirene. |
| | TEST_SIJALICE | I0.4 | Taster za proveru ispravnosti sijalice. |
| IZLAZNE | SIJALICA | Q0.0 | Svetlosni signal postojanja kvara. |
| | SIRENA | Q0.1 | Zvučni signal za postojanje kvara. |
| TAJMER | TAJMER_1 | T37 | Definiše dužinu trajanja uključenog stanja sijalice. |
| | TAJMER_2 | T38 | Definiše dužinu trajanja isključenog stanja sijalice. |
| BROJAČ | BROJAČ | C0 | Pomoćni element za rad RESET tastera. |

Tabela 12. Tabela promenljivih u zadatku 7.

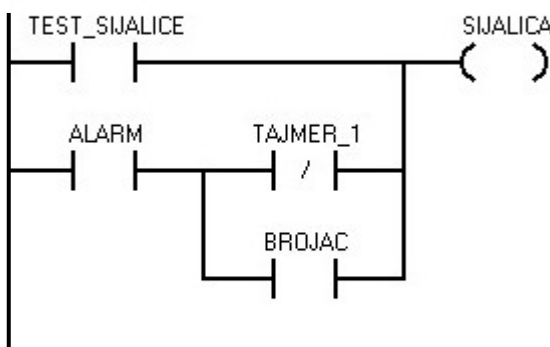
Pre puštanja u pogon, iz bezbedonosnih razloga, proverava se ispravnost alarmne sijalice. Pritiskom na taster TEST_SIJALICE, SIJALICA svetli.

Stanje tajmera i brojača je nula (OFF). Usled nastanka ALARMA, pali se SIJALICA (*Network 1*) i alarmna SIRENA (*Network 2*). Pali se TAJMER_1 (*Network 3*), i odbrojava predefinisanu vrednost koja određuje dužinu trajanja uključenog stanja sijalice (u ovom slučaju: $50 \times 100 \text{ ms} = 5 \text{ s}$).

Posledica uključenja ALARMA je paljenje alarmne SIJALICE sa prekidima (treperenje) i alarmne SIRENE. Pritiskom na RESET taster alarmna SIRENA se gasi, a alarmna SIJALICA pocinje da radi kontinualno dok se ne otkloni uzrok alarma. Taster TEST_SIJALICE služi za proveru ispravnosti sijalice.

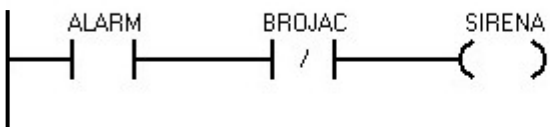
Network 1 Network Title

Network Comment



| Symbol | Address | Comment |
|---------------|---------|---|
| ALARM | I0.0 | Alarm za postojanje kvara. |
| BROJAC | C0 | Pomocni element za rad RESET tastera. |
| SIJALICA | Q0.0 | Alarmna sijalica - svetlosni signal postojanja kvara. |
| TAJMER_1 | T37 | Definise duzinu trajanja ukljucenog stanja sijalice. |
| TEST_SIJALICE | I0.4 | Taster za proveru ispravnosti sijalice. |

Network 2



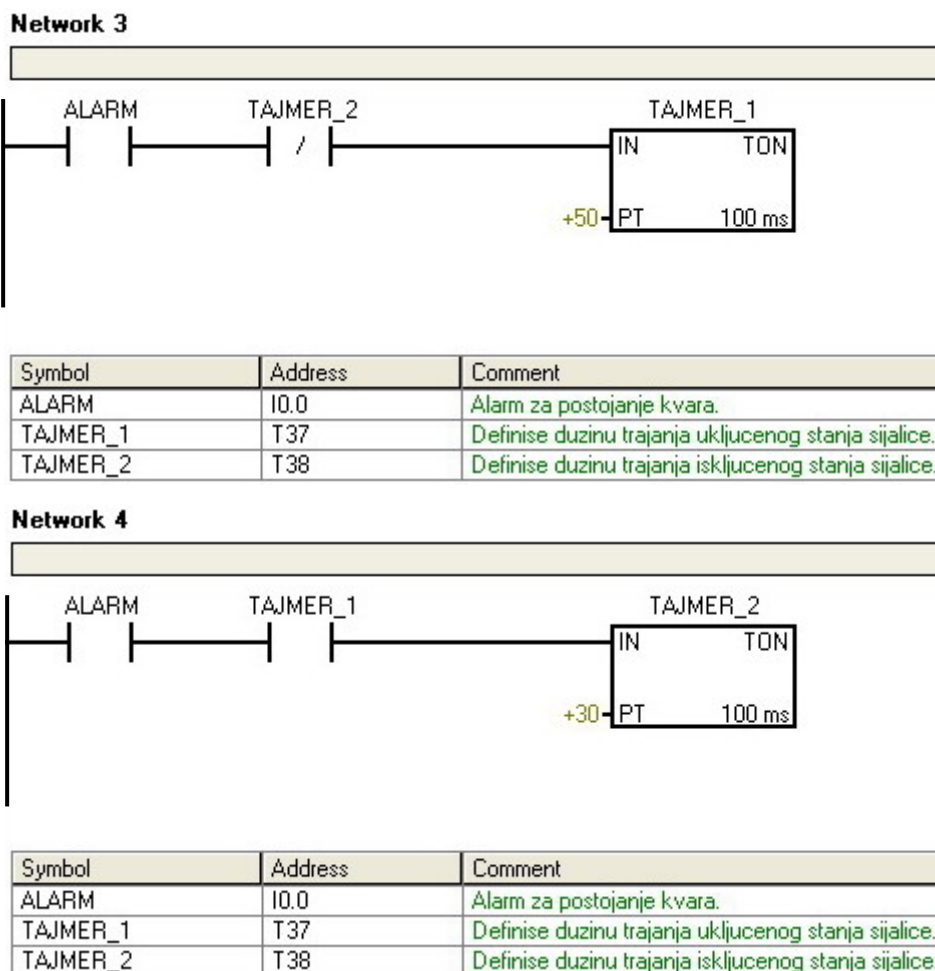
| Symbol | Address | Comment |
|--------|---------|---------------------------------------|
| ALARM | I0.0 | Alarm za postojanje kvara. |
| BROJAC | C0 | Pomocni element za rad RESET tastera. |
| SIRENA | Q0.1 | Zvucni signal za postojanje kvara. |

Slika 52. Zadatak 7_Network 1 i 2.

Nakon 5 sekundi, TAJMER_1 pali izaz TON (OFF u ON), gasi se SIJALICA (*Network 1*), startuje se TAJMER_2 (*Network 4*), i odbrojava predefinisanu vrednost koja određuje dužinu trajanja isključenog stanja sijalice (u ovom slučaju: $30 \times 100 \text{ ms} = 3 \text{ s}$).

Nakon 3 sekunde, sijalica je ugašena, TAJMER_2 pali svoj izlaz (*Network 4*), TAJMER_1 gasi svoj izlaz, vrednost njegovog brojača se

resetuje (*Network 3*), TAJMER_2 gasi svoj izlaz i takođe resetuje vrednost svog brojača (*Network 4*), i sijalica se ponovo pali (*Network 1*).



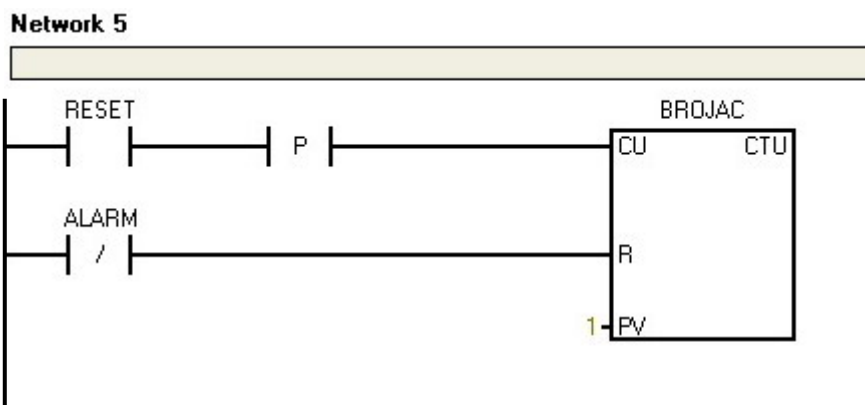
Slika 53. Zadatak 7_Network 3 i 4.

Ovaj proces se nastavlja do trenutka prestanka postojanja alarma (uzrok kvara je otklonjen ili je prestao da postoji) ili do trenutka pritiska RESET tastera. Sve vreme SIRENA ostvaruje kontinualni rad.

Odmah po pritisku tastera RESET, startuje se BROJAČ koji povećava svoju trenutnu vrednost na 1. Ovo je ujedno i vrednost za poređenje (predefinisana vrednost) pa se pali njegov izlaz (*Network 5*). Paljenje izlaza brojača drži sijalicu stalno upaljenom, bez obzira na stanje tajmera (*Network 1*), i takođe gasi sirenu (*Network 2*).

Ovo stanje se zadržava sve do prestanka postojanja alarma. Po ponovnom pojavljivanju alarma ceo proces se ponavlja počevši od početka (nema zadržke informacija – pamćenja).

Brojač je uveden da se omogući sprečavanje slučajnog isključenja sirene. Povećavanjem predefinisane vrednosti brojača, operater se obavezuje da određeni broj puta pritisne RESET taster kako bi isključio sirenu.



| Symbol | Address | Comment |
|--------|---------|---------------------------------------|
| ALARM | I0.0 | Alarm za postojanje kvara. |
| BROJAC | C0 | Pomocni element za rad RESET tastera. |
| RESET | I0.1 | Taster za resetovanje sirene. |

Slika 54. Zadatak 7_Network 5.

Zadatak 8

Nož se koristi za isecanje tri vrste proizvoda: A, B i C. Zbog istupljenosti, menja se posle odsecanja 1000 komada A, 500 komada B ili 100 komada C. Delovi mogu nasumično dolaziti na red za isecanje. Kada se nož istupi, aktivira se zujalica.

Za izvođenje ovog zadatka koriste se tri senzora, svaki za po jednu vrstu proizvoda, i jedan senzor koji je indikator završene operacije sečenja. Tasterom se startuje celokupan proces.

Rešenje

Kako delovi dolaze nasumično mora se utvrditi 'težina' koju sečenje nekog od proizvoda nosi. Ako se kao zbirni koeficijent i cifra merodavna za indikaciju istupljenosti noža uzme cifra 1000, koeficijenti sečenja određenog proizvoda su:

- proizvod A – 1,
- proizvod B – 2,
- proizvod C – 10.

Množenjem broja isečenih proizvoda i njihovih odgovarajućih koeficijenata, i njihovim sabiranjem i poređenjem sa 1000 dobijamo odgovor da li je nož istupljen i da li ga treba menjati ili ne.

Na ovaj način nakon sečenja npr. 9 proizvoda C, 4 proizvoda B i 2 proizvoda A imaćemo:

$$90 \times 10 + 40 \times 2 + 20 \times 1 = 1000.$$

Dakle treba signalizirati da je nož istupljen.

U tabeli 13, dati su nazivi i adrese korišćenih promenljivih.

| PROMENLJIVA | Naziv | Adresa | Komentar |
|-------------|----------|--------|--|
| ULAZNE | START | I0.0 | Startovanje mašine. |
| | RESET | I0.1 | Taster za resetovanje brojača nakon zamene noža. |
| | S_A | I0.4 | Senzor proizvoda A. |
| | S_B | I0.5 | Senzor proizvoda B. |
| | S_C | I0.6 | Senzor proizvoda C. |
| IZLAZNE | ZUJALICA | Q0.0 | Signal istrošenog noža. |
| | MAŠINA | Q0.1 | Status rada mašine. |
| | SEČENJE | Q0.3 | Operacija sečenja je u toku. |

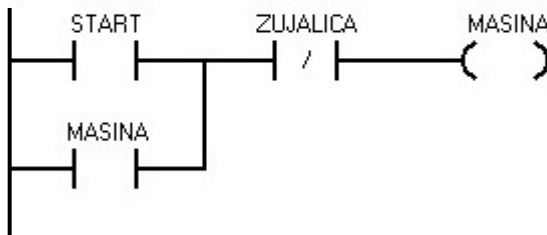
Tabela 13. Tabela promenljivih u zadatku 8.

Po startovanju mašine, ukoliko nije uključena zujalica, dakle nož nije istupljen, mašina se startuje (*Network 1*), i nakon registrovanja nekog od proizvoda, putem senzora, sečenje može da počne (*Network 2*).

Nož se koristi za isecanje tri vrste proizvoda: A, B i C. Zbog istupljenosti, menja se posle 1000 odsecanja komada A, 500 komada B ili 100 komada C. Delovi mogu nasumicno dolaziti na red za isecanje. Kada se nož istupi, aktivira se zujalica. Za izvođenje ovakvog zadatka koriste se tri senzora, svaki za po jednu vrstu komada, i jedan senzor koji je indikator završene operacije sečenja. Tasterom se startuje celokupan proces.

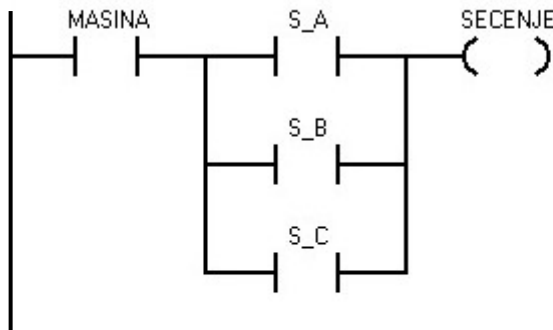
Network 1 Network Title

Network Comment



| Symbol | Address | Comment |
|----------|---------|-------------------------|
| MASINA | Q0.1 | Status rada masine. |
| START | I0.0 | Startovanje masine. |
| ZUJALICA | Q0.0 | Signal istrosenog noza. |

Network 2

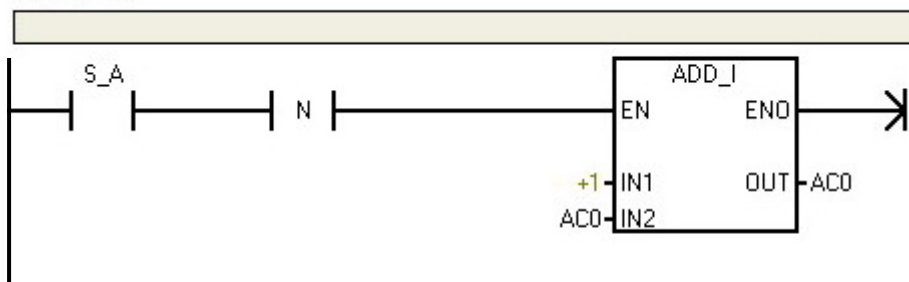


| Symbol | Address | Comment |
|---------|---------|------------------------------|
| MASINA | Q0.1 | Status rada masine. |
| S_A | I0.4 | Senzor proizvoda A. |
| S_B | I0.5 | Senzor proizvoda B. |
| S_C | I0.6 | Senzor proizvoda C. |
| SECENJE | Q0.3 | Operacija sečenja je u toku. |

Slika 55. Zadatak 8_Network 1 i 2.

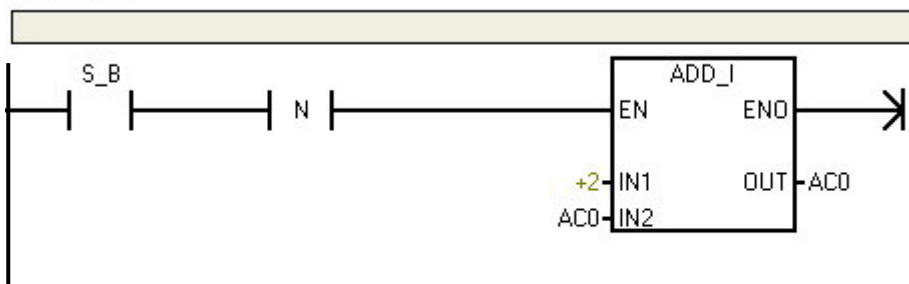
Odlaskom proizvoda (na silaznu ivicu signala senzora) u odgovarajućoj grani programa se, shodno vrsti proizvoda čije je sečenje registrovano, na vrednost unutrašnjeg brojača se dodaje vrednost koeficijenta sečenja: za proizvod A - jedan (*Network 3*), za proizvod B - dva (*Network 4*), za proizvod C - deset (*Network 5*).

Network 3



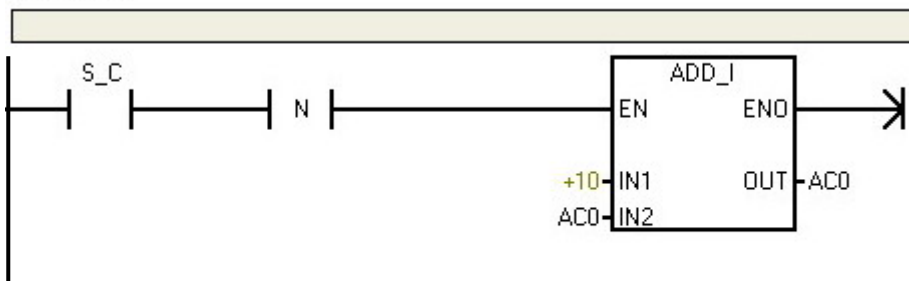
| Symbol | Address | Comment |
|--------|---------|---------------------|
| S_A | I0.4 | Senzor proizvoda A. |

Network 4



| Symbol | Address | Comment |
|--------|---------|---------------------|
| S_B | I0.5 | Senzor proizvoda B. |

Network 5

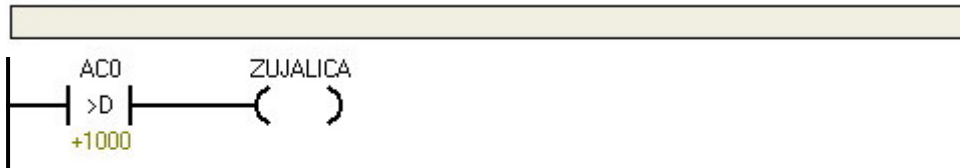


| Symbol | Address | Comment |
|--------|---------|---------------------|
| S_C | I0.6 | Senzor proizvoda C. |

Slika 56. Zadatak 8_Network 3, 4 i 5.

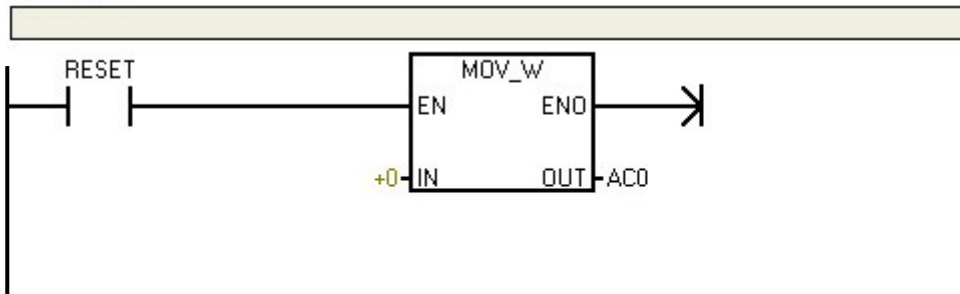
U sledećem koraku se proverava da li je trenutna vrednost unutrašnjeg brojača veća od 1000, i ako jeste, uključuje se zujalica (*Network 6*). Uključenje zujelice zaustavlja mašinu (*Network 1*) i time zahteva od operatera da nož zameni. Nakon zamene noža neophodno je protisnuti taster RESET, da bi se unutrašnjem brojaču dodelila vrednost nula, kako bi mogao da nastavi sa odbrojavanjem (*Network 7*).

Network 6



| Symbol | Address | Comment |
|----------|---------|-------------------------|
| ZUJALICA | Q0.0 | Signal istrošenog noza. |

Network 7

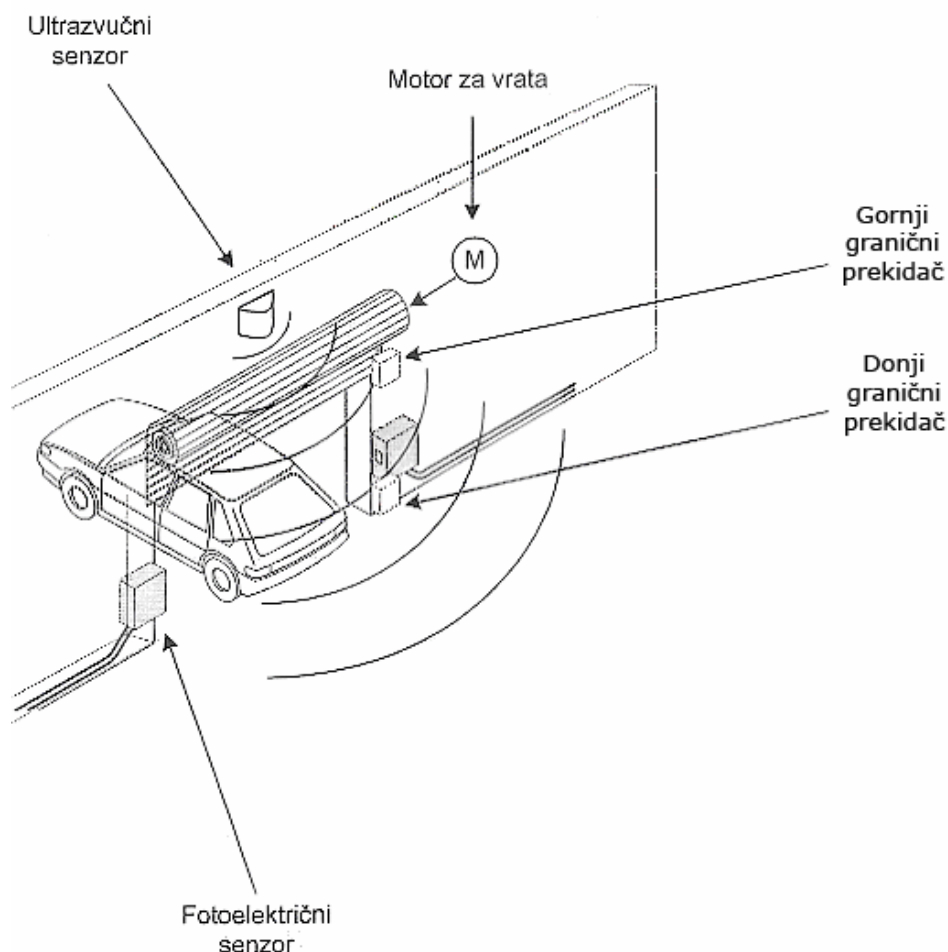


| Symbol | Address | Comment |
|--------|---------|--|
| RESET | I0.1 | Taster za resetovanje brojaca nakon zamene noza. |

Slika 57. Zadatak 8_Network 6 i 7.

Zadatak 9

Napisati LAD program koji se upravlja otvaranjem i zatvaranjem vrata skladišta. Kao detektor vozila koristi se ultrasonični senzor/prekidač. Odvojeni fotosenzori detektuju prolaz vozila preko svetlosnog zraka. Kao odziv ovih signala, upravljačka kola generišu signale koji pogone motor koji spušta ili podiže vrata.



Slika 58. Šema upravljanja podizanjem i spuštanjem vrata.

Rešenje

Dolazak vozila detektuje ultrasonični senzor (UltraSonic_switch) i, ukoliko Senzor_Gornjeg_Položaja – SGP nije uključen (vrata nisu u krajnjem gornjem položaju), šalje signal za uključivanje motora koji pokreće vrata na gore – MOTOR_G (*Network 1*). Paralelna grana održava motor uključenim sve dok se Senzor_Gornjeg_Položaja – SGP ne uključi.

Kada vrata dostignu krajnji gornji položaj, uključuje se Senzor_Gornjeg_Položaja – SGP, koji startuje TAJMER (*Network 2*). Tajmer omogućava da vrata po dostizanju krajnjeg gornjeg položaja ne nastave

odmah kretanje na dole, već da se zadrže u gornjem položaju neko vreme (u ovom slučaju 10 sekundi).

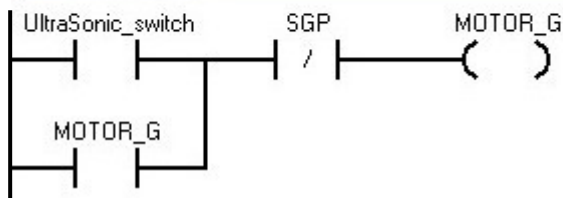
| PROMENLJIVA | Naziv | Adresa | Komentar |
|-------------|-------------------|--------|---|
| ULAZNE | UltraSonic_switch | I0.0 | Detektuje dolazak vozila, tereta, itd. |
| | SGP | I0.1 | Senzor Gornjeg Položaja vrata. |
| | Foto_senzor | I0.2 | Detektuje prolazak vozila, tereta, itd. |
| | SDP | I0.3 | Senzor Donjeg Položaja vrata. |
| IZLAZNE | MOTOR_G | Q0.0 | Motor pokreće vrata na gore. |
| | MOTOR_D | Q0.1 | Motor pokreće vrata na dole. |
| TAJMER | TAJMER | T37 | Vreme zadržavanja vrata u gornjem položaju. |

Tabela 14. Tabela promenljivih u zadatku 9.

Program upravlja podizanjem/spuštanjem vrata na dolazak/odlazak vozila, tereta, itd. Detektor prilaska objekta je ultrasonični prekidač. Fotosenzori detektuju prolazak vozila preko svetlosnog zraka. Tajmer omogućava da se podesi vreme nakon koga ce nakon prolaska objekta vrata krenuti da se spustaju.

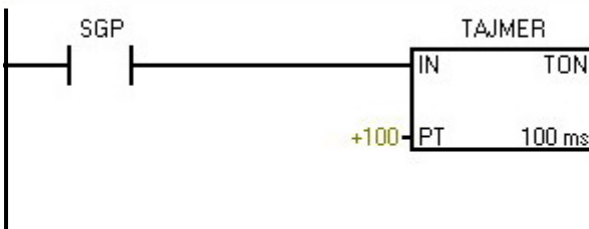
Network 1 Network Title

Network Comment



| Symbol | Address | Comment |
|-------------------|---------|--|
| MOTOR_G | Q0.0 | Motor pokrece vrata na gore. |
| SGP | I0.1 | Senzor Gornjeg Polozaja vrata. |
| UltraSonic_switch | I0.0 | Detektuje dolazak vozila, tereta, itd. |

Network 2



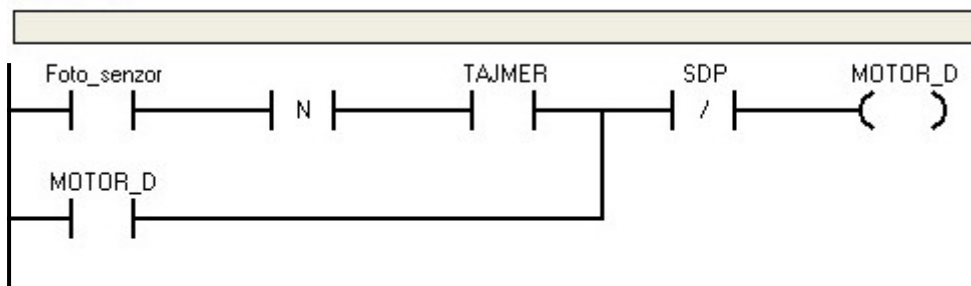
| Symbol | Address | Comment |
|--------|---------|---|
| SGP | I0.1 | Senzor Gornjeg Polozaja vrata. |
| TAJMER | T37 | Vreme zadržavanja vrata u gornjem položaju. |

Slika 59. Zadatak 9_Network 1 i 2.

Nakon isteka 10 sekundi i prolaska vozila, što se detektuje Foto_senzor-om (silazna ivica signala Foto_senzor-a), staruje se motor koji spušta vrata (*Network 3*). Po dolasku vrata u krajnji donji položaj, Sensor_Donjeg_Položaja – SDP daje signal i vrata prestaju sa spuštanjem.

Moguće je dodati još jedan tajmer kojim bi se definisalo vreme koje treba da prođe od trenutka prolaska vozila do trenutka odpočinjanja spuštanja vrata.

Network 3

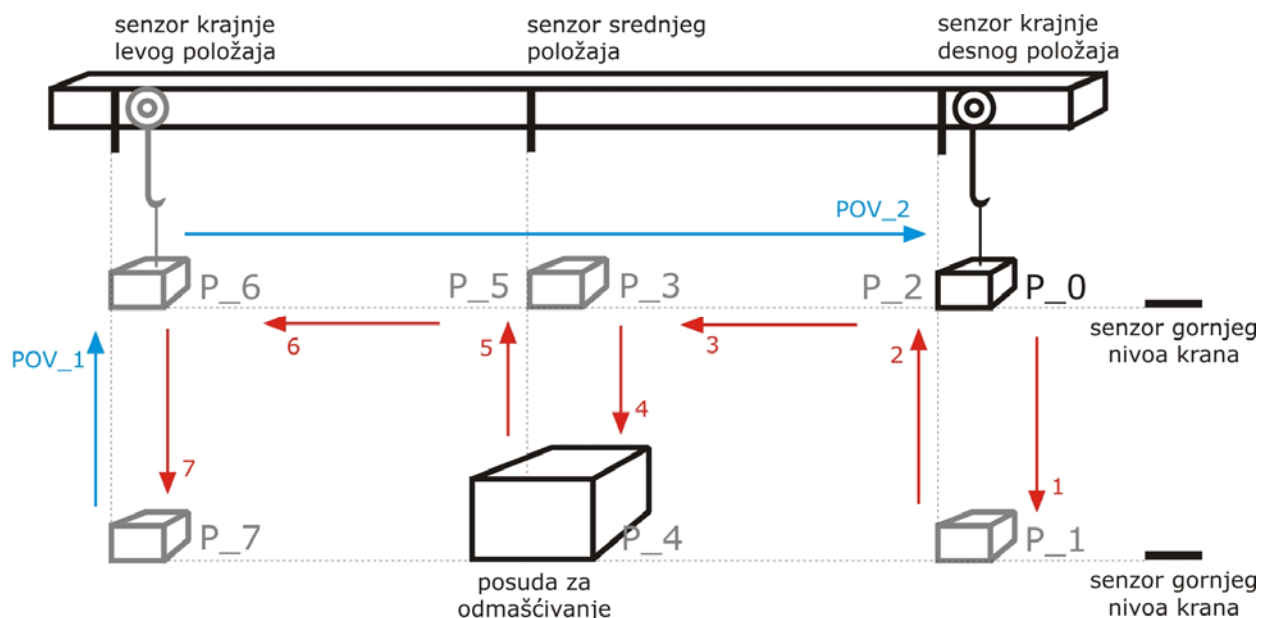


| Symbol | Address | Comment |
|-------------|---------|---|
| Foto_senzor | I0.2 | Detektuje prolazak vozila, tereta, itd. |
| MOTOR_D | Q0.1 | Motor pokrece vrata na dole. |
| SDP | I0.3 | Senzor Donjeg Polozaja vrata. |
| TAJMER | T37 | Vreme zadržavanja vrata u gornjem položaju. |

Slika 60. Zadatak 9_Network 3.

Zadatak 10

Napisati program za PLC kojim se upravlja kranom za proces odmaščivanja mašinskih delova prema postupku datom na slici 61.



Slika 61. Putanja mašinskog dela.

Rešenje

Kran počinje kretanje iz nultog položaja, P_0 – krajnji desni, gornji položaj. Po uključivanju procesa pomoću prekidača START, mašinu je potrebno resetovati, pritiskom na RESET taster. Potom kran kreće na dole, putanjom 1 (putanje su naznačene crvenim strelicama, a njihov redosled crvenim brojevima). Kada dođe u krajnji donji položaj, P_1, pali se senzor donjeg nivoa kрана S_D, sa adresom IO.3. Kretanje prestaje, a programom je predviđeno postojanje elektro-magneta, ili drugog mehanizma koji bi ostvario vezu između kрана i mašinskog dela, koji se pali. Kran zatim nastavlja kretanje na gore, ka položaju P_2, nakon čega se pali senzor gornjeg nivoa kрана, a kran nastavlja kretanje ka položaju P_3. Senzor srednjeg položaja zaustavlja dalje kretanje kрана u levo, i pali se motor na dole, ka položaju P_4. Po dostizanju krajnjeg donjeg položaja, deo se nalazi u posudi za odmaščivanje, i tu se zadržava onoliko koliko je potrebno da se proces odmaščivanja završi. Vreme zadržavanja dela u posudi za odmaščivanje se reguliše pomoću TAJMERA. Nakon toga kran kreće na gore ka položaju P_5, zatim P_6, i na kraju ka P_7. U ovom položaju elektro-magnet se isključuje i predmet se odvaja od kрана. Kran se potom vraća u početni, nulti položaj putanjom označenom plavim strelicama: POV_1, i zatim POV_2.

U tabeli 15, dati su nazivi i adrese korišćenih promenljivih.

| PROMENLJIVA | Naziv | Adresa | Komentar |
|-------------|--------|--------|---|
| ULAZNE | START | I0.0 | Prekidač za startovanje kрана. |
| | RESET | I0.1 | Vraćanje kрана u početni položaj 0. |
| | S_1 | I0.4 | Senzor krajnje desnog položaja. |
| | S_2 | I0.5 | Senzor srednjeg položaja. |
| | S_3 | I0.6 | Senzor krajnje levog položaja. |
| | S_G | I0.2 | Senzor gornjeg nivoa kрана. |
| | S_D | I0.3 | Senzor donjeg nivoa kрана. |
| IZLAZNE | M_G | Q0.0 | Motor na GORE. |
| | M_D | Q0.1 | Motor na DOLE. |
| | M_L | Q0.2 | Motor na LEVO. |
| | M_R | Q0.3 | Motor na DESNO. |
| | E_M | Q0.4 | Elektro-magnet ili drugi mehanizam hvatanja predmeta. |
| TAJMER | TAJMER | T37 | Odredjuje vreme zadržavanja u posudi za odmašćivanje. |

Tabela 15. Tabela promenljivih u zadatku 10.

Program korišćenjem pomoćnog brojača, AC0, i senzora, registruje u kom položaju se nalazi kran. Pri postizanju svakog od definisanih položaja, što se registruje kombinacijom senzora, trenutna vrednost brojača AC0 se uvećava za jedan. Dakle, trenutne vrednosti brojača AC0 odgovaraju položaju kрана u toku kretanja. Na taj način je omogućeno upravljanje motorima.

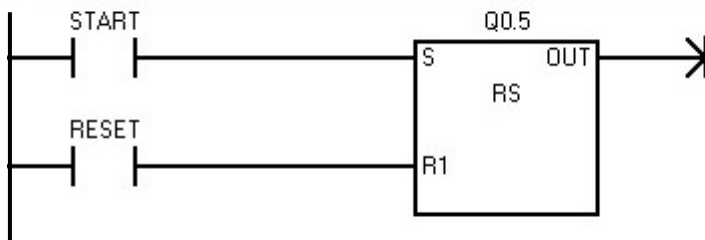
Startovanje celog procesa zahteva resetovanje, da bi se brojaču AC0 dodelila trenutna vrednost 0(nula). U slučaju da se kran nalazi u bilo kom položaju osim u nultom, od operatera se zahteva da pritiskom na taster RESET, kran dovede u početni, nulti položaj, P_0. Takođe, u slučaju kvara, ili neke druge situacije, pritiskom na taster RESET, kran se zaustavlja, bez obzira na trenutni položaj i vraća se u položaj P_0.

Kada kran dostigne položaj P_7, brojač se resetuje, dobija vrednost nula, a kran se vraća u položaj P_0, i cela operacija može da se ponovi.

Na sledećim slikama dat je izgled kompletnog programa.

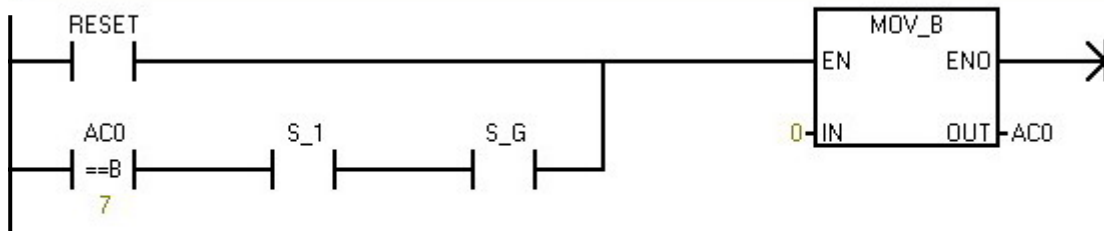
Program za PLC kojim se upravlja kranom za proces odmasćivanja masinskih delova prema datom postupku.

Network 1



| Symbol | Address | Comment |
|--------|---------|-------------------------------------|
| RESET | I0.1 | Vracanje kрана u pocetni položaj 0. |
| START | I0.0 | Prekidac za startovanje kрана. |

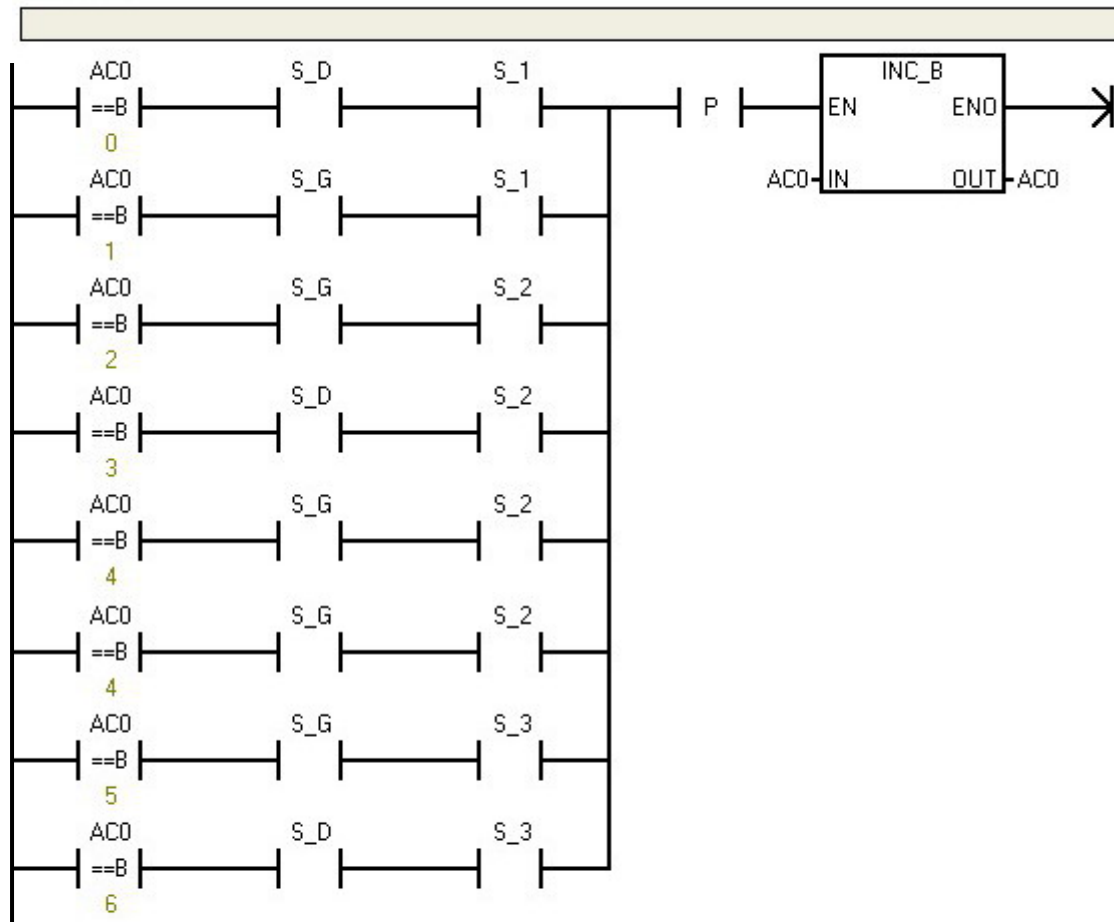
Network 2



| Symbol | Address | Comment |
|--------|---------|-------------------------------------|
| RESET | I0.1 | Vracanje kрана u pocetni položaj 0. |
| S_1 | I0.4 | Senzor krajnje desnog položaja. |
| S_G | I0.2 | Senzor gornjeg nivoa kрана. |

Slika 62. Zadatak 10_Network 1 i 2.

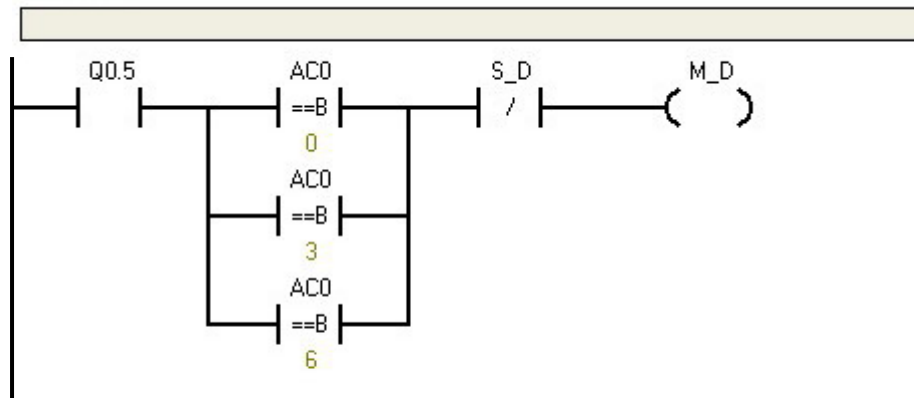
Network 3



| Symbol | Address | Comment |
|--------|---------|---------------------------------|
| S_1 | I0.4 | Senzor krajnje desnog položaja. |
| S_2 | I0.5 | Senzor srednjeg položaja. |
| S_3 | I0.6 | Senzor krajnje levog položaja. |
| S_D | I0.3 | Senzor donjeg nivoa krana. |
| S_G | I0.2 | Senzor gornjeg nivoa krana. |

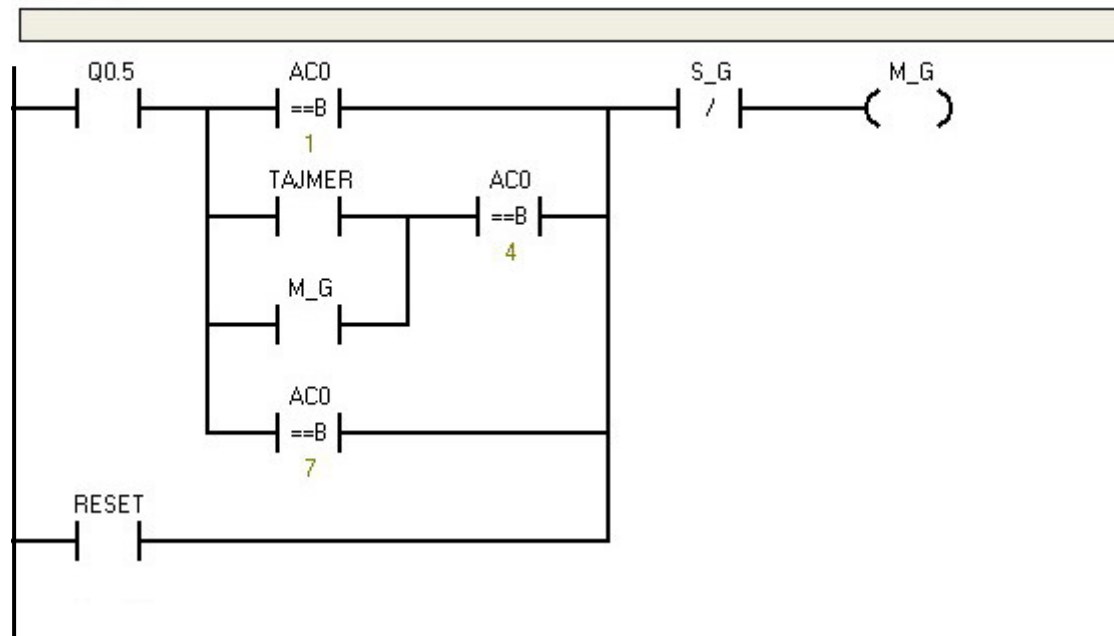
Slika 63. Zadatak 10_Network 3.

Network 4



| Symbol | Address | Comment |
|--------|---------|----------------------------|
| M_D | Q0.1 | Motor na DOLE. |
| S_D | I0.3 | Senzor donjeg nivoa krana. |

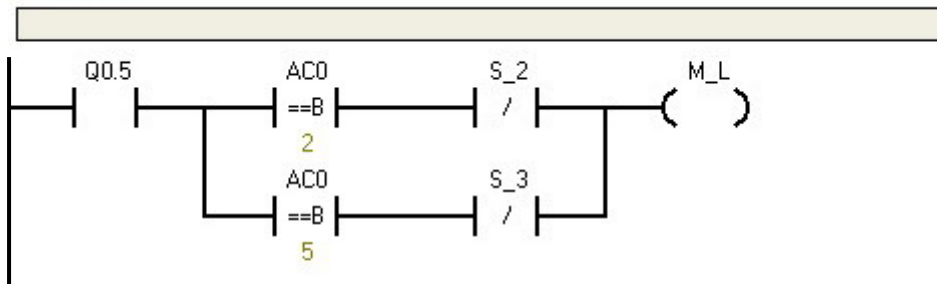
Network 5



| Symbol | Address | Comment |
|--------|---------|---|
| M_G | Q0.0 | Motor na GORE. |
| RESET | I0.1 | Vracanje krana u pocetni položaj 0. |
| S_G | I0.2 | Senzor gornjeg nivoa krana. |
| TAJMER | T37 | Odredjuje vreme zadržavanja u posudi za odmasćivanje. |

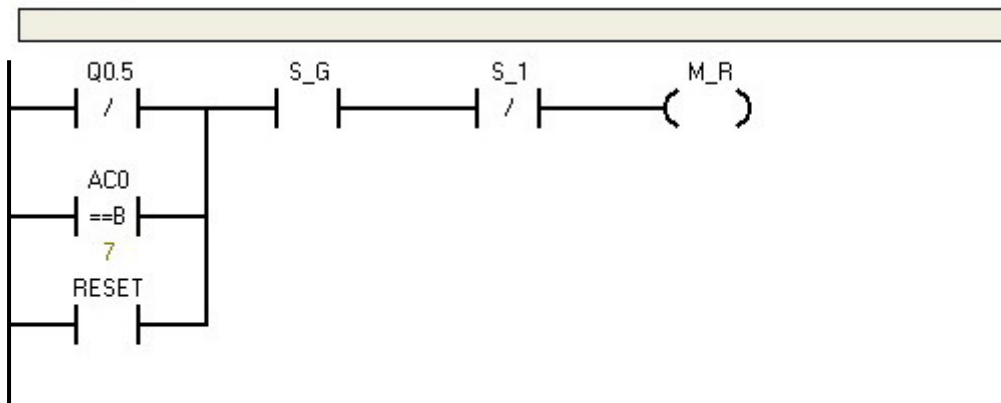
Slika 64. Zadatak 10_Network 4 i 5.

Network 6



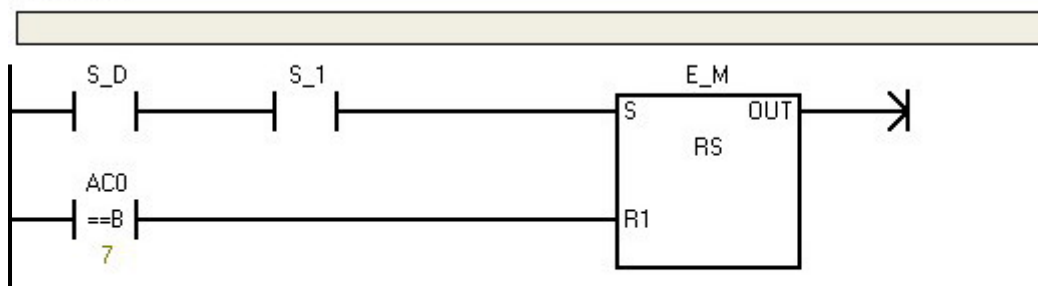
| Symbol | Address | Comment |
|--------|---------|--------------------------------|
| M_L | Q0.2 | Motor na LEVO. |
| S_2 | I0.5 | Senzor srednjeg položaja. |
| S_3 | I0.6 | Senzor krajnje levog položaja. |

Network 7



| Symbol | Address | Comment |
|--------|---------|-------------------------------------|
| M_R | Q0.3 | Motor na DESNO. |
| RESET | I0.1 | Vracanje krana u pocetni položaj 0. |
| S_1 | I0.4 | Senzor krajnje desnog položaja. |
| S_G | I0.2 | Senzor gornjeg nivoa krana. |

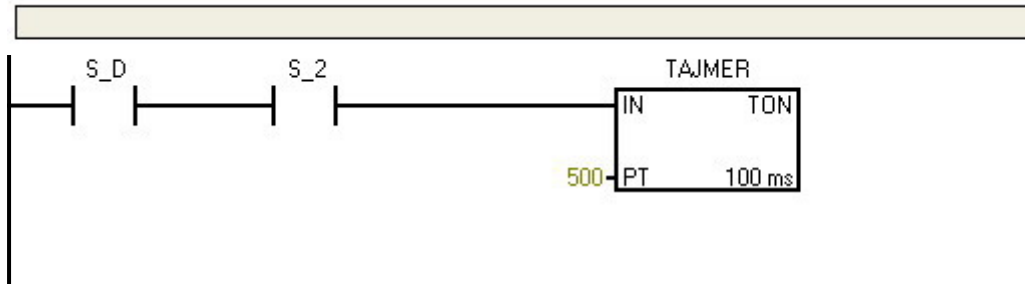
Network 8



| Symbol | Address | Comment |
|--------|---------|---|
| E_M | Q0.4 | Elektro-magnet ili drugi mehanizam hvatanja predmeta. |
| S_1 | I0.4 | Senzor krajnje desnog položaja. |
| S_D | I0.3 | Senzor donjeg nivoa krana. |

Slika 65. Zadatak 10_Network 6,7 i 8.

Network 9



| Symbol | Address | Comment |
|--------|---------|---|
| S_2 | I0.5 | Senzor srednjeg položaja. |
| S_D | I0.3 | Senzor donjeg nivoa krana. |
| TAJMER | T37 | Odredjuje vreme zadržavanja u posudi za odmasćivanje. |

Slika 66. Zadatak 10_Network 9.

LITERATURA

- [1] Milan Matijević, Goran Jakupović, Jelena Car, *Računarski podržano merenje i upravljanje*, Mašinski fakultet u Kragujevcu, 2005.
- [2] Siemens - Automation and Drives - Service& Support - Automation Service, Automation Support, Simatic Service, Simatic Support, Technical Support, Technical Consulting, <https://support.automation.siemens.com/WW/llisapi.dll?func=cslib.csinfo2&aktprim=99&lang=en>
- [3] PLC Programming Articles and Examples, <http://www.searcheng.co.uk/articles/plc/>
- [4] PLC Ladder Programs: KEYENCE Uk and Ireland, http://www.keyence.co.uk/downloads/plc_dwn.php#form
- [5] PLCs.net - FREE Online PLC Simulator, <http://www.plcsimulator.net/login.php>
- [6] ETF Beograd - Signali i sistemi / Signals&Systems Department, <http://automatika.etf.bg.ac.yu/>
- [7] Mr Milan Matijević, *RAZVOJ NOVIH STRUKTURA DIGITALNO UPRAVLJANIH ELEKTROMOTORNIH POGONA I INDUSTRIJSKIH PROCESA, DOKTORSKA DISERTACIJA*, Kragujevac, 2001.