

# Razvoj aplikacija za programabilne logičke kontrolere

---

Čugura, Leon

Undergraduate thesis / Završni rad

2024

Degree Grantor / Ustanova koja je dodijelila akademski / stručni stupanj: **University of Zagreb, Faculty of Organization and Informatics / Sveučilište u Zagrebu, Fakultet organizacije i informatike**

Permanent link / Trajna poveznica: <https://um.nsk.hr/um:nbn:hr:211:558129>

Rights / Prava: [Attribution-NonCommercial-NoDerivs 3.0 Unported](#) / [Imenovanje-Nekomercijalno-Bez prerada 3.0](#)

Download date / Datum preuzimanja: **2025-03-01**



Repository / Repozitorij:

[Faculty of Organization and Informatics - Digital Repository](#)



SVEUČILIŠTE U ZAGREBU  
FAKULTET ORGANIZACIJE I INFORMATIKE  
VARAŽDIN

Leon Čugura

RAZVOJ APLIKACIJA ZA  
PROGRAMABILNE LOGIČKE  
KONTROLERE

ZAVRŠNI RAD

Varaždin, 2024.

**SVEUČILIŠTE U ZAGREBU**  
**FAKULTET ORGANIZACIJE I INFORMATIKE**  
**V A R A Ž D I N**

**Leon Čugura**

**Matični broj: 0016156318**

**Studij: Informacijski i poslovni sustavi**

**RAZVOJ APLIKACIJA ZA PROGRAMABILNE LOGIČKE**  
**KONTROLERE**

**ZAVRŠNI RAD**

**Mentor:**

Prof. dr. sc. Neven Vrček

**Varaždin, rujan 2024.**

*Leon Čugura*

### **Izjava o izvornosti**

Izjavljujem da je moj završni/diplomski rad izvorni rezultat mojeg rada te da se u izradi istoga nisam koristio drugim izvorima osim onima koji su u njemu navedeni. Za izradu rada su korištene etički prikladne i prihvatljive metode i tehnike rada.

*Autor potvrdio prihvaćanjem odredbi u sustavu FOI-radovi*

---

## Sažetak

Rad istražuje razvoj aplikacija za programabilne logičke kontrolere, ključne komponente u automatizaciji industrijskih procesa. U teorijskom dijelu rada su analizirani osnovni koncepti programiranja programabilnih logičkih jedinica, uključujući programski jezik, logičke operacije i tipične aplikacije. Također će biti razmotrene prednosti korištenja programabilnih logičkih kontrolera u industriji uz njihovu ključnu ulogu u procesu automatizacije.

Ovaj će rad pružiti temeljno razumijevanje programabilnih logičkih kontrolera, koje je ključno za inženjere koji rade u industriji automatizacije.

**Ključne riječi:** programabilni logički kontroleri, PLC, automatizacija, releji

# Sadržaj

Sadržaj .....	iii
1. Uvod .....	1
2. Metode i tehnike rada .....	2
3. Programabilni logički kontroleri .....	3
3.1. Osnovni koncepti .....	4
3.1.1. Svrha PLC u usporedbi s relejnim sustavima .....	4
3.1.2. Struktura PLC .....	5
3.1.3. Adresiranje memorije PLC računala .....	11
3.1.4. Princip rada PLC .....	13
3.2. Programiranje programabilnih logičkih kontrolera .....	14
3.2.1. Simuliranje realnih scenarija .....	15
3.2.2. Ljestvičasti dijagrami .....	17
3.2.3. Instrukcijske liste .....	21
3.2.4. Funkcijski blokovski dijagrami .....	24
4. Zaključak .....	26
Popis literature .....	27
Popis slika .....	29
Popis tablica .....	30

# 1. Uvod

Svi su fizički uređaji i pomagala proizvedeni industrijskim procesima koji su u manjoj ili većoj mjeri automatizirani. Automatizacija pojedinih procesa ubrzava proizvodnju proizvoda, od njegove početne konceptualizacije, pa sve do finalizacije i prodaje istog. Programabilni logički kontroleri često su opisani kao „radni konji industrijske automatizacije“, samim time može se zaključiti da imaju ključnu ulogu u industrijskom procesu.

Programabilni logički kontroleri (PLC) predstavljaju specijalizirano računalo koje se koristi u industrijskoj automatizaciji. PLC je drukčija vrsta računala, nema zaslon ni tipkovnicu, kao ni miš, printer ili tvrdi disk. Iako ove, klasične komponente bez kojih krajnji korisnik ne bi mogao zamisliti svoj rad na računalu nedostaju, ovo je i dalje vrlo važno računalo koje se krije u upravljačkoj ploči, te ima svoju svrhu. PLC podiže razinu efikasnosti kod rada strojeva na puno višu razinu, pružajući brojne mogućnosti koje ubrzavaju proces proizvodnje proizvoda. U današnjem svijetu vrlo je teško ispuniti sve želje i potrebe korisnika, iz tog je razloga važno da svaki od procesa bude u najvišoj mjeri efektivan i efikasan, što uključuje automatizaciju pojedinih procesa. Zbog toga, PLC je relevantan i ključan dio industrijskog procesa proizvodnje. Upravljanje strojevima pomoću PLC-a uistinu poboljšava proces proizvodnje.

Ovaj rad istražuje i analizira programabilne logičke kontrolere, osnovne koncepte i svrhu istih, način programiranja, područje primjene i ostale relevantne informacije o PLC. U završetku će rad pružiti smislen zaključak o najčešćoj primjeni PLC-a, ključne komponente u procesu automatizacije.

## **2. Metode i tehnike rada**

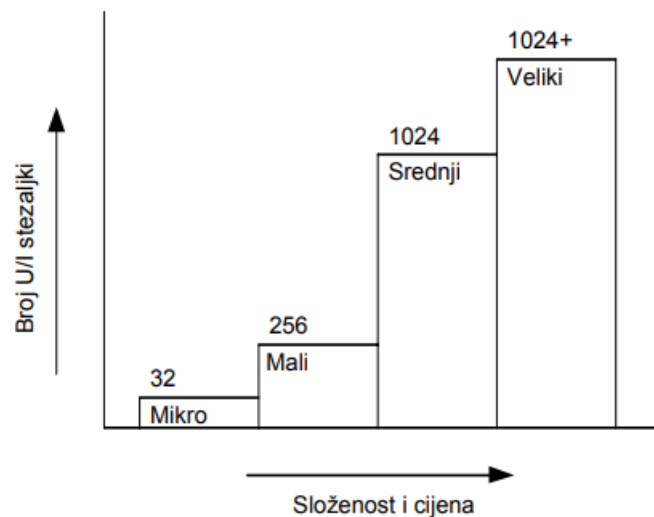
U vidu izrade ovog rada korištena je povezana literatura, istraživani su i razni članci navedeni u literaturi samog rada. Osim toga, korišteni su razni drugi znanstveni radovi i pouzdani izvori s interneta.



### 3. Programabilni logički kontroleri

Programabilni logički kontroleri koji su postali neizostavan dio procesa proizvodnje u industriji tema su ovog rada, točnije, razvoj aplikacija za iste. Međutim, kako bismo mogli programirati PLC, potrebno je shvatiti što je PLC. Naime, bilo koji sustav upravljanja može imati korist od PLC-a, kad je potrebno upravljanje u stvarnom vremenu ovi su upravljači (kontroleri) kvalitetan odabir.

Prema Malčiću (2015) programabilne logičke kontrolere možemo ugrubo podijeliti u 4 kategorije prema broju ulazno/izlaznih stezaljki, odnosno na mikro, mali, srednji i veliki programabilni logički kontroleri, što je vidljivo iz dolje priložene fotografije:



Slika 1. Kategorije PLC računala (Malčić, 8. kolovoza 2024.)

Iz slike 1 vidljivo je da se programabilni logički kontroleri koji imaju do 32 U/I stezaljki svrstavaju u mikro kategoriju, do 256 U/I stezaljki pokriva kategorija malih PLC računala, do 1024 je srednja kategorija, a sve više od 1024 U/I stezaljki pokriva kategorija velikih PLC računala.

Guedes (2010) navodi da je PLC dizajniran za specifične potrebe koje se javljaju kod procesa upravljanja u stvarnom vremenu, jer istovremeno funkcioniraju razni sustavi koji su u koheziji, a kako bi se izbjegli konflikti sustava i greške u proizvodnji i upravljanju, PLC ima razvijene podsustave za minimizaciju mogućnosti da se dogodi greška. Također, često dolazi potreba za promjenom, bila ona veća ili manja, te kako bi ta promjena prošla što bezbolnije potrebno je da upravljački sustav ima visoki stupanj agilnosti, a PLC ispunjava taj uvjet.

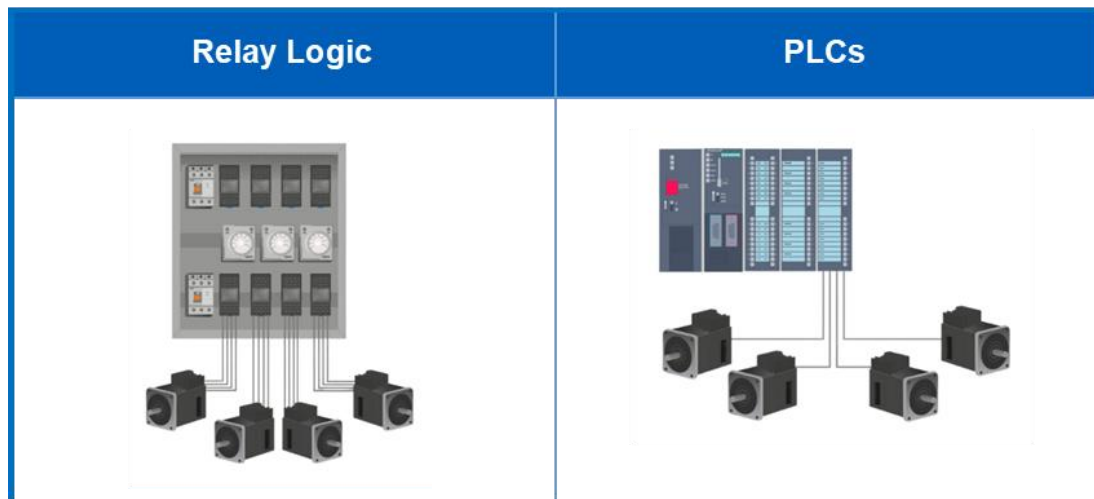
## 3.1. Osnovni koncepti

### 3.1.1. Svrha PLC u usporedbi s relejnim sustavima

Kao što je već rečeno, današnja industrija u kojoj se konstantno provodi neki oblik proizvodnje, gotovo je nemoguće istu zamisliti bez automatizacije u manjoj ili većoj mjeri.

„Pojam automatizacija označava upravljanje strojevima, procesom ili sustavom pomoću mehaničkih i elektroničkih uređaja koji zamjenjuju ljudski rad, odnosno nadzor i donošenje odluka u poslovima koji su za čovjeka previše složeni, opasni ili zamarajući. U industrijalizaciji, automatizacija je nastavak mehanizacije. Dok mehanizacija rada omogućava ljudima u pogonu lakše uvjete rada, automatizacija smanjuje potrebu za ljudskom prisutnošću u obavljanju određenih djelatnosti.“ („Fakultet elektrotehnike i računarstva [FER]“, 6. 4. 2021.). Pojam automatizacije ključan je kod uloge PLC računala u industriji, visok stupanj agilnosti podrazumijeva činjenicu da kod potrebe za promjenom, umjesto mukotrpne promjene ožičenja, PLC je samo potrebno reprogramirati. Osim toga, PLC je fleksibilnije, bolje i transparentnije rješenje od relejnih krugova.

Razumijevanjem pojma automatizacije, jasnija je i sama svrha programabilnih logičkih kontrolera koji su zamijenili relejne krugove. PLC računala su fizički imala manji opseg od relejnih krugova jer su pojedine komponente poput brojača bile obuhvaćene softverski.



Slika 2. Usporedba PLC računala s relejima (Electrical & Automation Solutions, 10. kolovoza 2024.)

Prema Malčiću (2015) krajem 60-tih godina gotovo su svi proizvodni pogoni bili upravljani pomoću relejnih krugova. Svakom promjenom proizvodnog programa nastala je potreba za prilagodbom upravljačkih sklopova, što je stvaralo dodatne troškove, a i duži period

„praznog hoda“ u proizvodnom pogonu. Ti su problemi nastajali iz razloga što kod svake vrste promjene, bila ona velika ili mala, nastala je potreba za promjenom ožičenja, a ponekad i sastavljanjem novog sklopa. U tom je periodu počela proizvodnja mikroprocesora, samim time i razvoj programabilnih logičkih kontrolera. Oni su se odmah pokazali kao isplativije rješenje od relejnih krugova, iz više razloga:

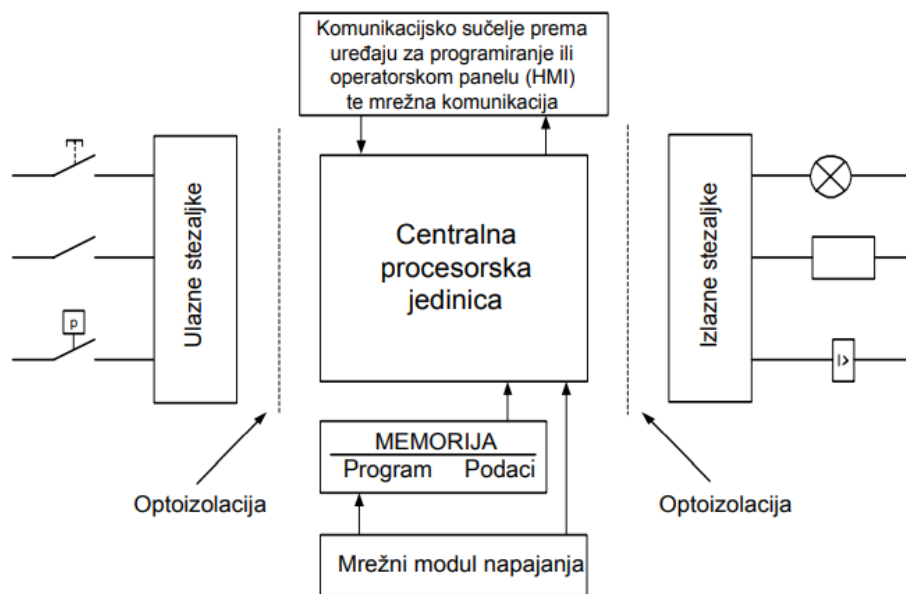
- **Pouzdanost** – ne postoje mehanički pokretni dijelovi, PLC otporan na razne elektromagnetske utjecaje, u slučaju nestanka napajanja, pri povratku istog PLC nastavlja s radom
- **Prilagodljivost** – za neku promjenu potrebno je vrlo malo vremena, pri raspisanom i testiranom PLC programu za upravljanje nekog uređaja se bez problema može prenijeti na drugi PLC u drugom uređaju
- **Komunikacija** – olakšano prikupljanje podataka od drugih PLC uređaja, kao i komunikacija s drugim upravljačkim panelima i obrada
- **Brzina** – kod automatiziranih strojeva brz odaziv na signale od strane aplikacija je od presudne važnosti, upravo to omogućuje PLC

Projektant upravljačkih sustava konstantno postavlja pitanje kada je u manjim sustavima bolje upotrijebiti PLC umjesto relejnog sklopa. Prema Malčiću (2015) je nekad to pitanje bilo opravdano visokom cijenom PLC-a, no u današnjici s obzirom na masovnu proizvodnju i nisku cijenu uporaba mikro PLC-ova (kategorija koja pokriva PLC računala do 32 U/I stezaljki) je svakako opravdana. Dovoljno je da u sustav upravljanja treba ugraditi najjednostavnije upravljačke i vremenske releje pa da se sa PLC-om dobije jeftiniji i fleksibilniji upravljački sustav, koji je prilagodljiviji od releja.

### 3.1.2. Struktura PLC

Jasno je da su programabilni logički kontroleri sastavni dio proizvodne industrije zbog svoje prilagodljivosti i široke lepeze mogućnosti u sferi proizvodnje i upravljanja strojevima. Struktura PLC, neovisno o njegovoj veličini, može biti podijeljena na osnovne dijelove (Paessler, 2022):

- **Centralnu procesorsku jedinicu**
- **Ulazni dio**
- **Izlazni dio**
- **Memorijski blok za program i podatke**
- **Napajanje i komunikacijsko sučelje**
- **Moduli za proširenje**



Slika 3. Struktura PLC računala (Malčić, 8. kolovoza 2024.)

### 3.1.2.1. Centralna procesorska jedinica

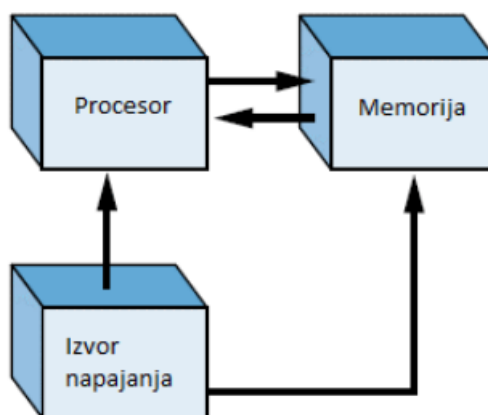
Centralna procesorska jedinica (CPU) predstavlja „mozak“ računala, a isto vrijedi i kod PLC računala. Sastoji se od 3 temeljne komponente:

- **Procesor**
- **Memorija**
- **Izvor napajanja**

Sve navedene komponente su u konstantnoj interakciji, procesorska jedinica konstantno čita stanja svih ulaza PLC-a, te u interakciji s programom razvijenim od strane programera ili arhitekta softvera logički obrađuje ulaze te dolazi do određenih rezultata. Konačno, u skladu s dobivenim rezultatima upravlja izlazima.

U globalu, mikroprocesor ima jedinstvenu ulogu izvođenja programa koji se nalazi u memoriji, odnosno, koji je pohranjen u memoriji PLC računala. Potom, izvor napajanja omogućuje i osigurava nesmetan i kontinuiran rad programabilnih logičkih kontrolera pružajući sav potreban napon. Bez izvora napajanja ni procesor, a ni memorija, nemaju mogućnost nesmetanog rada, samim time, bez napajanja rad PLC računala nije moguć. No, nestankom napajanja, ne nastaje šteta nad PLC računalom, normalno se vraća radu povratkom napajanja, zahvaljujući memoriji i mikroprocesoru unutar same centralne procesorske jedinice.

Sama arhitektura i dizajn centralne procesorske jedinice PLC računala je varijabilan, varira od proizvođača do proizvođača, no svaki se svodi na gore navedene 3 ključne komponente.

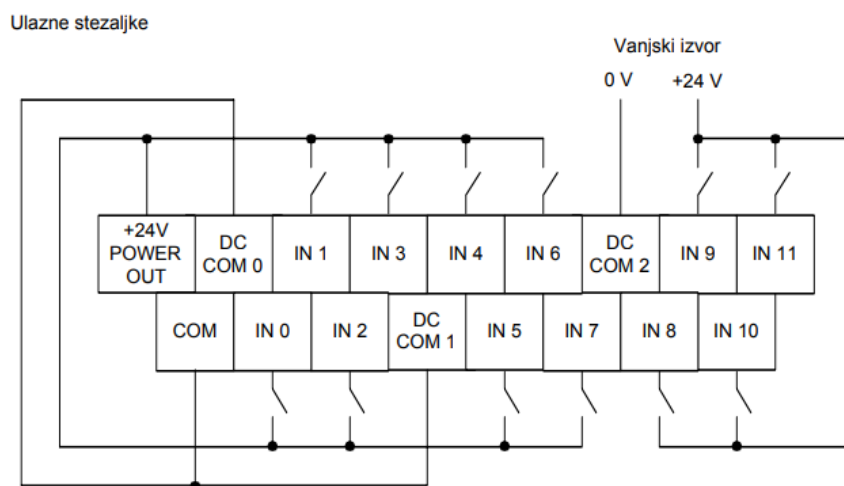


Slika 4. Struktura centralne procesorske jedinice PLC-a (Šepac, 8. kolovoza 2024.)

### 3.1.2.2. Ulazni dio

Prema Malčiću (2015) ulazni dio PLC-a predstavljaju priključne vijčane stezaljke na koje se spajaju dojavni signali iz procesa čijim se radom upravlja, te su mjesto od kojeg počinje prilagodba vanjskog signala iz radne okoline, signalu kojeg razumije procesorska jedinica PLC-a. Informacije koje PLC prima na svojim ulaznim stezaljkama mogu biti digitalne ili analogne.

„PLC prima ili očitava podatke s ulaznih uređaja kao što su senzori blizine i fotoelektrični senzori, tipkovnice, mjerači razine, mjerači vremena, brojači, svjetla na konzoli, električni motori te sklopke za temperaturu i tlak. Koncept senzorskih podataka odnosi se na prirodu PLC ulaznih podataka koji dolaze u obliku elektroničkih signala.“ („Paessler the monitoring experts“, 21. 4. 2022.).

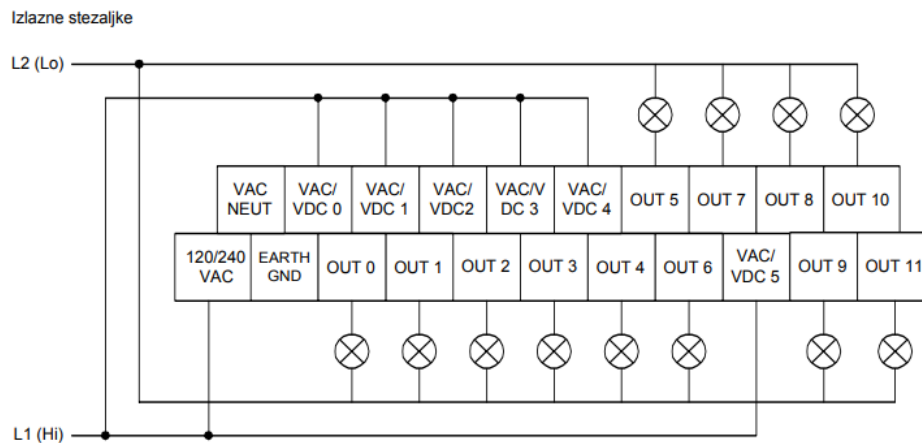


Slika 5. Primjer ožičenja ulaznih stezaljki PLC-a sa 12 digitalnih ulaza (Malčić, 8. kolovoza 2024.)

### 3.1.2.3. Izlazni dio

Prema Malčiću (2015) izlazni dio PLC-a su priključne vijčane stezaljke na koje se spajaju izvršni uređaji iz procesa kojima PLC šalje digitalne i analogne signale te na taj način upravlja procesom. Na digitalne izlaze iz PLC-a su najčešće spojeni magnetni svici, releji, sklopnici, signalne lampe, i sl., dok na analogne izlaze mogu biti spojeni npr. strujni signal za prikaz neke veličine na pokaznom instrumentu, referenca brzine za frekvencijski pretvarač itd. Izlazne stezaljke također su optoizolirane od procesorske jedinice radi galvanske izolacije električnih krugova.

Drugim riječima, kartice s digitalnim ulazom obrađuju diskretne signale, na primjer signale za uključivanje i isključivanje. Ovo se može odnositi na jednostavne radnje poput uključivanja ili isključivanja svjetla. Analogne ulazne kartice pretvaraju napon u brojeve koje CPU može razumjeti, primjerice za pogon nekih strojeva.



Slika 6. Primjer ožičenja izlaznih stezaljki PLC-a sa 12 digitalnih izlaza (Malčić, 8. kolovoza 2024.)

### 3.1.2.4. Memorijski blok za program i podatke

Memorija sustava mjesto je gdje su sve informacije, uz instrukcije i korisničke programe pohranjeni. Važno je istaknuti da je memorija podijeljena na dva različita dijela, a to su:

- **Izvršna memorija (Program Memory)**
- **Korisnička memorija (Variable Memory)**

Dok s jedne strane izvršna memorija sadržava stvari koje su trajno pohranjene, poput korisničkih programa koji su sastavni dio programabilnih logičkih kontrolera, korisnička memorija sadrži korisničke programe. Korisnik korisničkoj memoriji može pristupiti i manipulirati sadržajem, no situacija nije ista s izvršnom memorijom. Izvršna memorija sadrži nadzorne programe i komunikaciju s ranije spomenutim ulazno – izlaznim uređajima.

Postoje razne vrste memorije na koje se podaci i informacije pohranjuju:

- **Memorija za ulaze i izlaze (I/O Memory)**
- **Memorija za statusne bitove (Status Memory)**
- **Memorija za tajmere i brojače (Timer and Counter Memory)**
- **Kontrolna memorija korisničkih funkcija (User Function Control Memory)**

Memorija za ulaze i izlaze se koristi za pohranu podataka ulaza i izlaza unutar samog PLC sustava. Memorija za statusne bitove se koristi za provjeru rada PLC-a i omogućuje razne promjene u sustavu. Memorija za tajmere i brojače, kako i samo ime predlaže, koristi se za pohranu podataka vezanih za tajmere i brojače u sustavu, ovi primjeri uključuju „EN“, što znači Enable, „TT“ za Timer Timing itd... Na kraju, postoji i kontrolna memorija korisničkih funkcija koja predstavlja općenitu memoriju za složenije funkcije koje se ne mogu dovršiti u jednom ciklusu izvršavanja. („Info PLC“, 9. 8. 2024.).



Slika 7. Memorijska kartica PLC računala (PLC Market, 10. kolovoza 2024.)

### 3.1.2.5. Napajanje i komunikacijsko sučelje

Izvor napajanja je ključan i neizostavan dio PLC računala, zahvaljujući napajanju PLC računalo se može pokrenuti i obavljati svoje zadatke.

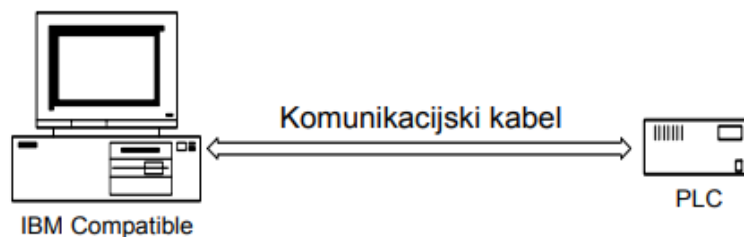
Prema Šepacu (2016) njegova uloga je puno šira od osiguranja potrebnog napona za pojedine komponente sustava. Druga mu je uloga nadgledanje i regulacija napona koji se dovodi uz upozorenje centralne procesorske jedinice u slučaju da nešto nije u redu. Odnosno, većina PLC uređaja radi na izmjenični izvor napajanja (AC), ali neki uređaji mogu raditi i na istosmjerni izvor napajanja (DC). Potreban napon napajanja je 220 VAC ili 24 VDC. Kako u

stvarnosti dolazi do odstupanja u vrijednostima napona, modul izvora napajanja mora biti u mogućnosti tolerirati 10% više ili manje od nominalnog iznosa.



Slika 8. Uređaj za napajanje PLC računala (Šepac, 8. kolovoza 2016.)

Prema vlastitoj intuiciji se da zaključiti da komunikacijsko sučelje ima ključnu ulogu u procesu komunikacije. Naime, komunikacijsko sučelje ima višestruku namjenu, osim osnovne i ključne komunikacije s nadređenim računalom, ima razne druge mogućnosti. Primjerice, komunikacijsko sučelje pruža mogućnost komunikacije s drugim sensorima, kao i ostalim PLC uređajima.



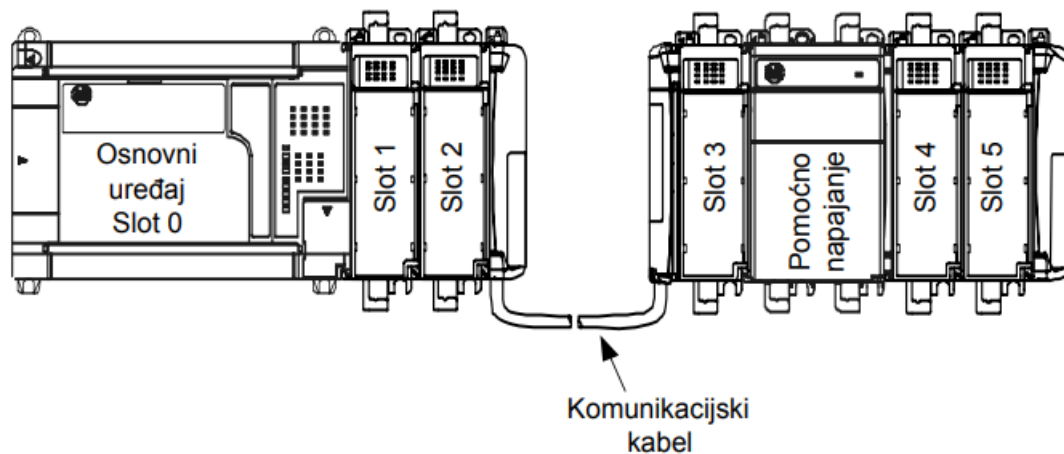
Slika 9. Komunikacija s nadređenim PC računalom (Malčić, 8. kolovoza 2024.)

### 3.1.2.6. Moduli za proširenje

PLC uređaj je u pravilu uređaj od jednog dijela, samim time ima ograničen broj dostupnih ulaza i izlaza, odnosno ulaznih i izlaznih stezaljki. Prema Malčiću (2015) kada je za



proces potrebno više ulaza ili izlaza od broja dostupnih, tada se koriste moduli za proširenje, odnosno slotovi. Modul za proširenje predstavlja poseban uređaj koji se spaja na programabilne logičke kontrolere te na sebi ima dodatne ulazne / izlazne stezaljke, samim time PLC uređaj može proširiti svoj kapacitet, bez potrebe za kupnjom novog PLC uređaja.



Slika 10. Osnovni PLC uređaj uz proširenje (Malčić, 8. kolovoza 2024.)

Moduli se napajaju el. energijom iz osnovnog uređaja, ali postoji i mogućnost da koriste posebna napajanja. Naravno, preporuka u praksi je da osnovni uređaj i moduli za proširenje koriste isti izvor napajanja. Prema Malčiću (2015) u pogonu moduli mogu biti fizički udaljeni od osnovnog uređaja, a veza se ostvaruje komunikacijskim kabelom, kao što je prikazano na slici.

### 3.1.3. Adresiranje memorije PLC računala

Kada postoji memorija, odmah se javlja pojam adresiranja. Svi su podaci u PLC memoriji podijeljeni u tri osnovne grupe zapisnika koji se naknadno dijele na podgrupe. Prema Malčiću (2015) postoje 3 najčešće grupe:

- **Podatkovni zapisnici (Data Files)**
- **Funkcijski zapisnici (Function Files)**
- **Programski zapisnici (Program Files)**

Naravno, svaka od ovih grupa ima razne datoteke koje su objašnjene u nastavku.

#### 3.1.3.1. Podatkovni zapisnici

Output file pohranjuje očitane vrijednosti za vrijeme pretraživanja, prikazuje stanje izlaza. Input file, kako i samo ime predlaže, pohranjuje vrijednosti koje su učitane također za vrijeme samog pretraživanja, ali u ovom slučaju stanje ulaza. Bit file pohranjuje vrijednosti

veličine bita, timer file pohranjuje vrijednosti timera. Counter file pohranjuje vrijednosti brojača, integer file pohranjuje podatak koji može biti do 16 bita veličine. Long word file pohranjuje podatak do 32 bita veličine, isto kao i float file, ali ovaj tip pohranjuje samo realne brojeve. Status file pohranjuje podatke o radu sustava, što je važno zbog dijagnostike sustava, također javlja i greške ukoliko do istih dođe. Na kraju, postoji message file koji pohranjuje poruke, kao i stanje poruka između više PLC uređaja.

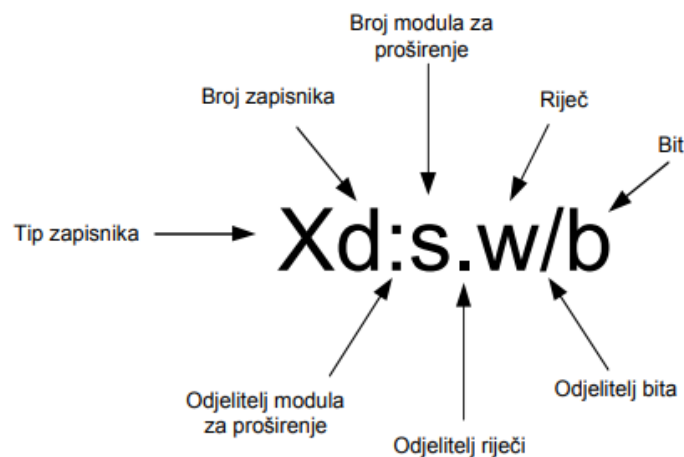
### 3.1.3.2. Funkcijski zapisnici

Base hardware information file uključuje podatke o PLC-u, odnosno tehničke pojedinosti. I/O status file prema intuitivnom imenu pruža informaciju da se radi o informacijama vezanima uz module za proširenje. Communications status file sadrži informacije o parametrima i aktivnosti prilikom komunikacije. Real time clock pruža informacije o vremenu. Trim pot information sadrži informacije o trimmer potencijometrima. Na kraju, memory module ima informacije o memorijskim karticama, ali onima koje su naknadno ugrađene u sustav.

### 3.1.3.3. Programski zapisnici

Program files datoteke sadrže programski kod koji je napisan na jedan od brojnih mogućih načina. Na kraju, system file posjeduje sve systemske zapisnike.

Memorijski registri u sferi programiranja PLC-a su veličine bita i riječi, drugim riječima 16 i 32 – bitne riječi. Kako bi se mogli upustiti u programiranje oznake ulaza, izlaza i ostalih relevantnih komponenti moraju biti jasno definirani, to uključuje način označavanja tih važnih komponenti. Naravno, kao i sa većinom ostalih proizvoda svaki proizvođač na drukčiji način određuje oznake tih komponenti, ali u globalu su prema Malčiću (2015) modificirani na način prikazan na slici.



Slika 11. Sustav adresiranja PLC-a (Malčić, 10. kolovoza 2024.)

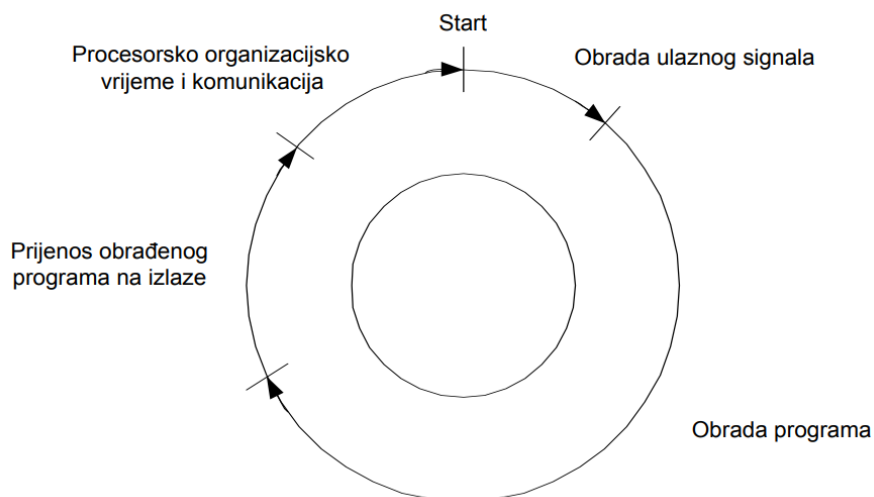
Adresa	Tip zapisnika	Broj zapisnika	Broj modula	Riječ	Bit
I:2/7	Input	-	2	0	7
O:5/10	Output	-	5	0	10
O:3.2/4	Output	2	3	1	4
I:4.1/8	Input	1	4	2	8
I:2.3/15	Input	3	2	1	15
B6:1/11	Bit	1	6	0	11
N10:2	Integer	10	-	2	-

Tablica 1. Primjeri sustava adresiranja (autorski rad)

### 3.1.4. Princip rada PLC

Baš kao i brojni drugi procesi u informatici, princip rada programabilnih logičkih kontrolera je zapravo kontinuirani, beskonačan proces koji se provodi u više koraka, odnosno iteracija.

Zbog principa rada programabilnih logičkih kontrolera, odnosno zbog efektivnosti i efikasnosti rada PLC računala upravo su ona poslužila kao zamjena za relejne krugove koji imaju mnogo mehaničkih dijelova koji uključuju brojne žice, sklopove itd... Kao što je ranije objašnjeno, pojedine komponente su u potpunosti izostavljene iz temeljne strukture PLC računala, te su softverski implementirane u programsko rješenje PLC računala. Iz tih, a i brojnih drugih razloga, PLC računala pokazala su se kao efikasnije rješenje od relejnih krugova. Princip rada PLC računala najjednostavnije je shvatiti kroz vizualizaciju rada istog, kao što je prikazano na slici.



Slika 12. Princip rada PLC računala u koracima (Malčić, 10. kolovoza 2024.)

Iz gore priložene slike jasno se vide koraci, te kojim redoslijedom idu. Kao što je ranije objašnjeno, princip rada PLC računala kontinuiran je proces, drugim riječima, beskonačan

petlja. Programabilni logički kontroleri prema promjeni stanja na ulazima mora kontinuirano i inkrementalno procesuirati informacije s dobivenih ulaza, te prema njima slati signale za izlaze. To znači da se za svaki ulaz u centralnoj procesorskoj jedinici obavljaju kalkulacije prema kojima PLC računalo vraća informacije kroz izlaze. Logika, odnosno koraci koji slijede nakon dobivenih i obrađenih ulaza nalazi se unutar korisničkog programa. Prema Malčiću (2015) osnovni ciklus obrade podataka podijeljen je u više iteracija:

- **Obrada ulaznog stanja**
- **Obrada programa**
- **Prijenos obrađenog programa na izlaze**
- **Procesorsko organizacijsko vrijeme i komunikacija**

Obrada ulaznog stanja uključuje provjeru svih ulaza, od senzora pa sve do drugih komponenti, te potom prikupljene podatke i informacije pohranjuje unutar ulaznog registra centralne procesorske jedinice. Nakon toga slijedi obrada programa, gdje dolazi do programske obrade prikupljenih artefakata iz prve faze, dakle obrađuju se ulazna stanja prema logici korisničkog programa razvijenog od strane programera. Nakon toga se rezultat obrade pohranjuje u izlazni memorijski registar centralne procesorske jedinice. Zatim slijedi treći korak, odnosno prijenos obrađenog programa na izlaze. Kao što samo ime koraka predlaže, pohranjeni se podaci iz izlaznog registra, oni obrađeni u koraku obrade, prosljeđuju na sve izlaze PLC računala. Na kraju se, u sklopu posljednjeg koraka odvijaju potrebne operacije za pravilan rad operativnog sustava, kao i nužna komunikacija sa svim vanjskim jedinicama. Vrijeme jednog ciklusa za otprilike 500 naredbi traje 1,5 ms.

## **3.2. Programiranje programabilnih logičkih kontrolera**

Programabilni logički kontroleri ključan su dio procesa automatizacije u modernoj industriji, no kako bi imali svoju zadaću, potrebno je iste isprogramirati. Kao i ostala osobna računala, PLC računala izvode niz naredbi u sklopu programa, te u skladu s programom upravljaju tekućim procesom. U ovom slučaju, upravlja se ulazima i izlazima. U praksi se najčešće PLC računala programiraju na način da se spoje s nadređenim PC računalom, te se na taj način uspostavlja komunikacija s PLC računalom.

U pravilu, pri kupnji PLC računala proizvođač pruža i softver koji predstavlja jedinstvenu kombinaciju programskog urednika, kompajlera i komunikacijskog softvera. U sklopu uređivača napiše se programski kod u jednom od odabranih jezika, a zatim slijedi provjera sintakse. Ukoliko je sve u redu sa napisanim programskim kodom, tada softver šalje program u RAM memoriju PLC-a. Što se komunikacije između nadređenog PC računala i PLC-a tiče, ona je serijska te se samim time može pratiti i tijekom izvođenja programa. Samim time uvijek

imamo mogućnost kontroliranja ulaza i upravljanja izlazima. Osim toga, naredbe se direktno mogu pisati korištenjem miša i tipkovnice.

Prema Malčiću (2015) postoje i ručni programatori koji imaju mali LCD zaslon uz tipkovnicu, takvi se uređaji direktno spajaju na PLC. Ovo je dobra alternativa u slučaju da je potrebno napraviti neke manje izmjene u programu, kada se izmjene moraju obaviti u pogonu. Drugim riječima, za neke jednostavnije izmjene koje je potrebno učiniti u procesu postoje PLC uređaji koji na sebi imaju nekoliko funkcijskih tipki i zaslon pa je moguće programirati na licu mjesta. Primjer takvog uređaja prikazan na slici.



Slika 13. Ručni PLC programator (Automation Direct, 11. kolovoza 2024.)

### 3.2.1. Simuliranje realnih scenarija

Prije puštanja PLC uređaja u industrijski pogon gdje će upravljati strojevima, potrebno je ispitati ispravnost napisanog programa. Dakle, kako bi se osigurala činjenica da će se strojevi spojeni na PLC ponašati kako je predviđeno, odnosno, da će se PLC kontroleri ponašati kako je zamišljeno, potrebno je provesti tzv. simulacije. Simulacijama ispitujemo i testiramo napisani programski kod.

Nadalje, simulator stanja omogućuje korisniku simulaciju ulaza i izlaza. Uređaj prikazan na sljedećoj slici je I/O sučelje koje, kao što je rečeno, pruža mogućnost simulacije ulaza i izlaza na PLC-u bez potrebe za povezivanje sa realnim i stvarnim procesima. Upravo se iz tog razloga koristi za testiranje u kontroliranim uvjetima prije spajanja PLC-a na upravljačku ploču.

U nastavku se nalazi slika PLC simulatora stanja, zajedno sa standardnim dijelovima koji imaju svoju svrhu i zadaću.



*Slika 14. PLC simulator stanja (Amatrol, 12. kolovoza 2024.)*

Na slici je vidljivo da uređaj posjeduje razne LED lampice, tipke, prekidače i konektore koji omogućuju povezanost s PLC uređajem. Sve su te stavke standardni dijelovi za PLC simulator. U praksi LED lampice označavaju izlaze, a tipke i prekidači povezani su s ulazima. U konačnici, takvi uređaji, odnosno simulatori omogućuju efikasan način simuliranja realnog rada PLC-a u raznim scenarijima, ali u kontroliranim uvjetima.

Za programiranje PLC uređaja postoji nekoliko različitih programskih jezika. Najrasprostranjeniji među njima je programiranje pomoću ljestvičastih dijagrama. Ovaj pristup programiranju vuče korijene iz prvih dana razvoja i osnivanja PLC uređaja i zasniva se na modifikaciji relajnih upravljačkih shema. Osim ljestvičastih dijagrama, često se koriste i instrukcijske liste koje omogućuju programiranje na razini assemblera (tekstualni programski jezik), kao i funkcijski blokovski dijagrami. Funkcijski blokovski dijagrami predstavljaju grafički način programiranja koji koristi logičke blokove iz Booleove algebre za prikazivanje logičkih operacija.

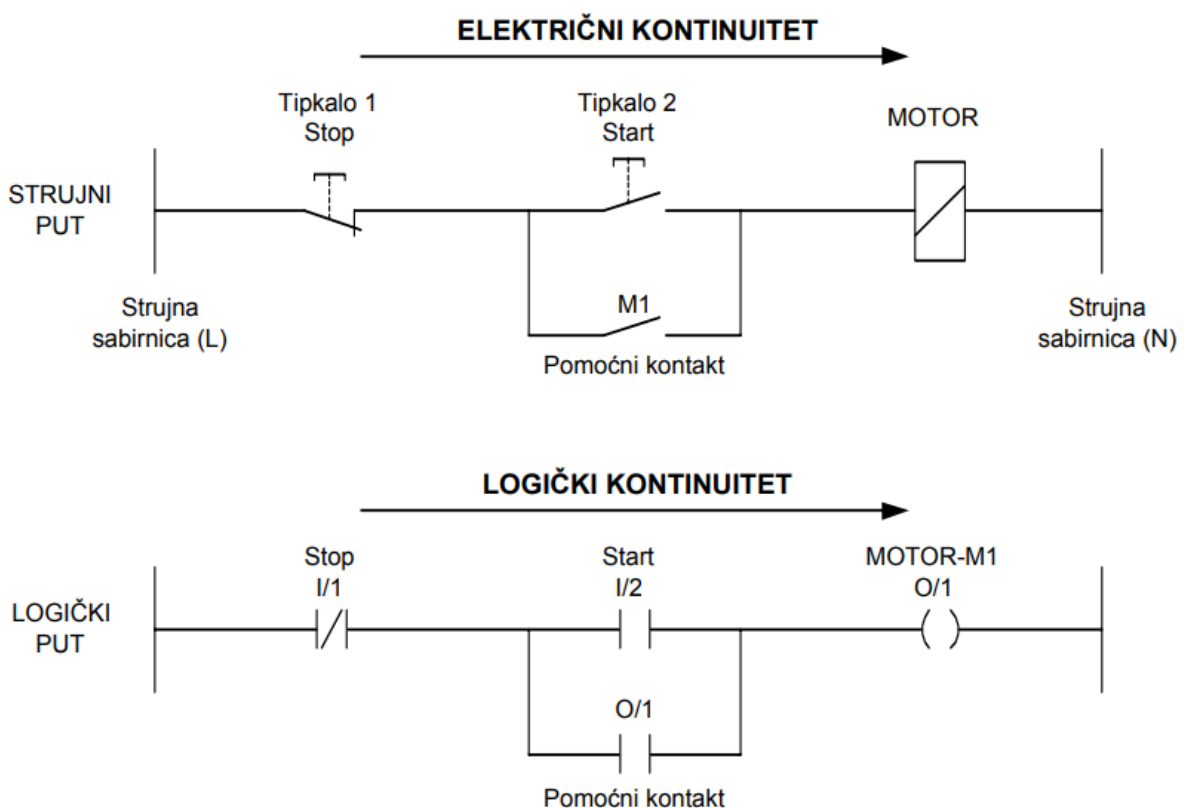
Kao što je objašnjeno, postoje brojni programski jezici pomoću kojih je moguće programirati PLC uređaje u skladu sa željama i potrebama korisnika. U vidu pružanja dubljeg

razumijevanja programiranja programabilnih logičkih kontrolera, u nastavku je svaki programski jezik, uz osnovne koncepte i primjere objašnjen.

### 3.2.2. Ljestvičasti dijagrami

Ljestvičasti dijagrami (engl. *ladder diagram*) nastali su u počecima razvoja PLC tehnologije i to na temelju strujnih upravljačkih shema. Zapravo, jedan od ključnih razloga za popularnost i široku primjenu PLC uređaja je mogućnost programiranja putem ljestvičastih dijagrama. Ova metoda programiranja odigrala je značajnu ulogu u uspjehu PLC tehnologije, jer se temelji na principima sličnim onima koji su korišteni u relejnoj ljestvičastoj logici, koja je prethodno dominirala u upravljanju proizvodnim pogonima. Upravo ta sličnost u načinu rada omogućila je lakšu i bržu tranziciju s tradicionalnih relejnih sustava na moderne sustave upravljanja pomoću PLC uređaja. Ovo olakšano prilagođavanje inženjera i tehničara na novu tehnologiju omogućilo je bržu implementaciju i širenje PLC sustava u industriji, čineći ih standardom u automatizaciji.

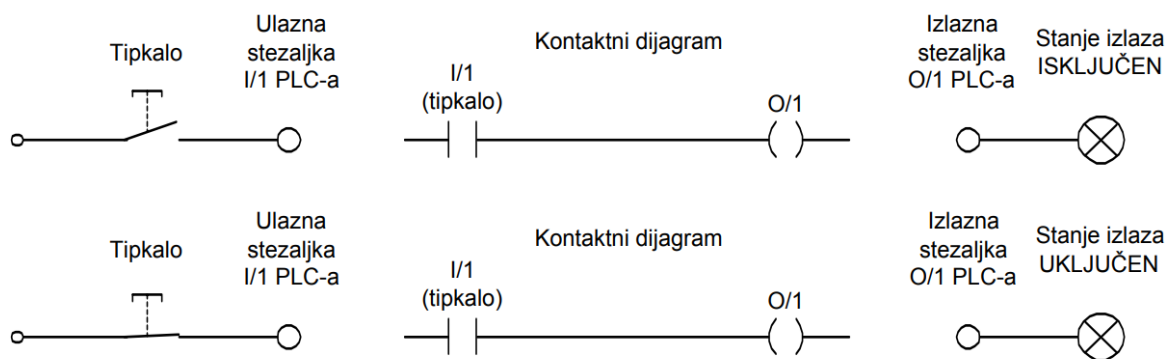
Analizom i usporedbom strujne sheme sa ljestvičastim dijagramima, postoje mnoge sličnosti, ali i razlike. U nastavku je uz vizualan prikaz objašnjeno upravo to.



Slika 15. Usporedba električnog i logičkog puta (Malčić, 12. kolovoza 2024.)

Prema Malčiću, u strujnim shemama simboli predstavljaju stvarne komponente (kontakte) i njihove fizičke veze, dok u ljestvičastim dijagramima slični simboli predstavljaju naredbe unutar programa. Ljestvičasti dijagram je sastavni dio upravljačkog softvera PLC-a, dok strujna shema prikazuje stvarni tok električne struje unutar kruga. Jedna od ključnih razlika između ljestvičastog dijagrama i strujne sheme leži u prikazu stanja: strujna shema pokazuje jesu li kontakti fizički otvoreni ili zatvoreni, dok se u ljestvičastom dijagramu provjerava istinitost određene naredbe, pri čemu '1' znači istinito, a '0' neistinito, bez nužne povezanosti s fizičkim stanjem kontakata spojenih na ulazne stezaljke PLC-a.

Prema Malčiću (2015) svaki logički put u ljestvičastom dijagramu mora sadržavati barem jednu izlaznu naredbu, a obično uključuje jedan ili više uvjeta koji moraju biti ispunjeni kako bi se ta naredba izvršila. Ovi uvjeti, odnosno uvjetne naredbe, najčešće predstavljaju signale dobivene od uređaja spojenih na ulaze PLC-a, zajedno sa statusima izlaza, pomoćnim memorijskim varijablama, te članovima kao što su vremenski i brojački elementi. Na kraju svakog logičkog puta nalazi se izlazna naredba koja se aktivira ili deaktivira ovisno o ispunjenju postavljenih uvjeta. Primjeri izlaznih naredbi uključuju 'uključiti izlaz', što aktivira izlazni relej PLC-a, interne PLC naredbe poput manipulacije bitovima, rad s vremenskim i brojačkim elementima, kao i matematičke operacije. U nastavku je pružen vizualan prikaz usporedbe između fizičkih kontakata na PLC-u i kontaktnog dijagrama.

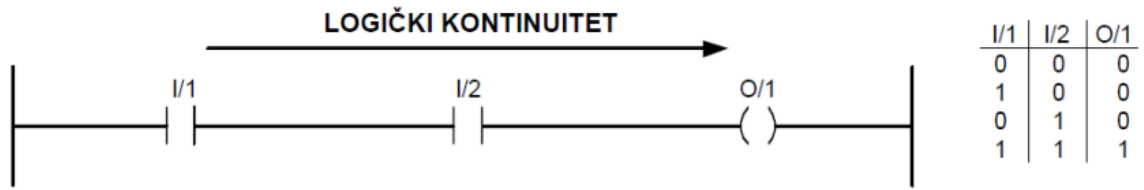


Slika 16. Fizički kontakti na PLC-u u usporedbi s kontaktnim dijagramom (Malčić, 12. kolovoza 2024.)

Program će kontinuirano pratiti stanje fizičkog ulaza I/1 i na temelju toga kontrolirati izlaz O/1. Ovo je najjednostavniji primjer kontaktnog dijagrama, ali u praksi se programiranje pristupom ljestvičastog dijagrama svodi na dvije osnovne logičke mogućnosti, odnosno operacije 'I' i 'IL' i njihove varijacije.

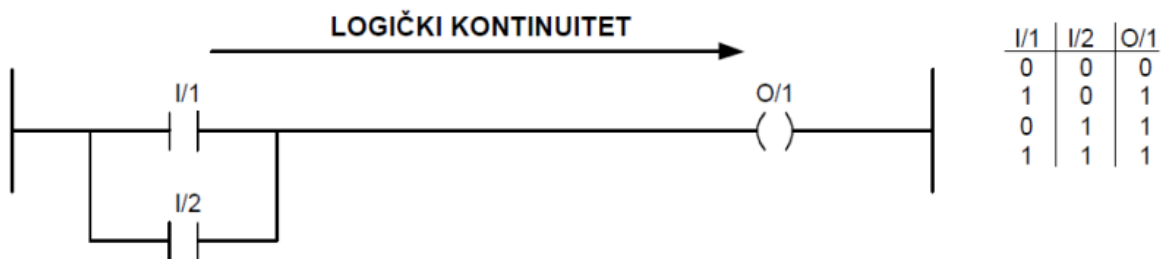
Nadalje, pružen je primjer gdje postoje dvije serijski spojene logičke naredbe 'I', što drugim riječima znači da oba uvjeta moraju biti zadovoljena kako bi se izlaz aktivirao, odnosno ako je samo jedan od dva logička uvjeta jednak nuli, tada izlaz neće biti aktiviran.





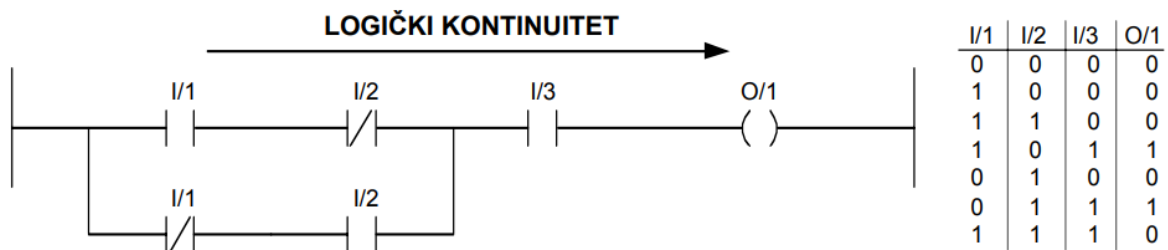
Slika 17. Ljestvičasti dijagram logičke operacije 'I' (Malčić, 12. kolovoza 2024.)

Kada se dvije naredbe povežu paralelno, rezultat je 'ILI' logička operacija. U tom slučaju, izlaz će se aktivirati ako je ispunjen barem jedan od dva uvjeta. Dakle, temeljna razlika između ljestvičastog dijagrama gdje imamo logičku operaciju 'ILI' u usporedbi s logičkom operacijom 'I', jest da je u ovom slučaju, 'ILI', dovoljno da je samo jedan uvjet zadovoljen kako bi se izlaz aktivirao. Drugim riječima, samo ako niti jedan od uvjeta nije zadovoljen, dolazi do prekida logičkog kontinuiteta te izlaz neće biti aktiviran. To je prikazano na sljedećoj slici:



Slika 18. Ljestvičasti dijagram logičke operacije 'ILI' (Malčić, 12. kolovoza 2024.)

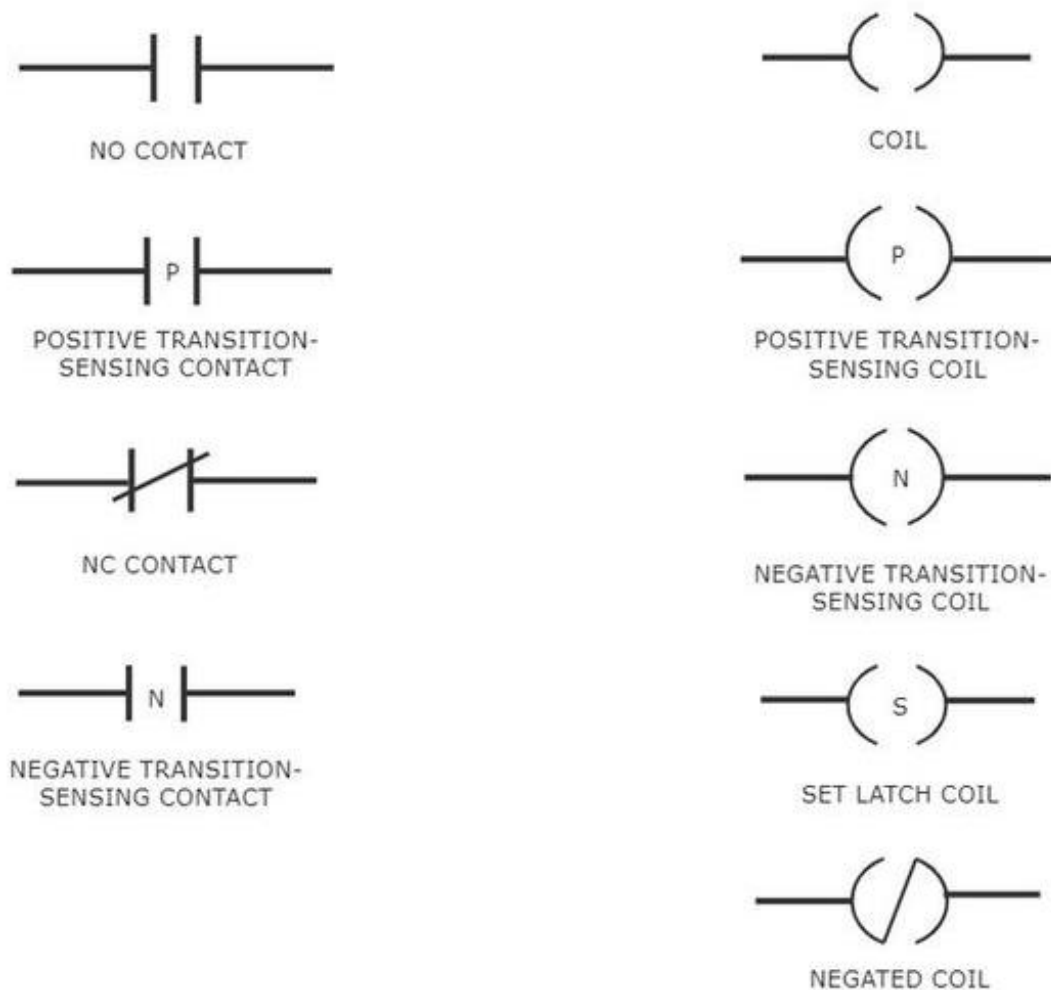
Osnovne logičke operacije se često kombiniraju, omogućujući nam da iz njih izvedemo cjelovitu Booleovu algebru. Ovakav pristup otvara neograničene mogućnosti prilikom programiranja. Primjer jedne takve kombinacije prikazan je na slici, gdje se koristi 'isključivo ILI' logička funkcija u kombinaciji s 'I' logičkom funkcijom.



Slika 19. Ljestvičasti dijagram logičke operacije 'isključivo ILI' u kombinaciji s logičkom operacijom 'I' (Malčić, 12. kolovoza 2024.)

Programi obično sadrže više logičkih krugova, a važno je napomenuti da se program izvršava sekvencijalno, odozgo prema dolje, počevši od prvog logičkog kruga pa sve do posljednjeg. Drugim riječima, program se izvršava prema Top – Down metodici.

Radi se o ljestvičastim dijagramima, samim time postoje jedinstveni simboli specifični za tu vrstu dijagrama. Na sljedećoj slici pružen je prikaz osnovnih simbola prema temeljnom standardu IEC 1131-3. Osim vizualnog prikaza osnovnih simbola, pruženo je objašnjenje značenja svakog od njih.



Slika 20. Osnovni simboli korišteni u programiranju ljestvičastim dijagramom (Automation, 12. kolovoza 2024.)

Jasno je da kao za dijagram klasa ili dijagram slučajeva korištenja, u ljestvičastom dijagramu također postoje jedinstveni simboli koji opisuju određeno stanje ili događaj. U nastavku su objašnjene instrukcijske liste uz funkcijsko blokovske dijagrame koji također predstavljaju jedne od standardnih načina i principa programiranja programabilnih logičkih kontrolera.

### 3.2.3. Instrukcijske liste

Instrukcijske liste (engl. *statement list*) predstavljaju programski jezik koji programerima omogućuje da kroz jednostavne korake pruže naredbe PLC uređaju koje je potrebno izvršiti. Naredbe su na razini assemblera, kao što je ranije rečeno. Prema tome, svaki redak koda u STL-u predstavlja jednu naredbu za centralnu procesorsku jedinicu. Modularna građa ovog programskog jezika pruža mogućnost efikasnog rješavanja složenih problema. Ovaj programski jezik ima jedinstvenu hijerarhijsku listu uz pomoć koje je program jednostavnije opisati, a i organizirati.

PROGRAM

KORAK (STEP)

NAREDBA

UVJETNI DIO

IZVRŠNI DIO

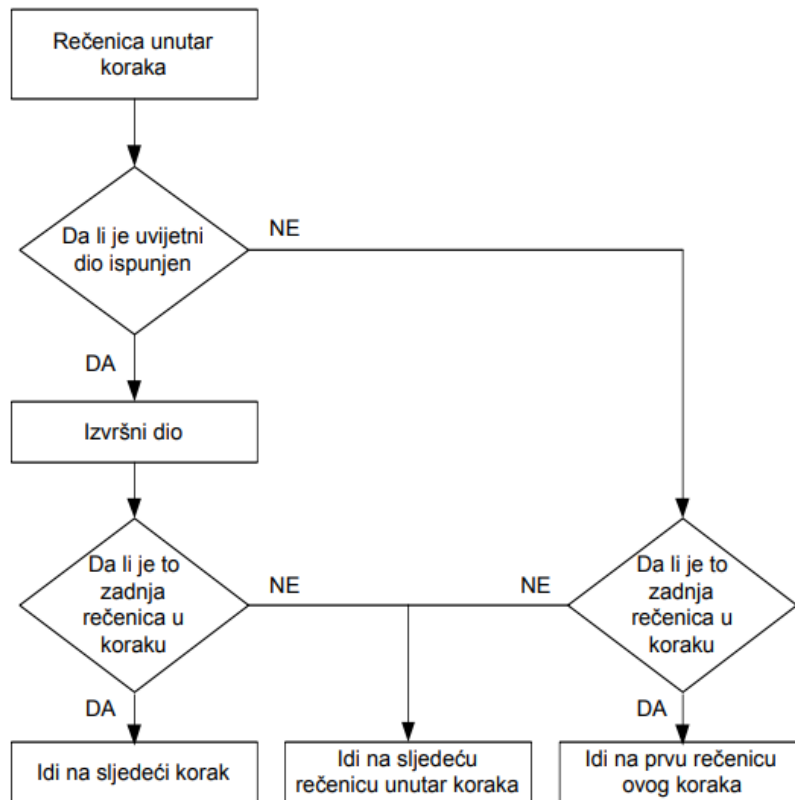
*Slika 21. Hijerarhijska lista programskog koda korištenjem instrukcijskih listi (Malčić, 13. kolovoza 2024.)*

Prema Šepacu (2016) STL program se sastoji od niza koraka koji se izvode sekvencijalno, od prvog do posljednjeg. Svaki korak unutar programa označava logički blok koji sadrži određeni dio koda, odnosno skup naredbi. Naredbe su strukturirane na način da se sastoje od uvjetnog dijela, gdje se provjeravaju određeni uvjeti, i izvršnog dijela koji sadrži operacije koje će se izvršiti ako su uvjeti zadovoljeni. Uvjetni dio obično počinje s ključnom riječju 'IF', nakon koje slijede uvjeti koji se provjeravaju, dok izvršni dio počinje s riječju 'THEN', koja ukazuje na operacije koje će biti izvršene.

Programiranje uz upotrebu koraka prati specifično pravilo koje određuje tok izvođenja programa. Ovo pravilo nalaže da, ako uvjet u posljednjoj rečenici unutar koraka nije ispunjen i izvršni dio nije izveden, program se ne nastavlja na sljedeći korak, već se vraća na početnu rečenicu tog istog koraka. Program se ponavlja u petlji sve dok se uvjet iz posljednje rečenice ne ispuni, čime se omogućava nastavak izvođenja sljedećih koraka.

Korištenje koraka u programiranju omogućuje veću kontrolu nad tokom informacije i učinkovito iskorištavanje mogućnosti koje pruža STL programski jezik. Iako je moguće cijeli

program napisati unutar jednog koraka, to otežava kontrolu nad procesom i ograničava fleksibilnost programiranja. Pravilo o povratku na početak koraka dok se uvjeti ne ispune može se najbolje ilustrirati blokovskim dijagramom koji pokazuje ovaj ciklički proces.



Slika 22. Ciklus izvođenja programa korištenjem instrukcijskih listi (Malčić, 13. kolovoza 2024.)

Budući da nije uvijek prikladno da se program kontinuirano izvršava u petlji unutar jednog koraka, uvedena je naredba 'NOP' (engl. *No Operation*), koja omogućuje da se uvjet automatski smatra zadovoljenim. Ova naredba osigurava da program ne ostane zaglavljen u beskonačnoj petlji, već da može nastaviti s izvođenjem sljedećih koraka, čak i ako uvjet nije eksplicitno ispunjen.

STEP 12				
IF			I:0/8	ako je ulaz 0/8 aktivan (visoko)
THEN	RESET		O:0/2	onda isključi izlaz 0/2
IF		N	O:0/1	ako je izlaz 0/1 nije aktivan
THEN	SET		O:0/9	onda uključi izlaz 0/9
IF	NOP			uvijek u stanju visoko
THEN	SET		B3:0/0	onda setiraj bit B3:0/0
	LOAD		N7:1	učitaj tu vrijednost u radni registar
	TO		N7:5	i spremi ga na adresu N7:5

Slika 23. Primjer naredbe NOP (Malčić, 13. kolovoza 2024.)

U primjeru funkcija uvijek postavlja stanje na visoko. Time se osigurava da, kada program dođe do zadnje rečenice unutar tog koraka, uvjet bude uvijek ispunjen, što omogućuje prelazak na sljedeći korak.

Osim naredbe uz 'NOP' koje igraju ključnu ulogu u STL programiranju, postoje još neke osnovne naredbe:

- **'=' (naredba pridruženja)** – ova se naredba koristi kod dodjele vrijednosti varijablama, primjerice može se postaviti vrijednost varijable 'x' na '5' korištenjem izraza: `x = 5`
- **'SET' (postavljanje bita na '1')** – naredba služi za promjenu vrijednosti bita s '0' na '1', primjerice ako želimo aktivirati neki od izlaznih releja, možemo koristiti: `SET O:0/1` kako bismo ga uključili
- **'RESET' (vraćanje bita na '0')** – suprotno od naredbe 'SET', ova naredba postavlja vrijednost bita nazad s '1' na '0', primjerice ako je potrebno nešto isključiti, moguće je taj učinak postići izrazom: `RESET O:0/1`
- **'OTHRW' (inače)** – ova naredba omogućuje izvršavanja određenog dijela koda u slučaju da uvjetni dio nije ispunjen, primjerice: `IF A > B THEN X = 1 OTHRW Y = 0`, gdje je 'Y = 0' samo ako 'A' nije veće od 'B'
- **'AND' (logičko 'I')** – naredba 'AND' koristi se u slučaju da se povezuje više uvjeta, gdje svi uvjeti moraju biti istiniti, odnosno ispunjeni kako bi se naredba izvršila, primjerice `IF A > B AND C < D THEN` provjerava jesu li oba uvjeta točna
- **'OR' (logičko 'ILI')** – naredba 'OR' omogućuje ispunjen uvjet čak i u slučaju gdje jedan od uvjeta nije ispunjen, odnosno, dovoljno je da je samo jedan uvjet ispunjen, primjerice `IF A > B OR C < D THEN` znači da će se naredba izvršiti čak i ako jedan od dva uvjeta nije ispunjen
- **'XOR' (isključivo 'ILI')** – naredba 'XOR' koristi se u situacijama kada je potrebno da je samo jedan od uvjeta ispunjen, ali ne i oba, primjerice `IF A > B XOR C < D THEN` znači da će se naredba izvršiti isključivo u slučaju da je jedan od uvjeta ispunjen, ali ne i oba

Također, osnovne matematičke operacije poput zbrajanja (+), oduzimanja (-), množenja (\*), i dijeljenja (/) koriste se uz naredbu 'LOAD TO' za proračune unutar programa. Na primjer, za dodavanje dviju vrijednosti koristi se naredba `LOAD TO X + Y`.

Naredbe usporedbe, kao što su veće (>), manje (<), jednako (=), veće ili jednako (≥), manje ili jednako (≤), i nejednako (<>), uvijek se koriste u uvjetnom dijelu zajedno s 'IF THEN' naredbom. Ove naredbe omogućuju programeru da provjerava odnose između varijabli i da na osnovu tih odnosa donosi odluke u toku izvođenja programa. Na primjer, `IF X > Y THEN` izvršit će određene naredbe ako je X veće od Y.

### 3.2.4. Funkcijski blokovski dijagrami

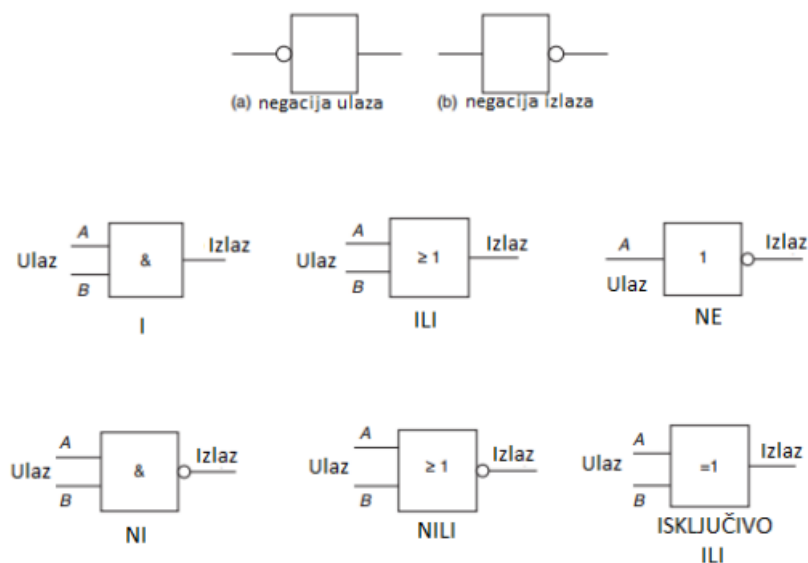
Funkcijski blokovski dijagram (engl. *function block diagram*) predstavlja jedan od načina programiranja PLC uređaja korištenjem blokova. Ovaj se programski jezik može najbolje opisati kao grafički jezik koji služi za prikaz toka signala uz podatke kroz funkcijske blokove, koji su ovdje centar pažnje.



Slika 24. Koncept funkcijskih blokovskih dijagrama (Šepac, 13. kolovoza 2024.)

Funkcijski se blok prikazuje kao pravokutnik, gdje se ulazi nalaze s lijeve strane, a izlaz je smješten s desne strane funkcijskog bloka. Ovi blokovi mogu uključivati standardne operacije i funkcije, kao što su brojači vremena (engl. *timers*) ili brojila (engl. *counters*), no s druge strane mogu biti prilagođeni i definirani od strane korisnika.

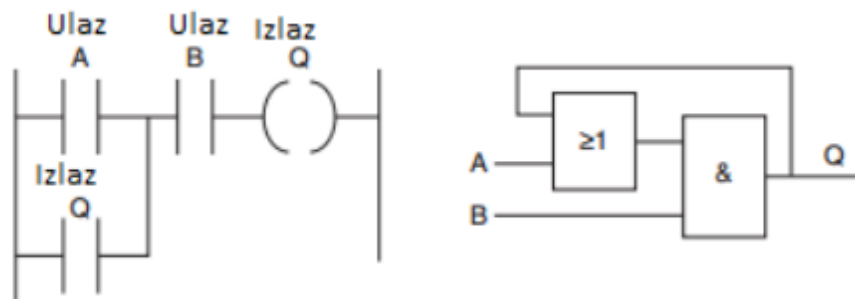
Na taj način se pruža fleksibilnost i prilagodba specifičnim željama i potrebama korisnika, samim time i programa. Na sljedećem vizualnom prikazu su vidljive osnovne funkcije koje se koriste u programiranju koristeći funkcijske blokovske dijagrame uključujući nekoliko primjera funkcijskih blokova koji olakšavaju shvaćanje logičkih operacija i upravljanje raznim procesima.



Slika 25. Osnovne operacije u funkcijskim blokovskim dijagramima (Šepac, 13. kolovoza 2024.)

U prvom redu prikazane su funkcije s negiranim ulazima i izlazima. Funkcija 'I' aktivira izlaz samo kada oba ulaza, A i B, imaju rezultat 'ISTINA', što odgovara logičkoj vrijednosti '1'. Funkcija 'ILI' aktivira izlaz kada je barem jedan od ulaza, A ili B, u stanju 'ISTINA', odnosno logičkoj '1'. Funkcija 'NE' aktivira izlaz ako je ulaz u stanju logičke '0'. Kombinacijom ovih osnovnih funkcija mogu se stvoriti izvedene funkcije poput 'NI', 'NILI' i 'ISKLUČIVO ILI'.

Na slici je prikazan ljestvičasti dijagram koji ilustrira funkciju u kojoj izlaz djeluje kao ulaz. Ista funkcija je implementirana i u funkcijsko blok dijagramu kao povratna veza, omogućujući složeniju logičku kontrolu.



Slika 26. Usporedni prikaz ljestvičastog dijagrama i funkcijskog blokovskog dijagrama (Šepac, 13. kolovoza 2024.)

Mnogi staromodni PLC programeri će gotovo uvijek koristiti ljestvičaste dijagrame bez da ispravno istraže ostale opcije, ali ponekad je FBD (engl. *function block diagram*) pristup najisplativiji. Ovo je slučaj jer ponekad može rezultirati bržim sustavom koji bolje reagira, što je u industriji ključan faktor. Automatizacija procesa dobro funkcionira s FBD-ovima jer se blok dijagrami koriste za dizajn i objašnjenje njihovog procesa kada se podaci dijele između raznih događaja i procesa. Jedan korak može napredovati i ostaviti svoje artefakte potrebne za ulaz u sljedeću fazu tek nakon što je prethodni proces unosa dovršen. (Dietrich, 2023).

Dobar primjer je pravljenje pizze, prvo je potrebno kupiti sastojke, pa ih pripremiti uz tijesto, ispeći, spakirati i izrezati pizzu na dijelove. Izlaz jedne funkcije izravno utječe na sljedeću, a neke funkcije poput pečenja mogu imati dva ulaza, primjer ovoga je zagrijana pećnica i pizza.

Prema Dietrichu (2023) situacije s ponavljajućim kodom unutar programa imaju koristi od FBD programa jednostavnim povlačenjem, ispuštanjem i dodjeljivanjem korisniku koji može vrlo brzo izgraditi program s dokazanim funkcijama. To može smanjiti obuku i zahtjeve za programerima napredne razine u radnji.

## 4. Zaključak

Današnja industrijska proizvodnja je nezamisliva bez automatizacije, zahvaljujući procesu automatizacije, brojni ponavljajući procesi koji bi oduzimali puno vremena za realizaciju, budu obavljeni u kraćem roku uz manji trošak. Kako bi automatizacija bila što efikasnija, postoje programabilni logički kontroleri koji pružaju razne mogućnosti od koristi za korisnika. U prošlosti su se koristili releji, ali su ih zamijenili PLC uređaji.

Nakon proučavanja načina programiranja PLC uređaja, vidljivo je da postoji širok spektar opcija za razvoj aplikacija za programabilne logičke kontrolere. Svaki od tih načina ima svoje prednosti i nedostatke, a i situacije u kojima je najisplativije koristiti određeni način. Drugim riječima, svaki programski jezik je pogodan u određenoj situaciji, ali najtransparentniji programski jezik za programiranje PLC uređaja je korištenje ljestvičastog dijagrama koji su zbog svoje sličnosti s relejnim sustavima pružili mogućnost brze tranzicije na bolji sustav. PLC uređaji imaju ključnu ulogu u modernoj industrijskoj automatizaciji, omogućujući precizno i točno upravljanje uz fleksibilnost u realnim uvjetima. Njihova prilagodljivost različitim industrijskim procesima te sama sposobnost da pouzdano funkcioniraju u kompleksnim i zahtjevnim okruženjima čine ih neizostavnim alatom za postizanje visoke razine efikasnosti i uštede. Programiranje i korištenje PLC uređaja ne samo da ubrzava proizvodne procese, već također omogućava kontinuirano poboljšanje proizvodnje.

Na kraju, PLC tehnologija predstavlja čvrst temelj na kojem počiva moderna automatizirana proizvodnja, osiguravajući da industrijski sustavi rade s maksimalnom količinom učinkovitosti i pouzdanosti.



# Popis literature

Guedes, L., A. (2010). *Programmable Logic Controller*. InTech.

Malčić, G. (2015). *Programirljivi Logički Kontroleri*. Zagreb: Tehničko Veleučilište u Zagrebu, Sveučilište u Zagrebu.

Fakultet elektrotehnike i računarstva [FER] (6. 4. 2021.) *Automatizacija*. Preuzeto 7. 8. 2024. sa <https://www.fer.unizg.hr/zesa/sk/automatizacija>

Paessler the monitoring experts (21. 4. 2022.) *IT Explained: PLC*. Preuzeto 8. 8. 2024. sa <https://www.paessler.com/it-explained/plc#:~:text=A%20typical%20PLC%20is%20connected,slide%20into%20a%20PLC%20rack.>

Šepac, M. (2016). *Programirljivi logički kontroleri (PLC) (Završni rad)*. Rijeka: Sveučilište u Rijeci, Filozofski fakultet. Preuzeto sa <https://repository.ffri.uniri.hr/islandora/object/ffri%3A816/datastream/PDF/view>

*Info PLC* (19.1.2012.) *PLC Memory*. Preuzeto 9.8.2024. s [https://www.infopl.net/files/descargas/rockwell/infopl\\_net\\_plc\\_memory.pdf](https://www.infopl.net/files/descargas/rockwell/infopl_net_plc_memory.pdf)

*Shawn Dietrich* (29. 8. 2023.) *PLC Programming With Function Block Diagrams*. Preuzeto 13.8.2024. s <https://control.com/technical-articles/plc-programming-with-function-block-diagrams/>

*Electrical & Automation Solutions* [Slika] (bez dat.) Preuzeto 10.8.2024. s [https://www.google.com/search?sca\\_esv=2fb4f24297e46be4&rlz=1C1GCEA\\_enHR969HR969&hl=en&sxsrf=ADLYWIKvGFIFdW6TydTm5HL\\_kVd9ZC7TmQ:1723299734759&q=plc+vs+releji&tbm=isch&source=lnms&fbs=AEQNm0Aa4sjWe7Rqy32pFwRj0UkWERaHdBms7t-tHL116ec0FnDlrxqGhNFSZEtYqV91QHsJxJfdWVTuD0IrLFirCeJ44XY3FEArMddqGqwXAI-D-tzLq2O82HVp5pZIJZm5LiJbd9wfStD6JC6xtsfYEOLyxkvjxNLUPS5a3jXHfgh4QtgYvQ&sa=X&ved=2ahUKEwj-i6Dkz-gHAXV0g\\_0HHRvGNh8Q0pQJeqQIERAB&biw=1707&bih=830&dpr=1.13#imgrc=NaWleIGc ewVrRM](https://www.google.com/search?sca_esv=2fb4f24297e46be4&rlz=1C1GCEA_enHR969HR969&hl=en&sxsrf=ADLYWIKvGFIFdW6TydTm5HL_kVd9ZC7TmQ:1723299734759&q=plc+vs+releji&tbm=isch&source=lnms&fbs=AEQNm0Aa4sjWe7Rqy32pFwRj0UkWERaHdBms7t-tHL116ec0FnDlrxqGhNFSZEtYqV91QHsJxJfdWVTuD0IrLFirCeJ44XY3FEArMddqGqwXAI-D-tzLq2O82HVp5pZIJZm5LiJbd9wfStD6JC6xtsfYEOLyxkvjxNLUPS5a3jXHfgh4QtgYvQ&sa=X&ved=2ahUKEwj-i6Dkz-gHAXV0g_0HHRvGNh8Q0pQJeqQIERAB&biw=1707&bih=830&dpr=1.13#imgrc=NaWleIGc ewVrRM)

*PLC Market* [Slika] (bez dat.) Preuzeto 10.8.2024. s <https://www.indiamart.com/proddetail/plc-memory-card-4566858955.html>

*Automation Direct* [Slika] (bez dat.) Preuzeto 11.8.2024. s [https://www.automationdirect.com/adc/shopping/catalog/programmable\\_controllers/directlogic\\_plcs\\_\(micro\\_to\\_small,\\_brick\\_-a-\\_modular\)/directlogic\\_accessories/d2-hpp](https://www.automationdirect.com/adc/shopping/catalog/programmable_controllers/directlogic_plcs_(micro_to_small,_brick_-a-_modular)/directlogic_accessories/d2-hpp)

*Amatrol* [Slika] (bwz dat.) preuzeto 12.8.2024. s <https://amatrol.com/product/plc-program-simulation-training/>

*Automation* [Slika] (bez dat.) Preuzeto 12.8.2024. s <https://www.automation.com/en-us/articles/2016-1/iec-1131-the-first-universal-process-control-langu>

# Popis slika

Slika 1. Kategorije PLC računala (Malčić, 8. kolovoza 2024.).....	3
Slika 2. Usporedba PLC računaa s relejima (Electrical & Automation Solutions, 10. kolovoza 2024.).....	4
Slika 3. Struktura PLC računala (Malčić, 8. kolovoza 2024.) .....	6
Slika 4. Struktura centralne procesorske jedinice PLC-a (Šepac, 8. kolovoza 2024.) .	7
Slika 5. Primjer ožičenja ulaznih stezaljki PLC-a sa 12 digitalnih ulaza (Malčić, 8. kolovoza 2024.).....	7
Slika 6. Primjer ožičenja izlaznih stezaljki PLC-a sa 12 digitalnih izlaza (Malčić, 8. kolovoza 2024.).....	8
Slika 7. Memorijska kartica PLC računala (PLC Market, 10. kolovoza 2024.) .....	9
Slika 8. Uređaj za napajanje PLC računala (Šepac, 8. kolovoza 2016.) .....	10
Slika 9. Komunikacija s ndređenim PC računalom (Malčić, 8. kolovoza 2024.).....	10
Slika 10. Osnovni PLC uređaj uz proširenje (Malčić, 8. kolovoza 2024.).....	11
Slika 11. Sustav adresiranja PLC-a (Malčić, 10. kolovoza 2024.).....	12
Slika 12. Princip rada PLC računala u koracima (Malčić, 10. kolovoza 2024.).....	13
Slika 13. Ručni PLC programator (Automation Direct, 11. kolovoza 2024.) .....	15
Slika 14. PLC simulator stanja (Amatrol, 12. kolovoza 2024.) .....	16
Slika 15. Usporedba električnog i logičkog puta (Malčić, 12. kolovoza 2024.).....	17
Slika 16. Fizički kontakti na PLC-u u usporedbi s kontaktnim dijagramom (Malčić, 12. kolovoza 2024.).....	18
Slika 17. Ljevičasti dijagram logičke operacije 'I' (Malčić, 12. kolovoza 2024.).....	19
Slika 18. Ljevičasti dijagram logičke operacije 'ILI' (Malčić, 12. kolovoza 2024.).....	19
Slika 19. Ljevičasti dijagram logičke operacije 'isključivo ILI' u kombinaciji s logičkom operacijom 'I' (Malčić, 12. kolovoza 2024.).....	19
Slika 20. Osnovni simboli korišteni u programiranju ljestvičastim dijagramom (Automation, 12. kolovoza 2024.) .....	20
Slika 21. Hijerarhijska lista programskog koda korištenjem instrukcijskih listi (Malčić, 13. kolovoza 2024.) .....	21
Slika 22. Ciklus izvođenja programa korištenjem instrukcijskih listi (Malčić, 13. kolovoza 2024.).....	22
Slika 23. Primjer naredbe NOP (Malčić, 13. kolovoza 2024.) .....	22
Slika 24. Koncept funkcijskih blokovskih dijagrama (Šepac, 13. kolovoza 2024.).....	24
Slika 25. Osnovne operacije u funkcijskim blokovskim dijagramima (Šepac, 13. kolovoza 2024.).....	24
Slika 26. Usporedni prikaz ljevičastog dijagrama i funkcijskog blokovskog dijagrama (Šepac, 13. kolovoza 2024.) .....	25

## Popis tablica

Tablica 1. Primjeri sustava adresiranja (autorski rad) .....	13
--	----