

PROGRAMIRANJE PLC-A



- 4.1 PROGRAMIRANJE PLC-A
 - 4.2 PROGRAMSKI JEZICI ZA PLC-OVE
 - 4.3 LESTVIČASTI DIJAGRAM
 - 4.4 REPERTOAR NAREDBI
 - 4.5 PREKIDI
 - 4.6 POTPROGRAMI
-

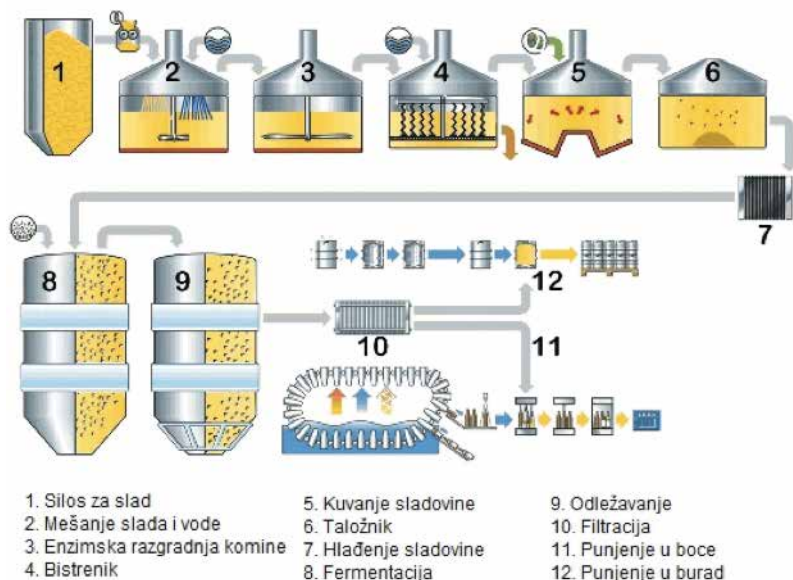
4.1 Programiranje PLC-a

Pod pojmom programiranje PLC-a podrazumevaju se svi postupci koji se obavljaju pri izradi softvera za PLC. Oni se mogu svrstati u dve faze:

- Izrada programa – pomoću posebnog inženjerskog softvera (alata) instaliranog na personalnom (ili industrijskom) računaru pravi se namenski aplikativni program koji PLC treba da izvršava.
- Prebacivanje programa u PLC (preuzimanje programa; engl. *download*) – program se iz računara prebacuje u PLC direktno preko kabla ili memorijske kartice.

Kako bi se pristupilo izradi programa prvo treba detaljno proučiti proces kojim PLC treba da upravlja, tj. tehnologiju samog procesa. Da bi se, recimo, automatizovao jedan proizvodni proces u pivari, tehnolog treba da dostavi detaljan opis procesa u tehnološkom smislu da bi programer stekao jasnu i kompletnu sliku o procesu – od toga kako se pripremaju sirovine i kako počinje sam proces, pa sve do glavnih i završnih faza, od procesa fermentacije i filtracije do punjenja flaša ili limenki (slika 4.1). Bez obzira na svoje znanje i veštine, programer je nemoćan bez poznavanja tehnologije proizvodnje ili pomoći tehnologa jer ne zna šta treba da uradi. Neki programeri su zahvaljujući dugogodišnjem iskustvu na istim ili sličnim programskim zadacima ovladali tehnologijom procesa koji automatizuju pa mogu izvesti programske zadatke i bez tehnologa. Oni su u izvesnom smislu i programeri i tehnolozi jer su se kroz višegodišnje iskustvo specijalizovali za određenu oblast.

U suštini, sve se svodi na to da treba sagledati i izdvojiti glavni programski zadatak ili više njih, i – ukoliko je to moguće – u okviru glavnog programskog zadatka izdvojiti i manje programske zadatke koji se mogu posmatrati kao posebne programske celine. Na taj način program će imati znatno jasniju strukturu. Da bi se napravio odgovarajući program, takođe je neophodno da programer tačno zna hardversku konfiguraciju PLC-a u smislu toga koliko je digitalnih i analognih ulaza i izlaza korišćeno, kao i to koji je senzor, prekidač, aktuator ili neki drugi uređaj priključen na koji ulaz ili izlaz PLC-a.



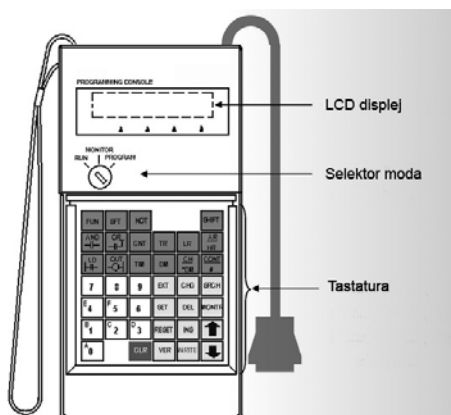
Slika 4.1 Tehologija proizvodnje piva

PLC se može programirati pomoću:

1. programskih konzola
2. personalnih računara
3. memorijskih kartica

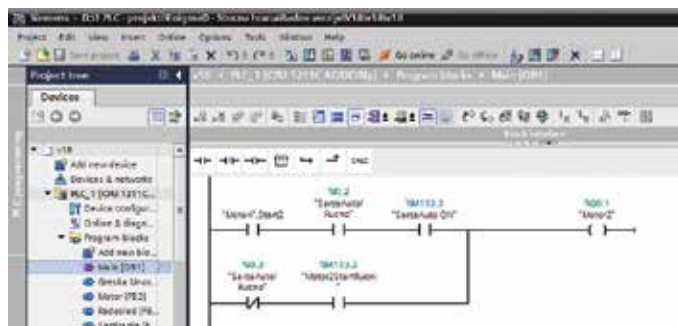
PLC se ranije mogao programirati i pomoću programskih konzola (slika 4.2) koje predstavljaju specijalnu vrstu terminala. One se ne mogu koristiti za izradu veoma složenih programa, ali su podesne za unos manjih izmena u programu ili monitoring nekih parametara. U suštini, većina proizvođača PLC-ova nikad nije ni imala programske konzole u ponudi, a danas su gotovo iščeznule s tržišta.

Danas se program za PLC pravi na personalnim računarima a potom prebacuje u PLC direktnim povezivanjem s računarom preko kabla ili mreže.



Slika 4.2 Programska konzola (OMRON)

Da bi se PLC programirao preko računara potreban je odgovarajući kabl za povezivanje PLC-a i personalnog računara, ali i odgovarajući softver (slika 4.3).



Slika 4.3 TIA portal – Inženjerski softver

Današnji laptop računari su idealno rešenje za programiranje i reprogramiranje PLC-a na licu mesta u samom fabričkom pogonu.

Proizvođači industrijske i procesne opreme vrlo često u svojoj ponudi imaju i specijalno prilagođene laptop računare za rad u industrijskom okruženju. Ti računari imaju zaštitu od elektromagnetnih smetnji, otporniji su na uslove rada u industrijskim halama, a imaju i instalirane licencirane softvere za programiranje PLC-ova, izradu SCADA aplikacija i sl. S obzirom na to da se koriste na terenu, odnosno u samom pogonu ili polju (engl. *field*), oni se često nazivaju i „field PG“ (slika 4.4).



Slika 4.4 Field PG

PLC se može programirati i pomoću memorijskih kartica. Taj način je izuzetno pogodan kad programiranje treba da izvrši operater na procesu, odnosno lice koje nema nikakvog iskustva u programiranju. Programer koji je napravio program za PLC može taj program da snimi na memorijsku karticu, i da je zatim pošalje kurirskom službom ili na neki drugi način na udaljenu lokaciju na kojoj se nalazi PLC (slika 4.5).

Da bi se PLC programirao pomoću memorijske kartice na kojoj je sačuvan program, treba samo



Slika 4.5 Memorijska kartica

ubaciti memorijsku karticu u PLC, a u većini slučajeva i ispoštovati određenu jednostavnu proceduru koja se uglavnom sastoji u tome da se memorijska kartica ubaci u PLC, zatim da se PLC isključi s napajanja i ponovo uključi.

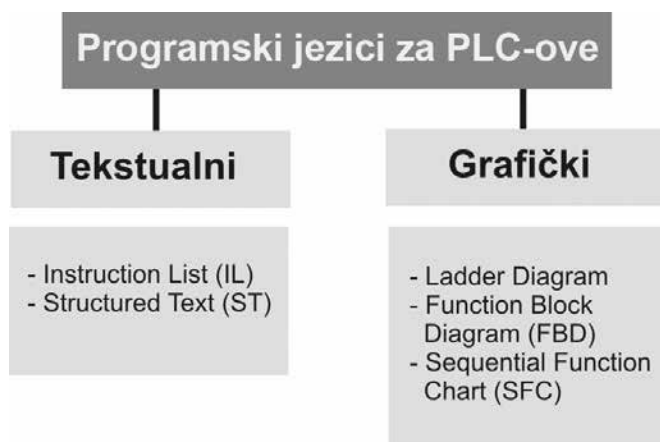
Kao preduslov za rad nekih PLC-ova neophodno je da u okviru CPU-a postoji memorijska kartica jer kod njih u okviru CPU-a ne postoji posebna memorija predviđena za smeštanje programa. Kod drugih tipova PLC-a, memorijska kartica služi samo za prebacivanje programa u PLC ili iz njega. Memorijska kartica PLC-a Simatic S7-1200 može se upotrebiti i u jednu i u drugu svrhu. Ukoliko se ona konfigurira kao programska kartica (engl. *program card*), u njoj CPU čuva aplikativni program koji izvršava, a kad se konfigurira kao prenosna kartica (engl. *transfer card*), služi samo za trenutno prebacivanje programa u PLC ili iz njega. Osim navedenih funkcija, ona može da se koristi i za ažuriranje/nadogradnju (engl. *update/upgrade*) firmvera, arhiviranje podataka itd.

4.2 Programski jezici za PLC-ove

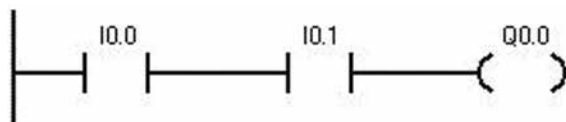
Stariji PLC-ovi su se uglavnom programirali pomoću lestvičastih dijagrama koji su po svojoj strukturi podsećali na šematske dijagrame kojima su se objašnjavale relejne mreže. Moderni PLC-ovi su doneli velike novine i poboljšanja, kako u svojoj hardverskoj strukturi, tako i u softverskim rešenjima. Današnji PLC-ovi se programiraju na više načina pa je zato bila potrebna i standardizacija programskih jezika. U skladu sa zahtevima koje su nametnuli tržište, korisnici, pa i sami proizvođači PLC-ova, usvojen je internacionalni standard IEC 61131-3 kojim je definisano i dozvoljeno sledećih pet programskih jezika za programabilne logičke kontrolere:

1. *Instruction List (IL)* – lista naredbi
2. *Structured Text (ST)* – strukturiran tekst
3. *Ladder Diagram* – lestvičasti dijagram
4. *Function Block Diagram (FBD)* – funkcijski blok dijagram
5. *Sequential Function Chart (SFC)* – sekvencijalni funkcijski grafikoni.

Prva dva jezika s ove liste pripadaju klasi tekstualnih jezika, pri čemu je IL programski jezik u klasi asemblerskih jezika, dok je ST programski jezik sličan Pascal programiranju. Ostala tri jezika pripadaju klasi grafičkih jezika, među kojima je jezik lestvičastih dijagrama (*Ladder Diagram*) i dalje ubedljivo najzastupljeniji programski jezik kada je u pitanju programiranje PLC-a (slika 4.6)



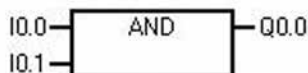
Slika 4.6 Programski jezici za PLC-ove



a.)

ID	I 0 . 0
∧	I 0 . 1
=	Q 0 . 0

b.)



c.)

Slika 4.7 Isti program napisan u vidu lestvičastog dijagrama (a), liste naredbi (b) i funkcijskih blok dijagrama (c)

Izbor programskog jezika

Izbor programskog jezika koji će programer koristiti za programiranje PLC-a zavisi od više faktora: od ličnih afiniteta programera, mogućnosti PLC-a, specifičnosti projektnog zahteva itd. U određenim okolnostima, pogodnije je koristiti jedan programski jezik, dok se u drugim situacijama kao bolje rešenje nameće neki drugi programski jezik (slika 4.8).

Lista naredbi (IL) programski je jezik blizak svakome ko ima iskustva s programiranjem mikroprocesora ili programiranjem u assemblyskom jeziku. Ovaj jezik se sadrži od koda s mnogo programskih linija, i u njemu svaka linija predstavlja jednu operaciju. Ukoliko programer koristi samo IEC definisane naredbe, program napisan na ovom jeziku lako se može prilagođavati različitom hardveru. IL je jezik najnižeg nivoa pa se kao takav i izvršava mnogo brže od grafički orijentisanih jezika, kao što je, recimo, jezik lestvičastih dijagrama. Pored toga, prednost ovog jezika je i to što program napisan na njemu zahteva mnogo manje prostora u memoriji. Međutim, bez obzira na prednosti koje obezbeđuje programerima, njegov veliki nedostatak je taj što nije omiljen među krajnjim korisnicima, inženjerima održavanja i servisnim



Slika 4.8 Programski jezici

inženjerima jer nema preglednu grafičku strukturu koja omogućava laku preglednost i brzo delovanje u slučaju kvarova i eventualnih izmena.

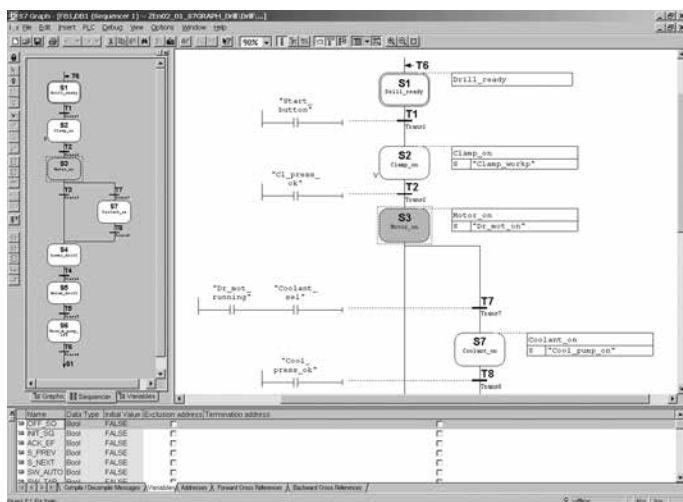
ST je sličan višim programskim jezicima kao što su PASCAL ili C. Trigonometrijska izračunavanja i analize podataka mogu se obavljati značajno lakše u ovom jeziku nego u jeziku lestvičastih dijagrama ili jeziku IL. Njegova prednost je u tome što danas većina tek diplomiranih inženjera ima više iskustva u računarskim programskim jezicima, nego u razumevanju relejnih i električnih mreža, pa samim tim mogu biti i produktivniji u programiranju na jeziku ST nego na jeziku lestvičastih dijagrama. Mana mu je to što nije blizak većini inženjera koji imaju dugogodišnje iskustvo u radu sa PLC-ovima, to jest, struktura programa napisanog u ST-u nije poznata većini inženjera koji rade na održavanju ili servisiranju PLC-ova.

Činjenica je da su lestvičasti dijagrami najzastupljeniji jezici za programiranje PLC-ova. Njihova ogromna prednost ogleda se u tome što su slični konvencionalnim električnim relejnim dijagramima, pa uz malo truda mogu biti razumljivi i ljudima koji nemaju programerska iskustva i sposobnosti, a poseduju znanja iz oblasti elektrotehnike. Njihova mana je to što se u slučaju značajnog povećanja veličine programa sve teže čitaju i razumeju, pa je neophodno dobro i promišljeno dokumentovati i komentarisati programske linije i delove programa.

Posle lestvičastog dijagrama, drugi po zastupljenosti je zasigurno jezik funkcijskih blok dijagrama (FBD). Pomoću njega se formira struktura programa koju je lako pratiti. Sa izlaza jednog bloka ide se na ulaz drugog bloka, tako da se praćenjem različitih putanja u okviru programa može lako ustanoviti funkcija određenih delova programa. Ovaj jezik predstavlja dobar izbor za početnike koji imaju samo osnovna znanja iz digitalne elektronike, a nikakvog iskustva u radu s relejnom logikom.

SFC ima jednostavan koncept. Ceo program je podeljen i grupisan u akcione blokove (slika 4.9). Akcioni blok u okviru koga se nalazi programski kôd, napisan na jeziku po izboru programera, aktivan je sve dok sledeći akcioni blok ne počne da se izvršava. Za prelazak na sledeći akcioni blok, neophodno je naravno da se ispune odgovarajući uslovi. S druge strane, ovaj stil programiranja nije pogodan za bilo koju aplikaciju jer korišćenje ovog jezika može u nekim slučajevima uneti nepotrebnu kompleksnost u strukturu programa. Dodatni nedostatak ovakvog programiranja jeste taj što ono iziskuje mnogo vremena za pripremu i planiranje pre nego što se uopšte počne s programiranjem jer, u suprotnom, SFC može biti izuzetno težak za praćenje. Program napisan na SFC-u izvršava se sporije od programa pisanih na drugim jezicima, a nemoguća je i njegova konverzija u druge jezike. IL, funkcijski blokovi i lestvičasti dijagrami lako se mogu prevoditi

jedan na drugi, dok sa SFC-om to nije slučaj. Sve u svemu, ovaj programski jezik može biti dobro rešenje u slučajevima kad je hardver PLC-a moćan, tj. kad ima brzinu i memoriju dovoljnu da isprati zahteve SFC-a, i naravno kad su lični afiniteti i sposobnosti programera ili krajnjeg korisnika više okrenute ovom programskom jeziku. Još jedna prednost ovog jezika je i to što je pogodan za krajnjeg korisnika i servisera koji brzo može da ustanovi u okviru kog se programskog akcionog bloka javio zastoj ili kvar.



Slik 4.9 Siemens-ov S7-Graph jezik (u klasi jezika SFC)

SIEMENS-ov programski jezik STEP 7, koji predstavlja inženjerski softver za programiranje PLC-ova, potpuno je kompatibilan sa standardom IEC 61131-3, tako da on programeru nudi mogućnost korišćenja svih programskih jezika definisanih ovim standardom. Osim programa Ladder, FBD i IL (Statement List), Step 7 omogućava i korišćenje jezika S7-SCL (koji je zapravo jezik ST) i S7-Graph (koji je po IEC 61131-3 standardu jezik SFC).

Kriterijum za izbor programskog jezika	Programski jezik
Jednostavna upotreba za krajnje korisnike	SFC
Univerzalno prihvaćen jezik	Ladder
Brzina izvršavanja u PLC-u	IL & ST
Aplikacije sa pretežno digitalnim I/O	Ladder & FBD
Lako naknadno menjanje programa	Ladder
Lako korišćenje kompleksnih matematičkih operacija	ST