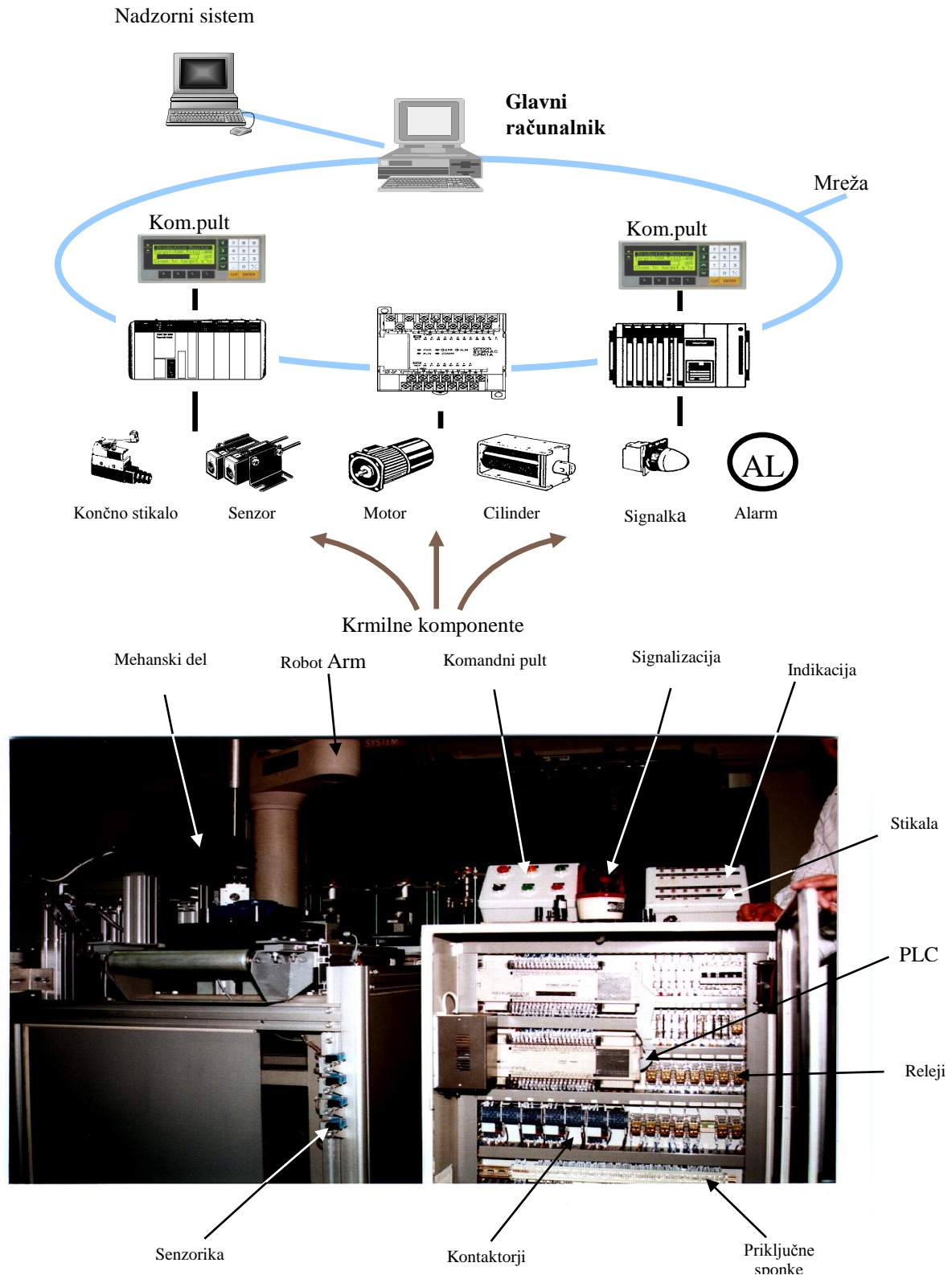


## II. KRMILNI SISTEM S PROGRAMIRLJIVIM KRMILNIKOM

Krmilni sistem sestavljajo elektronske komponente, naprave in druga oprema, s pomočjo katere neposredno vplivamo na nek tehnološki postopek ali proces. Krmilni sistem zajema upravljalni del s prosto programirljivim krmilnikom PLC (Programmable Logic Controller), nadzorni sistem s PC, komandne tabloje ali operativne panele, periferijo s senzoriko in izvršilne elemente (aktuatorje).



Na shemi je prikazana tipična aplikacija robotske krmilne naprave. Procesno sekvenco krmili PLC. Periferne enote (stikala, senzorji, ...) so povezane na vhodne kanale PLC preko vhodnih priključnih sponk. Izhodne enote – aktuatorji (releji, kontaktorji, signalizacija, ...) so priključene na izhodne kanale PLC. Proces krmilimo s pomočjo aplikacijskega

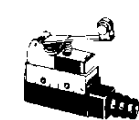
programa, naloženega v pomnilniku CPU krmilnika. PLC nadzoruje in krmili delovanje naprave v avtomatskem in ročnem režimu, varnostne akcije (izklop v sili) pa se vršijo direktno, mimo PLC.

## 1. Vloga programirljivega krmilnika PLC

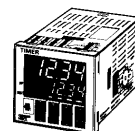
V avtomatskem krmilnem sistemu predstavlja PLC srce krmilnega sistema. S krmilnim aplikacijskim programom PLC nadzoruje stanje sistema in trenutna stanja vseh vhodnih in izhodnih kanalov (digitalnih in analognih). PLC lahko uporabimo za krmiljenje enostavnih, koračnih in zahtevnih kompleksnih procesov. Lahko jih povežemo preko ustreznih komunikacijskih linij v mreže (Profibus, Ethernet, ...) in tvorimo različne hierarhične nivoje PLC sistemov. Takšen princip izgradnje krmilnega sistema je značilen za kompleksne industrijske procese (robotika, krmilno-regulacijski sistemi).

### Vhodne enote - periferija

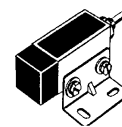
Inteligenca avtomatiziranega sistema je v največji meri odvisna od sposobnosti PLC, kako in koliko vhodnih informacij lahko zajema preko periferije. Med ročne vhodne enote prištevamo tipke, tastature in razna preklopna stikala. Za detekcijo stanj v procesu pa imamo na voljo veliko množico senzorjev, analognih in digitalnih (fotosenzorji, induktivni in kapacitivni senzorji, ...). Signale iz vhodnih enot pripeljemo na različne vhodne module PLC, ki galvansko ločijo periferijo od CPU.



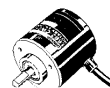
Končno stikalo



Časovni



Fotosenzor



Enkoder



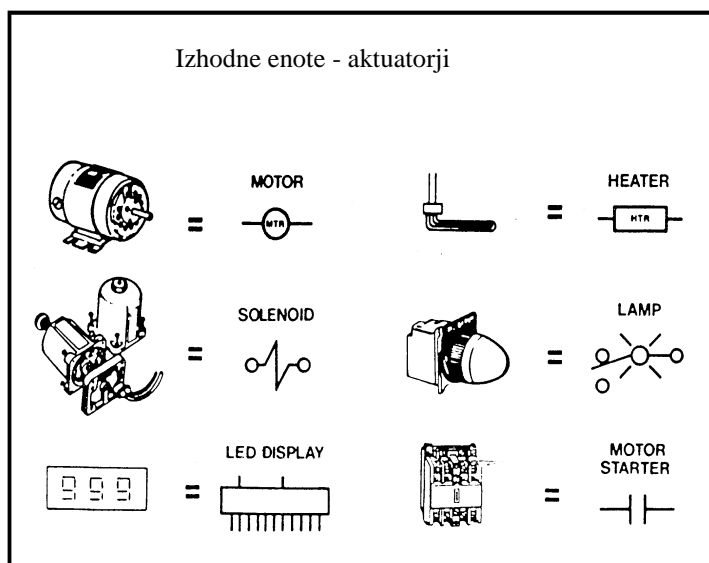
L, C stikalo

### Izhodne enote

Aktuatorji so vmesni elementi med PLC krmiljem in procesom. Iz izhodnih enot lahko vodimo povratne signale na vhod PLC.

Izhodne enote delimo:

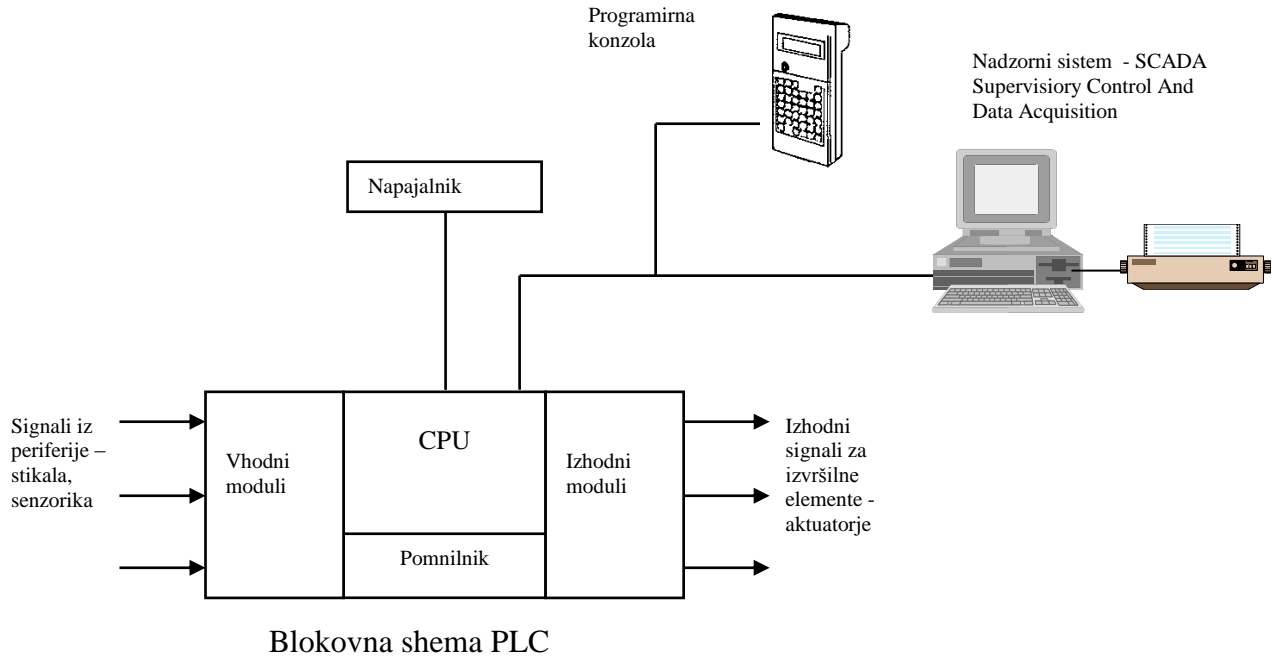
- končni izvršilni elementi (motorji, emg. ventili, ...)
- izvršilni členi ali aktuatorji (releji, kontaktorji, ...)
- signalizacija in alarmne enote (prikazovalniki, hupe, ...)



## 1.1. Komponente programirljivega krmilnika PLC

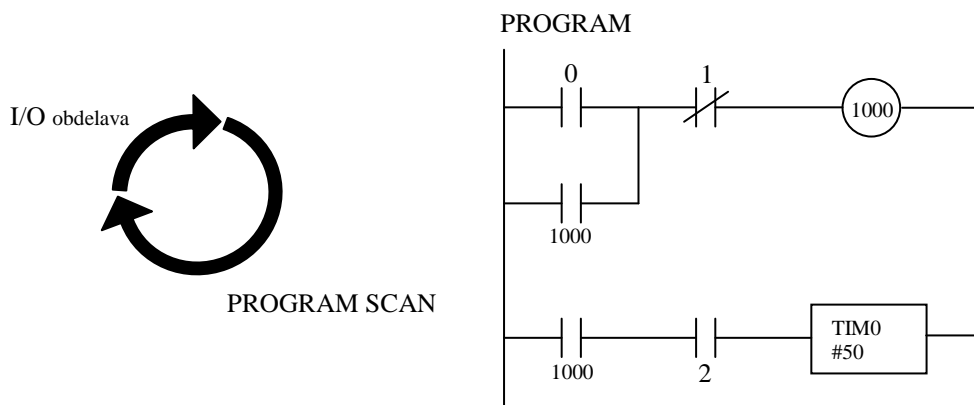
PLC sestavljajo naslednje osnovne komponente:

- centralna procesna enota CPU, ki vsebuje aplikacijski program in omogoča podporo ostalim enotam
- vhodno-izhodni vmesni moduli, ki so direktno priključeni na vhodno-izhodne naprave
- aplikacijski program, ki nadzira delovanje PLC tako, da ob aktiviranju nekega vhodnega signala ustrezno ukrepa in aktivira določeni izhodni signal.



## 1.2. Princip delovanja aplikacijskega programa (skaniranje)

Proces branja vhodov, izvrševanje samega programa in obdelavo ter postavljanje izhodov poznamo pod imenom 'skaniranje' (scanning). Čas obdelave programa je odvisen od dolžine aplikacijskega programa in traja nekaj milisekund.

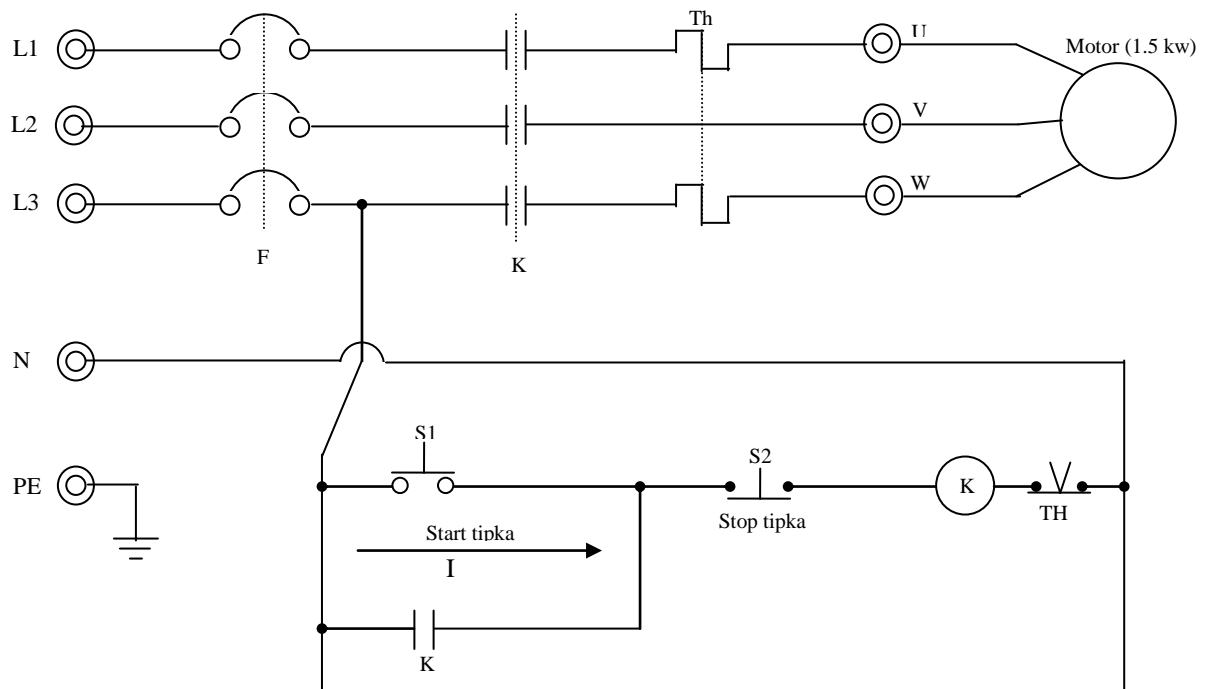


### 1.3. Prednosti izvedb krmilnih sistemov s PLC krmiljenjem

- manjša zahteva v ožičenju sistema (do 80%)
- manjša poraba električne energije krmiljenja
- možnost diagnostike napak, alarmov in statistike
- enostavno spreminjanje aplikacijskega programa in s tem delovanja krmiljenja
- PLC krmiljenje lahko izvedemo v primerno manjšem prostoru, privarčujemo pri časovnikih, relejih, ...
- hitrejše izvajanje krmilnih funkcij
- nižja cena pri srednjih in večjih krmilnih sistemih
- večja zanesljivost delovanja PLC sistema (digitalni števeci, časovniki)
- lažja in hitrejša izdelava dokumentacije.

## 2. Izvedba klasičnega krmilja s PLC

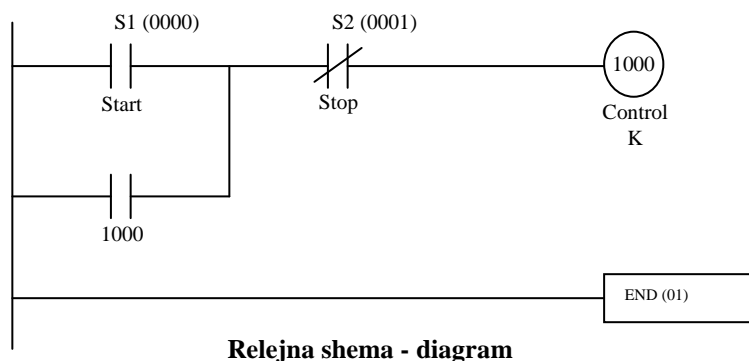
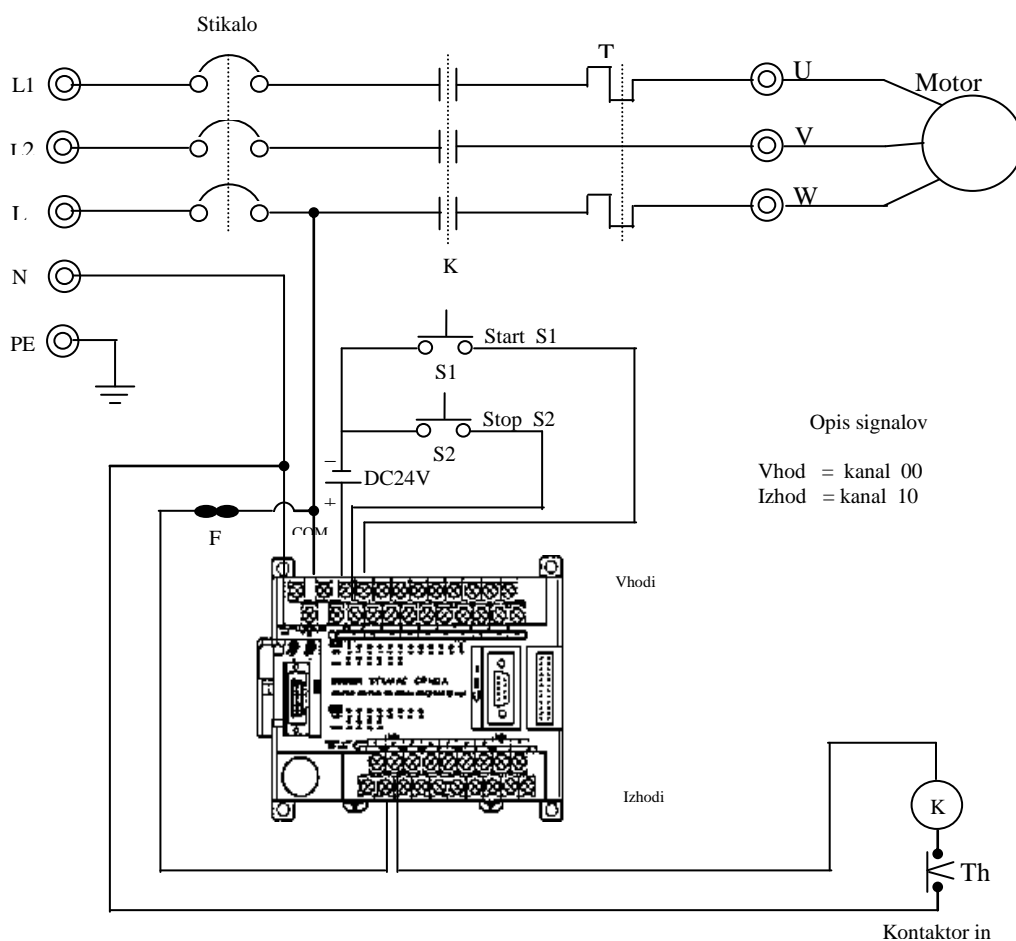
**Primer:** Vklon in izklop 3-faznega motorja



Ko pritisnemo tipko S1, bo stekel tok skozi krmilno vezje in aktiviral kontaktor K. Vzporedno vezani kontakt K k vklopni tipki je samodržni kontakt. Stikalo S2 je namenjeno izklopu pogona.

Dano krmilno vezje bomo izvedli s pomočjo PLC krmilnika. Povezava in primer aplikacijskega programa, zapisanega v obliki lestvičnega diagrama oz. relejne sheme, je prikazan na naslednjih shemah.

Primer vezalne sheme za vklop in izklop motorja preko dveh komandnih tipk je tehnološko najenostavnejši primer krmiljenja. Princip delovanja sloni na izvedbi krmiljenja s samodržnim kontaktom kontaktorja K. Ob pritisku na tipko start, bo navitje kontaktorja K dobilo krmilno napetost, kontaktor bo pritegnil svoje kontakte in s tem vklopil porabnik – motor, hkrati pa bo s svojim pomožnim delovnim kontaktom še premostil priključe tipke start (samodržni kontakt). Motor bo priključen na napajalno napetost toliko časa, dokler ne bomo aktivirali tipke stop, ki je svojim mirovnim kontaktom vezana zaporedno k start tipki. Krmilno ali relejno shemo (programska aplikacija) prikazuje spodnja slika.



## 2.1. Potek načrtovanja in izvedbe krmiljenja s PLC

Snovanje krmiljenja s PLC predstavlja izvajanje naslednjih procedur:

### 1. Definiranje tehnologije procesa

Določiti je potrebno, v katero skupino zahtevnosti spada obravnavni tehnološki proces ali naprava, ki jo želimo krmiliti. Od stopnje kompleksnosti krmiljenja je potem odvisna tudi izbira krmilne opreme, periferije in tipa PLC. Sistem, ki ga bomo rešili z navedeno opremo bomo imenovali PLC krmilni sistem. Akcije (pomiki, pogoni,...) nadzorujemo neprestano z vhodnimi enotami (periferijo oz. senzoriko) in specifične pogoje pošiljamo v PLC, ki na osnovi aplikacijskega programa ustrezno aktivira izhodne komponente.

### 2. Določanje vhodov in izhodov

Definirati moramo vse vhodne signale (zaporedna številka, priključna sponka, tip, vrsta kontakta /NC,NO, adresa v PLC), ki jih bomo v tehnološkem procesu uporabili. Prav tako moramo definirati vse izhodne signale (priključne sponke, vrsta, adresa - naslov) preko katerih bomo vodili aktivnosti v procesu (motorji, ventili, pnevmatski ali hidravlični cilindri, signalizacija,...).

### 3. Izdelava aplikacijskega programa

Zajema načrtovanje in zapis aplikacije s pomočjo relejnih shem, mnemoničnih ukazov (assembler) ali funkcijskih diagramov. Program izvedemo s pomočjo različnih operacijskih programov na PC in ga nato prenesemo v uporabniški pomnilnik PLC. Program zapišemo lahko na osnovi različnih algoritmov (opisni, časovni diagrami, digrami poteka, funkcijski načrti, logične enačbe,...).

### 4. Nalaganje programa v PLC

Po izdelavi aplikacijskega programa, le tega prenesemo v PLC in ga testiramo s simulacijskim programom, preverimo vse naslove vhodnih in izhodnih kanalov in preizkusimo vedno najprej delovanje v ročnem režimu delovanja naprave. Avtomatski režim delovanja preizkusimo po korakih in na koncu testiramo delovanje celotne aplikacije: ročno in avtomatsko delovanje, alarmne situacije, preklope med ročnim in avtomatskim režimom delovanja, diagnostiko in povezave s komandnim mestom.

### 5. Zagon krmilnega sistema

Delovanje krmilnega sistema moramo testirati dlje časa in v tem obdobju odpraviti vse možne napake. Predvsem moramo preizkusiti vse varnostne funkcije sistema (zasilni izklop, STOP, ...). Po končanem zagonu je potrebno ažurirati končni aplikacijski program in ga primerno shraniti.

## Adresni plan – popis vhodnih in izhodnih kanalov

Popis vhodnih kanalov

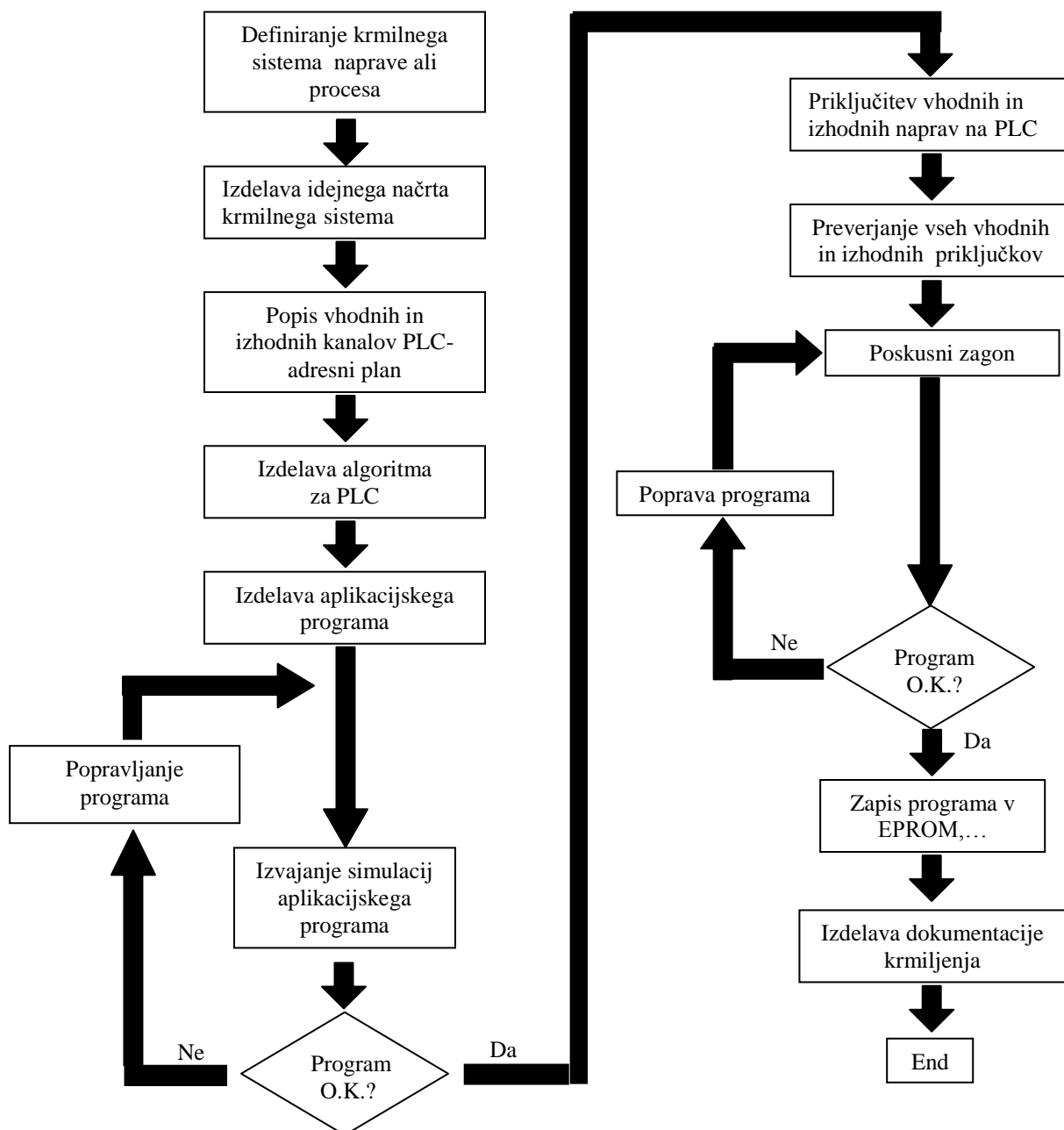
Adresa	Simbol	Opis	Kontakt	Opomba - sponke
000.00	S1	-X1:1 start tipka	delovni D	vklop motorja M1
000.01	S2	-X1:2 stop tipka	mirovni M	izklop motorja M1
000.02	S3	-X1:3 kontakt bim.releja	mirovni M	termozaščita M1

Popis izhodnih kanalov

Adresa	Simbol	Opis	Opomba
010.00	KM1	kontaktor	-X2:1,2,3 vklop motorja M1
010.01			
010.02			

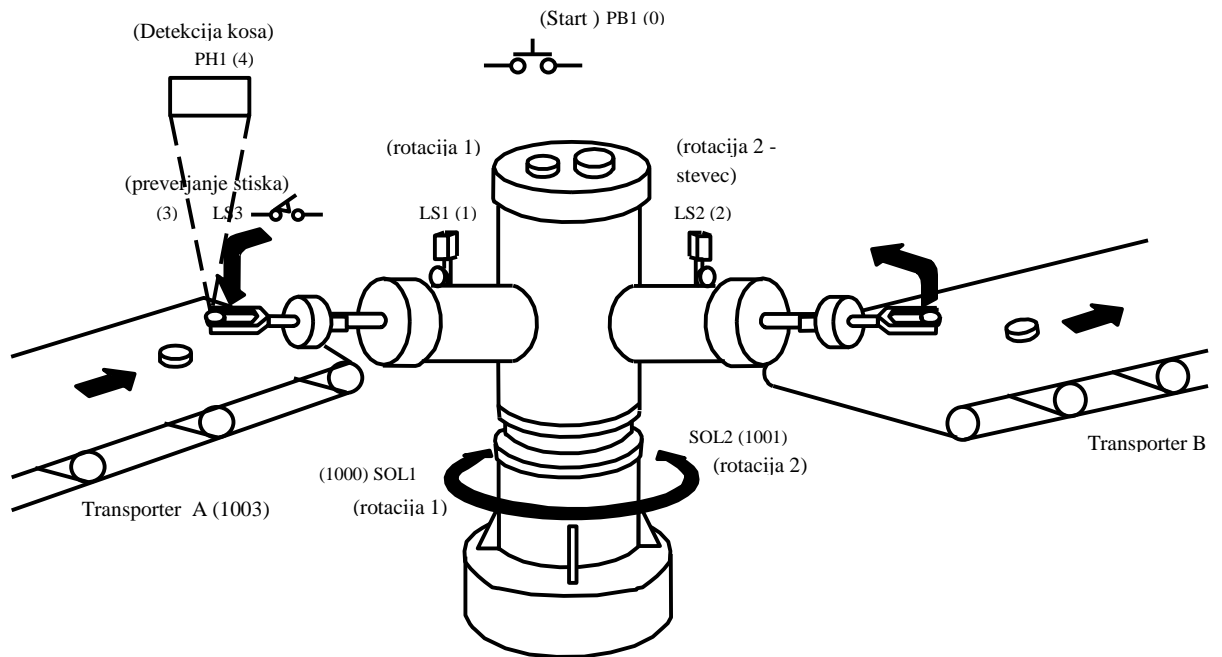
Popis uporabljenih podatkovnih pomnilniških lokacij, specialnih bitov, števcov, časovnikov ...

## Diagram poteka načrtovanja in izvedbe krmiljenja s PLC



## 2.2. Primer izdelave aplikacije krmiljenja mehanske roke

### Tehnološka shema naprave



### Opis sekvence delovanja

Iz tehnološke sheme naprave je razvidno, da mehanska roka prestavi obdelovanec, ki prispe po transporterju A, na transporter B. Sekvenco akcij lahko zapišemo kot zaporedje naslednjih korakov:

1. Ob pritisku na tipko Start, se mehanska roka zavrti v začetni položaj (nad transporter A) – rotacija 1
2. V trenutku pozicioniranja nad transporterjem A v delovni položaj, roka prime obdelovanec
3. Po prijemu obdelovanca, se roka zavrt nad transporter B – rotacija 2
4. Po pozicioniranju nad transporter B, roka izpusti obdelovanec

### Adresni plan

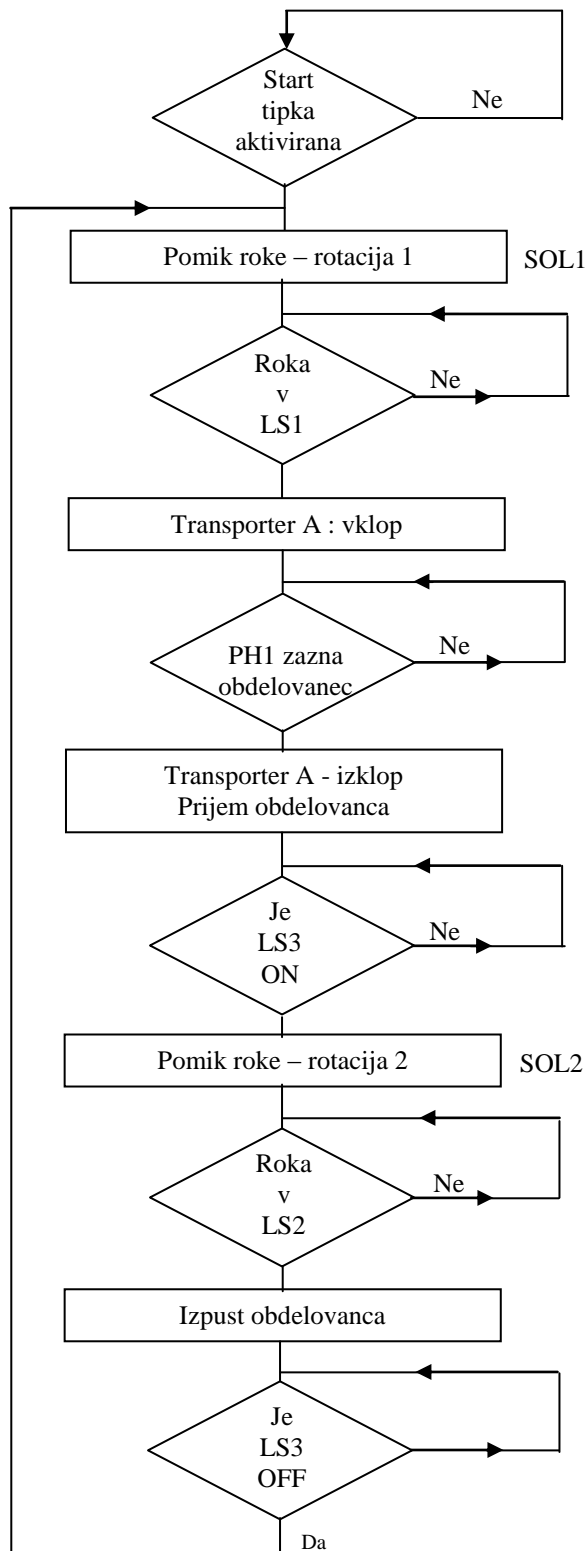
VHODI	OPIS	IZHODI	OPIS
00000	start tipka	1000	SOL 1
00001	LS 1	1001	SOL 2
00002	LS 2	1002	SOL 3
00003	LS 3	1003	transporter A
00004	PH 1		
00005	stop tipka		
00006	reset tipka		

### Zapis algoritma delovanja s pomočjo logičnih enačb

1. Rotacija 1:  $SOL-1 = (PB1 + Izpust) \cdot LS2 \cdot \overline{LS3}$ ; ( $\overline{SOL-1} = SOL-1 \cdot LS1$ )
2. Vklon A in čakanje na obdelovanec:  $TR-A = LS1 \cdot PH1 \cdot SOL3$
3. Pozicioniranje in prijem obdelovanca:  $TR-A = TR-A \cdot PH1$ ;  $SOL-3 = TR-A \cdot PH1 \cdot LS1$
4. Rotacija 2:  $SOL-2 = SOL-3 \cdot LS3 \cdot LS1$ ; ( $\overline{SOL-2} = SOL-2 \cdot LS2$ )
5. Izpust obdelovanca na transporter B:  $Izpust = \overline{SOL-3} = SOL-3 \cdot LS2$
6. Po izpustu obdelovanca se mehanska roka zavrti ponovno v začetni položaj



**Algoritem delovanja naprave kot diagram poteka**



**STOP POGOJI:**

V primeru STOP, se mora roka v kateremkoli položaju ustaviti;

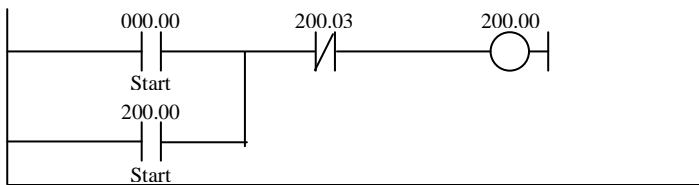
Po ponovnem aktiviranju START, roka nadaljuje pomik

**Zapis programa v obliki relejne sheme – Ladder diagram**

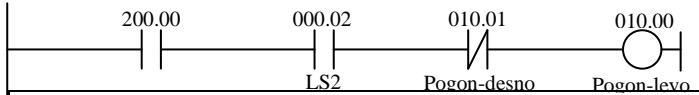
Ladder Diagram : Main 1 Network 1 / Blok 1

Main 1 – Krmiljenje robota – demonstracija robotskega delovanja: roka prestavlja obdelovance iz transporterja A na transporter B.

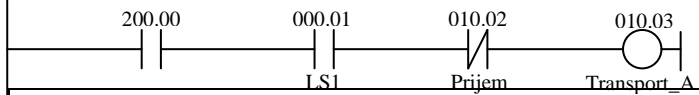
Network 1 – Startni pogoji



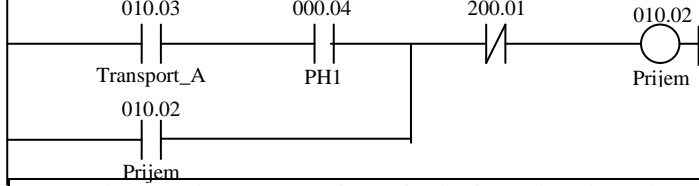
Network 2 – Roka v smeri urinega kazalca - levo



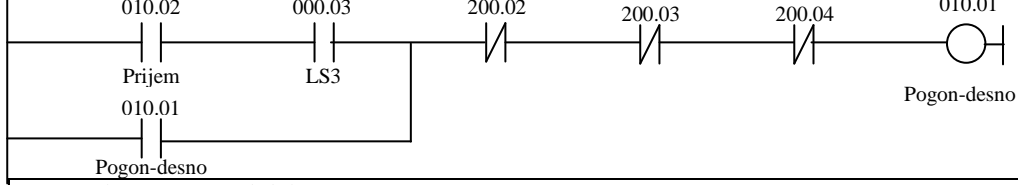
Network 3 – Transporter A



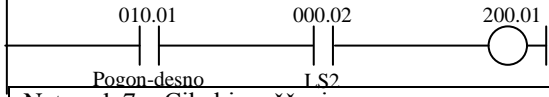
Network 4 – Prijemalka



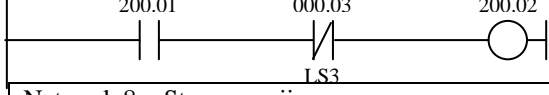
Network 5 – Roka v nasprotni smeri od urinega kazalca - desno



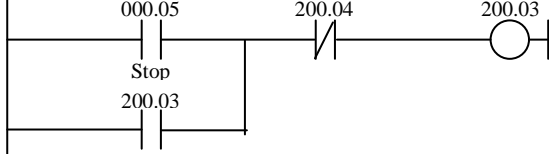
Network 6 – Izpust obdelovanca



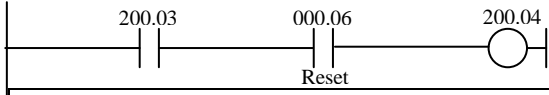
Network 7 – Cikel izpuščanja



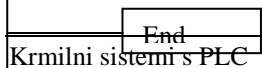
Network 8 – Stop pogoji



Network 9 – Reset pogoji - Reset po stop



Network 10 – End



### 2.3. Osnovno preverjanje krmilnega sistema

Osnovno preverjanje in inicializacijo sistema izvedemo po postavitvi in ožičenju sistema. Preverjanje sistema opravimo pred poskusnim zagonom (test run). Koraki osnovnega preverjanja so:

Korak	Preverjanje
Napajanje in I/O priključki	Ali je ožičenje pravilno – adrese? Ali so priključne sponke ustrezno privite? Preverjanje mogočih kratkih stikov!
Priključni kabli	Ali so priključni kabli (vhodni in izhodni signali) priključeni pravilno in učvrščeni?

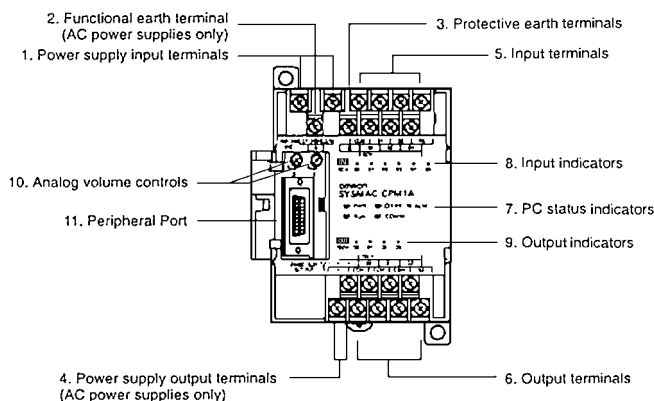
### 2.4. Procedura zagona sistema – test run

1. Napajalna aplikacija
  - a) Preverjanje napajanja PLC in priključnih sponk.
  - b) Preverjanje napajanja I/O enot in priključnih sponk.
  - c) Vklon napajanja PLC in preverjanje indikatorja POWER.
  - d) Vzpostavitev komunikacije med PLC in PC ter postavitve PLC v stanje za programiranje (MONITOR, RUN) .
  
2. Preverjanje I/O povezav
  - a) S simulacijskim programom preverimo izhodne povezave in kanale (Set in Reset) / način delovanja: Program.
  - b) Preverjanje vhodnih povezav in kanalov preko PLC vhodnih indikatorjev ali monitor operacij.
  
3. Test Run – zagon sistema
  - a) Preklop na izvajanje programa (Run) in preverjanje indikatorja RUN. V primeru napake (Error) je potrebno napako odkriti in odpraviti.
  - b) Preverjanje posameznih programskih sekvenc z uporabo simulacijskih programov v ročnem in avtomatskem režimu.
  
4. Preverjanje pravilnosti aplikacijskega programa  
Odprava vseh odkritih programskih napak.
  
5. Dokumentiranje programa
  - a) Zapis programa na disk.
  - b) Izdelava dokumentacije (listinga) programa.

### 3. Konfiguracija PLC sistema

#### 3.1. Centralna procesna enota

Primer krmilnika CPM1 / Omron



#### Statusni indikatorji krmilnika

Indikator	Status	Pomen
PWR (zelen)	ON	Vklopljeno napajanje PLC
	OFF	Napajanje PLC ni vklopljeno
RUN (zelen)	ON	PLC je v RUN ali MONITOR načinu delovanja
	OFF	PLC ni v RUN načinu ali pa je prisotna napaka
ERR/ALM (rdeča)	ON	Velika napaka (fatal), izvajanje programa se ustavi
	Utripa	Prisotna je napaka, izvajanje PLC operacije se nadaljuje
	OFF	Signalizacija normalnega delovanja
COMM (oranžn)	ON	Komunikacija – prenos podatkov preko periferne enote (serijska komunikacija)
	OFF	Ni prenosa podatkov

**Vhodni indikatorji** so aktivirani, ko je odgovarjajoči vhodni kanal aktiven (ON).

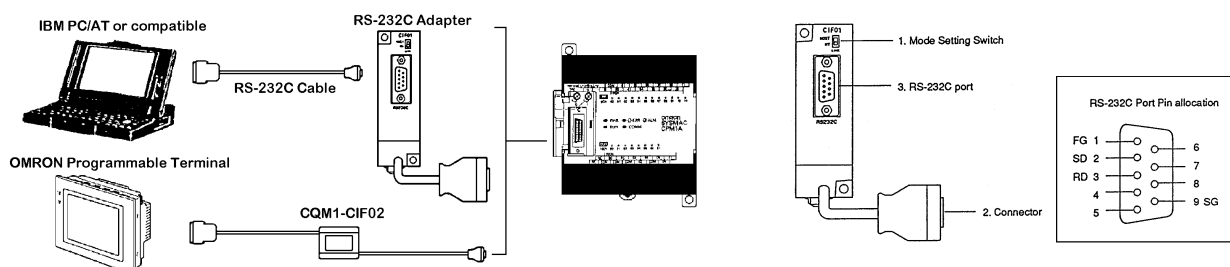
**Izhodni indikatorji** so aktivirani, ko je izhodni kanal (tranzistor, rele, ...) vklopljen.

#### Komunikacijske povezave (RS 232 ali RS 422)

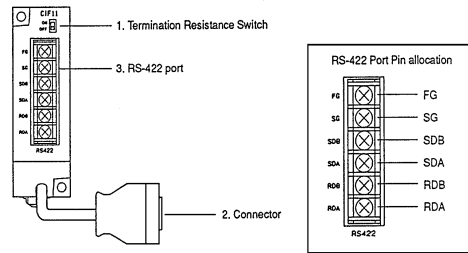
Periferni priključek je namenjen povezavi PLC s programirno napravo, računalnikom ali standardno zunanjo enoto.

RS-232C vmesnik je namenjen povezavi PLC s programirno napravo, računalnikom, programirljivim terminalom ali standardno zunanjo enoto.

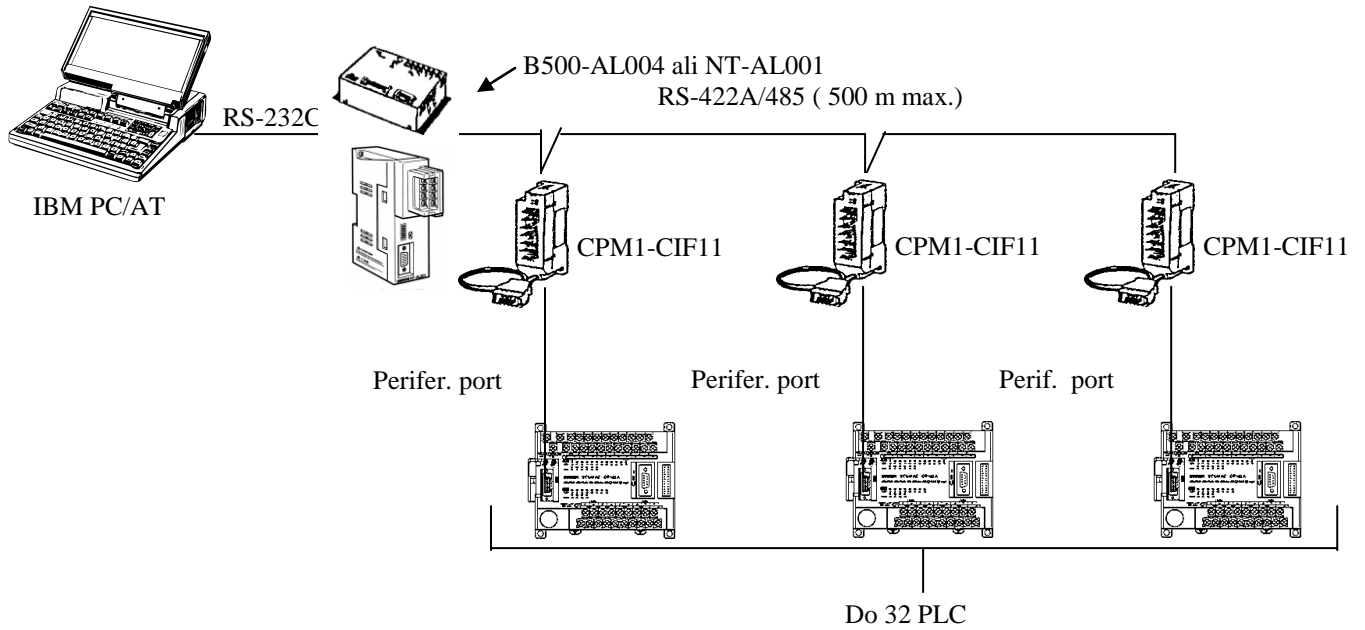
#### RS-232C vmesnik



## RS-422 vmesnik

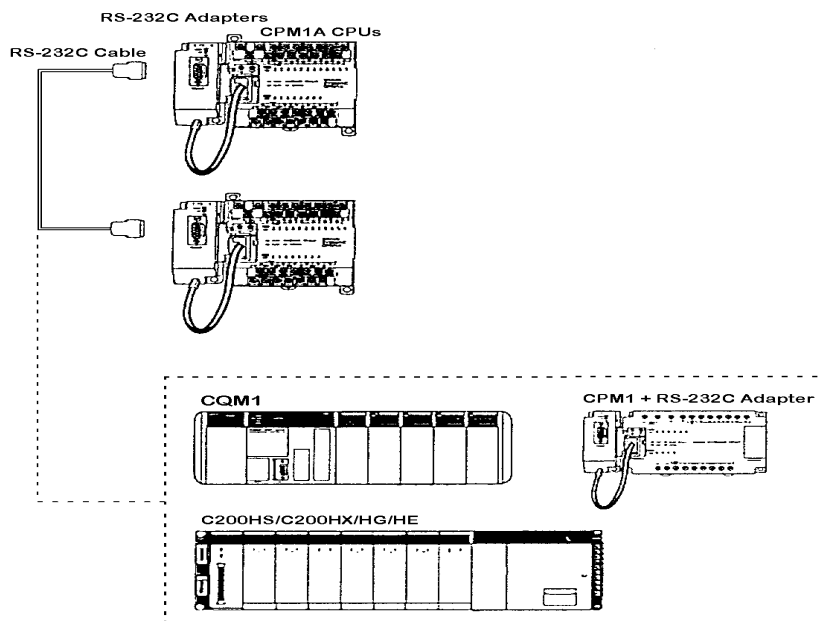


## Mreža PLC



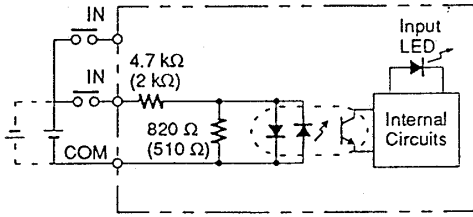
## Podatkovna povezava – komunikacija

Omogoča povezavo za prenos podatkov preko posebnih pomnilniških lokacij med več različnimi tipi PLC. S pomočjo te vrste komunikacije tvorimo PLC mrežo.



### 3.1. Digitalni vhodni in izhodni kanali

#### Specifikacija digitalnega vhodnega kanala (CPM1A)

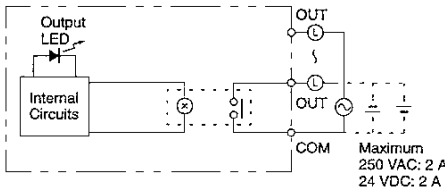
Parameter	Specifikacija
Vhodna napetost	24 VDC +10% / -15%
Vhodna irpedanca	IN00000 do IN00002: 2 k $\Omega$ ; ostali vhodi: 4.7 k $\Omega$
Vhodni tok	IN00000 do IN00002: 12 mA tipično; ostali vhodi: 5 mA tipično
ON napetost	14.4 VDC min.
OFF napetost	5.0 VDC max.
ON zakasnitev	2 ms max. / opomba
OFF zakasnitev	8 ms max. / opomba
Konfiguracija vezja	

Opomba: Vhodna časovna konstanta zakasnitve ON/OFF lahko nastavimo na 1, 2, 4, 8, 16, 32, 64, in 128 ms. V primeru uporabe vhodnih kanalov IN00, IN01 in IN02 kot hitrih števnih vhodov veljajo zakasnitve, ki so prikazane v spodnji tabeli.

Parameter	Inkrementalni način	Diferenčni fazni način (up-down)
IN00000 (A-phase)	5 KHz	2.5 KHz
IN00001 (B-phase) Normalni vhod	Normalni vhod	
IN00002 (Z-phase)	ON: 100 $\mu$ s max. ; OFF zakasnitev: 500 $\mu$ s max.	

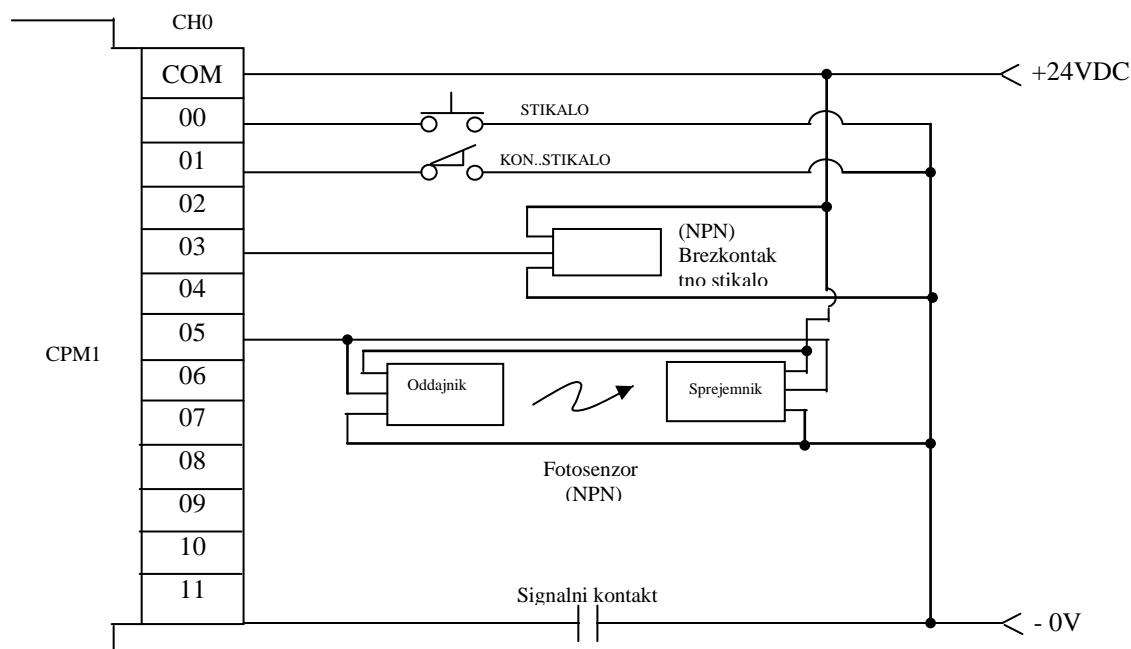
V primeru uporabe vhodnih kanalov IN 000.03 do IN 000.06 kot prekinitvenih vhodov, znaša maksimalna časovna zakasnitev 0.3 ms. Zakasnitev je merjena med pozitivno fronto vhodnega kanala in končno izvršitvijo prekinitvenega podprograma.

#### Specifikacija digitalnega izhodnega kanala (CPM1A)

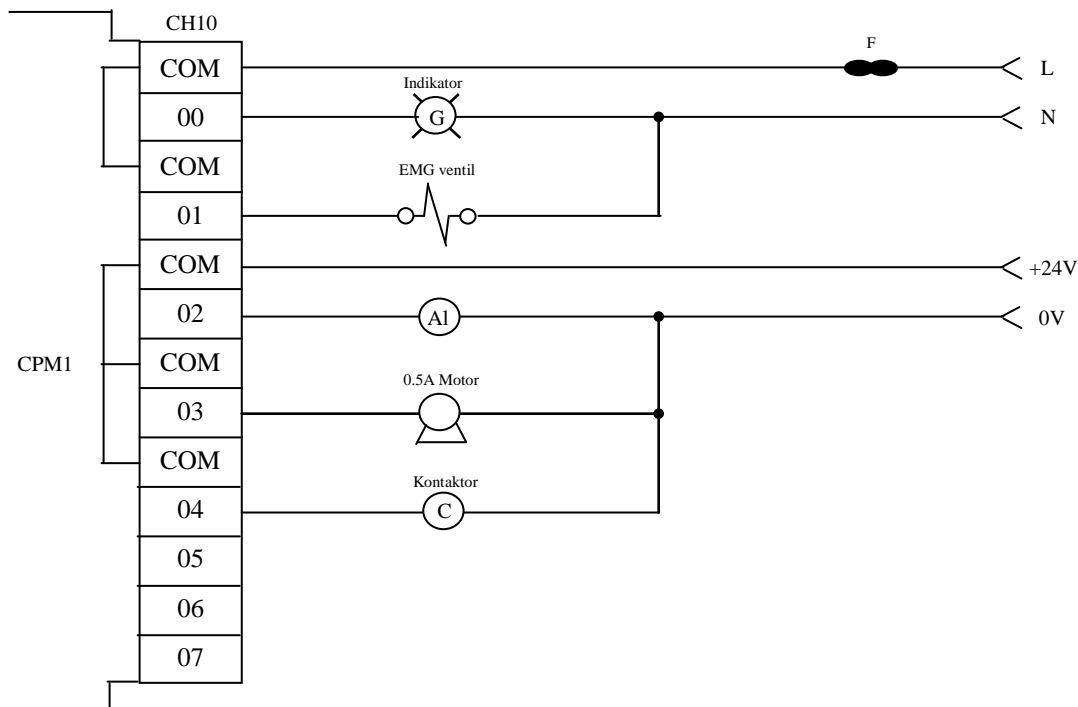
Parameter	Specifikacija
Izhodni tip	Vsi izhodi so relejski tipi (pri drugih PLC je možen še tranzistorski ali tiristorski izhod)
Max. preklopna moč	2 A, 250 VAC ( $\cos \phi = 1$ ) 2 A, 24 VDC (4 A/common)
Min. preklopna moč	10 mA, 5 VDC
Življenjska doba relejev	Električna : 300,000 operacij (uporovni porabnik); 100,000 operacij (induktivni porabnik) Mehanska : 20,000,000 operacij
ON zakasnitev	15 rns max.
OFF zakasnitev	15 ms max.
Konfiguracija vezja	

## Primeri vezave periferije in aktuatorjev na PLC

Vhodni kanali (DC24V)



Izhodni kanali

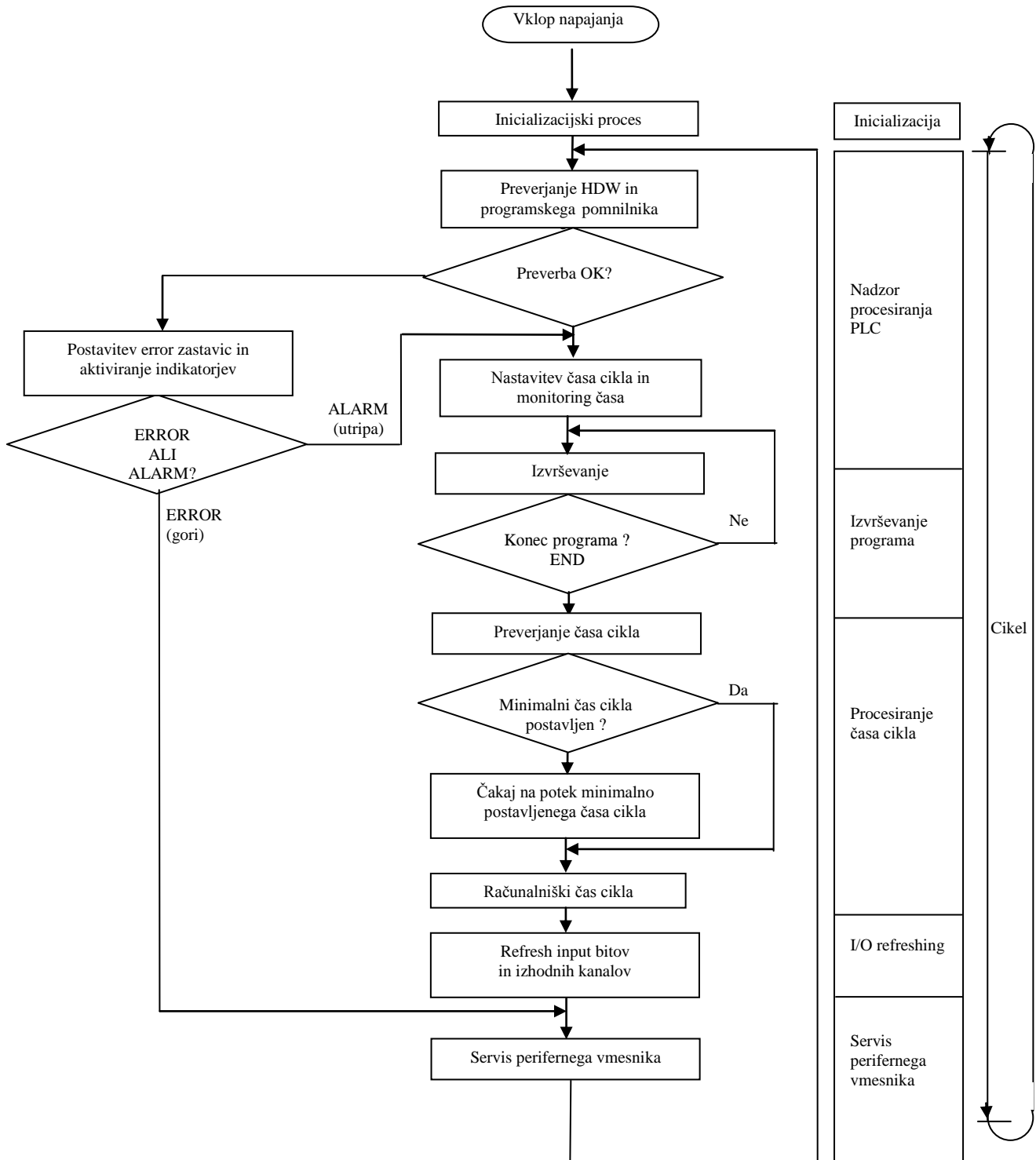


### Osnovne tehnične značilnosti CPM1

- CPM1 je kompaktni PLC z 10, 20, 30 ali 40 I/O priključki
- možnost razširitve osnovnega PLC z razširitvenimi I/O enotami s po 20 I/O točk (maksimalno 100 I/O točk)
- flash pomnilnik omogoča ohranitev podatkov brez baterije
- možnost 4 prekinitvenih (interrupt) vhodov, ki jih lahko uporabimo kot hitre števne vhode
- možnost uporabe hitrega intervalnega časovnika (nastavitve od 0,5 ms do 319968 ms)
- možnost uporabe hitrega števca, ki ga lahko formiramo v inkrementalni ali up/down obliki
- možnost uporabe 2 analognih ročnih enot (zunanje nastavljanje vrednosti časovnika ali števca)
- povezava CPM1 z PC ali programirnim terminalom (host link povezava)
- možnost uporabe vmesnika RS-232C za komunikacijo 1-1 in vmesnika RS-422 za komunikacijo 1 - n (mreža)

## 5. Cikel delovanja PLC

Ob vklopu napajalne napetosti se PLC interno inicializira. V primeru, da sistem ne odkrije nobenih napak, se začnejo ciklično odvijati procedura nadzora, aplikacijskega programa, preverjanje I/O kanalov (refresh) in strežba perifernih enot.





## Samo-diagnostične funkcije

PLC je opremljen z nekaj samo-diagnostičnimi funkcijami, ki omogočajo spoznavanje in korekcijo napak, ki vplivajo na izvajanje cikla PLC. PLC napake so na osnovi resnosti napake razdeljene v dve skupini: napake (non-fatal errors) in velike napake (fatal errors).

### Napake PLC (non-fatal errors)

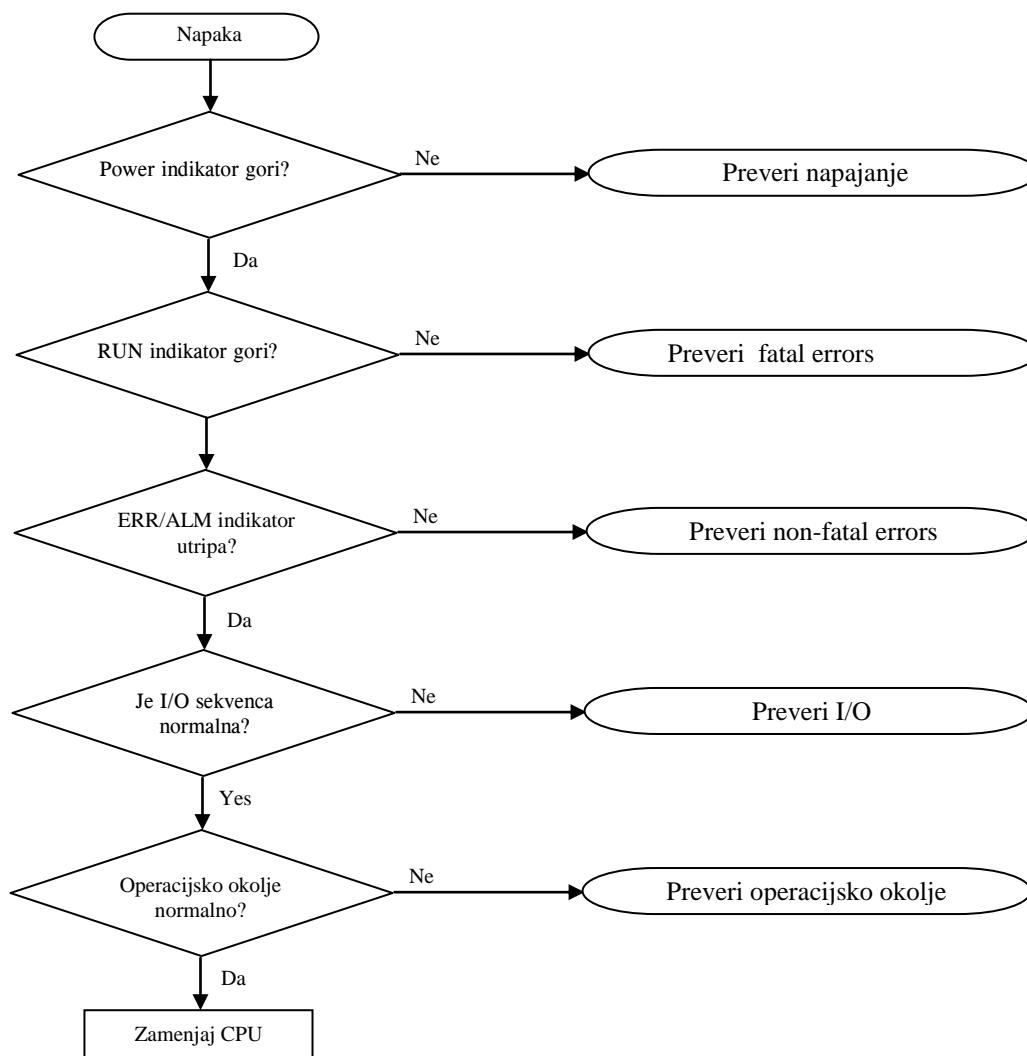
Pri pojavu takšne napake bo PLC deloval naprej, izvajanje aplikacijskega programa se bo nadaljevalo, indikatorja POWER in RUN gorita naprej. Utripati prične indikator ERR/ALM. Na programirni konzoli se lahko izpišejo sporočila kot npr.: SYS FAIL FAL ali SCAN TIME OVER. Odkriti je potrebno vzrok in napako odpraviti.

### Velike napake PLC (fatal errors)

V primeru velike napake se aktivnost PLC in izvajanje programa prekine, izhodni kanali so neaktivni. PLC lahko restartamo samo z izklopom in ponovnim vklopom ali pa preko periferne programirne naprave (konzola ali PC). Napako moramo nemudoma odkriti in popraviti. Napake so lahko systemske (SW in HDW), uporabniške (napačen zapis parametra) ali programerske. Ob pojavu velike napake lahko PLC posreduje naslednja sporočila:

- MEMORY ERR
- NO END INST
- I/O BUS ERR
- I/O UNIT OVER
- SYS FAIL FALS\*\*

## Diagram poteka odkrivanja napak



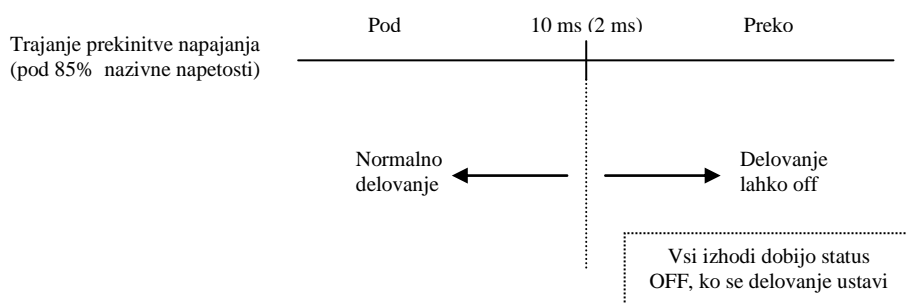
## 6. Instalacija krmilja v elektro razdelilni sistem

### 6.1. Napajanje in instalacija krmilnika

V primeru padca napajalne napetosti pod 85% nazivne vrednosti, bo PLC prenehal delovati in vsi izhodni kanali se bodo postavili na OFF.

V primeru kratke začasne prekinitve napajalne napetosti ( $t < AC - 10 \text{ ms}$  in  $DC - 2 \text{ ms}$ ) bo CPU krmilnika deloval nemoteno naprej. V primeru daljših prekinitvev bo CPU zaznal napako v napajanju in CPU bo prenehal delovati, izhodni kanali se bodo postavili na OFF. V praksi je priporočljivo, da po vsakem izpadu napajalne napetosti PLC resetiramo in ponovno izvedemo standardni zagon.

Avtomatski restart se izvede v primeru, ko naraste napetost nad 85% v predpisanem času 10 ms (2 ms).



Časovna zakasnitev začetka delovanja aplikacijskega programa od vklopa PLC znaša minimalno 300 ms.

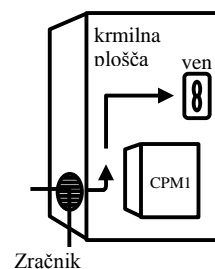
### Tehnični pogoji instalacije in montaže PLC

Krmilnika tipa PLC ne smemo instalirati oziroma izvesti montaže na mesta:

- z direktno sončno svetlobo
- kjer temperatura okolice ni v območju  $0^{\circ}\text{C}$  do  $55^{\circ}\text{C}$
- kjer relativna vlaga ni v območju 10% to 90% RH
- kjer se zaradi temperaturnih sprememb pojavlja možnost kondenza in povečane vlage
- z agresivnimi plini, tekočinami in soljo
- z večjim vplivom vode, olja ali kemijskih razpršil
- ki so ogrožena z neposrednimi vibracijami.

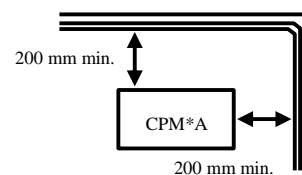
### Instalacija PLC v razdelilno omaro

V primeru neprimerne temperature v razdelilni omaro (izven območja  $0^{\circ}\text{C}$  do  $55^{\circ}\text{C}$ ) je potrebno urediti ogrevanje ali hlajenje oz. ventilacijo. V razdelilni omaro s PLC je potrebno zagotoviti dovolj prostora zaradi cirkulacije zraka in zaradi vzdrževalnih del. PLC ne smemo vgraditi nad naprave, ki proizvajajo toploto (frekvenčni pretvorniki, transformatorji, veliki upori,...). V primeru preseganja temperature v razdelilni omaro nad  $55^{\circ}\text{C}$ , je potrebno vgraditi hladilno (klimo) napravo z ventilatorjem, kot je prikazano na shemi.



### Električne motnje

Energetski vodniki in visokonapetostne naprave povzročajo motnje v PLC. Zaradi tega je potrebno pri vgradnji PLC paziti, da ga ne vgradimo v isto omaro kot visokonapetostne naprave (transformatorji, VN pretvorniki, napajalniki,...), ali vsaj ne njihovo v neposredno bližino. PLC namestimo vsaj 200 mm od močnostnih vodnikov (kanalov).

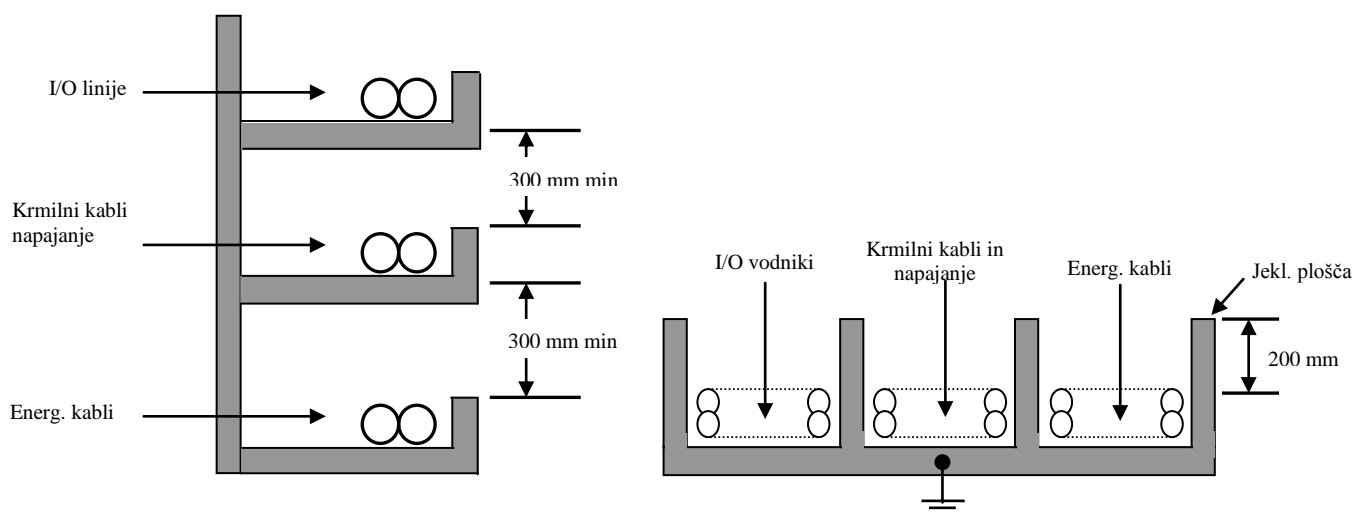
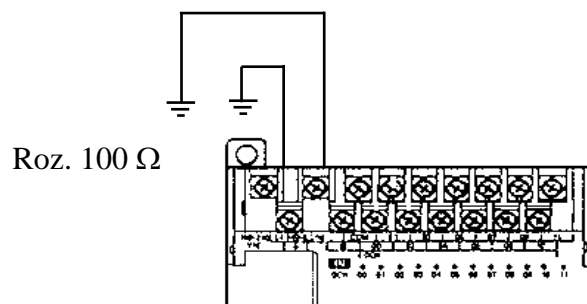


Priključek za funkcionalno ozemljitev in zaščitni priključek povežemo na ustrezne ozemljitvene točke z vodnikom minimalnega preseka  $1,5 \text{ mm}^2$

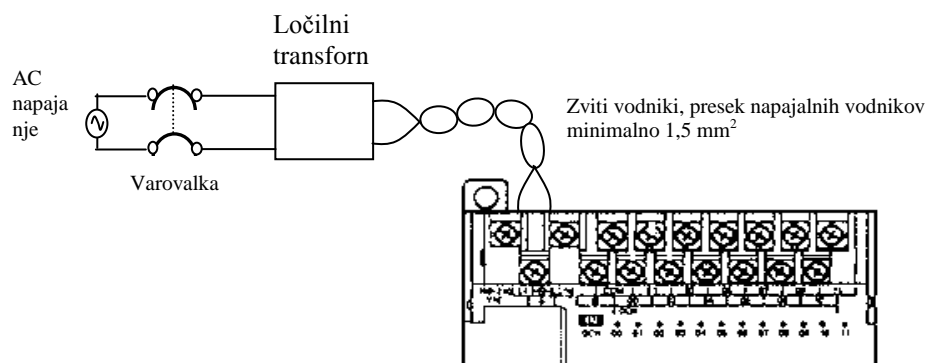
Priporočena izvedba priključka je objemna sponka za pri-vijačenje z vijakom M3.

I/O linij ne smemo voditi v istem kanalu kot napajalne ali energetske vodnike PLC. V primeru stranske vertikalne montaže kanalov, moramo med I/O ali drugimi krmilnimi vodniki in energetskimi vodniki zagotoviti vsaj 300 mm razmaka.

V primeru talnega polaganja kablov moramo zagotoviti 200 mm razmaka do vrha kanala, kot je prikazano na shemi.

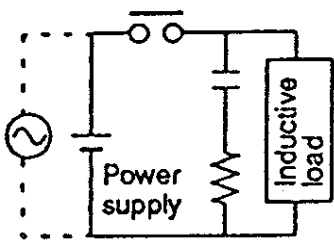
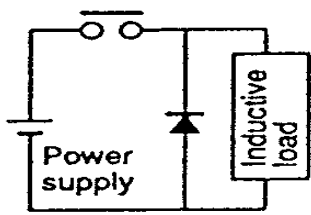
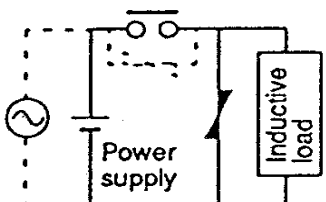


### Izvedba napajanja PLC



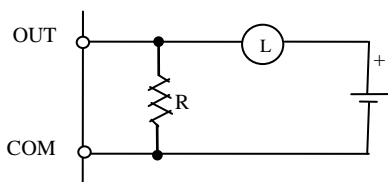
## 6.2. Vezava induktivnih bremen na izhodne kanale PLC

V primeru krmiljenja (vklapljanja) induktivnega bremena moramo vzporedno bremenu (vzbujevalna tuljavica releja, kontaktorja ali sam kontakt) priključiti zaščitni element, diodo,... Uporabo in karakteristike zaščitnih elementov kaže naslednja preglednica:

Vezje	Tok		Karakteristika	Zahtevani elementi
	AC	DC		
<p>CR metoda</p> 	Da	Da	V primeru releja kot bremena imamo opravka z zakasnitvijo med trenutkom odprtja porabniškega tokokroga in izklopom (reset) porabnika. Pri napajalni napetosti 24 V ali 48 V priključimo zaščitni RC element vzporedno navitju, pri napetosti 100 V do 200 V pa priključimo zaščitni RC element med kontakte.	Kapacitivnost kondenzatorja je 1 do $0.5\mu\text{F}$ na A kontaktnegega toka, upornost pa je 0.5 to $1\Omega$ na V kontaktne napetosti. Vrednosti so empirične in so odvisne tudi od karakteristik releja. Dielektrična zdržnost kondenzatorja mora biti 200 V do 300 V. Če je krmilno vezje AC, uporabimo nepolarizirani kondenzator.
<p>Diodna metoda</p> 	Ne	Da	Vzporedno navitju releja vezana dioda spremeni v tuljavici akumulirano energijo v tok, ki tako teče v navitje. Tok se spremeni v joulsko toploto. Zakasnilni čas med odprtjem vezja in izklopom porabnika (releja) se v primerjavi s CR metodo poveča.	Diodna reverzna dielektrična zdržnost mora biti najmanj 10 krat večja od napetosti v vezju. Tok diode je enak ali večji toku porabnika.
<p>Varistor metoda</p> 	Da	Da	Metoda prepreči prisotnost visoke napetosti med kontakti. Pri napajalni napetosti 24 V ali 48 V priključimo varistor vzporedno navitju, pri napetosti 100 V do 200 V pa priključimo zaščitni varistor med kontakte.	

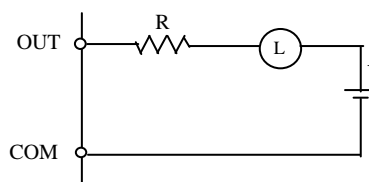
Kadar vklapljam porabnike z večjimi zagonskimi tokovi, jih zmanjšamo na naslednja dva načina:

Vezje 1



Uporaba ničelnega toka

Vezje 2



Uporaba omejevalnega upora

### 6.3 Ukrepi za odpravljanje motenj in njihovih vplivov

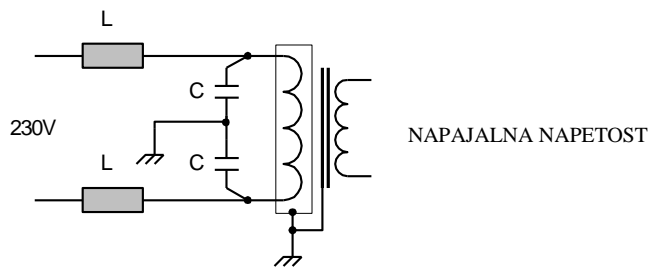
#### Motnje iz omrežja

Nastanejo zaradi stikalnih preklonov (vklop – izklop) in popačenja sinusne oblike omrežne napetosti. Pojavijo se napetostni motilni signali, ki imajo:

- širok frekvenčni in napetostni spekter
- maksimalne vrednosti amplitude 800V do 2000V
- napetostne konice (rele PR53,  $U = 24VDC$ , pri izklopu se pojavi v tuljavici napetostna konica 600V do 800V).

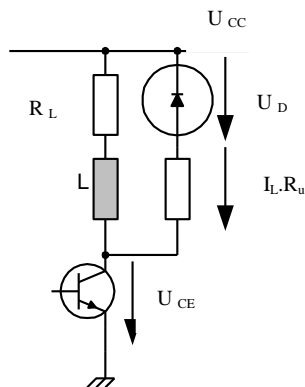
Ukrepi za odpravo motenj iz omrežja:

- ločeno napajanje elektronskega vezja (statični DC/DC in DC/AC pretvorniki)
- oklop in filtriranje na mrežnem transformatorju (LC filter dimenzioniramo za frekvenčni spekter večji od 100kHz)
- uporaba stabiliziranih usmernikov



- zmanjšanje inducirane napetosti na induktivnih porabnikih, ki nastane na vzbujevalnih tuljavah ob izklopu s čimer zaščitimo stikalne tranzistorje

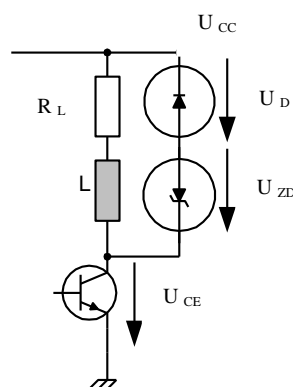
Ničelna dioda:



- povečanje odklopnih časov  
10 ms na 100 ms
- $I_{Dmax} > I_{Lmax}$
- $I_D \approx 0.5 I_{Lmax}$

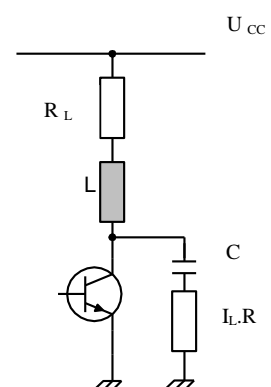
$$R_u = \frac{U_{CEmax} - U_{CC}}{I_{Lmax}}$$

Zener dioda:



- uporabno za nizke U in I
- $U_{ZD} < U_{CEmax} - U_{CC}$
- $P_{izg} \approx U_{ZD} \cdot I_{Lmax}$
- $U_i \approx U_{ZD} + U_D$

Zaščitni RC člen:



- R-C-L tvorijo serijski krog
- $R < U_{CEmax} / I_{Lmax}$
- $U_C \approx 2 \cdot U_{CC}$
- C (eksperimentalno)

### III. APLIKACIJSKO PROGRAMIRANJE KRMILNIKOV

Osnovni postopki za izdelavo aplikacijskega programa in zagon programa so sestavljeni iz naslednjih predstavljenih modulov. Programiranje lahko izvedemo v relejni shemi, mnemonični kodi ali funkcijskem diagramu. Pri izdelavi programa si lahko pomagamo s časovnimi diagrami ali algoritmi delovanja. Prenos programa omogoča posebni podprogram.

#### Programiranje v relejni shemi in mnemoničnem programskem jeziku

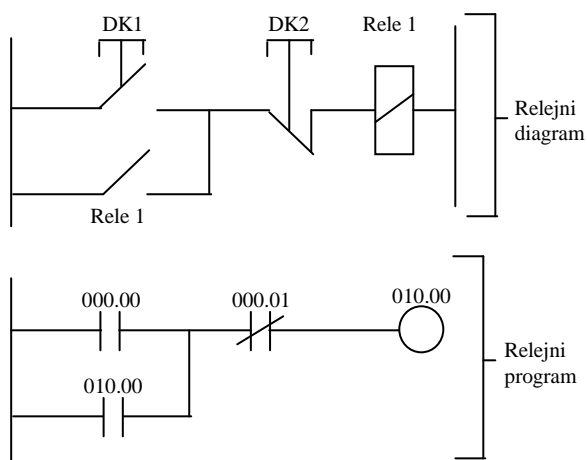
Zelo priljubljeno je pisanje aplikacijskih programov v relejnih shemah (Ladder diagram), ki zelo spominjajo na klasične relejske krmilne načrte in nam omogočajo zelo pregledno sliko nad delovanjem programa. Lahko rečemo, da gre za izvedbo krmilja, pri kateri prepišemo s pomočjo osebnega računalnika relejsko krmilno shemo v prosto programirljivi krmilnik. Vsi sodobni krmilniki imajo možnost takšnega programiranja, še zlasti pa je enostavno takšno programiranje v Windows okolju. V primeru terenskega dela oziroma servisiranja aplikacije lahko pri programiranju krmilnikov uporabimo tudi programirne konzole, s katerimi pa programiramo predvsem v mnemonični kodi. Takšno programiranje je seveda primerno za izdelavo manjših aplikacij ali pa za nastavljanje in spreminjanje parametrov (števc, časovniki,...). V nekaterih primerih je možno program napisati tudi kot klasični funkcijski načrt z uporabo logičnih elementov oz. simbolov, vendar je ta princip redkeje v uporabi.

Sam program razdelimo v posamezne vejitve (Network) in bloke (Main), ki predstavljajo določeno zaključeno celoto (npr. ročni režime delovanja). Večina programskih aplikacij ima naslednjo osnovno strukturo:

1. inicializacija krmilnika
2. postavitve osnovnih parametrov
3. zapis glavnega programa (ročno, avtomatsko delovanje,...) – pomnilniki, števc, časovniki,...
4. izvršilni del (ukazi OUT)
5. END.

#### 1. Osnovne instrukcije in programski algoritmi

Vhodne pogoje (periferija, izhodni pogoji, začasni delovni biti,...) vnašamo na levo stran diagrama, definiramo stanje vhodnega signala (NO - delovni ali NC - mirovni) in na desno stran diagrama zapišemo še izhodni signal (kanal, pomožni bit, števec,...). Program v zapisu mnemonične kode formiramo z zaporedno adresno, mnemonikom ustreznega ukaza in pripadajočim naslovom - podatkom, na katerega se ukaz nanaša.

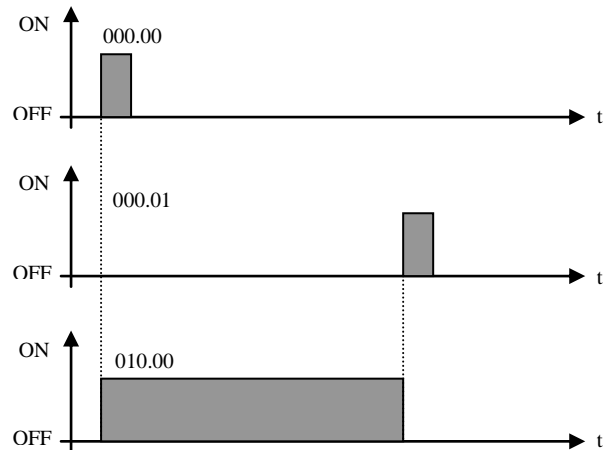
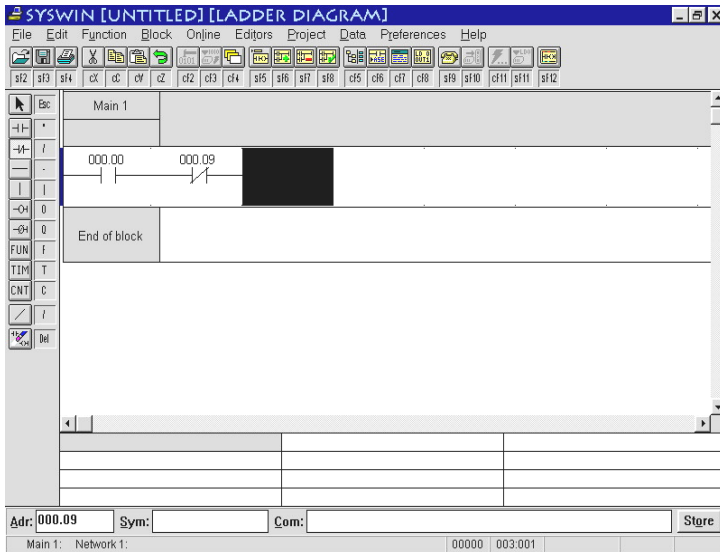


#### Mnemonična koda

Address	Instruction	Data
0000	LD	000.00
0001	OR	010.00
0002	AND NOT	000.01
0003	OUT	010.00
0004	END	
0005		
0006		
0007		

Zapis programa v operacijski sistem:

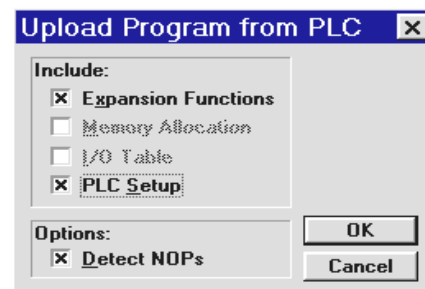
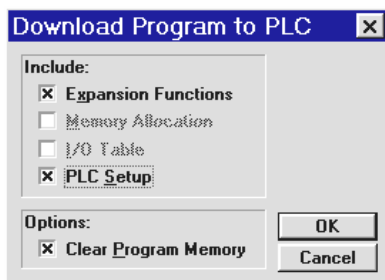
Časovni diagram delovanja:



### Nivojska struktura programa

<b>PROGRAM</b>	(PODPROGRAMI)
<b>BLOK</b>	(MAIN)
<b>POLJE</b>	(NETWORK)
<b>VHODNE IN IZHODNE SPREMENLJIVKE</b>	(VHODNO – IZHODNI KANALI, POMOŽNI BITI)

Prenos programa iz PC v krmilnik in izvajanje simulacijskih in monitor programov:



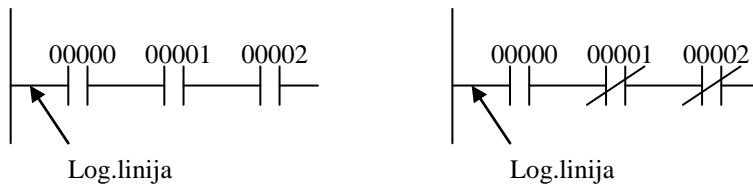
Po izdelavi aplikacijskega programa v enem izmed programskih oblik verificiramo sintaktično pravilnost programa, ga prevedemo v ustrežno kodo in prenesemo v pomnilnik krmilnika (postopek imenujemo: Download). Po uspešnem prenosu v PLC moramo program preskusiti. Običajno to počnemo s simulacijskimi 'monitoring' programi, s katerimi lahko aktiviramo določene vhodne ali izhodne signale, pomožne bite, vrednosti števcov,.... Program preskušamo po posameznih segmentih (networkih ali blokih), posebej npr. ročno delovanje in posebej avtomatsko delovanja neke naprave.

Po testiranju programa moramo aplikacijo testirati tudi v živo na napravi, pri čemer je potrebna največja pazljivost. Po izdelavi je potrebno izdelati tudi ažurirano dokumentacijo programa, tako da je možno izdelati tudi morebitne kasnejše rekonstrukcije programa.

## 1.1. Ukazi osnovnih logičnih funkcij

### Instrukcija AND in AND NOT

Relejna shema

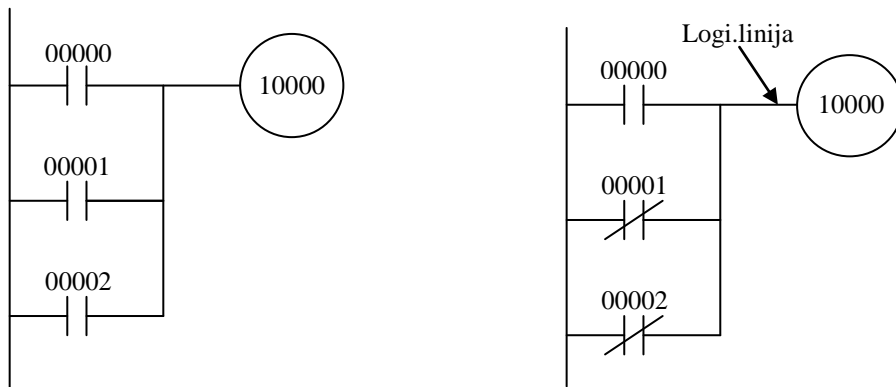


Mnemonična koda:

```
LD 000.00      LD      000.00
AND 000.01     AND NOT 000.01
AND 000.02     AND NOT 000.02
```

### Instrukcija OR in OR NOT

Relejna shema:

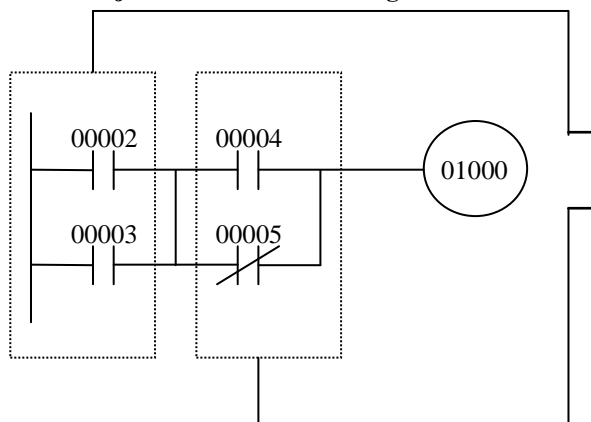


Mnemonična koda:

```
LD      000.00
OR NOT 000.01
OR NOT 000.02
OUT     100.00
```

Instrukcija AND LD - instrukcija poveže dva bloka zaporedno:

Relejna shema – Ladder diagram

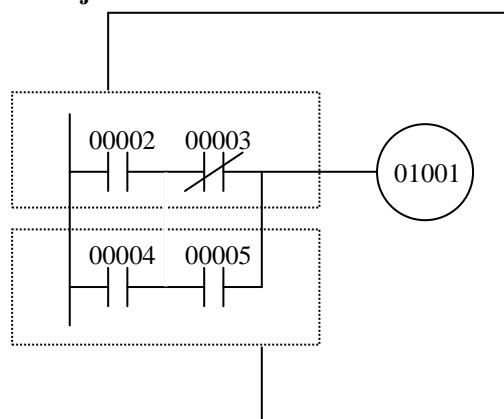


Mnemonična koda

Adresa	Ukaz	Podatek
00000	LD	00002
00001	OR	00003
00002	LD	00004
00003	OR-NOT	00005
00004	AND-LD	-----
00005	OUT	01000

Instrukcija OR LD - poveže dva bloka vzporedno:

Relejna shema

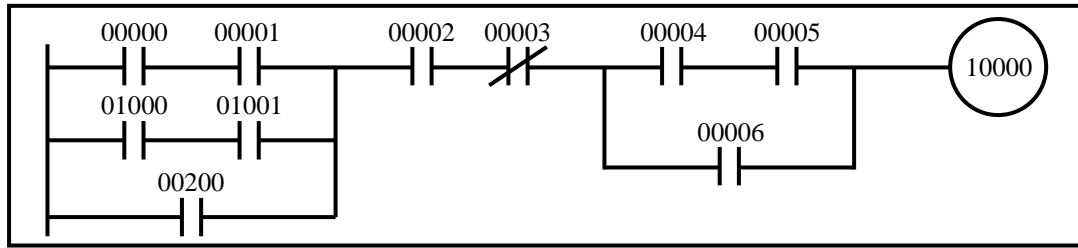


Mnemonična koda

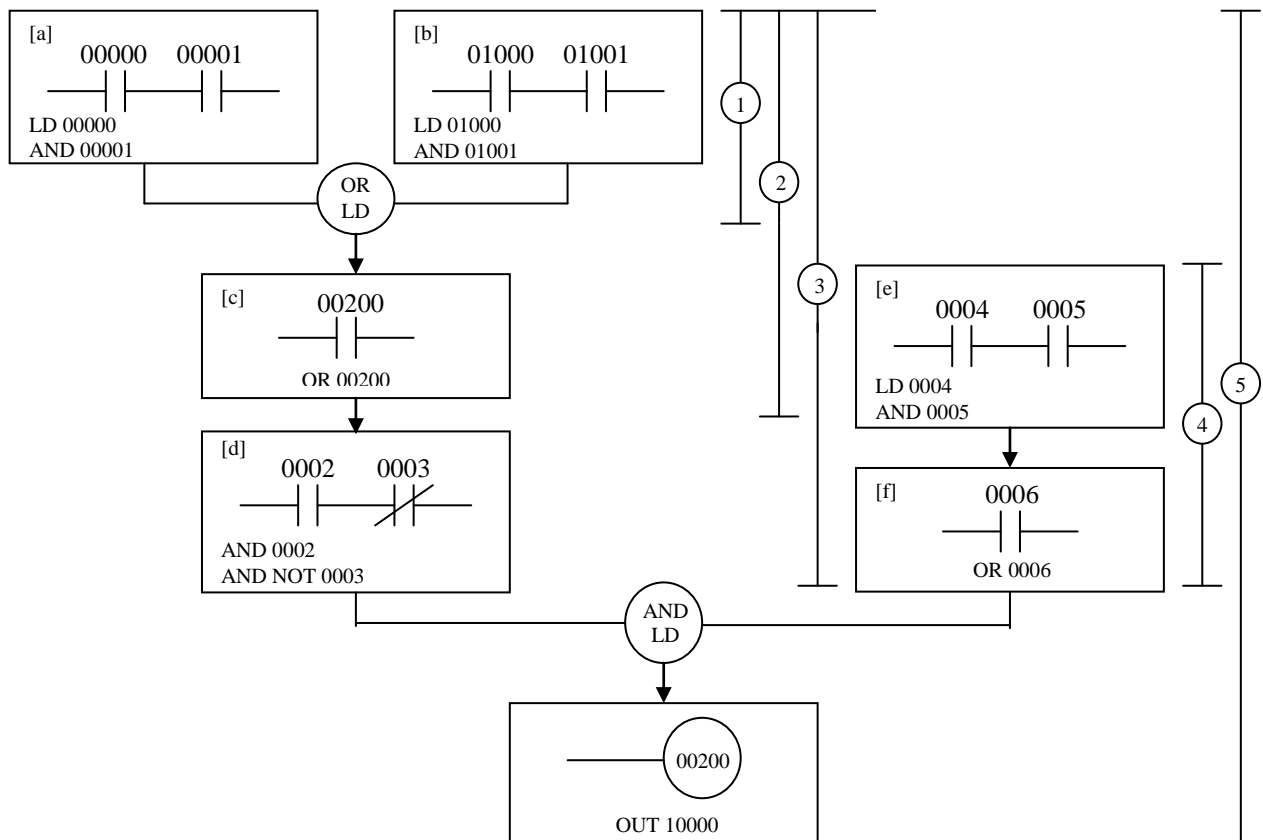
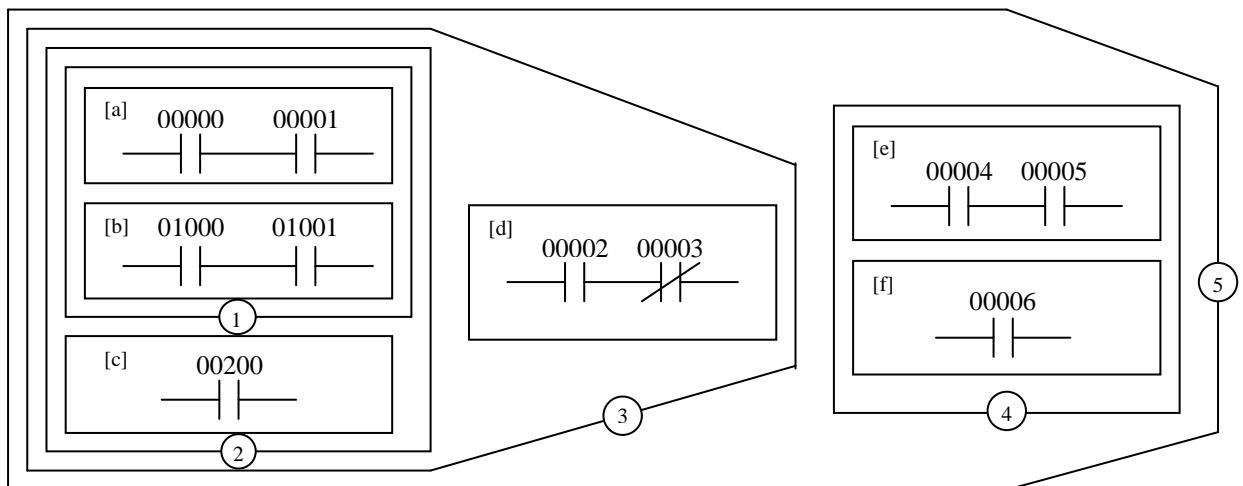
Adresa	Ukaz	Podatek
00000	LD	00002
00001	AND-NOT	00003
00002	LD	00004
00003	AND	00005
00004	OR-LD	-----
00005	OUT	01001



## Organizacija programa in programiranje v mnemonični kodi



Vezje razdelimo na manjše bloke [a] to [f]



### Primer: Zapis aplikacije enostavnega sekvenčnega krmilja

Ob pritisku na tipko **Start**, bo motor pognal voziček z leve proti desni. Ob aktiviranju končnega stikala **LS2** se motor ustavi, aktivira se časovnik, ki povzroči zakasnitev 5s . Po preteku 5s se voziček prične pomikati v levo, dokler ne doseže **LS1**. V tem trenutku ustavimo pogon, sekvenca je zaključena.

Tehnološka shema delovanja:

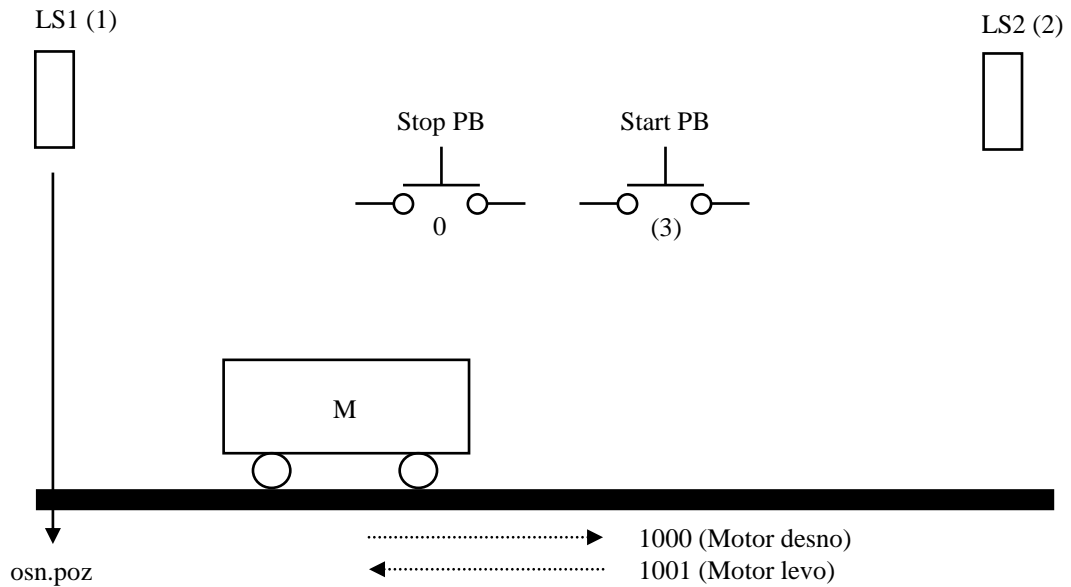


Tabela vhodnih in izhodnih signalov ( adresni plan)

Vhodi	Opis	Izhodi	Opis
00000	Stop PB	01000	Motor (desno)
00001	LS1 (osn.pozicija)	01001	Motor (levo)
00002	LS2		
00003	Start PB		

Aplikacija – relejna shema:

