

9. PROGRAMIBILNI LOGIČKI KONTROLERI

Do kraja šezdesetih godina prošlog vijeka sistemi upravljanja su bili zasnovani na primeni relejne logike (realizovani pomoću elektromehaničkih releja ili nešto kasnije poluprovodničkih diskretnih releja), zasnovane na relativno jednostavnim logičkim algoritmima. Različite logičke funkcije su realizovane načinom povezivanja između različitih releja. Svaki relej je izvršavao samo jednostavnu funkciju otvaranja ili zatvaranja kontakata kada su ispunjeni odgovarajuć uslovi na njegovom ulaznom namotaju. Programiranje bilo koje logičke funkcije je realizovano odgovarajućim načinom povezivanja releja.

Glavna mana takve relejne logike je da se pri bilo kakvoj promjeni u sistemu upravljanja ona mora mijenjati, promjenom ožičenja ili čak ubacivanjem u potpunosti novih sklopova.

Da bi se sa jednog mjesta lako pratilo stanje procesa kojima se relejno upravlja, svi releji su postavljani na međusobno malom rastojanju na odgovarajućim “upravljačkim” pločama.

Nadalje, širim korišćenjem i povećanjem kompleksnosti sistema upravljanja upravljački dio zasnovan na relejnom upravljanju postajao je sve glomazniji i nedovoljno pouzdan. To je značilo potrebu za većim industrijskim prostorom i zamjenom releja koji otkažu.

Te promjene izazivale su velike troškove ne samo za opremu već i dugotrajne zastoje potrebne za modifikaciju i testiranje.

Paralelno prethodnom, napredak tehnologije u izradi mikroprocesora, u to vrijeme, doveo je do unapređenju sistemima za vođenje različitih procesa.

Na prethodnim trendovima nastala je inicijativa za izradu elektronsko-mikroprocesorskog upravljačkog uređaju koji će biti zamjena i unapređenja relejnog upravljanja. Izrađeni su i prvi takvi uređaji, koji su

dobili naziv *programabilni logički kontroleri* (Eng. *Programmable logic controllers - PLC*).

Implementacija i razvoj ovih uređaja su bili vrlo brzi, pošto su pokazali izuzetne prednosti u odnosu na logičko upravljanje zasnovano na primjeni releja. Osnovne prednosti su: nemaju mehaničkih pokretnih dijelova, fleksibilniji su zbog mogućnosti programiranja/reprogramiranja, manja je moguća pojava grešaka tokom ožičavanja, manjih su dimenzija, imaju manju sopstvenu potrošnju i pouzdanost rada im je velika.

Prema standardima Udruženja proizvođača električne opreme (*The National Electrical Manufacturers Association – NEMA*) programabilni logički kontroler definisan je kao: “Digitalni elektronski uređaj koji koristi programabilnu memoriju za pamćenje naredbi kojima se zahtjeva izvođenje specifičnih funkcija, kao što su logičke funkcije, sekvenciranje, prebrojavanje, mjerenje vremena, izračunavanje i druge, u cilju vođenja različitih mašina i procesa”.

Naredna specifičnost *PLC* je da je primarno razvijen kao računar za korišćenje u industriji. To podrazumijeva da se instalira (i koristi) u neposrednom okruženju procesa sa kojim upravlja, tako da je otporan na razne nepovoljne uticaje: prašinu, vlagu, visoku temperaturu, vibracije i elektromagnetne smetnje. Na taj način je vrlo pogodan za izvršavanje decentralizovanog vođenja procesa, na samom mjestu upravljanja, gde se povezuje preko ulaza i izlaza sa uređajima kao što su operatorski paneli, motori, senzori, prekidači, ventili i sličnim.

PLC kao i svaki računar ima operativni sistem, koji svakako ima mnogo manje mogućnosti od operativnih sistema opšte namene, ali u današnje vrijeme opšte potrebe za komunikacijama, može u potpunosti da ih podrži. Stoga je moguće izvesti povezivanje programabilnih logičkih kontrolera (*PLC*-a) i eventualno centralnog računara ili drugih računara, radi rješavanja složenijih upravljačkih zadataka ili jednostavne akvizicije podataka i upravljanja na daljinu.

Mogućnosti komunikacije među *PLC* uređajima su tako velike da omogućavaju visok stepen iskorišćenja i koordinacije procesa, kao i veliku fleksibilnost u realizaciji upravljačkog procesa, tako da mogućnost komunikacije kao i fleksibilnost predstavljaju glavne prednosti primjene rješenja sa *PLC* uređajima.

9.1. Prednost upravljanja pomoću *PLC*-a u odnosu na upravljanje relejima

Početkom industrijske revolucije automatizovanim mašinama upravljali su pomoću releja, međusobno povezanih žicama unutar komandnog ormara. Za otkrivanje greške u sistemu bilo je potrebno mnogo vremena pogotovo kod složenih upravljačkih sistema. Vjek trajanja kontakata releja je ograničen, pa se vremenom moraju zamjeniti. Prilikom zamjene releja ili ostalih potrošnih dijelova, proces se mora zaustaviti a time i proizvodnja. Relejni upravljački orman koristio se samo za jedan određeni proces i nije ga bilo jednostavno izmeniti prema potrebama novog sistema. Prema ovome izloženom do sada relejno upravljanje se pokazalo veoma neefikasnim. Ovi nedostaci u velikoj mjeri su otklonjeni uvođenjem *PLC*-a u sisteme upravljanja, što je još i doprinelo poboljšanju kvaliteta, uvećavanju produktivnosti i fleksibilnosti.

Prednost upravljačkog ormara urađenog na bazi *PLC* kontrolera u odnosu na komandne ormare napravljenih na bazi releja ogleda se u nekoliko sledećih stavki:

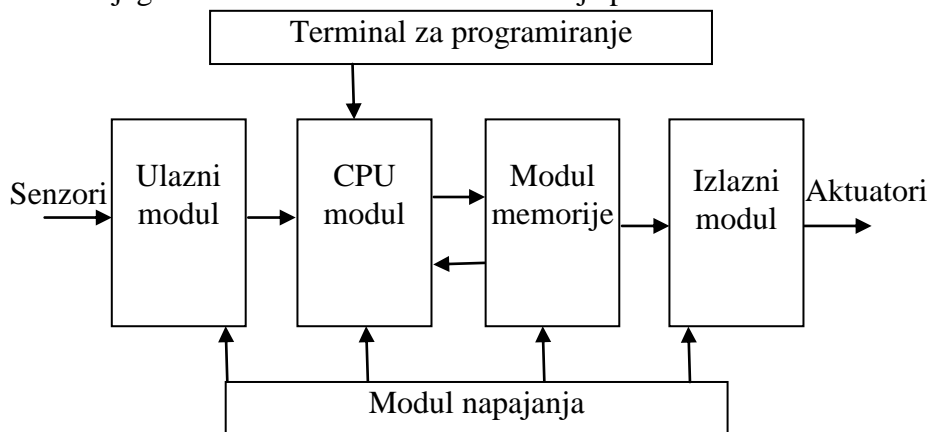
- Potrebno je 80% manje provodnih linija za povezivanje u poređenju sa konvencionalnim upravljačkim sistemom.
- Potrošnja je značajno smanjena jer *PLC* znatno manje troši od mnoštva releja.
- Dijagnostičke funkcije *PLC* kontrolera omogućavaju brzo i jednostavno otkrivanje grešaka.
- Izmjena u sekvenci upravljanja ili primjena *PLC* uređaja za drugi proces upravljanja, može se jednostavno izvršiti izmjenom programa preko konzole ili uz pomoć softvera na računaru (bez potrebe za izmjenama u ožičenju, sem ukoliko se ne zahtjeva dodavanje nekog ulaznog ili izlaznog uređaja).
- Potreban je znatno manji broj rezervnih dijelova.
- Mnogo je jeftiniji u poređenju sa konvencionalnim sistemom, naročito u sistemima gdje je potreban veliki broj *U/I* uređaja.
- Pouzdanost *PLC*-a je veća od pouzdanosti elektro-mehaničkih releja i tajmera.

9.2 Osnovna struktura PLC uređaja

Svi *PLC* bez obzira na veličinu imaju istu osnovnu hardversku strukturu (iste osnovne cjeline – module), sličnu drugim računarskim sistemima, prilagođenu industrijskom okruženju:

- CPU (centralna procesorska jedinica).
- Memorija za program i podatke.
- Komunikacioni modul.
- Mrežni modul za napajanje.
- Ulazni modul (digitalni, analogni).
- Izlazni modul (digitalni, analogni).
- Modul za proširenje.

Dijagram osnovne blok strukture *PLC* je prikazan na Sl.9.1.



Sl.9.1. Blok dijagram PLC-a

Modul sa CPU je obavezna komponenta svakog PLC i odgovoran je za usklađen zajednički rad svih ostalih komponentata *PLC*-a. U literaturi se kao sinonimi često za ovaj modul koriste termini: miktoprocesorski modul (Eng. *MicroProcessor Unit*- MPU), mikroprocesor (Eng. *microprocessor*) ili processor (Eng. *processor*). Ova jedinica: upravlja razmjenom podataka između *PLC*-a i spoljašnjeg svijeta (upravljanje komunikacijama), izvršava aritmetičke i logičke operacije, upravljanju memorijom (čita ili mijenja sadržaj memorijskih lokacija, prikuplja podatke sa ulaza i šalje signale za postavljanje izlaza. Ako se ne radi o velikom broju ulaza i izlaza tada je *CPU* realizovana sa jednim mikroprocesorom. Kod *PLC*-ova sa velikim brojem ulaza i izlaza *CPU* je realizovan kao multiprocesorski.

Memorija se sastoji od dvije cjeline: sistemske i korisničke:

Sistemska memorija se koristi od strane *PLC*-a za operativni sistem. U njoj se pored operativnog sistema nalazi i korisnički program u binarnom obliku. Ova memorija je obično *EEPROM* i može se mijenjati samo kad se radi o mjenjanju korisničkog programa. Korisnički program sa algoritmom obrade ulaznih informacija unosi se preko odgovarajućeg programatora, danas obično *PC* računara. Dobra praksa je da se program smješta i u *RAM* memoriju podržanu baterijom, tako da se izvršava iz *RAM*-a, odnosno da se učitava u *RAM* iz *EEPROM*-a svaki put kad se uključuje *PLC*, ili u slučaju gubitka podataka iz *RAM*-a.

Korisnička memorija je podjeljena u blokove koji imaju posebne funkcije. Razmjena podataka između *CPU* i eksternih uređaja ne izvodi se direktno nego preko memorije. Svakom eksternom uređaju se dodeljuje adresa. Jedan dio ove memorije se koristi za čuvanje stanja ulaza i izlaza, drugi dio se koristi za čuvanje vrijednosti promjenljivih kao što su vrijednosti tajmera i brojača.

Sa svih senzora podaci se direktno upisuju u određeni blok u memoriji, predviđen za ulaze. Kada *CPU* treba stanje nekog od senzora on ga čita iz određene lokacije u memoriji. Za sve *PLC*-ove je ovaj princip isti, ali formalan način zapisivanja varira od proizvođača do proizvođača. Za ilustraciju uzmimo da se radi o *PLC*-u proizvođača *SIEMENS*.

Neka želimo pristupiti podatku označenom sa "X" u bloku memorije rezervisanom za binarne ulaze prikazanom na SI.9.2. Da bi se pristupilo bitu u

	Blok memorijskih mjesta ulaza							
	7	6	5	4	3	2	1	0
Bajt 0								
Bajt 1			X					
Bajt 2								
Bajt 3								
Bajt 4								
Bajt 5								

SI.9.2. Pristup ulaznu/izlazu u memoriji

dijelu memorije, navodi se adresa, koja sadrži identifikator memorijskog bloka, adresu bajta i broj bita. U prikazanom primjeru se radi o:

blok ulaza, oznaka I (Eng. input – I)

bajt 1

bit 5.

Adrese bajta i bita se odvajaju tačkom (‘.’).

Dakle, imajući u vidu prethodno, adresa memorijske lokacije naznačene na Sl.9.2 bi se zapisala sa:

I01.5

Ekvivalentno vrijedi za binarne izlaze, pri čemu se za SIEMENS PJC-ove blok memorije rezervisan za izlaze označava sa Q. Na ovaj način se rezerviše mesto gde će biti pohranjena logička vrednost koja definiše stanje tog binarnog izlaza.

Na osnovu stanja ulaza, stanja vremenskih komponenata, brojača i memorisanih međustanja, dobija se rezultat algoritma koji se izvršava. Na taj način se dobijaju stanja izlaza, koja se prenose u odgovarajuće memorijsko područje a odatle preko interne magistrale do izlaza.

Komunikacioni dio obezbeđuje prije svega komunikaciju sa nadređenim programatorom ili PC računarom na kojem se piše upravljački program, šalje u PLC i zatim provjerava njegova funkcionalnost. Ostale mogućnosti su komunikacija sa drugim PLC uređajima i raznim senzorima, komunikacija sa operatorskim panelima, nadređenim računarima i modemsom vezom. Gotovo svi PLC-i imaju ugrađen serijski port za komunikaciju (RS232), a komunikacija se vrši preko protokola koji zavisi od proizvođača (najčešće full duplex serijska veza).

Modul napajanja obezbeđuje napajanje i otporan je na smetnje koje dolaze iz električne mreže kao i na kraće ispade mrežnog napona u trajanju od 10 do 15ms. Standardni naponi napajanja su 120/230VAC i 24VDC.

Ulazni moduli su namjenjeni za prikupljanje podataka sa senzora za praćenje stanja procesa. Starije verzije PLC su sadržavale ulazne module koji su samo mogli prenositi signale koje generišu senzori ON/OFF tipa (imaju dva stanja na izlazu: 0 ili 1, odnosno “uključen” ili “isključen”. Ovo ograničenje je omogućavalo praćenje samo nekih posebnih stanja procesa, a time i vrlo ograničene mogućnosti za vođenje. Današnji PLC-ovi sadrže ulazne module koji, bez posebnog prilagođavanja, mogu biti vezeni na sve tipove senzora. To je postignuto putem ulaznih modula, koji pored binarnih signala/informacija mogu prihvatati analogne i digitalne.

Izlazni moduli mogu generisati signale za vođenje procesa putem generisanje signala za djelovanje na aktuatora. Na taj način preko njih *PLC* djeluje na aktuatora i time vodi proces. Razvoj izlaznih modula je tekao po scenariju vrlo sličnom izloženom za ulazne module. Izlazni moduli prvih generacija, su mogli generisati samo signale *ON/OFF* tipa (dva stanja na izlazu: 0 ili 1, odnosno “uključen” ili “isključen”). Današnji izlazni moduli mogu generisati analogne i digitalne signale u skladu sa sadržajem memorije rezervisanom za izlaze.

Ulazni i izlazni moduli sadrže stepene za galvansko razdvajanje *CPU* od senzora i aktuatora.

Razmjena podataka/signala između *PLC*-a i procesa se realizuje putem razmjene informacija između *CPU* modula i dijela memorijskog modula rezervisanog za memorisanje stanja ulaza i izlaza.

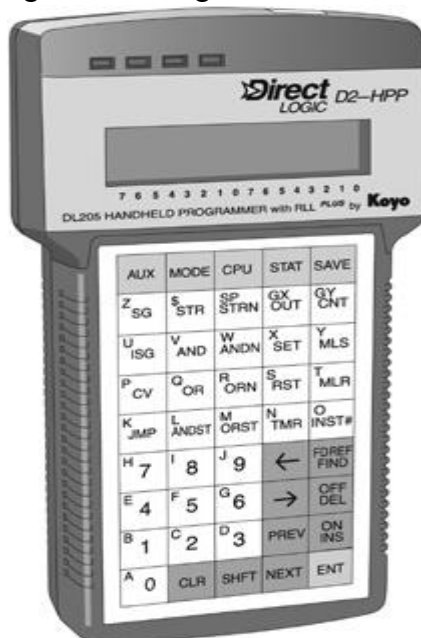
Terminal (modul) za programiranje *PLC* služi za unošenje potrebnih programa u memoriju *PLC*-a. Program se može unijeti u obliku relejne ljestvičaste logike (Eng. *relay ladder logic* - *RLL*), što je jedan od najpopularnijih jezika za programiranje. Ovaj jezik je nastao na osnovu ekvivalencije sa “programiranjem” relejnog upravljanja. Umjesto riječi, u ovom programskom jeziku koriste se grafički simboli. Program napisan na ovom jeziku ekvivalentan je šemi upravljačkog kola realizovanog pomoću releja. Pogodnost programiranja u ovom jeziku jeste što olakšava komunikaciju između programera *PLC*-a i ljudi koji izvode instalaciju *PLC* na process, a koji su po pravilu vrlo familijarni sa relejnom logikom.

Nakon priključivanja na *PLC*, terminali za programiranje se mogu koristiti za unošenje i praćenje rada programa (otklanjanje grešaka).

Ako terminal za programiranje nije u upotrebi, može se isključiti i razdvojiti od *PLC*-a. Razdvajanje uređaja za programiranje ne utiče na rad programa u *PLC*-u.

U proizvodnim pogonima, terminali za programiranje se često koriste za otkrivanje grešaka u opremi, mjenjanje programa i instaliranje programa na više mašina.

Za programiranje malih *PLC* sistema ponekad se koriste prenosivi programatori (slika 9.3) jer su jeftini i jednostavno se koriste.

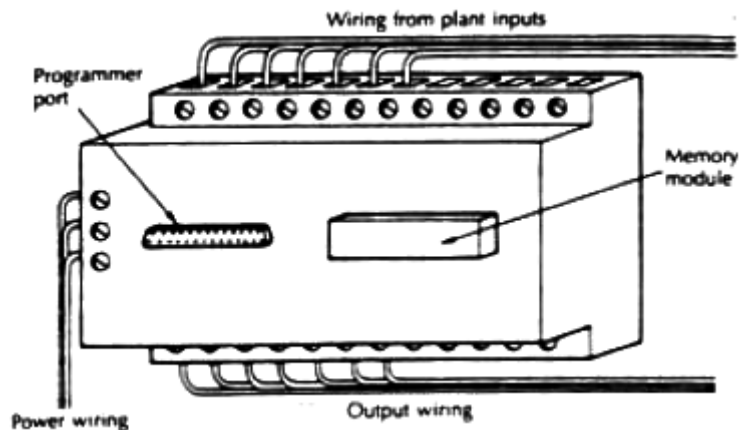


Sl.9.3 Izgled jednog namjenskog terminala za programiranje *PLC*

Za programiranje *PLC*, najčešće se koristi personalni računar (*PC*). Većina proizvođača personalnih računara nudi softver koji omogućava da se *PC* koristi kao uređaj za programiranje *PLC*-a. Taj softver omogućava korisniku da piše, mijenja, dokumentuje, čuva i ispravlja programe napisane u relejnoj ljestvičastoj logici.

Monitor prenosnog računara omogućava prikazivanje većeg dijela logike programa nego ekran prenosnog programatora, što olakšava tumačenje programa. Personalni računar razmjenjuje podatke s procesorom *PLC*-a preko serijskog ili paralelnog interfejsa ili preko Ethernet mreže.

Po pravilu su samo mikro *PLC*-i realizovani u jednom kompaktnom kućištu (Sl.9.4.) Mikro (kompaktni) *PLC* kontroleri su nezavisni, zatvoreni uređaji sa fiksnim brojem ulaza/izlaza, bez mogućnosti proširenja. U jednom kućištu, obično manjih dimenzija, smešetni su: izvor za napajanje, procesorska jedinica i ulazni i izlazni modul.



Sl.9.4. Skica izgleda jednog kompaktnog PLC

Kompaktni *PLC* kontroleri predstavljaju ekonomično rešenje, predviđeno za upravljanje sistemima i procesima male složenosti (Sl.9.5.)

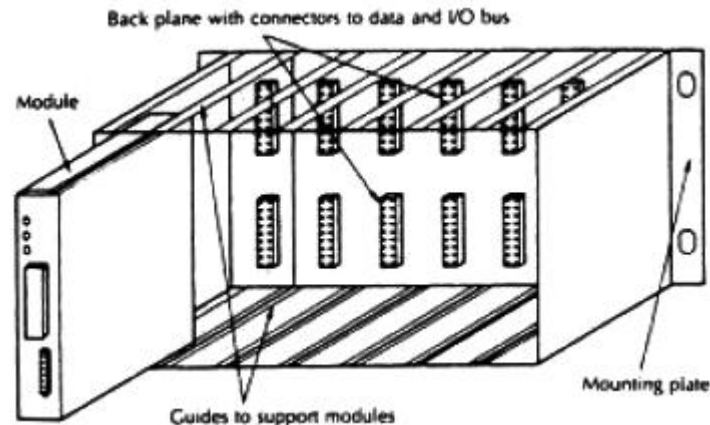
Tipično, poseduju do 16 ulaza i 16 izlaza i memoriju od nekoliko KB.



Sl.9.5. Izgled kompaktnog SIEMENS SIMATIC S7-200 mikro PLC

Svi PLC-ovi srednje i više klase se realizuju modularno. **Modularni *PLC*** sistemi se sastoje od većeg broja modula koji su smešteni unutar mehaničkog okvira, tj. kućišta (Eng. *rack*). Kućište poseduje određeni broj ležišta (Eng. *slot*) za smeštanje modula. Svaki slot čini par vođica duž gornje i donje stranice kućišta koje služe za mehaničko učvršćenje modula kao i konektor na zadnjoj ploči kućišta za priključivanje modula na

zajedničku magistralu izvedenu na štampanoj ploči zadnje stranice kućišta. Princip modularnog konfigurisanja *PLC*-a je ilustrovan na SI.9.6.



SI.9.6. Prikaz modularnog kućišta za konfigurisanje *PLC*

Prva dva *slot*a u kućištu zauzimaju modul napajanja i *CPU* modul, dok je raspored ostalih modula obično proizvoljan. Fizički položaj modula u kućištu, idući od *CPU* modula, je važan jer direktno određuje adresu tog modula, pri njegovoj komunikaciji sa *CPU*.

U zavisnosti od broja modula, *PLC* može imati i više od jednog kućišta.



SI.9.9. Prikaz nekoliko konfiguracija *SIEMENS* modularnih *PLC*-ova

Ovakav način konstrukcije omogućava lako proširenje ili modifikaciju sistema u zavisnosti od konkretne primjene. Na SL.9.9. je za ilustraciju prikazano nekoliko konfiguracija modularnih *PLC*-a koje proizvodi *SIEMENS*.

9.3. Programiranje *PLC*-a

Proizvođači *PLC*-a uz njih isporučuju namjenske programske jezike, koji su manje više u skladu sa standardom IEC 61131-1. Po tom standardu programski jezici se dijele na grafičke i tekstualne.

Najšire korišćen grafički programski jezik je, ako što je navedeno, forma ljestvičastog dijagrama (Eng. Ladder Diagram- LD). Neki proizvođači za programiranje *PLC*-ova daju i mogućnost programiranje pomoću funkcionalnih blok dijagrama (Eng. Functionial Block Diagram).

Tekstualni programski jezici su asemblerski jezik (Instruction List – IL) i proceduralni jezici (Structured Text – ST). Sintaksa ovih jezika značajnije varira od jednog do drugog proizviđača *PLC*-ova.

9.3.1. Grafičko programiranje

Programiranje *PLC*-ova pomoću ljestvičastik dijagrama se zasniva na korišćenju grafičkih simbola standardnih za relejne elemente upravljanja. Osnovni logički upravljački elementi relejnog upravljanja su releji i solenoid. Iz tog razloga će biti dat kratak osvrt na te elemente.

Upravljački relej sastoji se od pokretne kotve i elektromagneta. Primjer releja sa dva para kontakata je prikazan na Sl. 9.9.a.

Funkciju upravljanja relej realizuje preko kontakata. Jedno kontaktno pero je ulazno i podrazumijeva se da je u stanju “visokog” napona (obično logičko 1). Preko stanja drugog kontaktnog pera se izvršava funkcija upravljanja. Ako su kontakti zatvoreni tada se podrazumijeva da relej uključuje odgovarajući uređaj, odnosno da prenosi upravljačik signal sa prvog kontaktnog pera na drugo. Kada su kontakti otvoreni tada je pripadajući uređaj isključen.

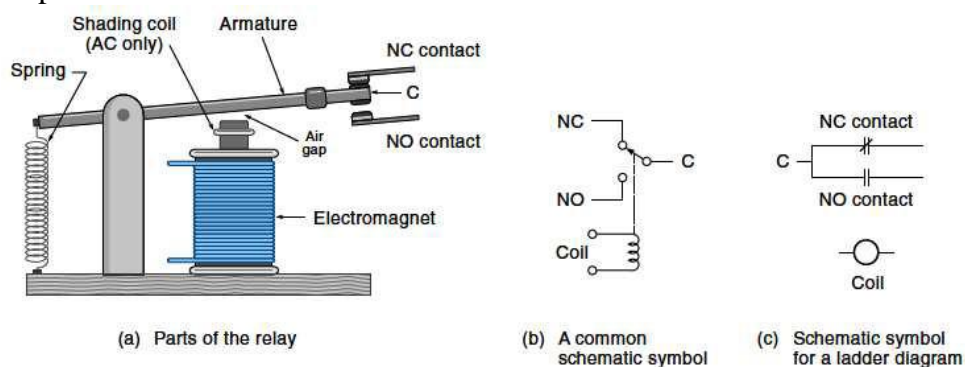
Stanje kontakata zavisi od stanja kalema elektromagneta. Kada kroz kalem ne protiče struje tada nema privlačenja kotve releja (oznaka “c” na Sl.9.9.). Kotva je u svom “normalnom” položaju. Na Sl.9.9.a će kotva, zbog privlačne sile opruge, biti u prikazanom položaju. Kontakt će biti ostvaren između kontaktnog pera kotve i gornjeg konatntnog pera, na Sl.9.9.a.

Istovremeno kontakt neće biti ostvaren između kontaktnog pera kotve i donjeg kontaktnog pera.

Treba primjetiti da će kod releja na slici gornji kontakt biti normalno zatvoren (engl. *Normally Closed* – NC), a da će istovremeno donji kontakt biti normalno otvoren (Eng. *Normally Open* – NO).

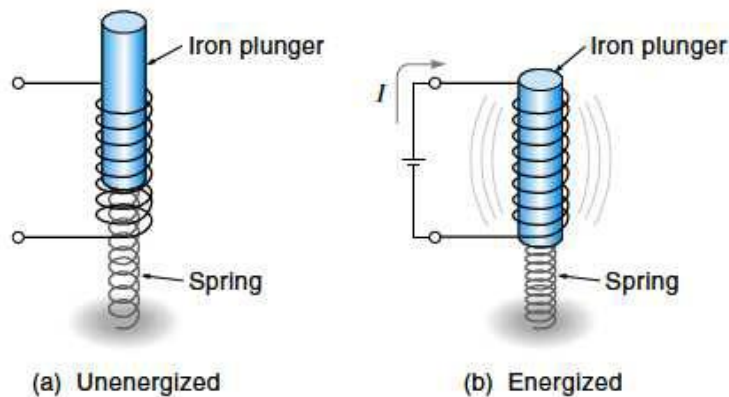
Za relejno i PLC upravljanje je veoma važno da je svaki relej bude u stanju NC ili NO zavisno od konkretne primjene, jer su to stanja koja će postojati kada iz bilo kakvog razloga nestane napajanja relejnog panela ili PLC-a, respektivno.

Na SI.9.9.b i SI.9.9.c su prikazani standardni simbli i simboli koji se koristi u ljestvičastim dijagramima za prikazivanje realnog releja sa SI.9.9.a, respektivno.



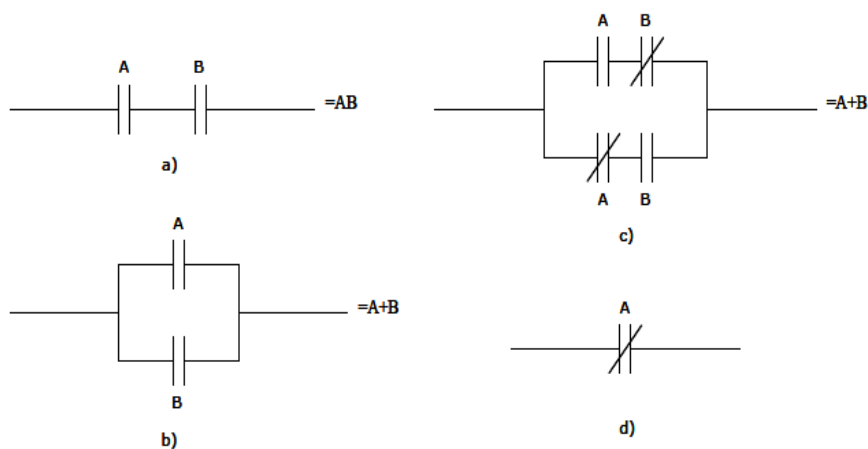
SI.9.9. Prikaz izgleda relea sa dva kontakta (a), simbla koji se koriste za stvarna električna kola (b) i simbola koji se koriste u ljestvičastim dijagramima

Solenoid je namotaj unutar kojeg se nalazi pokretno feritno jezgro. Ako kroz solenoid ne protiče strua jezgro je u “normalnom” položaju (SI.9.9.a). Kada kroz solenoid protiče struja tada elektromagnetna sila pomjera jezgro u drugi položaj (SI.9.9.b). Obično se koristi za pokretanje različitih vrsta aktuatora (električnih, pneumatskih, hidrauličkih i drugih).



Sl.9.9. Solenoid u “normalnom” (neaktivnom) položaju (a) i pobuđenom (aktivnom) položaju (b)

Odgovarajućom šemom povezivanja releja, odnosno granama ljestvičastih dijagrama mogu se realizovati različite logičke funkcije. Na Sl.9.9. su prikazane konfiguracije grana za realizaciju osnovnih logičkih funkcija. Na sličan način se mogu realizovati druge potrebne logičke funkcije.

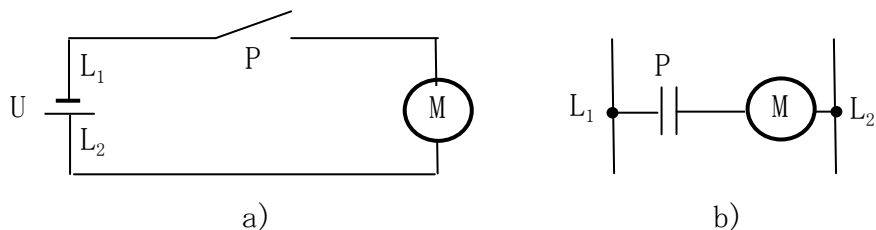


Sl.9.9. Konfiguracija kontakata za realizaciju osnovnih logičkih funkcija:

a) AND; b) OR; c) EXCLUSIVE; d) NOT

U cvilju pojašnjenja osnovnih pravila za crtanje ljestvičastih dijagrama prvo pogledajmo jedno elementarno električno kolo prikazano na Sl.9.11.a. Dijagram prikazuje kolo za uključivanje i isključivanje elektromotora. Kada je prekidač P uključen napon na krajevima istosmjernog izvora će preko linija za

napajanje L_1 i L_2 pokretati elektromotor M . Isto kolo se koristeći usvojenu notaciju za crtanje ljestvičastih dijagrama može predstaviti kao na Sl.9.11.b. Na ljestvičastim dijagramima se napajanje za električna kola predstavlja sa dvije vertikalne linije (šine). Horizontalnu liniju (prečagu) čini linija na kojoj se nalaze prekidač i motor.



Sl.9.11. Prikazivanje jednosatnog električnog kola na dva različita načina

Standardizovani simboli koji se koriste u ljestvilastim dijagramima definisani su standardom IEC 11331-1(International Electrotechnical Commision - IEC) i predstavljeni na Sl.9.12.

- — | | — ● Normalno otvoren kontakt (*NO*)
- — | / | — ● Normalno zatvoren kontakt (*NC*)
- — () — ● Izlazni namotaj (protiče struja kada je priključen naoin)

Sl.9.12. Simboli definisani standardom IEC 11331-1

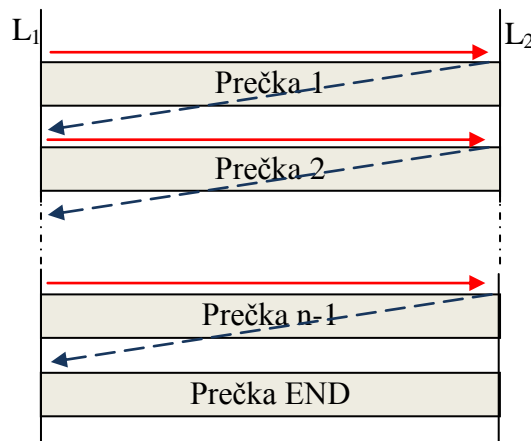
Ljestvičasti dijagrami se široko koriste u grafičkom programiranju PLC-ova. Pisanje programa je ekvivalentno crtanju električne prekidačke šeme. Pored osnovnih simbola za crtanje ljestvičastog dijagrama vrijede i druga usaglašena pravila.

Vertikalne linije na dijagramu predstavljaju šine za napajanje kola postavljenih između njih. Podrazumijeva se da struja teče od lijeve vertikale kroz prečku (Eng. rung).

Svaka prečka definiše jednu operaciju u procesu upravljanja.

PLC izvodi program tako što ljestvičasti dijagram prolazi sa lijeva na desno i odozgo prema dolje.

Na SI.9.13 je prikazano izvođenje programa koji realizuje PLC kroz ciklus skeniranja (Eng. *scan cycle*).



SI.9.13. Ilustracija ciklusa skeniranja PLC-a.

Kada je PLC u režimu izvršavanja programa (engl. *run*), prolazi kroz čitav leder program do kraja poslednje prečke, a zatim odmah nastavlja izvršavanje programa skokom na početak. Jedan prolazak kroz ljestvičasti dijagram naziva se ciklus skeniranja. Kraj jednog ciklusa skeniranja mora biti markiran blokom sa riječi *END* ili *RET*.

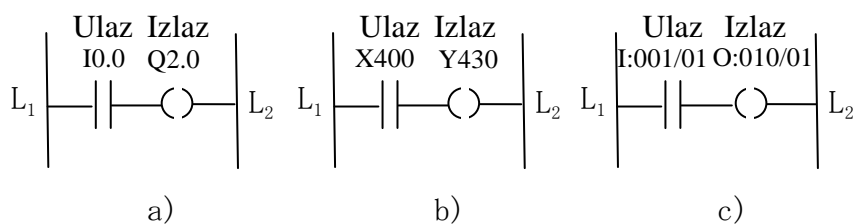
Svaka prečka mora početi sa ulazom (ulazima) i mora završiti sa najmanje jednim izlazom. Termin ulaz se koristi za akciju upravljanja (vođenja), kao što je zatvaranje kontakata prekidača, koji se koriste kao ulaz u PLC. Termin izlaz se koristi za uređaje povezane na izlaz PLC, na primer motor.

Električni uređaji (elementi) su prikazani u njihovom normalnom stanju. Tako je *NO* prekidač na ljestvičastom dijagramu prikazan kao otvoren.

Svaki uređaj se može pojaviti u više od jedne prečke ljestvičastog dijagrama. Na primjer, ako se radi o releju koji se koristi kao prekidač za više uređaja. Za jedan uređaj se u tim slučajevima koriste isti simboli za obilježavanje.

Svi ulazi i izlazi se identifikovani njihovim adresama, koje odgovaraju njihovim adresama u memoriji *PLC*. Način obilježavanja je definisan kod svakog proizvođača *PLC*.

Za ilustraciju su na Sl.9.14. prikazani ljestvičasti dijagrami, za primjer sa Sl. 9.11. u skladu sa načinom predstavljanja kod različitih proizvođača *PLC*-ova: Siemens (a), Mitsubishi (b), Allen-Bradley, (c).



Sl.9.14. Ljestvičasti dijagrami za Siemens (a), Mitsubishi (b), Allen-Bradley (c) *PLC*, prema primjeru sa Sl. 9.11.

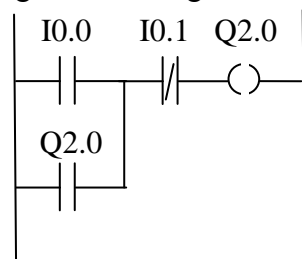
U *SIEMENS* notaciji ulazi i izlazi su raspoređeni u grupe po 8 (bajtova). Svaki ulaz ili izlaz u okviru bajta odgovara bitu. Na taj način svi ulazi i izlazi imaju svoje adrese koje se direktno pridružuju fizičkim ulazima ili izlazima na odgovarajućem modulu. Notacija je u obliku XXX.X. Prvi simbol označava da li se radi o ulaznom (Eng. input – I) ili o izlaznom (engl. output – Q) modulu. Nakon toga, prije tačke su jedan ili dva simbola rezervisana za oznaku bajta na koji je povezan modul. Na kraju, nakon tačke slijedi oznaka bita na kojem je smješten ulaz ili izlaz.

U skladu sa prethodnim adresama (I0.0 i Q2.0) na Sl.9.14.a imaju sljedeće značenje.

Adresa I0.0 znači da se ulaz nalazi na mjestu bajta 0 (modul neposredno pored *CPU* modula), na poziciji bita 0.

Adresa Q2.0 znači se izlaz nalazi na mjestu bajta 2 (modul je na trećem mjestu od *CPU* modula) i bita 0.

Često postoje situacije u kojima je potrebno da se održi izlaz vezan na napajanje, čak i kada ulaz prestane. Jednostavan primer takvog situacija je motor koji se pokreće tasterom. Iako taster ne ostaje zatvoren, potrebno je da motor nastavi da radi sve dok se ne aktivira taster za zaustavljanje. Za obavljanje ovakve funkcije se koriste kola za samo-držanje (Eng. latch circuit). Primjer kola za samo-držanje je prikazan na slici 9.15. Kada se kontakti ulaza I0.0 zatvore, biće aktiviran izlaz Q2.0. Da bi izlaz i dalje bio aktivan postavljani su drugi kontakti koji su vezani na izlaz (Q2.0). Na taj



Sl.9.15. Ljestvičasti dijagram sa kolom za samo-držanje

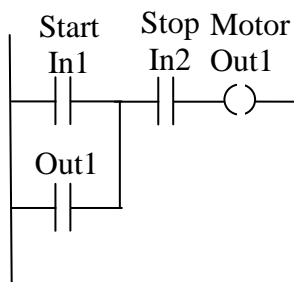
način ovi kontakti zajedno sa ulaznim konatktima čine logičko ILI kolo (u skladu sa S.9.9.b). Zato će izlaz biti aktivan i kada se ulaz I0.0 otvori. Izlaz se može isključiti aktiviranjem drugog ulaza (I0.1).

Primjer

Kreirati ljestvičasti dijagram za upravljanje elektromotorom tako da se postignu naredni zahtjevi. Motor se pokreće i zaustavlja sa dva tastera (Start, Stop). Kada je motor uključen svijetli crvena *LED* dioda (engl. *LED on*), a kada je motor isključen svijetli zelena *LED* dioda (engl. *LED off*). U cilju sigurnijih mjera zaštite od posledica neregularnog rada, kada nema napajanja upravljačkog ormara tada kontakti Stop tastera moraju biti otvoreni.

Rješenje

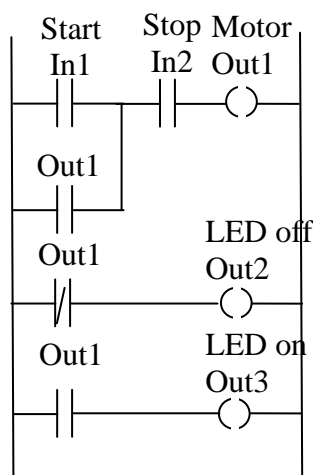
Iz uslova zadatka slijedi da prva linija ljestvičastog dijagrama treba biti kao na Sl.9.16.



Sl.9.16. Pokretanje i zaustavljanje motora sa NO tasterima

Za indicaciju rada motora potrebno je dijagram dopuniti sa dodatne dvije linije koje pokreću *LED* diode. Kompletan dijagram izgleda kao na Sl.9.17.

Svaka horizontalna linija (prečka) na ljestvici predstavlja instrukciju u programu koju će *PLC* realizovati. Kompletan ljestvičasti dijagram predstavlja program. Postoji nekoliko načina na koje se mogu koristiti za upisivanje programa u terminal za programiranje (namjenski prenosivi programatori ili *PC*). Bilo koji metod se koristi za unos programa putem terminala za programiranje program u memoriju mora biti



Sl.9.17. Ljestvičasti dijagrami za upravljanje motorom sa indicijom uključenosti

smješten u formi mašinskog jezika (samo binarni kod).

Ako se program unosi putem namjenskog terminala tada njegova tastatura sadrži tipke na kojima su naznačen simboli karakteristični za ljestvičasti dijagram ili operator asemblerskog jezika (tek slijedi detaljnije objašnjenje).

Ako se *PC* koristi za unošenje programa tada je potrebno da taj *PC* sadrži softvera za tu namjenu. Svaki proizvođač *PLC* ima vlastiti softver za tu namjenu, naprimjer: *STEP 7 - Micro/WIN V4* za *SIEMENS*, *RSLogix* za *Allen-Bradley*, *MELSOFT- GX Developer* za *Mitsubishi PLC*-ove. Softver radi na *Windows* operativnom sistemu tako da se elementi selektuju sa menija ekrana.

9.3.2. Tekstualno programiranje

Asemblerski jezik za tekstualno programiranje je po opštoj strukturi vrlo blizak ljestvičastim dijagramima.

Može se smatrati da program u ovom jeziku odgovara unošenju linija ljestvičastog dijagrama u obliku tekstualnih instrukcija (naredbi). Program čini niz instrukcija, tako da se svaka instrukcija nalazi u novom redu. Svaka instrukcija se sastoji od operatora praćenog jednim ili više operandima (subjekata operacije). U poređenju sa ljestvičanim programiranjem u osnovi se može smatrati da jednom operator instrukcije odgovara elementu ljestvičastog dijagrama. Staka instrukcija može koristiti ili upisivati podatke u odgovarajućoj memorijskoj oblasti. Za to se koriste mnemonički kodovi koji definišu operaciju nad operandom. Kodovi se međusobno razlikuju od proizvođača do proizvođača, iako je preporučeno da se i za to koristi standard IEC 1131-3.

Za ilustraciju su u tabeli 9.1 prikazani kodovi za neke funkcije prema IEC 1131-3 standardu i kodovi za *SIEMENS PLC*-ove.

IEC 1131-3 operatori	SIEMENS operatori	Operacija	Ljestvičasti dijagram
LD	A	Upisuje operand u registar	Započinje liniju sa (NO) otvorenim kontaktom
LDN	AN	Upisuje negativan operand u registar	Započinje liniju sa (NC) zatvorenim kontaktom
AND	A	Bulovo I (Eng. AND)	Serijska veza elemenata sa otvorenim kontaktima
ANDN	AN	Bulovo I sa negativnim operandom	Serijska veza elemenata sa zatvorenim kontaktima
OR	O	Bulovo ILI	Paralelna veza elemenata sa otvorenim kontaktima
ORN	ON	Bulovo ILI sa negativnim operandom	Paralelna veza elemenata sa zatvorenim kontaktima
ST	=	Smješta rezultat registra u operand	Izaz

U skladu sa prethodnim će za ljestvičasti dijagram prikazan na Sl.9.14. odgovarajući assemblerski program za *SIEMENS PLC*-ove biti:

```

A      I0.0
=      Q2.0

```

U prethodnom kratkom programu su A i Q operatori, a I0.0 i Q2.0 označavaju operande.

Za ljestvičasti dijagram na S1.9.15 će tekstualni program biti:

```
A      I0.0
O      Q2.0
AN     I0.1
=      Q2.0
```

Kada se pišu instrukcije za ljestvičasti dijagram sa više prečki, tada = u SIEMENS notaciji označava kraj svake prečke (i početak nove).

Prethodni primjeri sa dati prije svega u cilju islustracije pisanja tekstualnog programa na osnovu ljestvičastog dijagrama. Pošto su sintaksa programa i neke funkcionalnosti različiti kod različitih proizvođača PLC-ova, za pisanje programa za složenije zadatke potrebno je prethodno se detaljnije upoznati sa konkretni programom.