

Programirljivi logički kontroleri (PLC)

Šepac, Marin

Undergraduate thesis / Završni rad

2016

Degree Grantor / Ustanova koja je dodijelila akademski / stručni stupanj: **University of Rijeka, Faculty of Humanities and Social Sciences / Sveučilište u Rijeci, Filozofski fakultet u Rijeci**

Permanent link / Trajna poveznica: <https://um.nsk.hr/um:nbn:hr:186:747767>

Rights / Prava: [In copyright](#)/[Zaštićeno autorskim pravom.](#)

Download date / Datum preuzimanja: **2024-12-01**



Repository / Repozitorij:

[Repository of the University of Rijeka, Faculty of Humanities and Social Sciences - FHSSRI Repository](#)



SVEUČILIŠTE U RIJECI
FILOZOFSKI FAKULTET U RIJECI

Odsjek za politehniku

Marin Šepac

Programirajući logički kontroleri
(PLC)

(završni rad)

Rijeka, 2016. godine

SVEUČILIŠTE U RIJECI
FILOZOFSKI FAKULTET U RIJECI

Studijski program: sveučilišni preddiplomski studij politehnike

Marin Šepac
0248019648

Programirajući logički kontroleri (PLC)

-završni rad -

Mentor : mr. sc. Gordan Đurović

Rijeka, 2016. godine

FILOZOFSKI FAKULTET U RIJECI

Odsjek za Politehniku

U Rijeci, 11. ožujka 2016. godine

ZADATAK ZA ZAVRŠNI RAD

Pristupnik: **Marin Šepac**

Studij: **Sveučilišni preddiplomski studij politehnike**

Naslov završnog rada: **Programirajući logički kontroleri (PLC) / Programmable Logic Controllers (PLC)**

Znanstveno područje: **2. Tehničke znanosti;**

Znanstveno polje: **2.03. Elektrotehnika;**

Znanstvena grana: **2.03.06 automatizacija i robotika**

Kratak opis zadatka: Opisati povijesni razvoj programirajućih logičkih kontrolera. Prikazati strukturu PLC sustava, njegove komponente te način njihovog povezivanja. Opisati vrste programskih jezika namijenjenih programiranju PLC-ova, njihove međusobne prednosti i razlike. Na primjeru jednostavnog sustava s automatskim upravljanjem koji upravlja garažnim vratima prikazati programski kôd u ladder dijagramu s detaljnim pojašnjenjima pojedinih elemenata programa.

Zadatak uručen pristupniku: **23. ožujka 2016. godine**

Ovjera prihvatanja završnog rada od strane mentora: _____

Završni rad predan: _____

Datum obrane završnog rada: _____

Članovi ispitnog povjerenstva: 1. predsjednik - _____

2. mentor - _____

3. član - _____

Konačna ocjena: _____

Mentor

mr. sc. Gordan Đurović

Izjava

Izjavljujem da sam završni rad izradio samostalno, isključivo znanjem stečenim na Odsjeku za politehniku Filozofskog fakulteta u Rijeci, služeći se navedenim izvorima podataka i uz stručno vodstvo mentora mr.sc. Gordana Đurović.

Sažetak

Ovaj rad se bavi programirljivim logičkim kontrolerima koji su postali temelj današnje moderne industrije i automatizacije. Programirljivi logički kontroleri (PLC) unaprijedili su sustav upravljanja te su zamijenili dosadašnji sustav koji je bio baziran na elektromehaničkim relejima. Prvi dio ovog rada bavi se povijesnim pregledom razvoja automatiziranih sustava upravljanja i samih programirljivih logičkih kontrolera. Dana je usporedba automatiziranih sustava upravljanih PLC-uređajima sa onim sustavima upravljanim relejima te su navedene mnogobrojne prednosti uvođenja PLC uređaja u proizvodne procese. Navedeno je hardversko sklopovlje koje sačinjava jedan PLC uređaj. Na jednostavan način je opisan princip rada samog uređaja. Na kraju su navedeni programski jezici koji se koriste u programiranju PLC uređaja te osnove i karakteristike svakog pojedinog programskog jezika. U drugom dijelu rada izrađen je jedan praktičan primjer automatskog upravljanja garažnim vratima.

Ključne riječi: automatizacija, proizvodni proces, sustav upravljanja, programibilni logički kontroleri, programiranje

Summary

This paper analyses programmable logic controllers that have become the foundation of today's modern industry and automation. Programmable logic controller (PLC) improved the production system management and replaced the previous system which was based on electromechanical relays. The first part of this paper deals with the historical overview of the development of automated control systems and programmable logic controllers themselves. There is a comparison of automated systems managed by PLC devices with those systems controlled by relays and listed numerous advantages of PLC devices in production processes. There are mentioned all the components of one PLC device. In a simple manner there has been described the working principle of the device. In the end there is a list of programming languages that are used in the PLC programming and the basics and characteristics of each programming language. The second section is made of one practical example of automatic control garage door.

Keywords: automation, manufacturing process, management system, programmable logic controllers, programming

SADRŽAJ

1. UVOD	1
2. POVIJEST PLC UREĐAJA	2
2.1. Osobine prvog PLC uređaja	3
2.2. Prednost upravljanja PLC uređajima nad klasičnim relejnim sustavom upravljanja ..	4
3. PLC UREĐAJ	5
3.1. Centralna procesorska jedinica (CPU)	6
3.2. Mikroprocesor	6
3.3. Izvor napajanja	7
3.4. Komunikacijsko sučelje.....	8
3.5. Memorija uređaja.....	8
3.6. Moduli za proširenje.....	9
3.7. Ulazi u PLC uređaj	10
3.8. Izlazi iz PLC uređaja	10
3.9. Princip rada PLC uređaja.....	11
4. PROGRAMIRANJE PLC UREĐAJA	12
4.1. Testiranje programa – simulacija stvarnih stanja	13
4.2. Ljestvičasti dijagram (<i>ladder diagram</i>) – LD.....	14
4.3. Instrukcijske lista (<i>statement list</i>) – STL.....	17
4.4. Funkcijski blokovski dijagram (<i>function block diagram</i>) – FBD.....	19
5. PRIMJER UPRAVLJANJA GARAŽNIM VRATIMA	21
6. ZAKLJUČAK	24
7. LITERATURA	25
8. POPIS SLIKA	26

1. UVOD

Moderna industrija kakvu danas znamo nezamisliva je bez automatike i automatizacije. Pod pojmom automatizacije smatramo prijenos čovjekova rada na strojeve i usko je vezana uz mehanizaciju. Dok mehanizacija oslobađa čovjeka fizički zahtjevnih poslova, automatizacija zamjenjuje čovjeka u fizičkom ali i umnom radu.

Stari proizvodni sustavi bili su vođeni relejnim sustavom upravljanja. To je bio spoj mnoštva elektromehaničkih releja povezanih žicom. Kada bi se proizvodni proces i najmanje promijenio, to bi značilo mukotrpano preslagivanje i ponovno spajanje releja u jednu funkcionalnu cjelinu. Za to vrijeme proizvodnja bi stajala i došlo bi do praznog hoda, što bi značilo ogromne gubitke po kompaniju.

Kako je danas težnja na što većoj produktivnosti i što većem profitu, relejni sustav upravljanja nije se uklapao u tu priču. Tako se rodila ideja o sustavu upravljanim programirljivim logičkim kontrolerom - PLC uređajem. PLC uređaji zamijenili su stari sustav upravljanja te su postavili novi standard u modernoj industriji. Glavna odlika ovih sustava jest da su modularni i fleksibilni, tako da se brzo mogu prilagoditi današnjoj dinamičnoj proizvodnji.

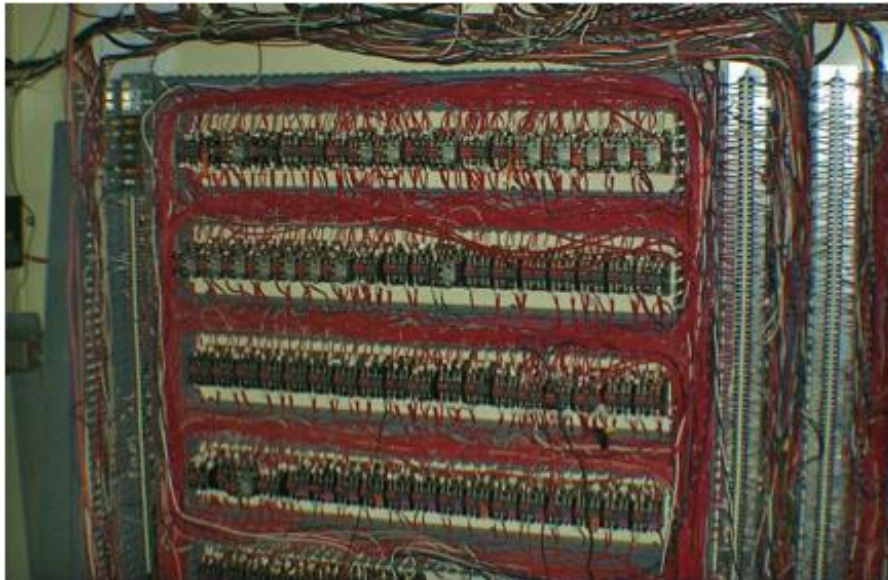
U prvom dijelu ovog rada bit će riječi o razvoju automatiziranih sustava kroz povijest te o počecima sustava upravljanja zasnovanim na programirljivim logičkim kontrolerima. Isto tako, napraviti ću usporedbu između relejnih sustava upravljanja i sustava upravljanim PLC uređajem, te istaknuti koje su to bitne prednosti prelaska na novi sustav upravljanja.

U drugom dijelu rada navest ću koje to komponente čine jedan cjeloviti PLC kontroler, koje su njihove karakteristike te na koji način sve to funkcionira.

Dalje, spomenut ću na koji način se programiraju PLC uređaji, koji su programski jezici koji se koriste u programiranju PLC uređaja te koji su njihovi prednosti i nedostaci. U zadnjem dijelu rada izradit ću jedan praktičan zadatak, primjer automatskog upravljanja garažnim vratima koji će sažeti sve napisano o PLC uređajima.

2. POVIJEST PLC UREĐAJA

Krajem šezdesetih i početkom sedamdesetih godina, procvatom industrijske proizvodnje, javlja se potreba za unapređenjem samih industrijskih pogona te procesa koji se u njima odvijaju. Glavni razlog bio je poboljšanje uvjeta i kvalitete rada, povećanje produktivnosti i fleksibilnosti. U to doba, industrijski pogoni su bili upravljani sustavima koji su se temeljili na relejnim krugovima. To je bio ogroman sustav koji je sadržavao mnoštvo elektromehaničkih releja, ponekad i preko stotinu, fizički povezanih žicama. Dakle, inženjer bi projektirao čitavu logiku sustava upravljanja dok bi električar sve te releje povezo prema shematskom planu logike.



Slika 2.1 - Relejni sustav upravljanja

Najveća mana ovog sustava jest u samim relejima. Releji su mehanički uređaji, a mehanički uređaji sadrže pokretne dijelove koji su podložni trošenju i kvarenju. Ako bi samo jedan relej "ispao" iz sustava, bilo je potrebno ispitati cijeli sustav ne bi li se našao istrošeni relej te se zamijenio novim. Sustav bi tokom tog perioda bio van funkcionalnosti, kao i proizvodnja. Nadalje, ukoliko bi se i najmanje promijenili parametri proizvodnje, trebale su se napraviti i prilagodbe upravljačkih sklopova. To je opet zahtijevalo čitavo projektiranje novog sustava te mukotrpno povezivanje mnoštva releja. Rezultat toga bio je prazan hod u proizvodnji koji za posljedicu ima gubitak proizvodnog vremena te ogromne gubitke po kompaniju. Potencijalni problemi bili bi ukoliko inženjer krivo projektira sustav ili električar krivo poveže sustav ili se pak u sustav ugradi mehanički neispravan relej. Jedini način da se provjeri ispravnost sustava jest da se sustav pusti u pogon.

Najmanje neispravnosti u sustavu donosile bi ogromne gubitke. Nije teško za zaključiti da su ovi sustavi bili veoma "skupi" i nefleksibilni. Trebalo je pronaći bolje rješenje za automatizaciju proizvodnih pogona.

Velika konkurencija u automobilskoj industriji natjerala je inženjere na razmišljanje i kreativnost. Cilj je bio biti što uspješniji u tržišnoj utakmici, a tu je prvi iskorak napravila američka kompanija 'General Motors'. Javlja se zamisao o implementiranju elektroničko – kompjuterskog upravljačkog sustava koji bi zamijenio postojeći neefikasni sustav. Takav sustav bi se dao reprogramirati jednostavnom izmjenom proizvodnog programa, bez dugotrajnih zastoja i bez zamornih povezivanja releja. 1968. godine 'General Motors Hydromatic Division', odjel kompanije "General Motors", izdaje smjernice za dizajniranje prvog logičkog kontrolera. Zahtjevi su bili da takvo mikroračunalo bude otporno na industrijske uvjete (povišena temperatura, niska temperatura, prašina, vibracije, vlaga, elektromagnetski utjecaji...), da se jednostavno programira te da ima mogućnosti da se reprogramira i ponovno koristi za neke druge zadatke. Nakon izdanih smjernica, 'General Motors' je tražio potencijalne kompanije koje bi razvijale proizvod. Kompanija 'Gould Modicon' je prva izradila uređaj koji je odgovarao izdanim specifikacijama.

2.1. Osobine prvog PLC uređaja

Novi uređaj je pružao funkcionalnost koju su do tada obavljali relejni krugovi, zamjenjujući njihovu ožičenu logiku. Isto tako, bili su modularni i proširivi prema potrebi. Uređaje je jednostavno bilo implementirati u zahtjevno industrijsko okruženje i bili su jednostavni za korištenje. Najveća prednost novog uređaja bila je ta što se za njegovo korištenje nije morao učiti novi programski jezik. Programirao se jezikom koji su inženjeri i električari već bili znali – ljestvičastim dijagramom (*ladder diagram*). Uređaj je prvotno nazvan PC kontroler (*Programmable Controllers*).

Pojavom osobnih računala (*Personal Computers*), programibilni kontroleri dobivaju naziv programibilni logički kontroleri (*Programmable Logic Controllers*) kako bi se izbjegla zabuna kod krajnjih korisnika. U veoma kratkom vremenu, PLC uređaji postaju široko prihvaćeni te se počinju koristiti za vođenje procesa proizvodnje i u drugim industrijama, ponajviše u prehrambenoj industriji.

2.2. Prednost upravljanja PLC uređajima nad klasičnim relejnim sustavom upravljanja

- U usporedbi sa klasičnim relejnim sustavom, ožičenje se smanjuje za 80%
- Potrošnja struje potrebne za upravljanje također se uvelike smanjuje
- PLC-ova funkcija za samodijagnosticiranje omogućuje brzo otkrivanje problema te uklanjanje grešaka
- Zbog brzine PLC-a smanjilo se potrebno vrijeme za jedan ciklus što je dovelo do povećanja produktivnosti
- Reprogramibilnost
- Veća pouzdanost sustava
- Jednostavno i lako pohranjivanje i čuvanje programa
- Kod složenih sustava, gdje su upravljačke funkcije složene, a broj ulaza/izlaza veliki, povećava se i isplativost korištenja PLC sustava u usporedbi sa klasičnim relejnim sustavom upravljanja

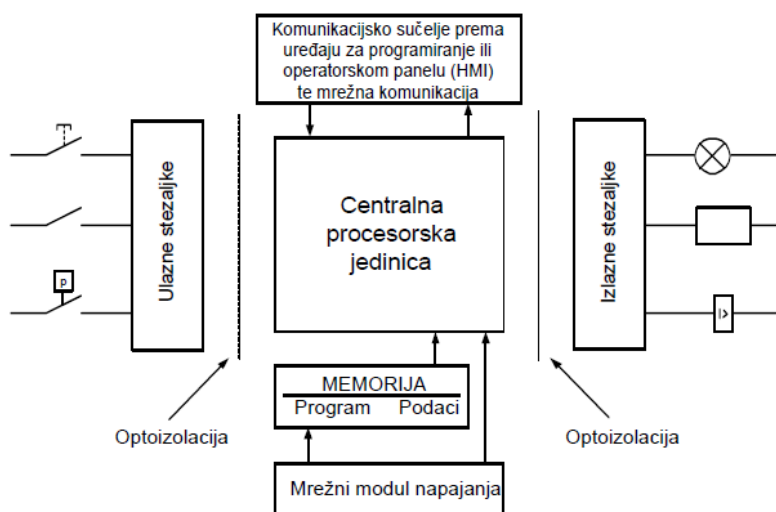
3. PLC UREĐAJ

Ukratko, PLC je programirajući logički kontroler, tj. industrijsko računalo koje je izumljeno prvenstveno kako bi zamijenilo postojeće sekvencijalne relejne krugove u upravljanju proizvodnim pogonima u industriji. U sustavu upravljanja, PLC se često naziva srcem upravljačkog sustava.



Slika 3.1 - Siemens PLC SIMATIC S7 - 300

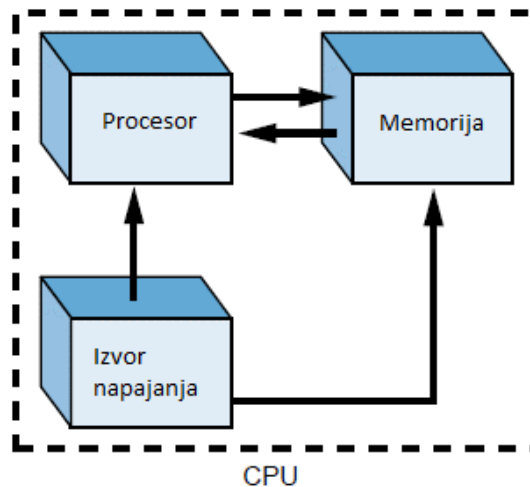
Na tržištu postoje mnogi uređaji od raznoraznih proizvođača, no njihova hardverska struktura je uglavnom ista. Jedan tipičan predstavnik PLC uređaja sastoji se od centralne procesorske jedinice (CPU), ulaznih i izlaznih dijelova koji mogu biti digitalni ili analogni, memorijskog bloka, mrežnog modula koji služi za napajanje i komunikaciju te modula za proširenje.



Slika 3.2 - Shema PLC uređaja

3.1. Centralna procesorska jedinica (CPU)

Centralna procesorska jedinica (CPU) je najvažniji element PLC uređaja. Ona je "mozak" sustava upravljanja. Sastoji se od tri komponente: mikroprocesora, memorije i izvora napajanja koji su međusobno u interakciji. Generalno, mikroprocesor izvodi program koji je pohranjen u memoriji, dok izvor napajanja osigurava sav potreban napon kako bi osigurao nesmetani rad procesora i memorije. Arhitektura centralne procesorske jedinice može varirati od proizvođača do proizvođača, međutim sve se svodi na te tri komponente.



Slika 3.3 - Shema CPU

3.2. Mikroprocesor

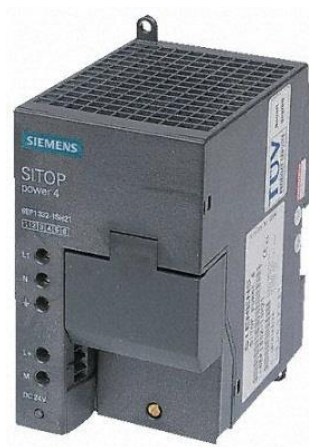
Veoma mali mikroprocesor koji se nalazi u centralnoj procesorskoj jedinici ustvari je integrirani krug koji ima zapanjujuću sposobnost računanja i upravljanja čitavim sustavom. Izvode matematičke operacije, upravlja operacijama, brine se o međusobnoj interakciji ostalih dijelova PLC uređaja, izvršava programe koje zadaje korisnik, nadgleda ulaze te na temelju njih postavlja izlaze...Mikroprocesori koji se upotrebljavaju u PLC uređajima kategoriziramo na osnovu duljine zapisa koje upotrebljavaju pri izvođenju operacija. Standardne veličine jesu 8, 16 i 32 bita. Veličina zapisa koji procesor može obraditi bitno utječe na brzinu izvođenja operacija. Za primjer, 32-bitni procesor obavlja operacije puno brže od 16-bitnog procesora iz razloga što obrađuje duplo više podataka u jednoj operaciji.



Slika 3.4 - Mikroprocesor

3.3. Izvor napajanja

Izvor napajanja je također jedan od ključnih elemenata čitavog sustava. Njegova uloga nije samo osigurati potreban napon za pojedine komponente sustava, već nadgledati i regulirati napon koji se dovodi te upozoriti centralnu procesorsku jedinicu ukoliko nešto nije kako treba biti. Većina PLC uređaja radi na izmjenični izvor napajanja (AC), međutim neki uređaji mogu raditi i na istosmjerni izvor napajanja (DC). Potreban napon napajanja je 220 VAC ili 24 VDC. Kako u stvarnosti dolazi do oscilacija u vrijednostima napona, modul izvora napajanja mora biti u mogućnosti tolerirati $\pm 10\%$ od nominalnog iznosa. To znači da ukoliko napon bude 198 VAC i 242 VAC, sustav bi trebao normalno funkcionirati. Sve izvan navedenog intervala trebalo bi javiti grešku procesoru.



Slika 3.5 - Uređaj za napajanje PLC-a

3.4. Komunikacijsko sučelje

Glavna zadaća komunikacijskog sučelja jest komunikacija sa računalom na kojem se piše upravljački program. Program se putem komunikacijskog kabla transferira u PLC uređaj. Komunikacijsko sučelje nam još nudi mogućnost povezivanja PLC uređaja sa ostalim PLC uređajima i raznoraznim sensorima. Komunikacija se vrši preko njihove interne veze.



Slika 3.6 - Shematski prikaz komunikacijskog sučelja

3.5. Memorija uređaja

Memorija sustava je mjesto gdje se pohranjuju sve instrukcije i svi korisnički programi. Cjelokupna memorija sustava podijeljena je na dva segmenta: izvršnu memoriju i korisničku memoriju. Izvršna memorija sadrži trajno pohranjene programe i potprograme koji su sastavni dijelovi PLC uređaja. Ti takozvani nadzorni programi zaslužni su za izvođenje korisničkih programa kao i komunikaciju sa ulazno – izlaznim uređajima. Ovaj dio memorije nije dostupan niti vidljiv krajnjem korisniku. Korisnička memorija je mjesto gdje se pohranjuju korisnički programi. Korisnik može pristupiti toj memoriji i mijenjati sadržaj.

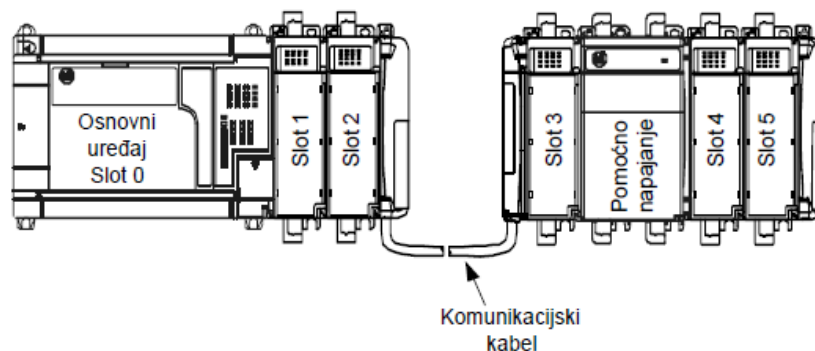
Kako postoje instrukcije i programi koji moraju trajno biti spremljeni u memoriji, tako postoje i podaci koji su podložni promjenama. Prema tome, postoje različite vrste memorije na koje se podaci spremaju:

- **ROM (Read – Only Memory)** je memorija na koju se podaci trajno spremaju, bez mogućnosti mijenjanja. Imuni su na promjene tokom nestanka struje. PLC uređaji rijetko upotrebljavaju ROM memoriju.
- **RAM (Random – Access Memory)** je memorija koja je dizajnirana na način da se informacije na nju mogu zapisivati i čitati. Ne zadržava podatke ukoliko ostane bez izvora napajanja. RAM memorija u PLC uređaju obično je podržana baterijom kako se po nestanku napajanje podaci ne bi izgubili.

- **PROM (Programmable read – only memory)** je posebna vrsta ROM memorije koja se može programirati samo jednom. Rijetko je u upotrebi kod PLC uređaja.
- **EPROM (Erasable programmable read – only memory)** je posebna vrsta PROM memorije koja se može izbrisati u cijelosti kada je izložena UV svjetlu te se ponovno programirati.
- **EAROM (Electrically alterable read – only memory)** je memorija slična EPROM memoriji, samo se podaci umjesto UV svjetlom brišu električnim putem.
- **EEPROM (Electrically erasable programmable read - only memory)** je vrsta memorije za trajnu pohranu podataka. Podaci se mogu brisati i ponovno programirati električnim putem. Dakle, nudi trajnu pohranu podataka koji se jednostavno mogu promijeniti programatorom. EEPROM memorija se koristi u PLC uređajima.

3.6. Moduli za proširenje

Svaki PLC uređaj ima ograničen broj ulaza i izlaza. Ukoliko je potrebno, taj broj se može povećati preko modula za proširenje. Modul za proširenje je poseban uređaj koji se spaja na PLC. Sadrži dodatne ulazne i/ili izlazne stezaljke koje mogu biti drugačijeg tipa od onih na PLC uređaju. Napajaju se uglavnom iz PLC uređaja na kojeg su priključeni iako mogu imati i zasebno napajanje što nije preporučljivo. Veza PLC uređaja i modula za proširenje ostvaruje se komunikacijskim kablom.



Slika 3.7 - Modul za proširenje

3.7. Ulazi u PLC uređaj

Učinkovitost automatiziranog postrojenja uvelike ovisi o sposobnosti samog PLC uređaja da čita i obrađuje podatke koje dobiva na ulazu. Ulaz u PLC čine priključne vijčane stezaljke na koje se spajaju različite vrste senzora te ulazni uređaji. Tu se dešava prilagodba vanjskog signala, signala iz okoline u signal razumljiv procesoru PLC-a. Senzori koji se koriste u automatiziranim pogonima jesu senzori temperature, senzori pritiska, senzori brzine, senzori razine tekućine itd. Ulazni uređaji preko kojih se odvija komunikacija čovjek – PLC jesu raznorazne sklopke, tipkala i tipkovnice. Signal koji PLC prima na svojim ulazima mogu biti digitalne i analogne. Analogni je signal npr. naponski signal s nekog od prethodno navedenih senzora, dok digitalni signal može biti signal sa ulaznih uređaja poput sklopke ili tipkala.

Između ulaznih stezaljki i centralne procesorske jedinice (CPU) nalazi se optoizolacija. Svrha optoizolacije jest da se galvanski odvoje strujni krugovi ulaza te procesora kako bi se spriječio protok struje uslijed potencijalnih razlika strujnih krugova, te sprečavanje visokofrekventnih smetnji filtriranjem signala.

3.8. Izlazi iz PLC uređaja

Izlazi iz PLC uređaja su vijčane stezaljke na koje se spajaju izvršni uređaji kojima se upravlja. To su najčešće motori, zvučna i svjetlosna signalizacija, pneumatski razvodnici i sl. Izlazi mogu biti analogni ili digitalni. Digitalni izlazni signal radi kao prekidač, dok se analogni izlaz koristi za generiranje analognog signala (npr. reguliranje brzine motora preko iznosa napona koji odgovara željenoj brzini).

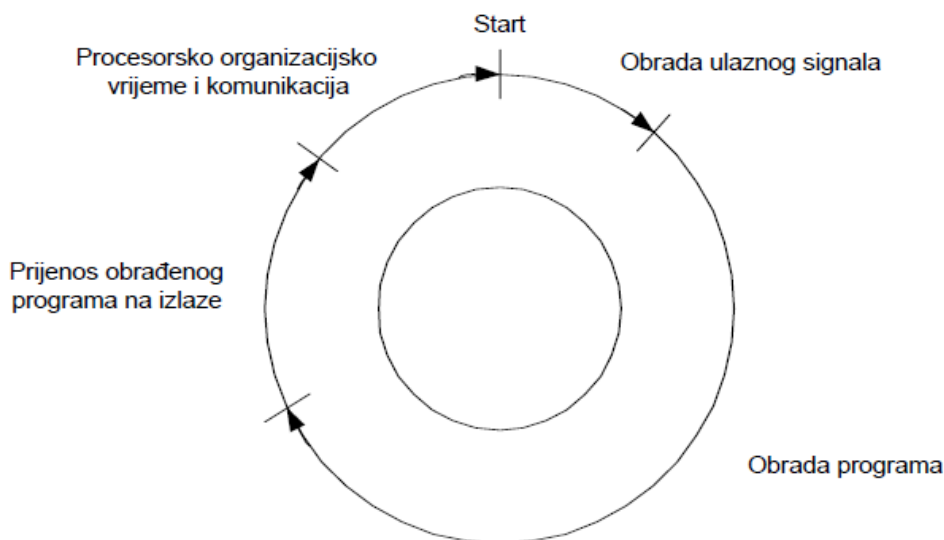
Između izlaznih stezaljki i centralne procesorske jedinice (CPU) se također nalazi optoizolacija radi galvanske izolacije vanjskog i unutarnje električnog kruga.

3.9. Princip rada PLC uređaja

Princip rada PLC uređaja najzornije možemo prikazati slikom. Osnova rada je kontinuirano skeniranje i obrada podataka koje se odvija ciklički u beskonačnoj petlji.

PLC uređaj korigira stanje izlaza na osnovu promjene ulaza, kako je to određeno logikom u korisničkom programu. Cijeli taj ciklus podijelili smo na četiri faze:

1. Obrada ulaznog stanja – PLC provjerava svaki od ulaza (senzor, prekidač i sl.) te podatke koje prikupi sprema u memoriju centralne procesorske jedinice.
2. Obrada programa – Prema stanju ulaznih signala i na osnovu logike korisničkog programa, rezultat se sprema u izlazni memorijski registar centralne procesorske jedinice.
3. Prijenos obrađenog programa na izlaze – Spremljeni podaci iz izlaznog memorijskog registra, obrađeni u prethodnom koraku, prosljeđuju se na izlaze PLC uređaja.
4. Procesorsko organizacijsko vrijeme i komunikacija –Odvijaju se potrebne operacije za ispravno funkcioniranje operativnog sustava te komunikacija sa vanjskim jedinicama.



Slika 3.8 - Ciklus rada PLC uređaja

4. PROGRAMIRANJE PLC UREĐAJA

Kako su se s vremenom PLC uređaji sve više razvijali i nadograđivali, tako su se paralelno s njima razvijali i programski jezici. Programski jezici omogućuju korisnicima da kreiraju kontrolni program za automatizirani sustav upravljan PLC uređajem. PLC uređaj potom izvodi programski kod te prema njegovim instrukcijama upravlja određenim procesom, tj. upravlja ulazima i izlazima.

Pisanje programa uglavnom se izvodi na osobnim računalima na kojem postoji programska podrška za određeni PLC uređaj. Kako danas na tržištu postoji mnoštvo proizvođača PLC uređaja, svaki proizvođač uz kupljeni PLC uređaj isporučuje i odgovarajuću programsku podršku koja se sastoji od:

- **Uređivača teksta (*Text editor*)**, koji služi za pisanje korisničkog programa, liniju po liniju,
- **Prevoditelja (*Compiler*)**, koji provjerava sintaksu napisanog programskog koda,
- te **komunikacijske programske podrške**, koja služi za prijenos ispravno napisanog programskog koda sa osobnog računala u RAM memoriju PLC uređaja.

Drugi način programiranja PLC uređaja jest preko ručnih programatora koji se spajaju na PLC uređaj. Ručni programatori posjeduju mali LCD zaslon te tipkovnicu. Uglavnom se koriste za manje izmjene programa kada je to potrebno obaviti trenutno i na licu mjesta, u pogonu.

Također, postoje PLC uređaji koji posjeduju mali LCD zaslon i nekoliko funkcijskih tipki. Takvi se mali PLC uređaji koriste za programiranje nekih jednostavnijih procesa.



Slika 4.1 - Siemens LOGO!

4.1. Testiranje programa – simulacija stvarnih stanja

Kako bi sam proces programiranja PLC uređaja bio uspješan, potrebno je na neki način ispitati napisani program. Ispitivanje se vrši na način da se na ulaze dovedu stanja veličina koje odgovaraju onima iz stvarnih uvjeta u samom procesu. Za takav način ispitivanja programa koriste se takozvani simulatori stanja (*states simulators*). Simulator stanja je ustvari niz prekidača na ulazu pomoću kojih se mijenja stanje ukoliko se radi o digitalnim ulazima, odnosno potencijometar ako se radi o analognim ulazima, te kontrolnih lampica (led dioda) na izlazu.

Kako to funkcionira? Kada npr. želimo testirati rad protusmrzavajućeg senzora, koji preklopi kada temperatura koja se mjeri padne ispod ništice (0°C), na njegovo mjesto postavimo prekidač kojeg preklopimo i on sada umjesto senzora simulira da je temperatura pala ispod ništice. Ako se aktivira očekivani *izlaz* na kojem je spojena kontrolna lampica i ona zasvijetli, znači da se ustvari dogodio očekivani scenarij i simulacija je izvršena uspješno kao i testiranje programa.

Također, danas se ispitivanje može vršiti i softverski što je praktičnije jer nema potrebe za izradom simulatora.

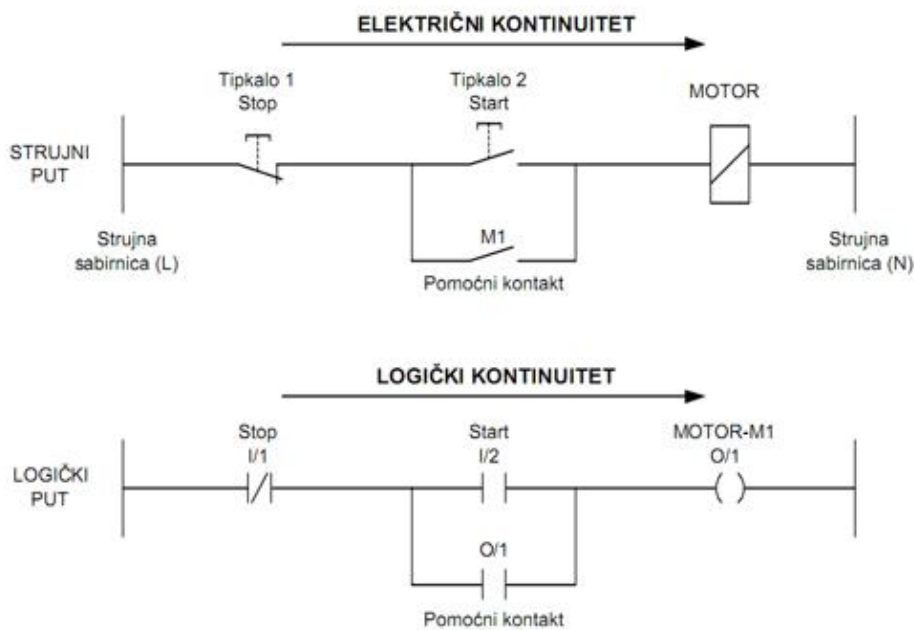
Za programiranje PLC uređaja postoje mnogi programski jezici. Najviše u upotrebi jest način programiranja ljestvičastim ili kontaktnim dijagramima (*ladder diagram*). Ovaj način programiranja datira još od samih početaka PLC uređaja, a temelji se na prilagodbi relejnih upravljačkih shema. Uz ljestvičaste dijagrame, u upotrebi su još i instrukcijske liste (*statement list*) koje predstavljaju programiranje na nivou assemblera (tekstualni programski jezik) te funkcijsko blokovski dijagrami (*function block diagram*) – grafičko programiranje koji za prikazivanje logike koriste logičke kutije iz Booleove algebre.

Postoji još i mogućnost programiranja pomoću BASIC i C programskih jezika, no ove metode nemaju široku zastupljenost i sve ostaje na samoj mogućnosti.

4.2. Ljestvičasti dijagram (*ladder diagram*) – LD

Kao što je već navedeno, ljestvičasti dijagrami nastali su na bazi strujnih upravljačkih shema. U upotrebi su od samih početaka PLC uređaja. Mogućnost programiranja ljestvičastim dijagramima jedan je od glavnih razloga uspjeha PLC uređaja. Sličnosti među načinom programiranja PLC uređaja ljestvičastim dijagramima i relejnom ljestvičastom logikom kakva se prije koristila u upravljanju proizvodnim pogonima, olakšala je sami prelazak s relejnom sustava upravljanja ka sustavima upravljanim PLC uređajima.

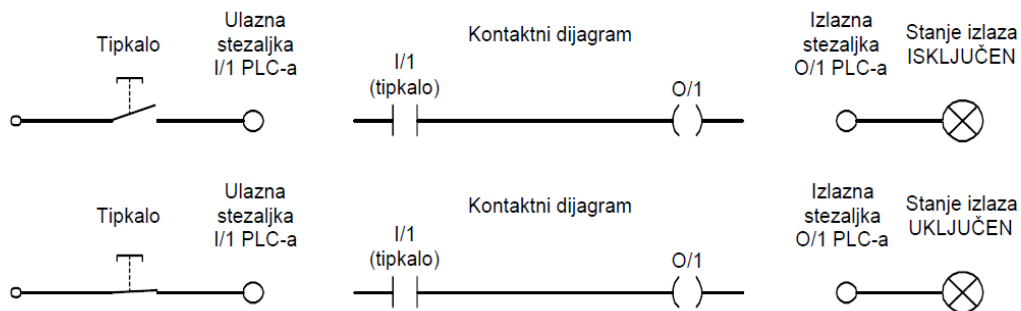
Uspoređujući strujne sheme sa ljestvičastim dijagramima PLC programskog jezika vidimo mnoge sličnosti. Na slici je prikazana shema jednostavnog START – STOP strujnog upravljačkog kruga. Rad upravljanog uređaja, u ovom slučaju motora, ovisi o poziciji tipkala, odnosno pomoćnog kontakta.



Slika 4.2 - Usporedba strujnog i logičkog puta

Ljestvičasti dijagram vrlo je sličan shemi strujnog upravljačkog kruga. Kod sheme strujnog kruga, simboli predstavljaju stvarne uređaje i njihovo ožičenje, dok kod ljestvičastih dijagrama oni predstavljaju naredbe u programu. Također, shema strujnog kruga prikazuje stanje kontakta (otvoreno/zatvoreno) dok se u ljestvičastom dijagramu logikom ispituje da li je neka naredba istinita (true) što predstavlja logičko '1' ili je neistinita (false) što predstavlja logičko '0'.

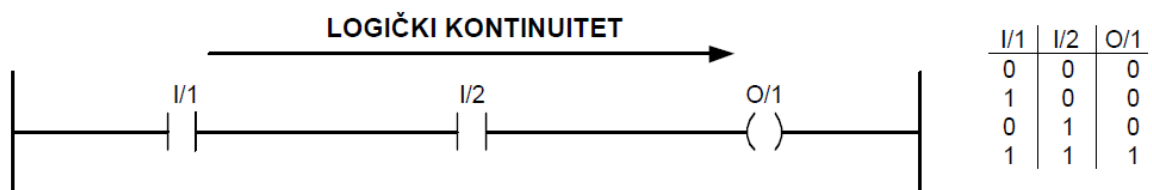
Svaki logički put u ljestvičastom dijagramu ima najmanje jednu izlaznu naredbu, a da bi se ona izvršila najmanje jedan ili više uvjeta mora biti zadovoljeno. Uvjeti su signali koje PLC uređaj procesuiraju sa ulaza. Na desnoj strani logičkog puta nalazi se izlazna naredba koja ima status UKLJUČENO/ISKLJUČENO, ovisno o stanju uvjeta. Na slijedećoj slici možemo vidjeti primjer logičkog puta u ljestvičastom dijagramu koji prikazuje ovisnost izlaza o ulazu.



Slika 4.3 - Primjer ljestvičastog dijagrama

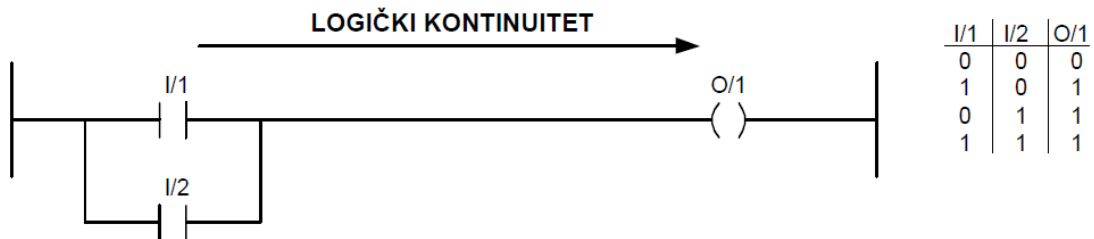
Programiranje ljestvičastim dijagramom možemo svesti na dvije osnovne kombinacije logičkih naredbi 'I' i 'ILI' te njihovim izvodima.

Na sljedećoj slici dan je primjer dvije serijski spojene logičke naredbe 'I'. Da bi se izlaz aktivirao, moraju biti zadovoljena oba logička uvjeta. Ako je samo jedan od dva logička uvjeta u stanju '0', prekida se logički kontinuitet i izlaz neće biti aktiviran.



Slika 4.4 - Ljestvičasti dijagram logičke operacije 'I'

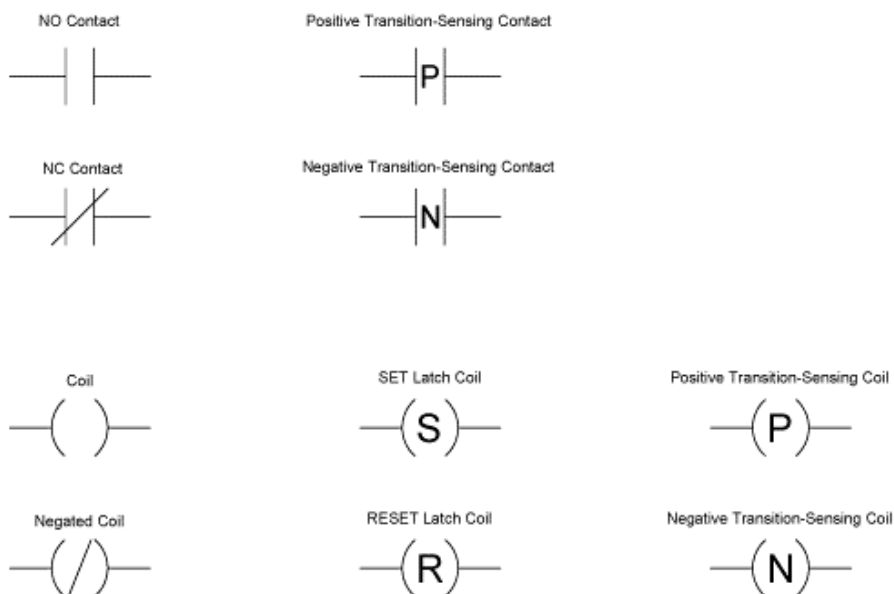
Ako se dvije naredbe povežu paralelno, dobijemo 'ILI' logičku operaciju. Izlaz će se aktivirati ako sam jedan od dva uvjeta bude ispunjen. Kada niti jedan od dva uvjeta nije ispunjen, gubi se logički kontinuitet i izlaz neće biti aktiviran, što prikazuje sljedeća slika:



Slika 4.5 - Ljestvičasti prikaz logičke operacije 'ILI'

Kombinacijom osnovnih logičkih operacija možemo izvesti čitavu Booleovu algebru, što nam daje hrpu mogućnosti kod samog programiranja. Programi se uglavnom sastoje od više logičkih krugova i treba naglasiti da se program izvodi odozgo prema dolje, od prvog logičkog kruga prema zadnjem.

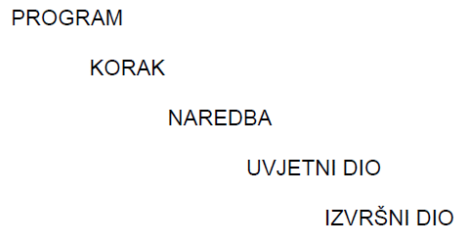
Na sljedećoj slici prikazani su osnovni simboli prema standardu IEC 1131-3 te je navedeno što oni predstavljaju.



Slika 4.6 - Osnovni simboli korišteni u programiranju ljestvičastim dijagramom

4.3. Instrukcijske lista (*statement list*) – STL

Instrukcijske liste (*statement list*) je programski jezik namijenjen programiranju PLC uređaja služeći se naredbama na nivou asemblerskog jezika. Svaka linija koda predstavlja jednu naredbu za procesor. Hijerarhiju programa možemo prikazati na sljedeći način:

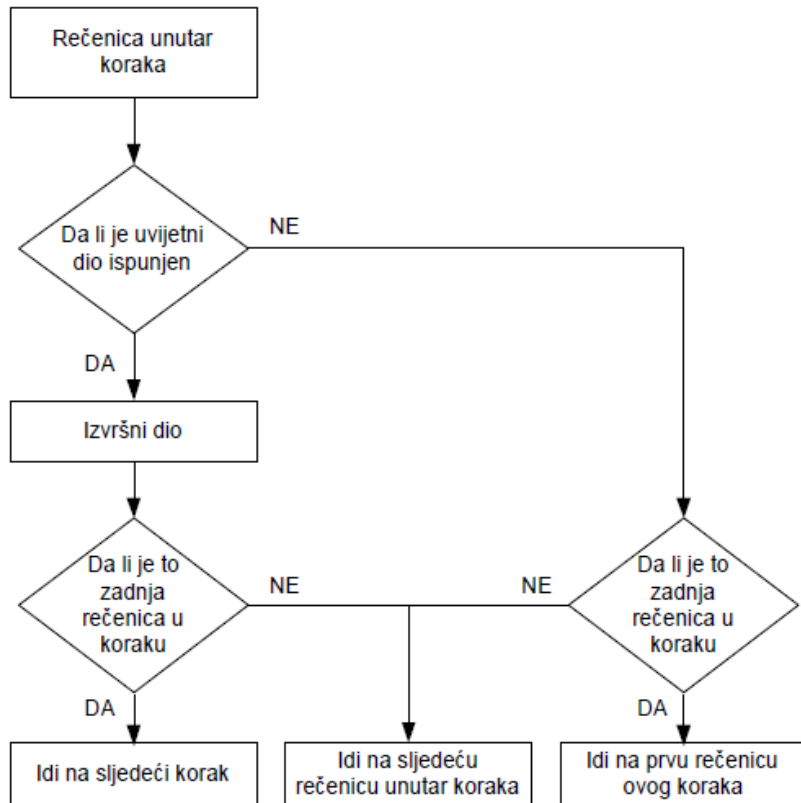


Slika 4.7 - Hijerarhija programa pisanog instrukcijskim listama

Program se sastoji od niza koraka. Koraci se izvršavaju redom, od prvoga prema zadnjem. Korak predstavlja logički blok unutar kojeg je napisan neki dio programskog koda, tj. niz naredbi. Naredbe se zadaju tako da imaju uvjetni i izvršni dio.

Uvjetni dio sadrži jedan ili više uvjeta čije se stanje provjerava u toku programa. Uvjetni dio počinje riječju 'IF' iza koje slijede uvjeti koji se provjeravaju, dok izvršni dio počinje riječju 'THEN'. U konačnici, to znači da ako su ispunjeni uvjeti u uvjetnom dijelu naredbe, šalje se logičko stanje u izvršni dio koji potom izvršava zadane zadatke.

Programiranje sa koracima ima jedno pravilo kojim se određuje tok izvođenja programa, a glasi: 'Ako uvjetni dio zadnje rečenice unutar nekog koraka nije ispunjen i izvršni dio se ne izvršava, program ne ide na sljedeći korak nego se vraća na prvu rečenicu istog koraka'. Program se vrti u petlji sve dok se uvjet iz zadnje rečenice ne zadovolji. To možemo prikazati na sljedećem dijagramu:



Slika 4.8 - Proces izvođenja programa instrukcijskim listama

Pošto nije uvijek poželjno da se program vrti u petlji unutar jednog koraka, napravljena je naredba 'NOP' zbog koje uvjet uvijek biva zadovoljen.

Ostale osnovne naredbe:

'=' - naredba pridruženja; 'SET' – postavlja vrijednost veličine bita iz '0' u '1'; 'RESET' – postavlja vrijednost veličine bita iz '1' u '0'; 'OTHRW' – ukoliko uvjetni dio nije zadovoljen, izvršava se jedan dio izvršnog dijela; 'AND' – logičko 'I'; 'OR' – logičko 'ILI'; 'XOR' – isključivo 'ILI'.

Osnovne matematičke operacije (zbrajanje (+), oduzimanje (-), množenje (*), dijeljenje (/)) se koriste uz naredbu LOAD TO.

Naredbe usporedbe (veće (>), manje (<), jednako (=), veće ili jednako (≥), manje ili jednako (≤), nejednako (<>)) uvijek dolaze u uvjetnom dijelu uz IF THEN naredbu.

4.4. Funkcijski blokovski dijagram (*function block diagram*) – FBD

Funkcijski blokovski dijagram je jedan od grafičkih načina programiranja PLC uređaja, baš kao i programiranje ljestvičastim dijagramima. Ovaj programski jezik možemo opisati kao grafički jezik za prikaz toka signala i podataka kroz funkcijske blokove, što simbolično možemo prikazati sljedećom slikom:

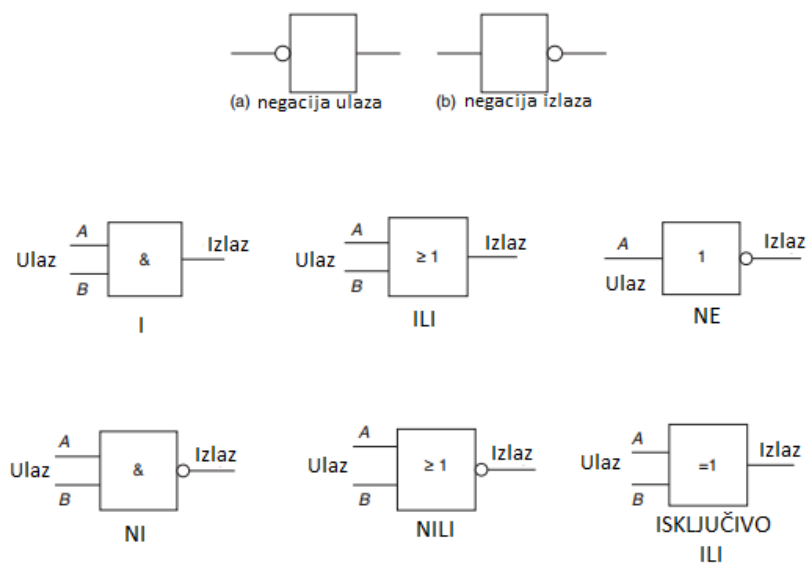


Slika 4.9 - Funkcijski blok

Funkcijski blok je prikazan kao pravokutni blok sa ulazima s lijeve strane i izlazom smještenim na desnoj strani.

Funkcijski blokovi mogu imati standardne funkcije poput brojača vremena ili mogu biti definirane od strane korisnika.

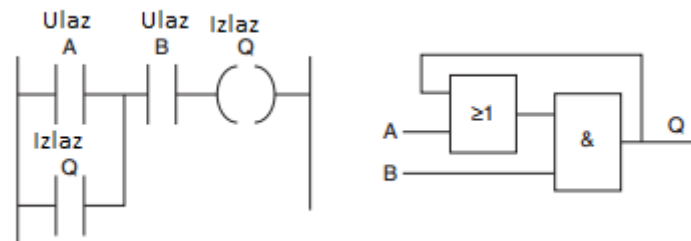
Na sljedećoj slici prikazane su osnovne funkcije u ovom načinu programiranja.



Slika 4.10 - Osnovne funkcije u funkcijsko blokovskom dijagramu

U prvom redu prikazane su funkcije sa negiranim ulazom i negiranim izlazom. Funkcija 'I' aktivira izlaz kada i ulaz A i ulaz B daju rezultat 'ISTINA', što odgovara logičkoj '1'. Funkcija 'ILI' funkcionira na način da se izlaz aktivira ukoliko je ili ulaz A ili ulaz B jednak 'ISTINI', tj. logičkoj '1'. Funkcija 'NE' aktivira izlaz ukoliko je stanje na ulazu jednako logičkoj '0'. Kombinacijom osnovnih funkcija možemo dobiti izvedene funkcije poput 'NI', 'NILI' i 'ISKLJUČIVO ILI'.

Na slici možemo vidjeti ljestvičasti dijagram koji prikazuje funkciju gdje izlaz ima ulogu ulaza. Ta ista funkcija implementirana je i u funkcijsko blokovskom dijagramu kao povratna veza.

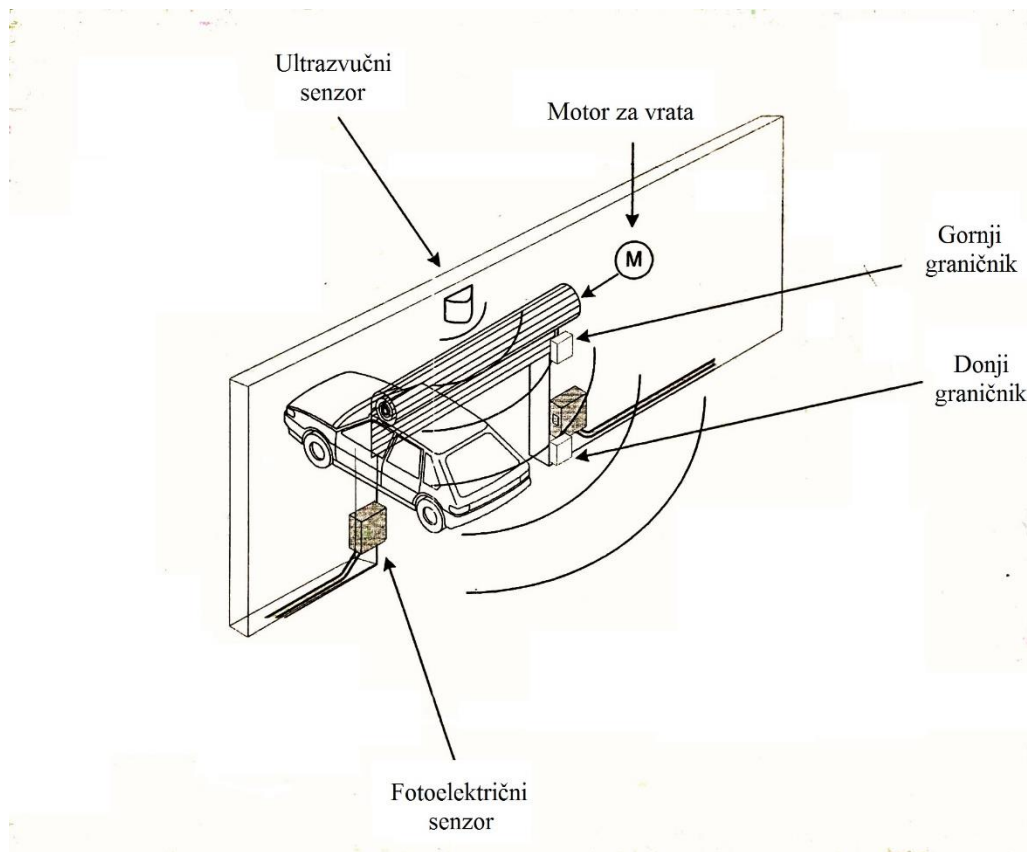


Slika 4.11 - Usporedni prikaz ljestvičastog dijagrama i funkcijsko blokovskog dijagrama

5. PRIMJER UPRAVLJANJA GARAŽNIM VRATIMA

Garažna vrata mogu se automatizirati tako da se o njihovom otvaranju i zatvaranju ne mora brinuti čovjek. Cilj ovog praktičnog zadatka jest osmisliti sustav automatskog upravljanja garažnim vratima te prikazati programski kôd za implementaciju istoga.

Princip rada je sljedeći: Ultrazvučni senzor koji je postavljen iznad garažnih vrata detektira nadolazeće vozilo čime se stvara uvjet za podizanje vrata. Prolaskom vozila kroz vrata, prekida se svjetlosna zraka koju emitira fotoelektrični senzor te se tako signalizira sustavu da se vrata sigurno mogu zatvoriti. Na vratima se još nalaze dvije krajnje sklopke; gornja koja signalizira da su vrata potpuno otvorena i donja koja signalizira da su vrata potpuno zatvorena.



Slika 5.1 – Shematski prikaz automatskog sustava upravljanja garažnim vratima

Prvi korak ka programiranju takvog sustava automatskog upravljanja garažnim vratima bio bi odrediti koji su nam ulazni signali u sustav (senzori), koji su nam izlazni signali iz sustava te im dodijeliti imena.

Ulazi u sustav:

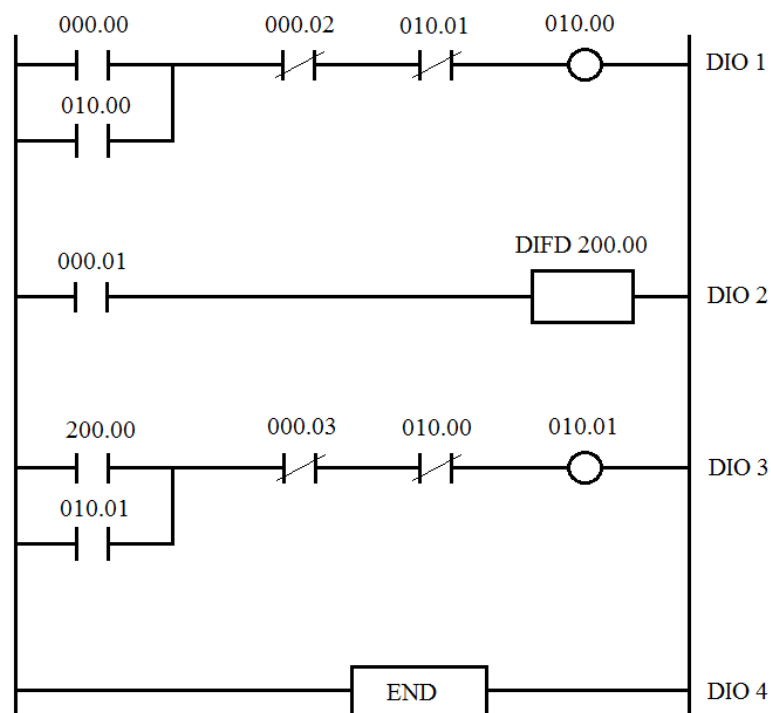
- Ultrazvučni senzor – 000.00
- Fotoelektrični senzor – 000.01
- Gornji graničnik – 000.02
- Donji graničnik – 000.03

Izlazi iz sustava:

- Motor za podizanje vrata – 010.00
- Motor za spužtanje vrata – 010.01

U zadatku ćemo još koristiti pomoćnu instrukciju DIFD (Differentiate Down) koja ima svojstvo da promjeni izlazno stanje iz 'OFF' u 'ON' za vrijeme trajanja jednog ciklusa, te se opet postavi u stanje 'OFF'.

Programski kôd zapisan ljestvičastim dijagramom za izneseni problem izgleda ovako:



Slika 5.2 - Programski kod - ljestvičasti dijagram

U prvom dijelu (DIO 1) ultrazvučni senzor (000.00) detektira vozilo čime se aktivira izlaz kojeg predstavlja motor za podizanje vrata (010.00). Da bi se vrata podignula, moraju se zadovoljiti još dva uvjeta:

1. Gornji graničnik (000.02) mora biti isključen što predstavlja da su vrata u krajnjem donjem položaju,
2. Motor za spuštanje vrata (010.01) mora biti isključen.

Kada su svi uvjeti zadovoljeni, vrata se počinju podizati.

Prolaskom vozila kroz vrata (DIO 2) prekida se zraka svjetlosti koju emitira fotoelektrični senzor (000.01). Postavlja se pomoćni signal (200.00).

U trećem dijelu (DIO 3) pomoćni signal (200.00) nam služi kao ulaz i signalizira da je vozilo ušlo u garažu te se može započeti proces spuštanja vrata (010.01). Prije toga, moraju biti ispunjena još dva uvjeta:

1. Donji graničnik (000.03) mora biti isključen što predstavlja da su vrata u krajnjem gornjem položaju,
2. Motor za podizanje vrata (010.00) mora biti isključen.

Kada su uvjeti zadovoljeni, vrata se spuštaju.

U zadnjem dijelu (DIO 4) kôd-a prikazan je kraj izvođenja ciklusa programa.

6. ZAKLJUČAK

Današnja industrija gotovo je nezamisliva bez automatizacije. PLC uređaji podigli su ljestvicu automatizacije na jedan potpuno novi nivo. Izbacili su stare relejne sustave koji su bili glomazni i neefikasni. Skratili su prazan hod u proizvodnim pogonima, koji je nastao kada bi se mehanički releji istrošili ili pokvarili. Smanjili su potrošnju struje koja je bila potrebna za upravljanje sustavom. Zbog brzine PLC uređaja, smanjilo se vrijeme ciklusa na strojevima što u konačnici rezultira većom produktivnošću.

U ovom radu navedene su sve prednosti uvođenja ovakvog sustava upravljanja u proizvodne pogone. Obradena je tematika hardverskog sklopovlja koji čine jedan PLC uređaj. Naveden je svaki dio pojedinačno te je uz opis rečeno i kakvu funkciju ima u cijelom sustavu. Objašnjen je princip rada PLC uređaja kroz četiri karakteristične faze. Navedeni su programski jezici koji se koriste u programiranju PLC uređaja te način kako se ti programi izrađuju te u konačnici i testiraju. Izneseno teorijsko znanje prenio sam na jedan praktičan primjer automatskog upravljanja garažnim vratima gdje sam, korak po korak, objasnio način izrade i implementacije programskog kôd-a za takav jedan jednostavan automatizirani sustav.

7. LITERATURA

1. Matić, N. (2001). *Uvod u industrijske PLC kontrolere*. mikroElektronika, Beograd.
2. Bryan L.A., Bryan, E.A. (1997). *Programmable Controllers*, An Industrial Text Company, Atlanta, SAD.
3. Bolton, W. (2006). *Programmable Logic Controllers*, Elsevier Newnes, Burlington, Velika Britanija.
4. Malčić, G., *Programibilni logički kontroleri – interna skripta*, Tehničko Veleučilište u Zagrebu, Zagreb.
5. Rajić, F. (2001)., *Automatizacija postrojenja*, Školska knjiga, Zagreb.

8. POPIS SLIKA

Slika 2.1 - Relejni sustav upravljanja	2
Slika 3.1 - Siemens PLC SIMATIC S7 - 300	5
Slika 3.2 - Shema PLC uređaja	5
Slika 3.3 - Shema CPU	6
Slika 3.4 - Mikroprocesor	7
Slika 3.5 - Uređaj za napajanje PLC-a	7
Slika 3.6 - Shematski prikaz komunikacijskog sučelja	8
Slika 3.7 - Modul za proširenje	9
Slika 3.8 - Ciklus rada PLC uređaja	11
Slika 4.1 - Siemens LOGO!	12
Slika 4.2 - Usporedba strujnog i logičkog puta	14
Slika 4.3 - Primjer ljestvičastog dijagrama	15
Slika 4.4 - Ljestvičasti dijagram logičke operacije 'I'	15
Slika 4.5 - Ljestvičasti prikaz logičke operacije 'IL1'	16
Slika 4.6 - Osnovni simboli korišteni u programiranju ljestvičastim dijagramom	16
Slika 4.7 - Hijerarhija programa pisanog instrukcijskim listama	17
Slika 4.8 - Proces izvođenja programa instrukcijskim listama	18
Slika 4.9 - Funkcijski blok	19
Slika 4.10 - Osnovne funkcije u funkcijsko blokovskom dijagramu	19
Slika 4.11 - Usporedni prikaz ljestvičastog dijagrama i funkcijsko blokovskog dijagrama	20
Slika 5.1 – Shematski prikaz automatskog sustava upravljanja garažnim vratima	21
Slika 5.2 - Programski kod - ljestvičasti dijagram	22