

1. PRIMENA I PROGRAMIRANJE PROGRAMIBILNIH LOGIČKIH KONTROLERA (PLC)

1.1 Uopšte o PLC kontrolerima

Uopšteno, sistem upravljanja u elektrotehnici čini skup elektronskih uređaja i opreme koji obezbeđuju stabilnost, tačnost i eliminaciju štetnih prelaznih stanja u proizvodnim procesima. Sistem upravljanja može biti različitog oblika i implementacije, od energetske postrojenja do mašina. Sistemi upravljanja su se razvijali tokom vremena. U ranom periodu razvoja sami ljudi su obavljali upravljačke zadatke. Krajem šezdesetih godina prošlog veka sistemi upravljanja su bili zasnovani na primeni relejne logike, zasnovane na relativno jednostavnim logičkim algoritmima. Glavna mana relejne logike je da se pri bilo kakvoj promeni u sistemu upravljanja ona mora menjati, promenom ožičenja ili čak ubacivanjem u potpunosti novih sklopova. Te promene izazivale su velike troškove ne samo za opremu već i dugotrajne zastoje potrebne za modifikaciju i testiranje.

Napredak tehnologije u izradi mikroprocesora, u to vreme, doveo je do revolucije u sistemima upravljanja. Pojavila se ideja o izradi elektronsko-mikroprocesorskom upravljačkom uređaju koji bi se mogao jednostavno reprogramirati u slučaju izmene u upravljačkim zadacima. Izrađeni su i prvi takvi uređaji, koji su dobili naziv *programabilni logički kontroleri* (*Programmable logic controllers*) ili skraćeno *PLC*. Dalji razvoj ovih uređaja je bio vrlo brz, pošto su pokazali izuzetne prednosti u odnosu na logiku zasnovanu na primenu releja, jer nemaju mehaničkih pokretnih delova, fleksibilniji su zbog mogućnosti programiranja, 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 Association – NEMA*) programabilni logički kontroler definisan je kao: “Digitalni elektronski uređaj koji koristi programabilnu memoriju za pamćenje naredbi kojima se zahteva izvođenje specifičnih funkcija, kao što su logičke funkcije, sekvenciranje, prebrojavanje, merenje vremena, izračunavanje, u cilju upravljanja različitim mašinama i procesima”.

PLC kao industrijski računar samim svojim dizajnom predviđen je za primenu u neposrednom okruženju procesa sa kojim upravlja, tako da je otporan na razne nepovoljne uticaje, prašina, vlaga, visoka temperatura, vibracije i elektromagnetne smetnje, tako da se obično primenjuje za rešavanje decentralizovanih upravljačkih zadataka, na samom mestu 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 vreme 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,

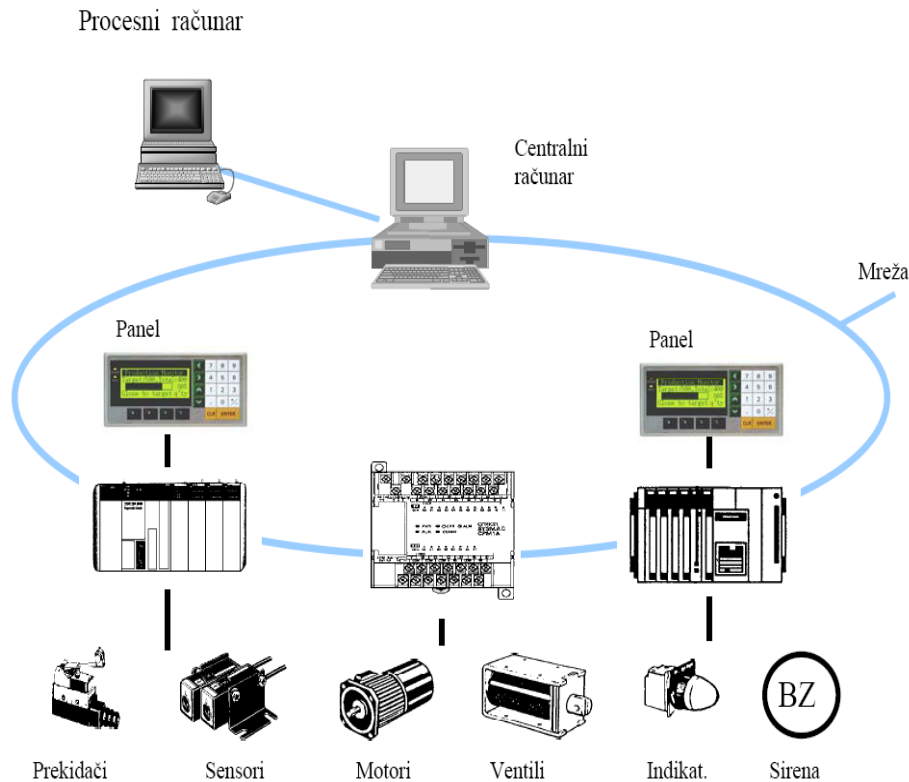
radi rešavanja složenijih upravljačkih zadataka ili jednostavne akvizicije podataka i upravljanja sa daljine.

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 primene rešenja sa *PLC* uređajima.

1.2 *PLC* kontroler

PLC kontroler je element automatizovanog sistema, koji na osnovu prihvaćenih ulaznih signala sa ulaznih uređaja, po određenom programu, formira izlazne signale sa kojima upravlja izlaznim uređajima. U automatizovanom sistemu, *PLC* kontroler je obično centar upravljanja.

Izvršavanjem programa smeštenog u programskoj memoriji, *PLC* neprekidno posmatra stanje sistema preko ulaznih uređaja. Na osnovu logike implementirane u programu *PLC* određuje koje akcije trebaju da se izvrše na izlaznim uređajima. Za upravljanje složenim procesima moguće je povezati više *PLC* kontrolera među sobom ili sa centralnim računarom.



Prema broju ulazno/izlaznih priključaka *PLC* uređaji mogu se načelno podeliti na mikro sa maksimalno do 32, male do 256, srednje do 1024 i velike *PLC*-e preko 1024 ulazno/izlaznih

priključaka. Sa povećanjem broja priključaka mora se povećati i brzina procesora kao i količina memorije a samim tim i složenost i cena samog uređaja raste.

1.3 Ulazni uređaji

Ulazni uređaji čije signale prihvata *PLC*, mogu biti vrlo različiti. Po tipu signala koji daju na svom izlazu mogu se podeliti na digitalne (kontaktne prirode *ON / OFF*) i analogne. Karakteristični digitalni ulazni uređaji su tasteri, prekidači, krajnji prekidači, fotoćelije, presostati, temostati i drugi. Karakteristični analogni ulazni uređaji su termoelementi, otpornički termometri i drugi pretvarači električnih i neelektričnih veličina u standardne strujne i naponske signale. Ulazni signali pri tome se moraju prilagoditi sa odgovarajućim *PLC* ulaznim modulima. Ulazni moduli konstruišu se za prihvatan jednosmernih i naizmeničnih električnih signala naponskih nivoa od 10 do 250V, strujnih nivoa od 0(4) do 20mA, signala na *TTL* nivou, impulsnih ulaza sa brojačkim ili interapt prihvatom i slično.

1.3.1 Tipični ulazni uređaji



Granični prekidač



Tajmer



Fotoćelija



Enkoder

1.4 Izlazni uređaji

Izlazni uređaji kojima upravlja *PLC* na osnovu programa i stanja na ulazima mogu biti releji, kontaktori, elektromagnetni ventili, elektromotori, step motori, pneumatski cilindri i drugi slični uređaji. Takođe kao ulazni uređaji mogu se podeliti na digitalne i analogne, pri čemu se izlazi *PLC* kontrolera moraju prilagoditi potrebnim naponskim i strujnim nivoima. Digitalni izlazi *PLC* kontrolera su obično galvansko izolovani kontakti, ili bez povećane galvanske izolovanosti sa triakom ili tranzistorom sa otvorenim kolektorom ili TTL izlazom.

1.4.1 Tipični izlazni uređaji



Motor



Solenoid



Displej

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

Početkom industrijske revolucije automatizovanim mašinama upravljali su pomoću releja, međusobno povezanih žicama unutar komandnog ormana. Za otkrivanje greške u sistemu bilo je potrebno mnogo vremena pogotovo kod složenih upravljačkih sistema. Vek trajanja kontakata releja je ograničen, pa se vremenom moraju zameniti. Prilikom zamene releja ili ostalih potrošnih delova, mašina se morala zaustaviti a time i proizvodnja. Komandni 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 su u velikoj meri otklonjeni uvođenjem *PLC*-a u sisteme upravljanja, što je još i doprinelo poboljšanju kvaliteta, uvećavanju produktivnosti i fleksibilnosti. Prednost komandnog ormana urađenog na bazi *PLC* kontrolera u odnosu na komandne ormene napravljenih na bazi releja ogleda se unekoliko sledećih stavki:

- Potrebno je 80% manje žica 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.
- Izmena u sekvenci upravljanja ili primena *PLC* uređaja na drugi proces, upravljanja može se jednostavno izvršiti izmenom programa preko konzole ili uz pomoć softvera na računaru (bez potrebe za izmenama u ožičenju, sem ukoliko se ne zahteva dodavanje nekog ulaznog ili izlaznog uređaja).
- Potreban je znatno manji broj rezervnih delova.
- Mnogo je jeftiniji u poređenju sa konvencionalnim sistemom, naročito u sistemima gde je potreban veliki broj *U/I* uređaja.
- Pouzdanost *PLC*-a je veća od pouzdanosti elektro-mehaničkih releja i tajmera.

1.6 Opis sistema upravljanja sa *PLC* uređajem

Sistem koji se automatizuje odnosno na koji se želi primeniti automatsko upravljanje naziva se objekat upravljanja. Rad objekta upravljanja se konstantno prati ulaznim uređajima (senzorima) koji daju informaciju *PLC* uređaju o zbivanju u sistemu. Kao odgovor na to *PLC* šalje signal spoljnim izvršnim elementima koji zapravo kontrolišu rad sistema na način kako je to programer programom odredio. Programer *PLC* uređaj programira na osnovu zahteva i postavljenih kriterijuma definisanih tehnološkim zadatkom. Program se piše u namenskom programskom jeziku, koji svaki proizvođač daje uz svoj *PLC*, a koji pretstavlja kombinaciju programskog editora, kompajlera i komunikacionog softvera. U editoru se program piše prateći redosled operacija upravljanja, a zatim se proverava njegova sintaksa i vrši kompajliranje. Ako je sve u redu, komunikacionom vezom softver se šalje u memoriju *PLC*-a gde se smešta i pokreće. Ulazni i izlazni uređaji, koji se povezuju sa *PLC* uređajem, optimalno se odabiraju na osnovu zahteva i postavljenih kriterijuma definisanih u tehnološkom zadatku koje treba da zadovolje.

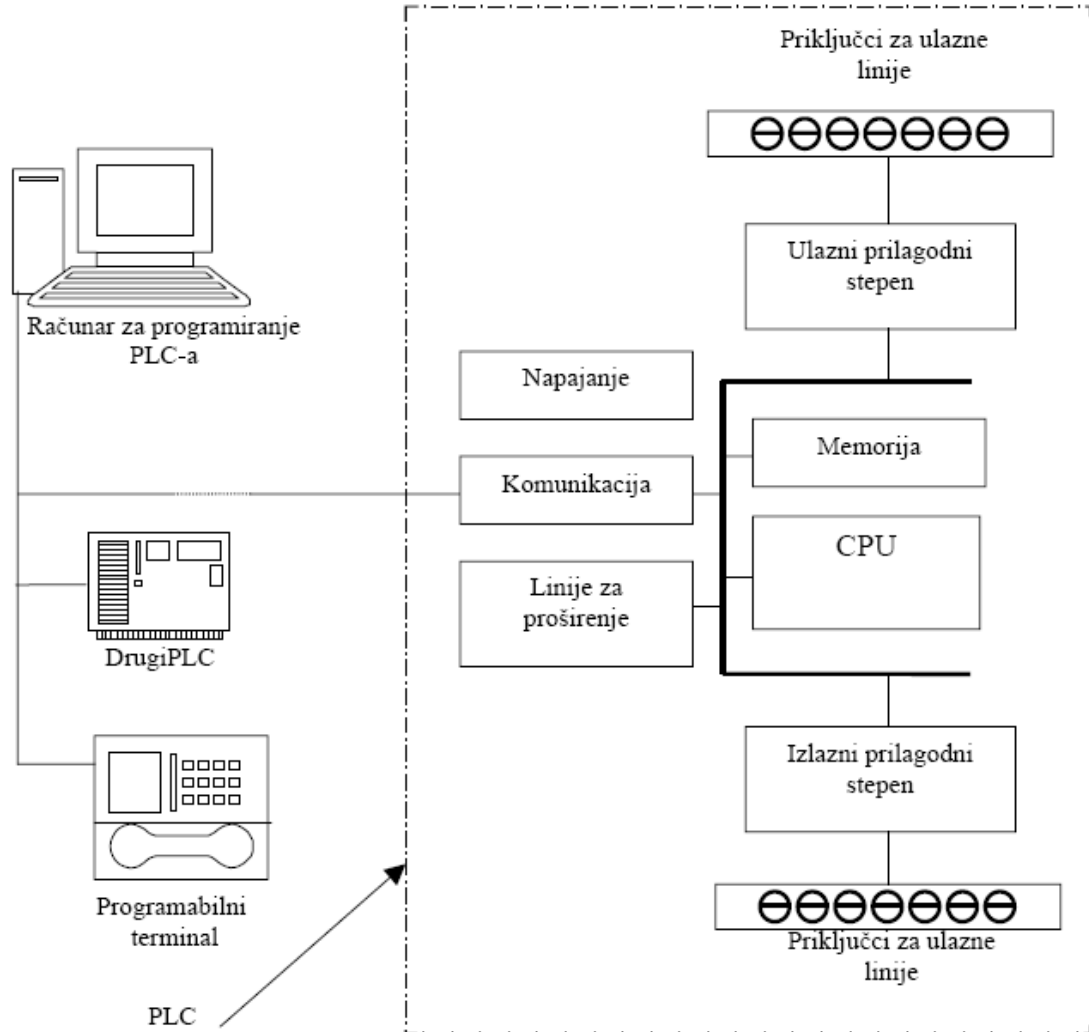
Ulazni uređaji su prekidači, senzori i davači. Izlazni uređaji mogu biti solenoidi, releji, elektromagnetni ventili, motori, kontektori kao i uređaji za svetlosnu i zvučnu signalizaciju.

1.7 Osnovna struktura *PLC* uređaja

Svi *PLC* kontroleri bez obzira na veličinu imaju istu hardversku strukturu, sličnu drugim računarskim sistemima, adaptiranu industrijskom okruženju, koja ima iste osnovne celine:

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

- Deo za proširenje



Osnovni elementi PLC kontrolera

CPU centralna procesorska jedinica je mozak *PLC* kontrolera koji odlučuje šta da se radi: brine o komunikaciji, međusobnoj povezanosti ostalih delova ovog kontrolera, izvršavanju programa, upravljanju memorijom, nadgledanjem ulaza i postavljanjem izlaza. Centralna procesorska jedinica izvodi se sa mikroprocesorom ili mikrokontrolerom kod uređaja sa manjim i srednjim brojem ulaza i izlaza ili kao multiprocesorska kod uređaja sa većim brojem ulaza i izlaza. *PLC* kontroler komunicira sa upravljačkim procesom preko analognih, digitalnih i brojačkih ulaza i izlaza. Informacije o stanju ulaza primarno se obrađuju i smeštaju u memoriju stanja ulaza i izlaza.

Memorija se može podeliti na sistemsku i korisničku:

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

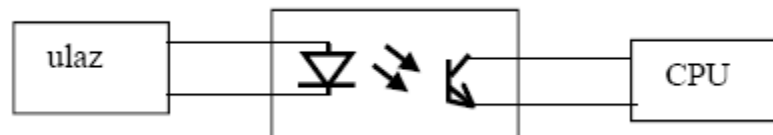
EEPROM i može se menjati samo kad se radi o menjanju korisničkog programa. Korisnički program sa algoritmom obrade ulaznih informacija unosi se preko odgovarajućeg programatora, danas obično *PC* računara. Dobra praksa je da se program smešta i u *RAM* memoriju podržanu baterijom, tako da se izvršava 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 podeljena u blokove koji imaju posebne funkcije. Jedan deo ove memorije se koristi za čuvanje stanja ulaza i izlaza, drugi deo se koristi za čuvanje vrednosti promenljivih kao što su vrednosti tajmera i brojača. Na osnovu stanja ulaza, stanja vremenskih članova, brojača i memorisanih međustanja, vrši se obrada koja formira stanja izlaza, koja se prenose u odgovarajuće memorijsko područje a odatle preko internog basa ka izlazu.

Komunikacioni deo obezbeđuju pre svega komunikaciju sa nadređenim programatorom ili *PC* računarom na kojem se piše upravljački program, šalje u *PLC* i zatim proverava 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).

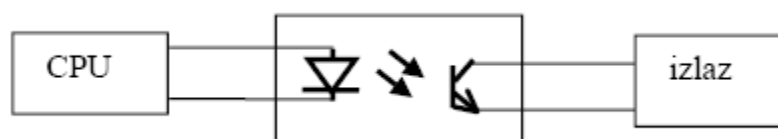
Moduo napajanja obezbeđuje napajanje i neosetljiv je na smetnje koje dolaze iz električne mreže kao i na kraće ispade mrežnog napona u trajanju od 10 do 15ms. Standardni naponi napajanja su 120/230VAC i 24VDC.

Ulazni prilagodni stepen štiti *CPU* od mogućih prevelikih signala na ulazu. Ulazni prilagodni stepen pretvara nivo stvarne logike u nivo logike koji odgovara *CPU* jedinici. Ovo se najčešće obavlja pomoću optokoplera kod digitalnih ulaza.



Ulazni prilagodni stepen

Izlazni prilagodni stepen takođe mora biti galvanski odvojen. Kod digitalnih izlaza, odvajanje je slično kao i kod ulaza. *CPU* dovodi signal na *LED* i uključuje ga. Svetlost pobuđuje foto tranzistor koji aktivira izlazni uređaj, obično rele koji je sposoban da vrši prekidanje jačih naponskih i strujnih signala.



Izlazni prilagodni stepen

Mikro *PLC*-i izvedeni su mehanički obično u okviru jednog kućišta, dok se ostali sastoje iz šasije (*rack*) koji ima određeni broj slotova u koji se stavljaju pojedini funkcionalni moduli. Prva dva slotova u kućištu zauzimaju napajanje i procesorski modul, dok je raspored ostalih modula obično proizvoljan. U zavisnosti od broja modula, *PLC* može imati i više od jednog kućišta. Dodatni moduli proširenja povezuju se preko dela odnosno **linija za proširenje**.

2 Programiranje *PLC*-a

Proizvođači *PLC*-a uz njih isporučuju namenske programske jezike, koji su manje više u skladu sa standardom *IEC 61131-1* (*IEC = International Electrotechnical Commission*). Po tom standardu programski jezici za kodiranje dele se na tekstualne i grafičke. Tekstualni programski jezici su *IL – Instruction List* (klasa asemblerских jezika) i *ST – Structured Text* (klasa proceduralnih jezika). Grafički programski jezici su *LD – Ladder Diagram* (lestvičast dijagram) i *FBD – Function Block Diagram* (funkcionalni blok dijagram). Neki proizvođači nude i mogućnost programiranja pomoću *BASIC* i *C* programskih jezika, ali ti jezici nemaju širu zastupljenost.

Najčešće upotrebljavan *PLC* programski jezik je kontakti lestvičast dijagram. Ovaj način programiranja ima za osnovu relejnu upravljačku šemu, odnosno njen grafički izgled, prilagođen principima rada *PLC* kontrolera. Ovaj način programiranja korišćen je već kod prvih primena *PLC*-a, kako bi korisnici navikli za izradu šema u relejnoj tehnici, bezbolno prešli na primenu *PLC*-a. Kako je ovaj grafički način programiranja lako shvatljiv i onima koji se nisu bavili relejnim upravljanjem, on se široko odomaćio.

Programski jezici se obično instaliraju na *PC* računar pod *WINDOWS* ili *DOS* platformom, tako da se dobija pristupačna platforma programatora za editovanje, kompajliranje i prenos programa na *PLC*. Komunikacija programatora sa *PLC*-om može biti aktivna i tokom izvođenja programa u njemu. Na taj način na ekranu programatora možemo pratiti stanje ulaza i izlaza tokom rada i zadavati eventualno nove naredbe na jednostavan način.

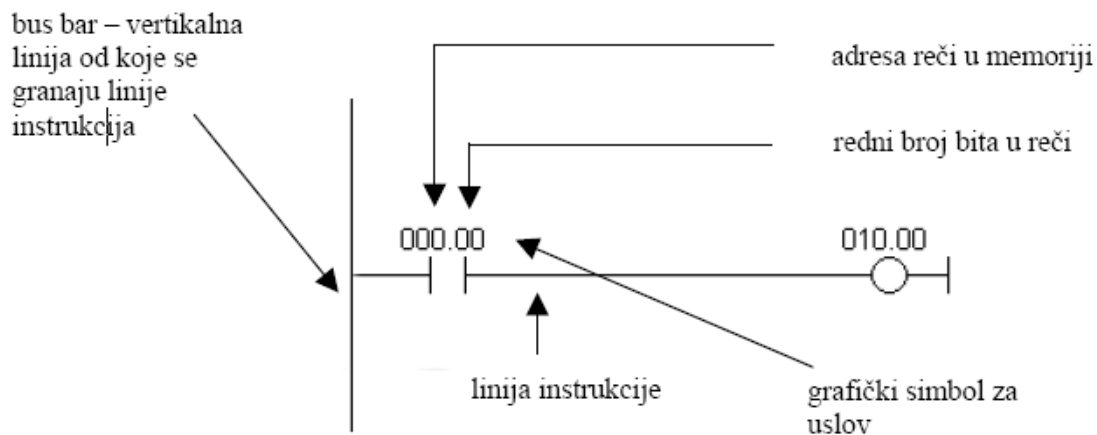
PLC se takođe može programirati i preko namenskih programatora, obično ručnih koji poseduju mali *LCD* ekran i tastaturu. Takvi uređaji se direktno spajaju na *PLC* i koriste se za kraće programe ili za manje izmene programa, kada se to vrši u pogonu. Za neke jednostavnije primene postoje čak i *PLC* kontroleri koji na sebi poseduju displej i nekoliko funkcijskih tastera, čime se obezbeđuje njihovo programiranje na mestu ugradnje. Neki *PLC*-i su opremljeni izmenljivim *EEPROM* memorijskim karticama, što olakšava programiranje odnosno izmene programa tokom rada. Dovoljno je ugastiti *PLC*, izmeniti memorijsku karticu unapred napunjenu sa novim programom i ponovo uključiti *PLC* koji automatski prihvata novi program.

2.1 Lestvičasti dijagram

Programabilni kontroleri su uglavnom i pre svega programiraju u takozvanom lestvičastom dijagramu (pored ovog prisutan je i naziv *ladder* ili relejni dijagram) koji nije ništa drugo do grafičko simbolično predstavljanje električnih kola. Izabrani simboli zapravo izgledaju slično šematskim simbolima električnih uređaja, što je olakšalo prelazak korisnika na programiranje *PLC* kontrolera.

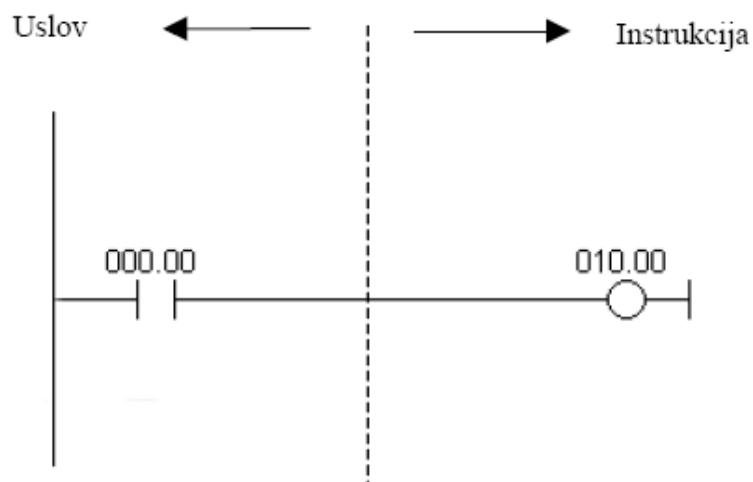
Korisnik koji nikad nije video *PLC* može da razume lestvičasti dijagram. Lestvičasti dijagram sastoji se od jedne vertikalne linije, koja se nalazi na levoj strani, i linija koje se granaju prema desnom delu. Linija sa leve strane naziva se *bus bar* a linije koje se granaju na desno su linije instrukcija. Duž linija instrukcija smešteni su uslovi koji vode do instrukcija pozicioniranih na desnom kraju dijagrama. Logička kombinacija ovih uslova određuje kada i na koji način se instrukcija na desnoj strani izvršava. Osnovni elementi lestvičastog dijagrama vide se na sledećoj slici, na primeru programske linije napisane u programskom jeziku SYSWIN za kontroler CPM1A.

Najveći broj instrukcija zahteva korišćenje najmanje jednog operanda, a često i više njih. Operand može biti neka memorijska lokacija, jedan bit memorijske lokacije ili neka numerička vrednost - broj. U gornjem primeru operand je bit 0 memorijske lokacije IR000. U slučaju kada se za operand želi proglasiti konstanta, koristi se oznaka # ispred numeričkog zapisa (da bi kompajler znao daje u pitanju konstanta a ne adresa).



Osnovni elementi lestvičastog dijagrama

Na osnovu gornje slike treba primetiti da se lestvičasti dijagram sastoji iz dva osnovna dela. Levi deo koji se naziva uslovni i desni koji sadrži instrukcije. Logika je u sledećem, kada se ispuni uslov instrukcija biva izvršena.



Uslov i instrukcija u lestvičastom dijagramu

Gornja slika predstavlja primer lestvičastog dijagrama u kome se aktivira relej u *PLC* kontroleru kada se pojavi signal na ulaznoj liniji 00. Parovi vertikalnih linija nazivaju se uslovi. Svaki uslov u lestvičastom dijagramu ima vrednost *ON* ili *OFF*, zavisno od statusa bita koji mu je dodeljen. U ovom slučaju taj bit je i fizički prisutan kao ulazna linija (klima) u *PLC* kontroler. Ukoliko se priključi taster na klemu koja mu odgovara, moguće je menjati stanje bita iz stanja logičke jedinice u stanje logičke nule i obratno. Stanje logičke jedinice se najčešće označava kao *ON* a stanje logičke nule kao *OFF* (po engleskim rečuma *on* i *off* koje bi u bukvalnom prevodu značile *uključeno* i *isključeno*).

Desni deo lestvičastog dijagrama je instrukcija koja se izvršava u slučaju da je levi uslov ispunjen. Postoji više vrsta instrukcija koje bi se najlakše mogle podeliti na jednostavne i složene. Primer jednostavne instrukcije je aktiviranje nekog bita u memorijskoj lokaciji. U gornjem primeru taj bit ima i fizičko značenje jer je povezan na relej unutar *PLC* kontrolera. Kada *CPU* aktivira neki od prvih četiri bita u reči IR010 kontakti releja se pomeraju i vrše spajanje linija koje su povezane na njega. U ovom slučaju to su linije spojene na klemu obeleženu sa 00 i jednu od COM klemu.

2.2 Normalno otvoreni i normalno zatvoreni kontakti

Pojmove *normalno otvoren* i *normalno zatvoren* veoma je važno razumeti jer se često sreću u industrijskoj praksi. Oba pojma se primenjuju na reči kao što su kontakti, ulaz, izlaz itd (sve kombinacije imaju isto značenje bez obzira da li se radi o ulazu, izlazu, kontaktu ili nečemu drugom). Suština je veoma jednostavna, *normalno otvoren* prekidač neće provesti struju dok nije pritisnut a *normalno zatvoren* prekidač će provoditi sve dok nije pritisnut. Dobri primeri za oba slučaja su zvono na ulaznim vratima i alarm za kuću.

Ako se izabere *normalno zatvoren* prekidač, zvono bi stalno radilo sve dok neko ne bi pritisnuo prekidač. Pritiskom na prekidač, otvaraju se kontakti i zaustavlja se protok struje do zvona. Naravno, tako koncipiran sistem ne bi nikako odgovarao vlasniku kuće. Bolji izbor bi svakako bio *normalno otvoren* prekidač, njegovim korišćenjem zvono neće raditi sve dok neko ne pritisne

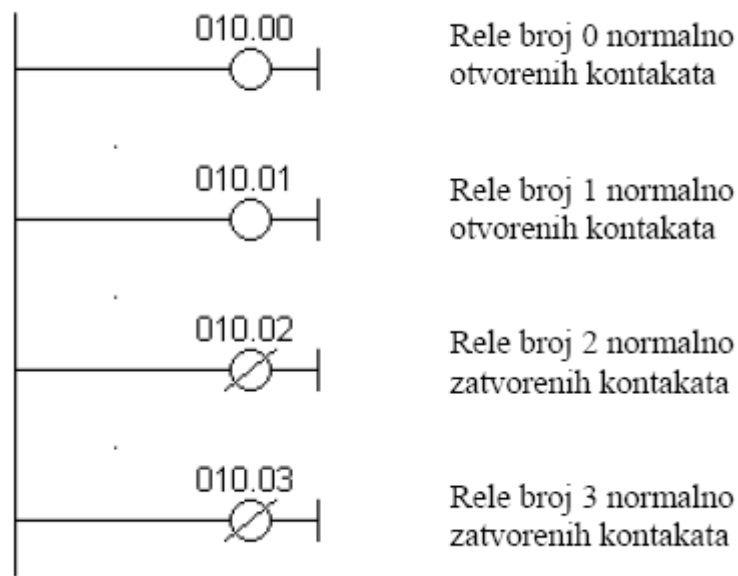
dugme na prekidaču i time označi svoje prisustvo pred vratima. Kućni sistem bezbednosti (kućni alarmni sistem) je primer upotrebe *normalno zatvorenog* prekidača. Pretpostavimo da je alarmni sistem namenjen nadgledanju ulaznih vrata u kuću. Jedan od načina da se ožiči kuća bi bio da se sprovede jedan normalno otvoren prekidač od svakih vrata do alarma (upravo kao i prekidač za zvono). Tada, ako bi se vrata otvorila, to bi zatvorilo prekidač i alarm bi se aktivirao. Ovako izveden sistem bi radio ali bi bilo problema. Neka se pretpostavi da prekidač ne radi, da je žica slučajno u prekidu ili se recimo prekidač polomi, itd (ima mnogo načina na koje bi sistem mogao da postane nefunkcionalan). Problem je što domaćin nikad ne bi znao da sistem ne radi. Provalnik bi mogao da otvori vrata, prekidač ne bi radio i alarm se ne bi aktivirao. Očigledno ovo nije dobar način kako napraviti sistem. Sistem treba da se postavi tako da se alarm aktivira od strane provalnika ali i sam od sebe ako neka od komponenti ne funkcioniše (domaćin svakako želi da zna ako sistem ne radi). Obzirom na ove nove okolnosti bolje je koristiti prekidač sa *normalno zatvorenim* kontaktima koji će detektovati neovlašćen ulaz (otvaranje vrata prekida tok struje i taj signal se koristi za aktiviranje zvučnog signala) ili kvar na sistemu kao što je prekid žice.

Razmatranja kao što su ova su još značajnija u industrijskom okruženju gde bi kvar mogao da prouzrokuje povredu nekog radnika. Jedan od takvih primera gde se koriste izlazi sa normalno zatvorenim kontaktima je sigurnosna ograda kod mašina za sečenje. Ukoliko se vrata ograde otvore prekidač deluje na izlaz sa normalno *zatvorenim kontaktima* i prekida kolo za napajanje čime mašina staje i time sprečava povređivanje radnika.

Pojmovi normalno otvoren i normalno zatvoren se mogu primeniti i na senzore. Senzori se koriste da bi se osetilo prisustvo fizičkih objekata, izmerila neka količina ili veličina. Na primer, jedna vrsta senzora može da se koristi da bi se detektovalo prisustvo kutije na pokretnoj traci, druga vrsta može da se koristi za merenje fizičke veličine kao što je toplota itd. Ipak, većina senzora je tipa prekidača. Njihov izlaz je u stanju *ON* ili *OFF* u zavisnosti od toga šta senzor *oseća*. Neka se kao primer uzme senzor koji je napravljen da oseti metal kada metalni deo prolazi kraj senzora. Za tu namenu mogao bi se upotrebiti senzor sa *normalno otvorenim* ili sa *normalno zatvorenim* kontaktom

na izlazu. Ako bi bilo potrebno obavestiti *PLC* svaki put kada deo prođe kraj senzora, trebalo bi izabrati senzor sa *normalno otvorenim* izlazom. Izlaz senzora bi se aktivirao samo ako bi metalni deo bio ispred senzora i odmah isključio kad bi deo prošao. *PLC* bi onda mogao da izračuna broj puta koliko se *normalno otvoren* kontakt na izlazu senzora aktivirao i time znao koliko je metalnih delova prošlo kraj senzora.

Pojmove *normalno otvoren* i *normalno zatvoren* kontakt treba i konkretno pojasniti na primeru ulaza i izlaza samog *PLC* kontrolera. Najlakše ih je objasniti baš na primeru releja.



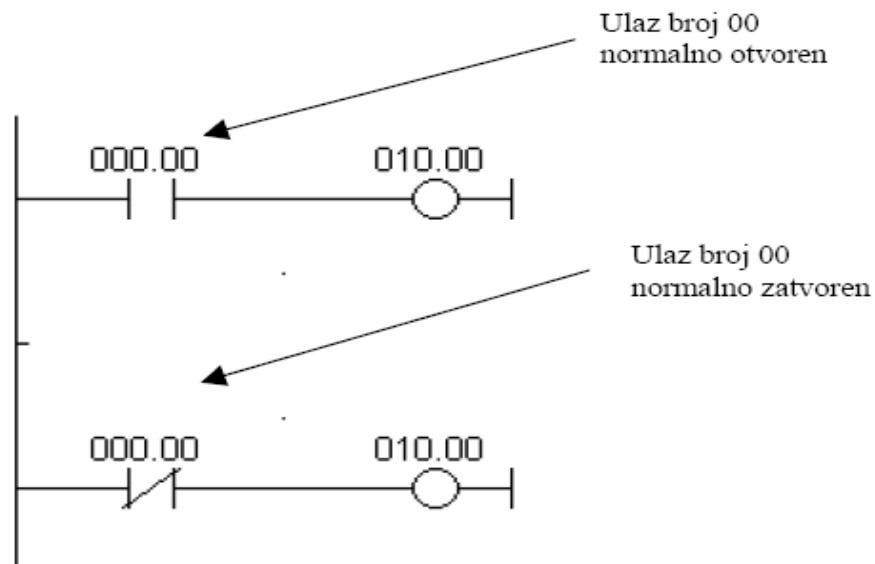
Primer normalno otvorenih i zatvorenih kontakata na izlazu PLC kontrolera

Normalno otvoreni kontakti bi predstavljali kontakte releja koji će po dovođenju signala izvršiti spoj. Za razliku od njih kod *normalno zatvorenih* kontakata signal će prekinuti kontakt tj. isključiti relej. Na prethodnoj slici se vidi kako to izgleda u praksi. Prva dva releja su definisani kao *normalno otvoreni* a druga dva kao *normalno zatvoreni*. Svi releji reaguju na pojavu signala! Prvi rele (00) ima signal i zatvara svoje kontakte. Drugi relej (01) nema signal i ostaje otvoren. Treći relej (02) ima signal i otvara svoje kontakte s obzirom da je definisan kao zatvoreni kontakt. Četvrti relej (03) nema signal i ostaje zatvoren jer je tako i definisan.

Pojmovi *normalno otvoren* i *normalno zatvoren* se mogu odnositi i na ulaze *PLC* kontrolera. Neka se kao primer ulaza u *PLC* kontroler iskoristi taster. Ulaz na koji je taster priključen se može definisati kao ulaz sa *otvorenim* ili *zatvorenim* kontaktima. Ako je definisan kao ulaz sa *normalno otvorenim* kontaktom pritisak na taster će aktivirati instrukciju koja se nalazi iza uslova. U ovom slučaju to će biti aktiviranje releja 00.

Ako je ulaz definisan kao ulaz sa *normalno zatvorenim* kontaktom pritisak na taster će prekinuti izvršenje instrukcije koja se nalazi iza uslova. U ovom slučaju to će prouzrokovati deaktiviranje releja 00 (relej je aktivan sve dok se ne pritisne taster). Na slici ispod je prikazan način povezivanja tastera i relejni dijagrami za oba slučaja.

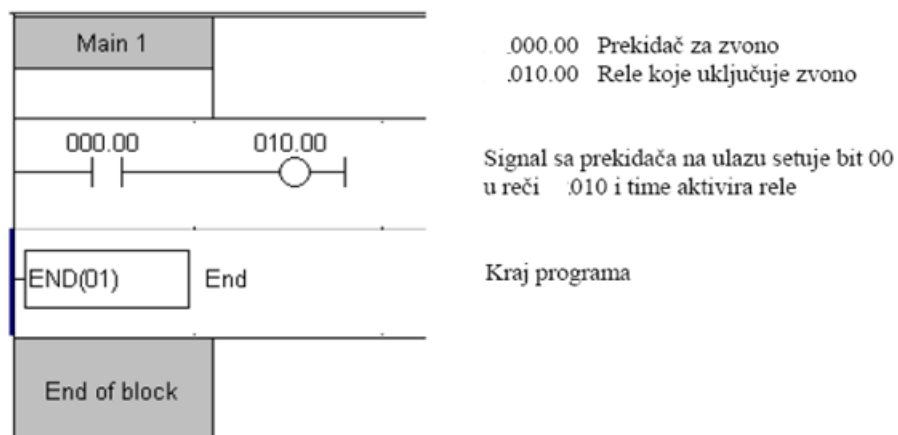
Normalno otvoreni / zatvoreni uslovi se u lestvičastom dijagramu razlikuju po dijagonalnoj liniji preko simbola. Ono što određuje uslov izvršenja (eng. *execution condition*) za instrukciju jeste status bita naznačenog ispod svakog uslova na liniji instrukcije. Normalno otvoreni uslov je *ON* ako njegov operand bit ima status *ON*, odnosno, njegov je status *OFF*, ako je takav i status njegovog operand bita. Normalno zatvoreni uslov je *ON* kada je njegov operand bit *OFF*, odnosno on ima status *OFF* kada je status njegovog operand bita *ON*.



Primer normalno otvorenih i zatvorenih kontakata na ulazu PLC kontrolera

2.3 Osnovni primer

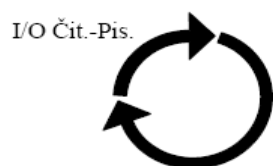
Donji primer predstavlja elementaran program. Primer se sastoji od jednog ulaznog uređaja i jednog izlaznog vezanog na izlaz *PLC* kontrolera. Ulazni uređaj je taster a izlazni zvonice koje se napaja preko kontakta releja 00 na izlazu *PLC* kontrolera. Ulaz 000.00 predstavlja uslov za izvršenje instrukcije nad bitom 010.00. Pritiskom na taster aktivira se bit 000.00 i ispunjava uslov za aktiviranje bita 010.00, čime zvono počinje sa radom. Za ispravan rad programa potrebna je još jedna linija programa sa *END* instrukcijom i time je program završen.



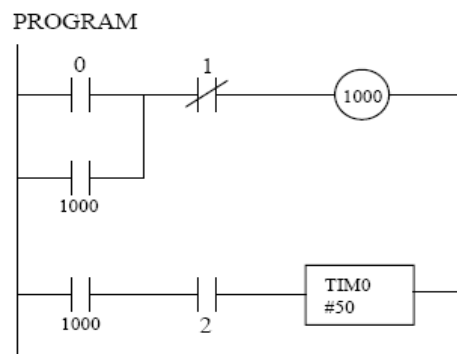
Lestvičasti dijagram kratkog primera

2.4 Princip rada PLC uređaja

Prihvat ulaza, obrada i ispis izlaza se ciklički ponavlja u skladu sa unesenim programom, određene dužine trajanja, zavisne od broja ulaza i izlaza i složenosti algoritma i vrste primenjenog procesora

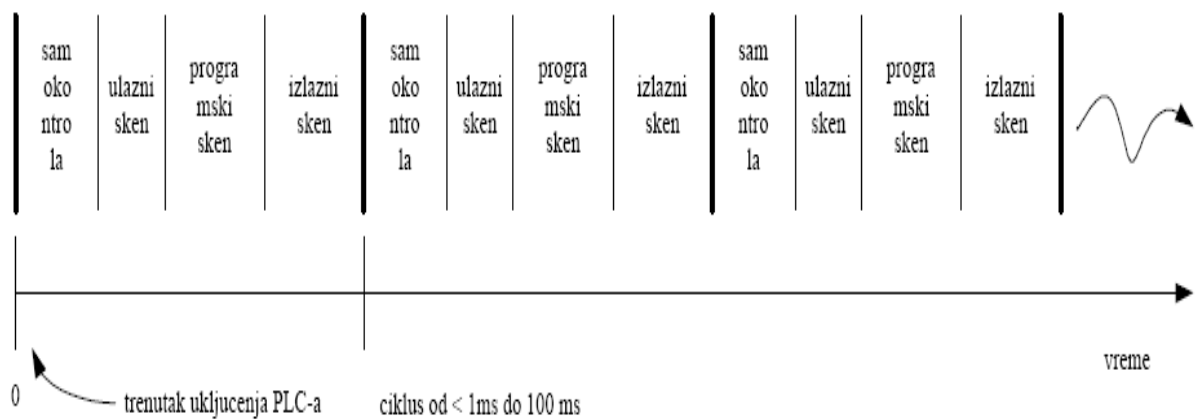


Programski ciklus



Programski ciklus obrade

Programski ciklus se sastoji od četiri faze. Pri inicijalizaciji, pri uključenju, *PLC* prvo proverava moguće greške u svom hardveru i softveru. Ako ih ne pronade, preuzima stanja ulaza (iz registara ulaza) i kopira njihove vrednosti u memoriju na zato predviđene lokacije. Taj postupak se naziva ulazni sken a podaci u memoriji se nazivaju slika ulaza. Koristeći ulazne podatke, odnosno njihovu sliku, procesor izvršava programske naredbe kojima su definisane odgovarajuće aritmetičko logičke funkcije u fazi koja se naziva programski sken. Pri tom se rezultati obrade smeštaju u zato predviđeno memorijsko područje nazvano slika izlaza. Po završetku programskog skena u fazi nazvanoj izlazni sken podaci iz slike izlaza prenose se na izlaze (registre izlaza). *PLC* nakon izlaznog skena pokreće nanovo čitav ciklus, proverava greške itd.



Ciklus rada PLC-a

Provera da li se programski ciklus izvodi pravilno, izvodi se uz pomoć hardverskog *watchdog* tajmera, koji se proziva u svakom skenu i predstavlja osnovnu garanciju sigurnog rada. Ako se to ne dogodi signalizira se greška u samom programu ili kvar na opremi kontrolera. Na taj način štiti se sistem upravljanja, naprimer od ulaska u beskonačnu petlju. U zavisnosti od primenjenog tipa procesora u *PLC*-u, ulazni i izlazni sken izvršavaju se u vremenu reda milisekundi (0.1 do 3ms), tako da se ciklus obrade ponavlja 10 do 100 puta u sekundi. Trajanje skena ciklusa obrade, posebno programskog dela zavisi od veličine programa.

3 GX-WORKS programski paket za programiranje *PLC* kontrolera

GX-WORKS programski paket spada u aplikacije koje su namenjene za rad sa *Mitsubishi* -ovim *PLC* kontrolerima .