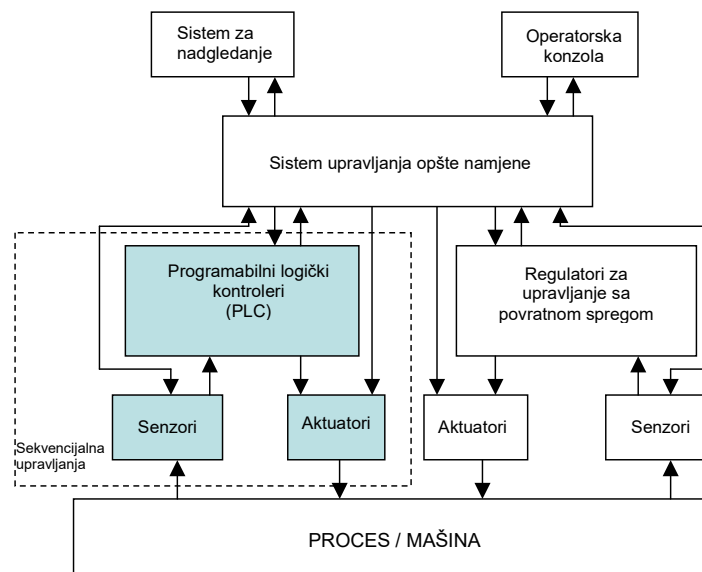


Sekvencijalni sistemi automatskog upravljanja

Kada se u praksi sriječemo sa ON/OFF funkcijama ili ON/OFF stanjima neke opreme u procesima kojima upravljamo, takav tip upravljanja se naziva logičko ili prekidačko upravljanje. Stanju ON najčešće odgovara logička 1, a stanju OFF logička 0. Jednostavnost ovakvog tipa upravljanja ga čini pogodnim za upotrebu u upravljanju automatskih mašina i procesa u kojima se zahtijeva da proces ili mašina slijede sekvencu operacija. Primjena logičkog upravljanja u sekvencama rada dovela je do termina sekvencijalno upravljanje.

Sekvencijalno upravljanje može da se realizuje pomoću elektromehaničkih releja, raznih pneumatskih i fluidnih komponenti, opreme na bazi poluprovodnika (tranzistori, mikroprcesori...) i naravno presonalnih računara. Računari specijalne namjene koji se koriste za realizaciju sekvencijalnog upravljanja se zovu programabilni logički kontroleri- PLC.

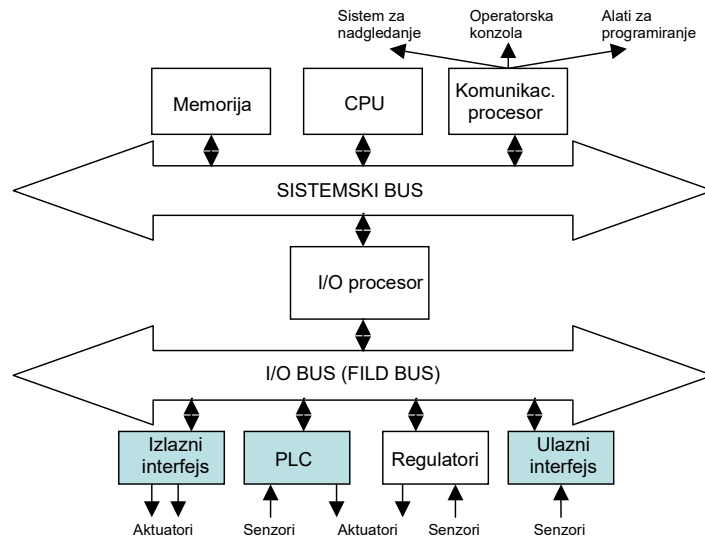
Sistem upravljanja opšte namjene



Glavna odgovornost sistema upravljanja opšte namjene je da koordinirano upravlja jednim ili više procesa ili mašina. Sistem upravljanja opšte namjene obavlja sledeće funkcije:

1. Generiše komande i prima informacije o statusu PLC i lokalnih regulatora za upravljanje u povratnoj sprezi individualnih mašina/procesa, kao i informacije o procesnim varijablama poput brzine, pozicije, temperature....
2. Da izdaje komande i prima informacije o statusu aktuatora i senzora bilo da oni opslužuju PLC ili pojedinačne regulatore za upravljanje u povratnoj sprezi
3. Prijem komandi iz i slanje status informacija u operatorsku konzolu za potrebe operatora procesa ili mašine
4. Prijem komandi iz i slanje status informacija u ručnu komandu i automatski sistem za nadgledanje rada čitavog sistema ili dijelova sistema unutar zaokruženog procesa ili grupe mašina.
5. Izmjena informacija o statusu unutar samog sistema opšte namjene između dijelova sistema i podsistema.

Tipična arhitektura sistema upravljanja opšte namjene



Memorija obezbeđuje storiranje programa i uvođenje podataka u sistem preko komunikacionog procesora.

Centralna procesorska jedinica (CPU) generiše upravljačka dejstva izvršavanjem programa i vrši koordinaciju drugih funkcija.

Ulazno-izlazni procesor (I/O procesor) obezbeđuje da CPU izdaje komande i prima informacije o statusu opreme i signala iz PLC i regulatora i razmjenjuje podatke neophodne za upravljačka dejstva preko ulaznog i izlaznog interfejsa.

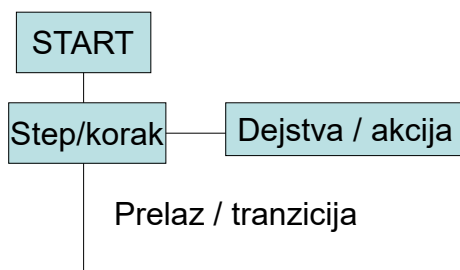
Komunikacioni procesor prikuplja komande iz procesa i obezbeđuje status informacija za sistem nadgledanja i operatorsku konzolu i omogućava interakciju sa razvojem programa i konfiguracionih alata.

Sistem predstavljen na prethodnoj slici može da obavlja kompleksne sekvence upravljačkih dejstava kada se primjenjuju na koordinaciju i upravljanje transporta materijala i rada mašina u izradi i sklapanju djelova ili u šaržnim i semišaržnim procesima, kao što su proizvodnja sirovog gvožđa ili proizvodnja lijekova u farmaceutskoj industriji.

Sekvencijalno upravljanje ima sve veći značaj u upravljanju "kontinualnih" procesa, jer oni nijesu stvarno kontinualni. Oni moraju startovati i zaustavljati se sa prethodno precizno definisanim sekvencama akcija. Primjer za ovo je startovanje termoelektrane koja radi na ugalj ili mazut, za čije je puno opterećenje potrebno vrijeme od nekoliko sati. Taj prelazni režim, ili režim upuštanja termoelektrane od stanja mirovanja do punog funkcionisanja vodi se sekvencijalnim sistemom upravljanja. Isti je slučaj kada se elektrana zaustavlja; jednostavno mora biti ispunjen veliki broj sekvenci upravljanja prema tačno definisanom redosledu i programu.

Porast značaja sekvencijalnog upravljanja doveo je do povećane potrebe za grafičkim programiranjem i tehnikama dokumentovanja velikih i kompleksnih planova sekvencijalnih upravljanja.

Međunarodna Elektrotehnička Komisija (IEC) je standardizovala sekvencijalne funkcionalne karte/dijagrame (SFC) za predstavljanje tih planova. Svaka SFC komponovana je na osnovu tri bazična elementa i to: korak, akcija i tranzicija.



Korak predstavlja tekuće stanje PLC i upravljanog sistema unutar sekvencijalnog plana upravljanja. Na svakom koraku odvija se skup **akcija**. Uslov **prelaza/tranzicije** određuje se kada stanje PLC i upravljanog sistema evoluira u sledeći korak ili korake.

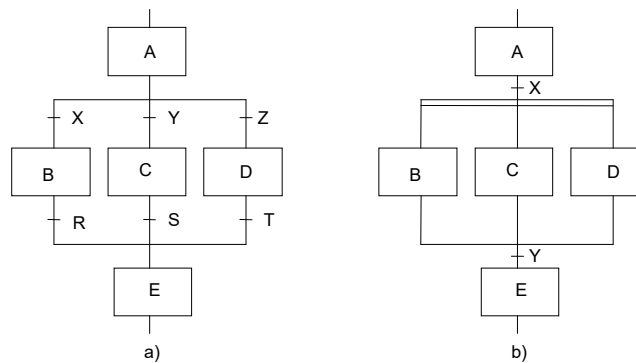
Kod SFC se razlikuju dva slučaja:

izbor-selekcija jednog broja alternativnih nasljednika na sljedeći korak bazirano na uslovima

inicijalizacija dvije ili više nezavisnih sekvenci na izvršavanje bazirano na uslovu tranzicije (paralelno odvijanje sekvence)

Svala sekvenca mora imati svoj početak, koga označavamo sa „Start“.

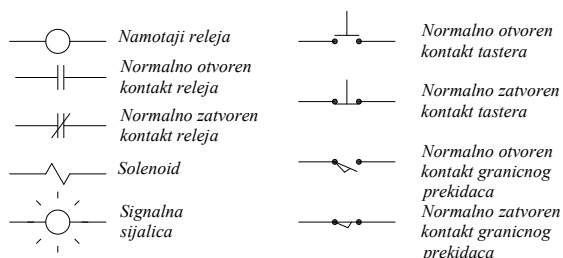
Mehanizmi selekcije i paralelnog odvijanja skvenci dati su na slici.



Sekvencijalni funkcionalni dijagrami; a) izbor koraka i konvergencija na korak E; b) Paralelne sekvence (koraci B,C i D) i konvergencija u korak E; na slici X,Y,Z su tranzicije iz jednog koraka u drugi

LEDDER DIJAGRAMI

Sekvencijalni sistemi upravljanja obično su projektovani upotrebom LADDER (lestvičastih) dijagrama, zbog široke primjene elektromehaničkih releja. Mnogi PLC imaju mogućnost programiranja sa ladder dijagramima. Ladder dijagrami uvedeni od programera, transliraju se u ekvivalentne Boolove jednačine sa PLC tako da logičke funkcije mogu biti realizovane.



Osnovne komponente koje se koriste u ladder dijagramima i od kojih se ovi sastoje su: namotaji releja, kontakti releja, ulazi kao što su granični kontakti raznih prekidača, te aktuatori kao što su namotaji i indikacione lampe, led diode, itd.

Upravljački releji. Namotaj releja sadrži pokretno jezgro i elektromagnet. Obično relej ima dva para kontakata, jedan je zatvoren a drugi otvoren, u zavisnosti od toga ima li napona na namotajima ili nema. Obično je kontakt releja normalno otvoren kada je namotaj pod naponom a normalno zatvoren kada je namotaj bez napona.

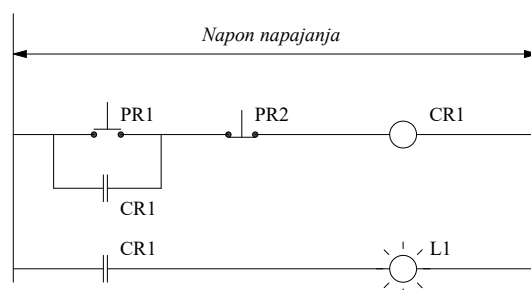
Solenoid. Solenoid je elektromagnetni aktuator. Obično se koristi za akucaciju, pokretanje pneumatskih i hidrauličkih ventila koji kontrolišu protok fluida u cilindrima.

Sijalica, led dioda. Sijalica ili led dioda često se koriste u sekvencijalnim sistemima upravljanja da indiciraju operatoru stanje mašine ili procesa.

Prekidači. Kontaktni prekidači dopuštaju da neki električni krug bude otvoren ili zatvoren. Normalno otvoreni prekidač (taster) zatvara krug između dva kraja kada se prekidač otpusti.

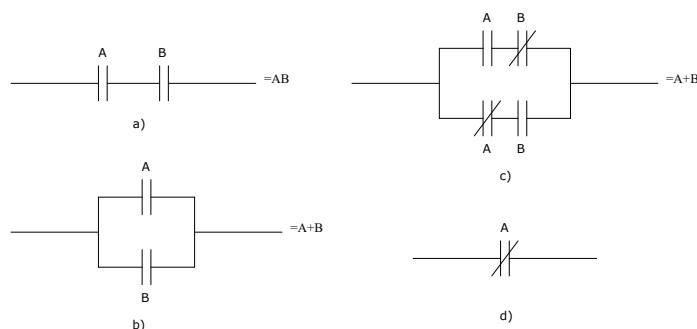
Granični prekidač. Drugi tip prekidača je granični prekidač koji je aktiviran mehaničkim kontaktom sa manipulativnom polugom. Normalno otvoren granični prekidač uspostavlja krug između dva kraja kada je prekidač aktiviran i otvara krug kada je prekidač deaktiviran.

Električni dijagrami za klasičnu relejnu logiku mogu biti konstruisani crtanjem dvaju vertikalnih linija koje predstavljaju električne provodnike (sabirnice) sa kontrolnim naponom priključenim između njih. Horizontalne linije koje se pojavljuju na dijagramu rezultat su različitih komponenti pomoću kojih se realizuje određena funkcija. Rezultat realizacije tih logičkih funkcija je Ladder (ljestvičasti) dijagram, jer podsjeća na ljestve.



*Indikatorska sijalica (L1) upravljana sa dva tastera/prekidača
(predstava pomoću ladder dijagrama)*

Logičke funkcije. Logičke funkcije koje se realizuju relejnom logikom određene su kontaktima prekidača tj. konfiguracijom kontakata u Ladder dijagramu.



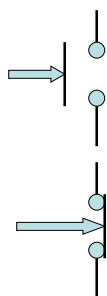
Konfiguracija kontakata za razne logičke funkcije:

a) AND; b) OR; c) EXCLUSIVE OR; d) NOT

Normalno otvoren (radni) i zatvoren (mirni)?

Tasteri

Šta je radno stanje, akcija, nenormalno stanje kod tastera? **To je pritisak na taster!**



Radni tester vodi u radnom stanju, a normalno (kada ga ne pritiskamo) je otvoren → **normalno otvoren**. Tako mu definiše i sam grafički simbol i **NO**.

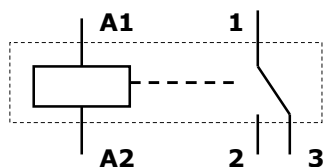
Mirni tester vodi u neradnom stanju, koje je normalno jer ga ne pritiskamo, on je → **normalno zatvoren**. Tako mu definiše i sam grafički simbol i **NC**.

Suština ovdje je da su vas pogrešno učili

- 1) da je normalno da ne radimo.**
- 2) da smo mirni samo kad ne radimo.**

Normalno otvoren (radni) i zatvoren (mirni)?

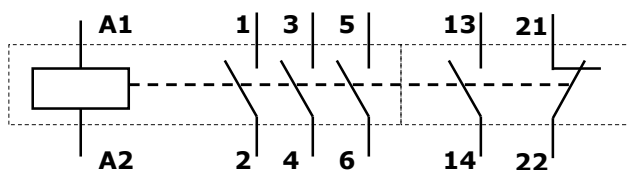
Rele



12 – radni (NO)

13 – mirni (NC)

Kontaktor



**namotaj
elmagneta
(špulna)**

**glavni
kontakti**

**pomoćni
kontakti**

Normalno otvoren (radni) i zatvoren (mirni)?

Moderni kontaktori

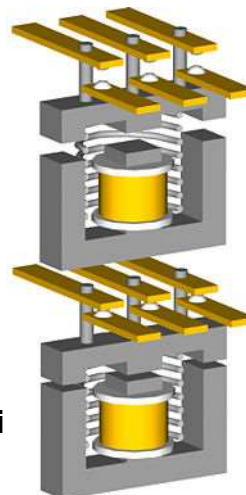


Normalno otvoren (radni) i zatvoren (mirni)?

Princip rada kontaktora

Nema napona na namotaju, el. magnet ne radi, nema privlačenja kotve, kontakti razdvojeni.

Napon na namotaju ,el.magneta
Privlači se kotva, kontakti zatvoreni



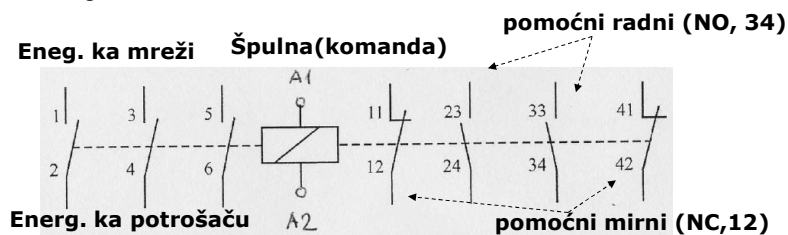
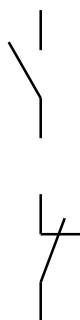
Normalno otvoren (radni) i zatvoren (mirni)?

Kontakti releja ili kontaktora

Šta je radno stanje, akcija, nenormalno stanje kod releja/kontaktora? **To je napon na špulni !**

Radni kontakt vodi u radnom stanju, a normalno (kada nema komande) je otvoren → **normalno otvoren**. Tako mu definiše i sam grafički simbol i **NO**.

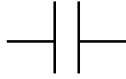
Mirni kontakt vodi u neradnom stanju, koje je normalno jer kad nema napona, on je → **normalno zatvoren**. Tako mu definiše i sam grafički simbol i **NC**.



Normalno otvoren (radni) i zatvoren (mirni)?

Bitovi u PLC

Šta je radno stanje, akcija, neaktivno, nenormalno stanje kod bitova ? **To je stanje 1, stanje ON !**



Radni bit vodi u radnom stanju (1), a normalno (kada nije 1) je otvoren → **normalno otvoren kontakt.**

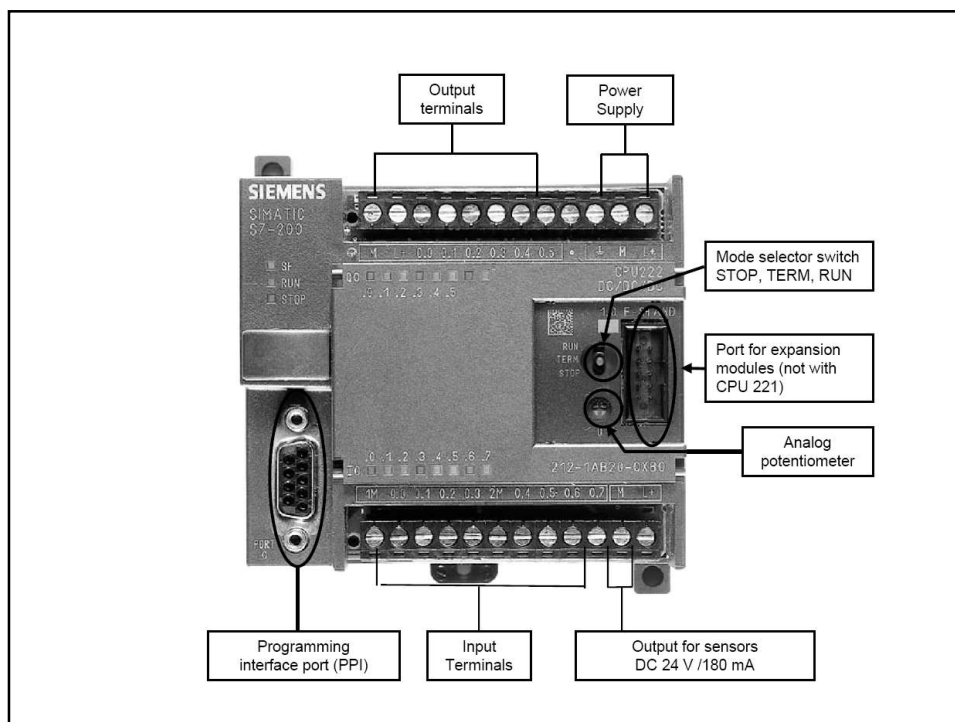
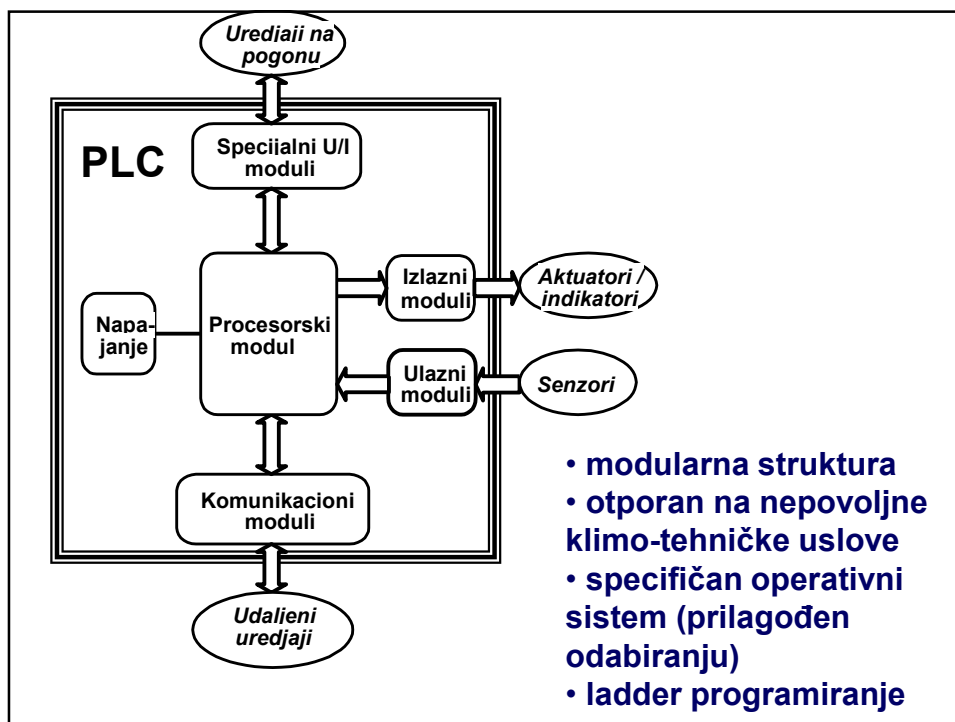


Mirni bit vodi u neradnom stanju (0), normalnom stanju on je → **normalno zatvoren kontakt.**

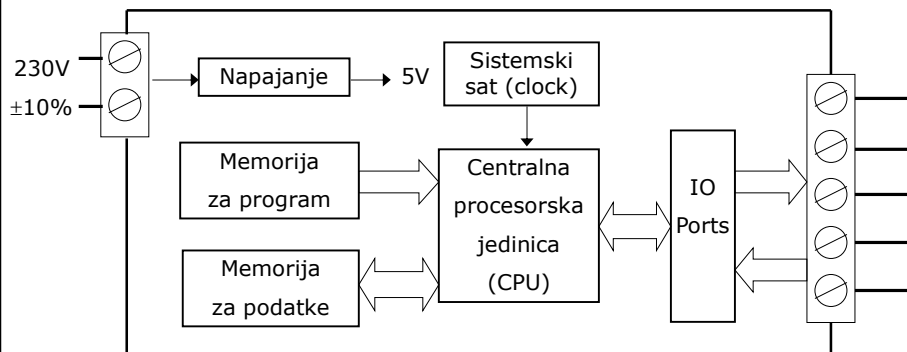
Bit može biti ulaz (na primer I0.0) koji je 1 ako je napon na ulazu, ili izlaz (na primer Q0.0) koji je 1 ako je napon na izlazu, ili čak i samo bit iz memorije (na primer M0.0) koji je 1 ako je prethodno setovan ili 0 ako je prethodno resetovan.

PLC – Programmable Logic Controller





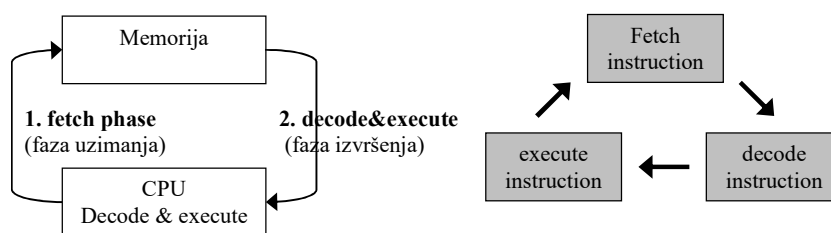
Osnovni elementi PLC sistema



µP sistem izvršava naredbe kao delove programa, jednu po jednu

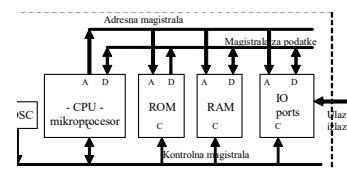
- ko čuva naredbu? – **memorija**
- ko prenosi naredbu? – **magistrala**
- ko dekoduje naredbu? – **CPU**
- ko izvršava naredbu? – **CPU**
- ko sinhronizuje sistem? – **sistemski sat** (clock ili oscilator)

Princip rada mikroprocesorskog sistema



uP sistem izvršava naredbe kao delove programa, jednu po jednu

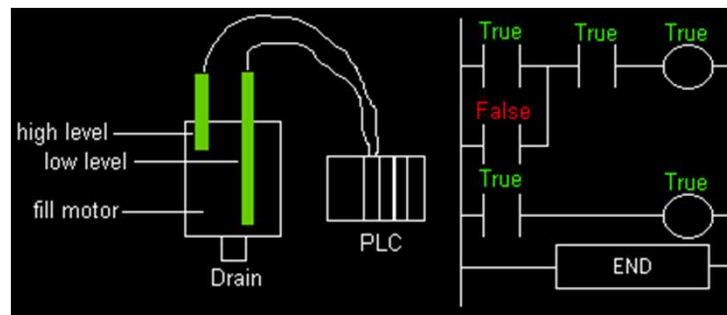
- ko čuva naredbu? – **memorija**
- ko prenosi naredbu? – **magistrala**
- ko dekoduje naredbu? – **CPU**
- ko izvršava naredbu? – **CPU**
- ko sinhronizuje sistem? – **sistemski sat** (clock ili oscilator)



Programiranje na višem nivou

Lestvičasti dijagrami – Ladder diagrams

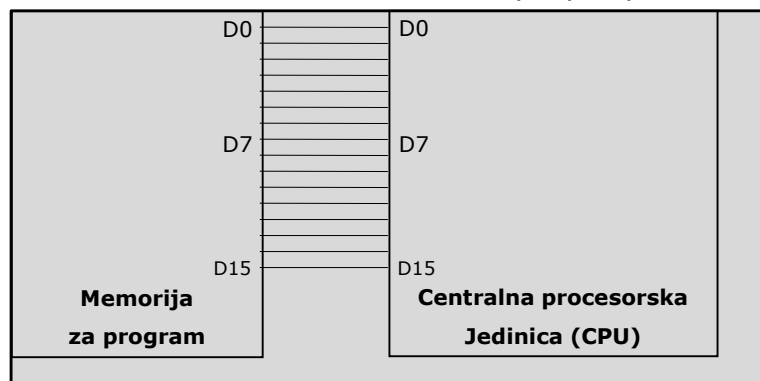
- Programski jezik višeg nivoa
- Specijalizovane šeme često korišćene da se dokumentuje kontrolna logika nekon industrijskog sistema
- Bliski relejnim šemama



Komunikacija unutra *uP* sistema

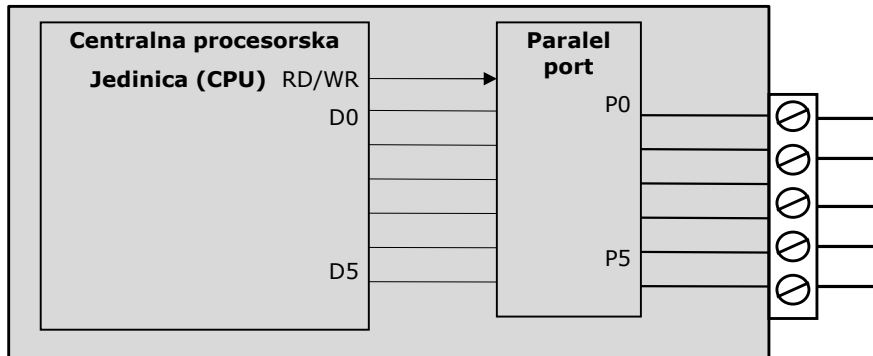
Unutar sistema podaci se prenose paralelno, svi bitovi odjednom. To je brže i efikasnije.

Postoje magistrale (skup paralelnih linija) za instrukcije, podatke i komande. Prenos može biti 8 , 16, 32 , 64 – bitni



Sl. Primer magistrale preko koje podatke razmjenjuju CPU i memorija za podatke

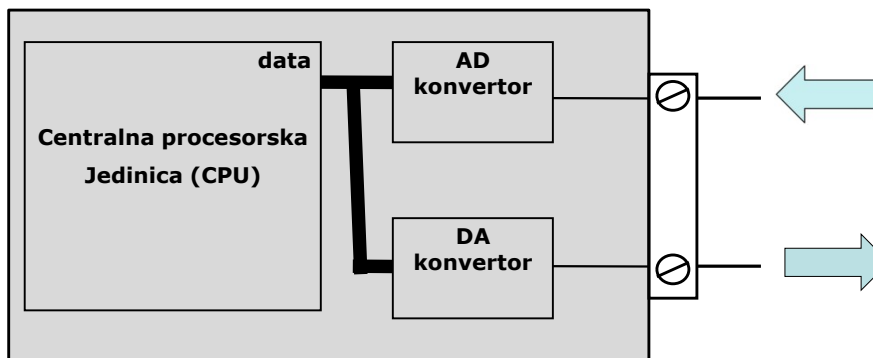
Komunikacija sa spoljnjim svijetom -paralelna



Paralelna komunikacija

- dobra je u lokalu (svi podaci odmah ulaze-izlaze)
- loša na daljinu
 - skupa (N žica, N pojačavača, N detektora nivoa ...)
 - neotporna na šum (signalna masa, greška bilo kog bita...)
 - nije lako sinhronizovati dva udaljena uređaja

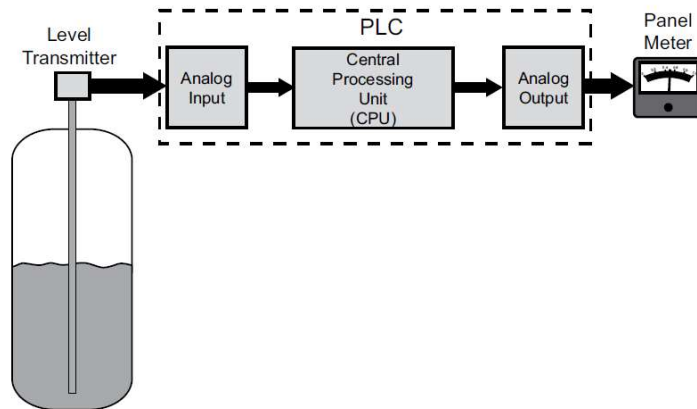
Komunikacija –paralelna ali analogni signal



Paralelni analogni prenos signala

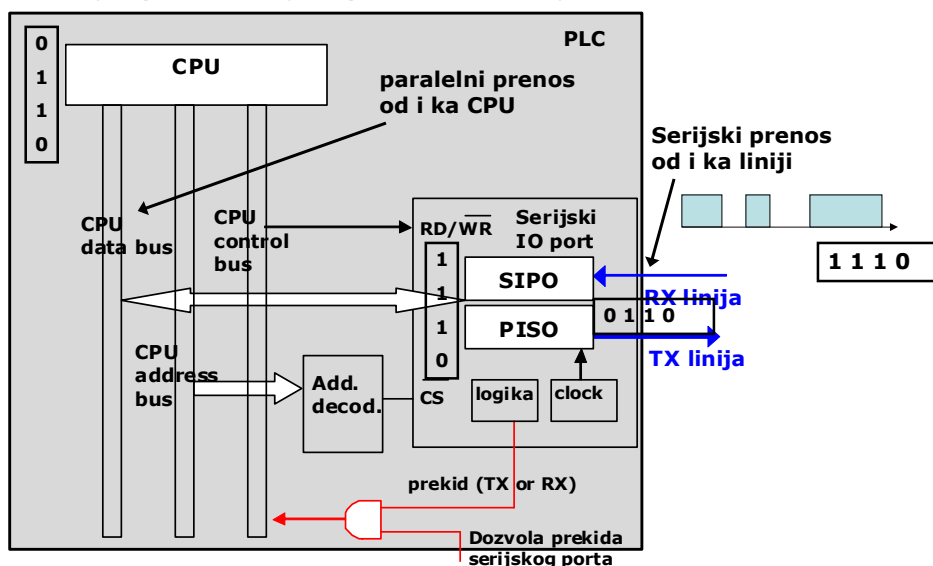
- Ulazni analogni signal (napon 0V – 12V, struja 0 – 20mA) pretvaramo u broj preko analognog/digitalnog konvertora – AD
- Na izlazu dobijamo analogni signal (napon 0V – 12V, struja 0 – 20mA) tako što broj pretvaramo u napon preko digitalnog/analognog konvertora - DA

Primjena ADC i DAC unutar PLC



Komunikacija sa spoljnim svijetom -serijska

Za prenos na daljinu. Manje žica, otporna na šum ..
Serijski port vrši serijsko/paralelnu konverziju



Šta su PLC-ovi ?

Programabilni logički kontroler (**Programmable Logic Controller**) je mikrokontrolerski sistem u kome su hardver i softver specifično adaptirani industrijskom okruženju.

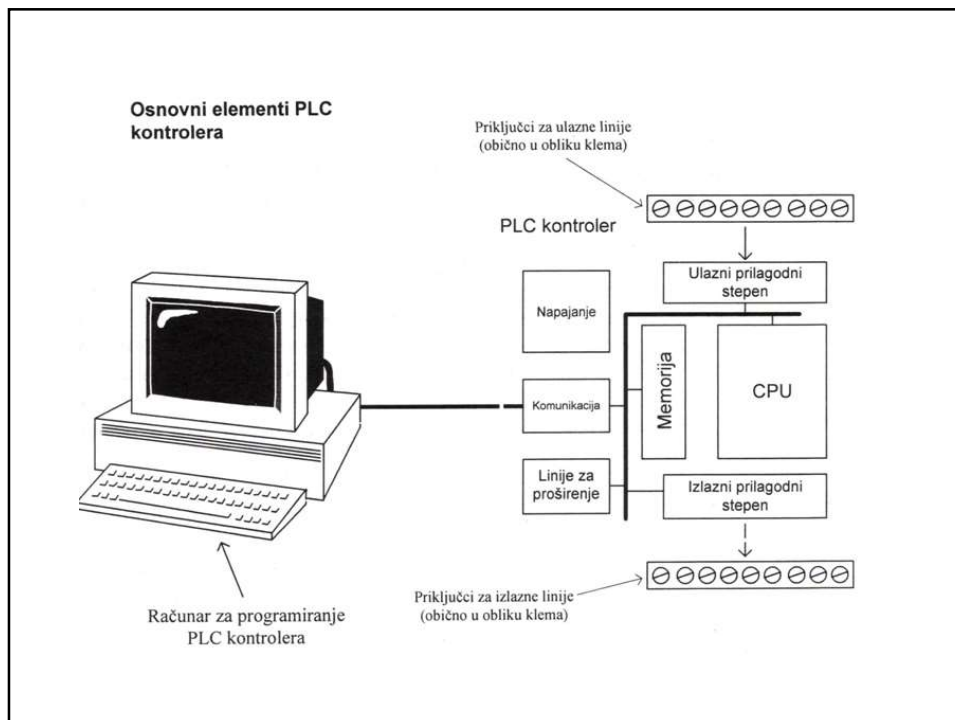
PLC je oklopljen mikrokontroler sa svakim dijelom dobro provjerenim, vrijeme razvoja aplikacije se značajno umanjuje.

PLC je originalno namijenjen kao zamena za relejne šeme. PLC se bazira na programu koji se može mijenjati tako da omogućava brze izmjene i proširenja.

Savremeni PLC kontroleri su sposobni za sve vrste komunikacija, RS485, MODBUS, PROFIBUS, Ethernet , IT (internet TCP/IP)

PLC koristi "ladder" ljestvičaste dijagrame za programiranje.

PLC ima odvojene ulaza i izlaze.



Primjena savremenog PLC

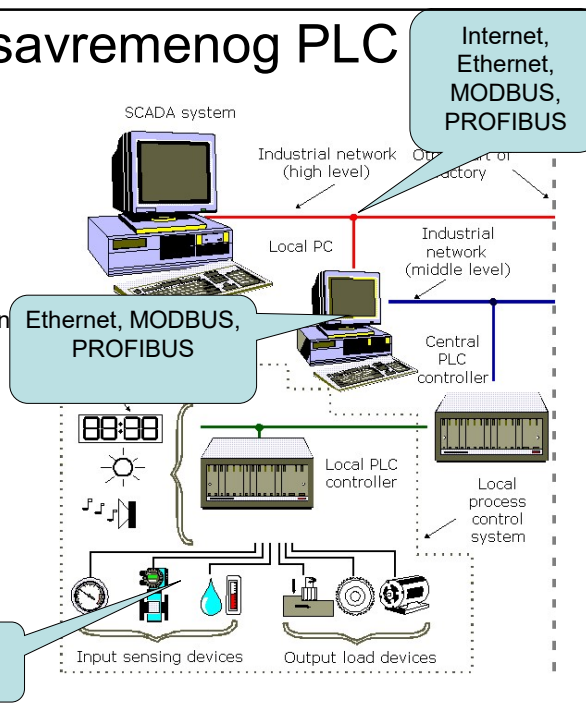
Kontrolu procesa čini čitava grupa elektronskih uređaja

Konekcija PLC kontrolera ka centralnom kompjuteru ili internom

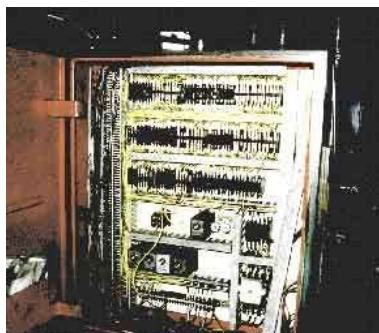
Konekcija više distribuiranih PLC između sebe

Konekcija PLC sa senzorima, Aktuatorima kao i sa raznim komandama i HMI

Analogni ulazi, digitalni ulazi/izlazi

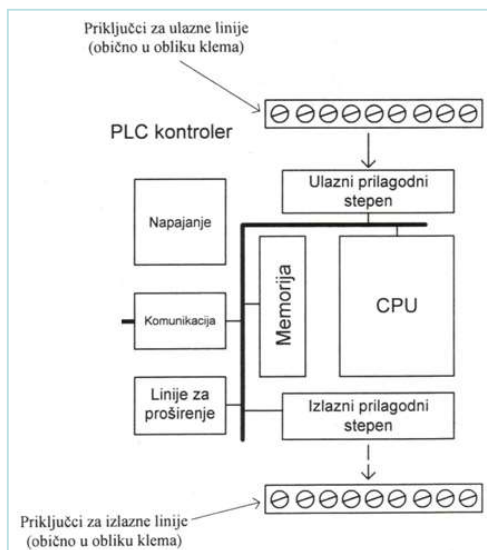
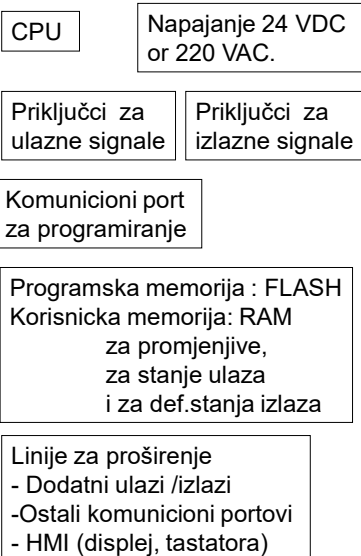


Kontrolni orman sa i bez PLC



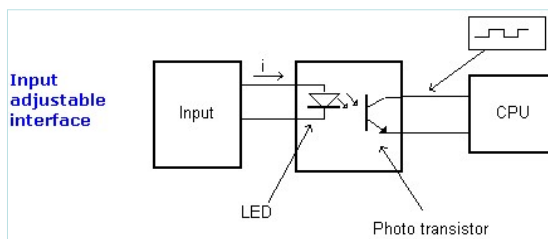
1. Broj žica umanjen za 80%
2. Umanjena potrošnja snage u odnosu na relejnu šemu
3. Dijagnostičke PLC funkcije omogućuju laku detekciju greške
4. Promena operacione sekvence cijelog sistema upravljanog sa PLC je veoma jednostavna, treba samo promijeniti i upisati novi program.
5. treba manje rezervnih delova
6. mnogo je jeftiniji
7. PLC je mnogo pouzdaniji u odnosu na elektromehaničke releje.

Osnovni elementi PLC kontrolera



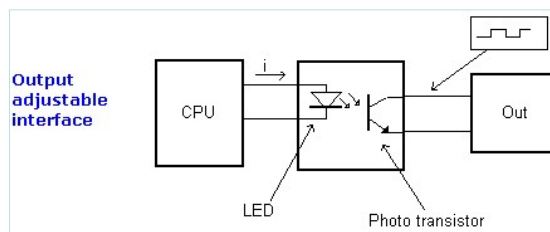
Digitalni ulazi/izlazi

Digitalni ulazi (odvojeni)



Ovo su digitalni ulazi (tasteri, prekidači, tastature) ali postoje i analogni

Digitalni izlazi (odvojeni i pojačani)



Upravlja ju kontaktorima, motorima, relejima, LED diodama

Analogni ulazi i izlazi

Analogni ulazi

Ulazi sa raznih senzora (temper, protok, pritisak..)

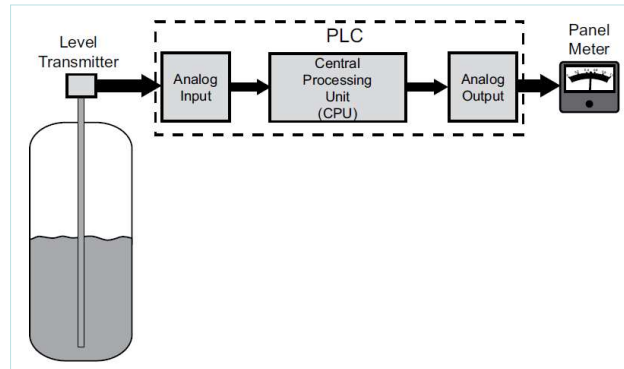
Uglavnom podržavaju dva indust. standarda

- strujni ulaz 4mA – 20mA
- naponski mV ulaz

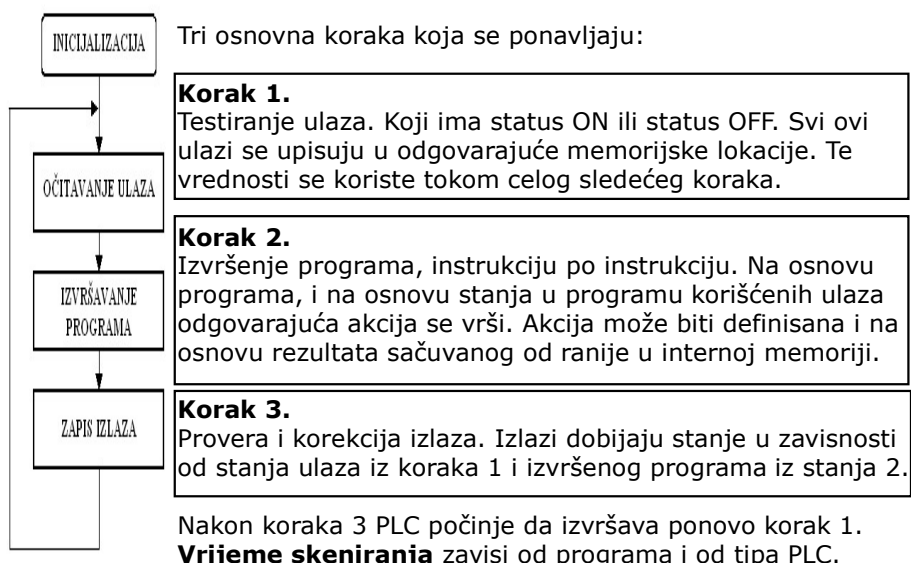
Analogni izlazi

Generišu promjenjivi analogni signal za regulaciju na primjer ugla zakretanja ventila, kazaljke ...

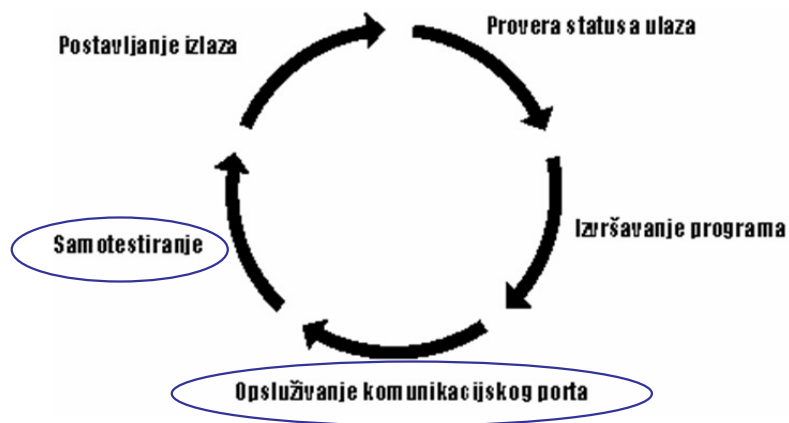
Uglavnom podržavaju ista dva indust. standarda



Osnovne tri faze izvršenja PLC programa

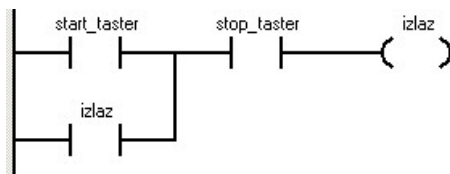


Sve faze izvršenja PLC programa



Način programiranja PLC

Ljestvičasta logika (ladder logic) je način crtanja el. logičkih šema. To je grafički jezik, veoma popular kod PLC. Originalno je izmišljen da zamijeni relejnu logiku. Ime je dobio jer program podseća na merdevine.



Alternativa

STL (statement list) - Instruction List language

```
LD start_taster
O izlaz
A stop_taster
= izlaz
```

Način programiranja PLC

Stara relejna šema
(napajanje, žice, špulne, mirni i radni kontakti releja)

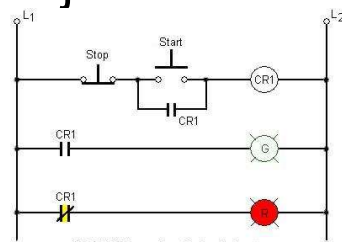
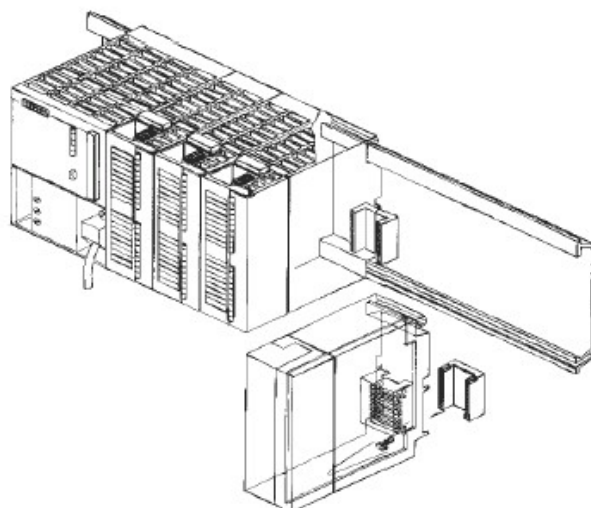
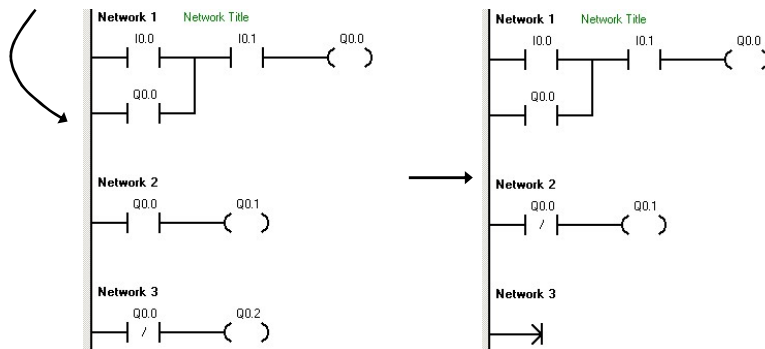


FIGURE 1 Example of relay logic diagram.

Ljestvičasti PLC programi
koji zamjenjuju ovu šemu



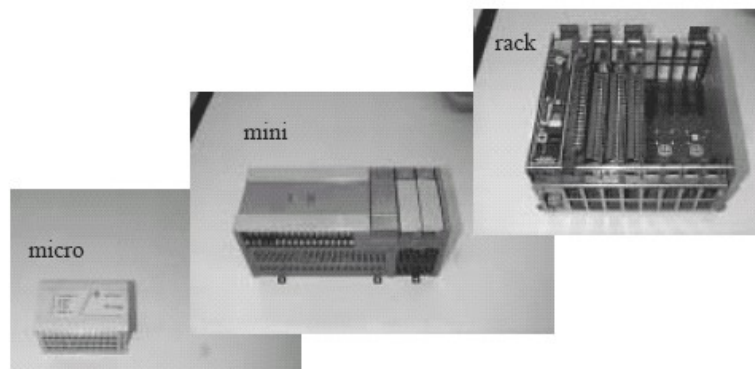
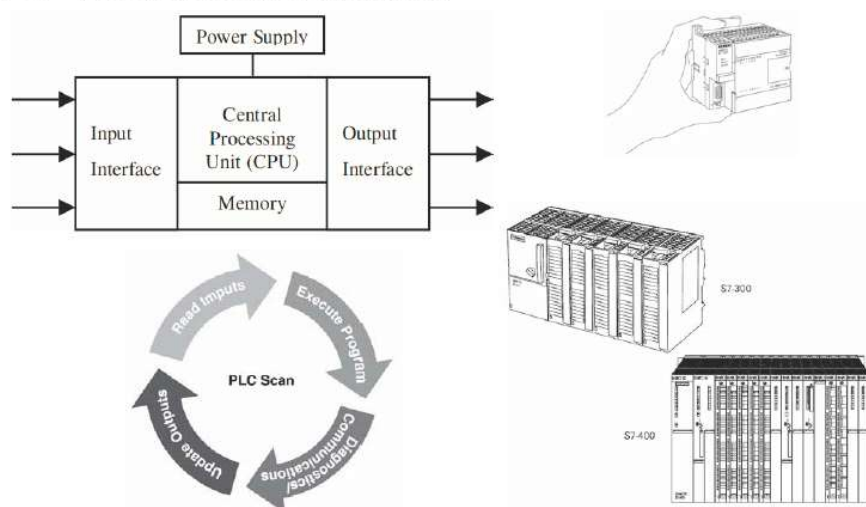
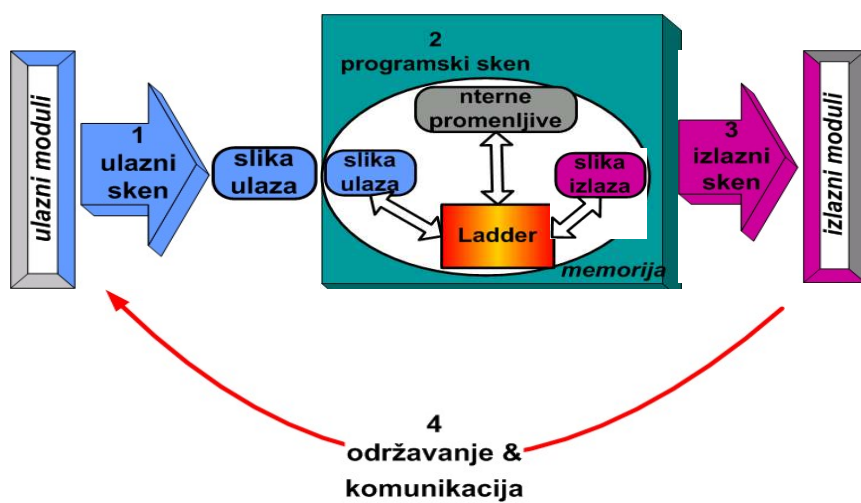


Figure 3.1 Typical Configurations for PLC

PLC – izgled, unutrašnjost i način rada



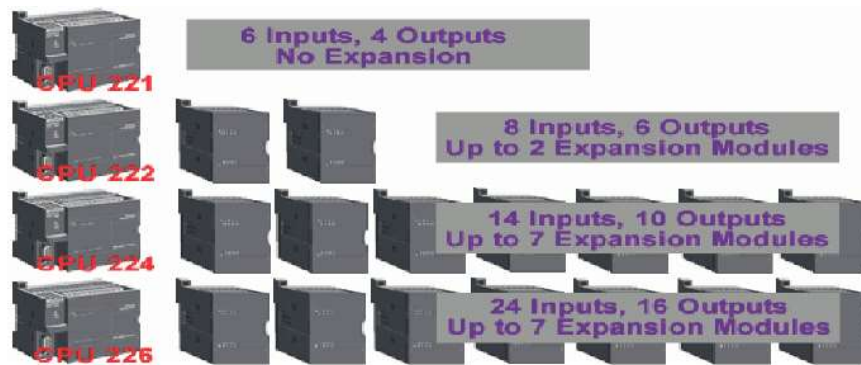
Operativni sistem



2. Komponente

Modularnost – najznačajnije svojstvo PLC-a,

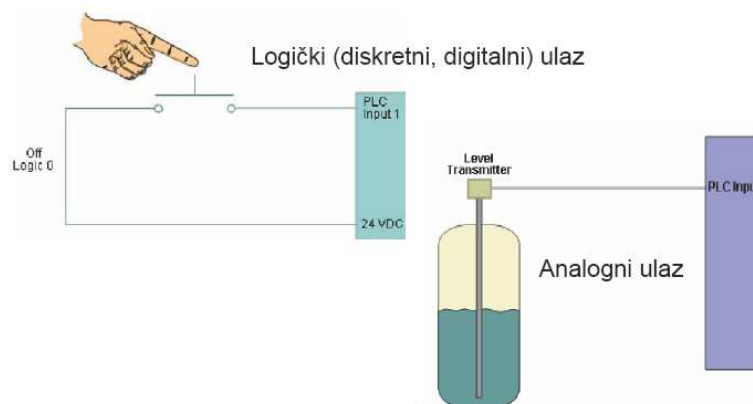
– dodavanje modula zavisi od funkcija koje treba obaviti PLC

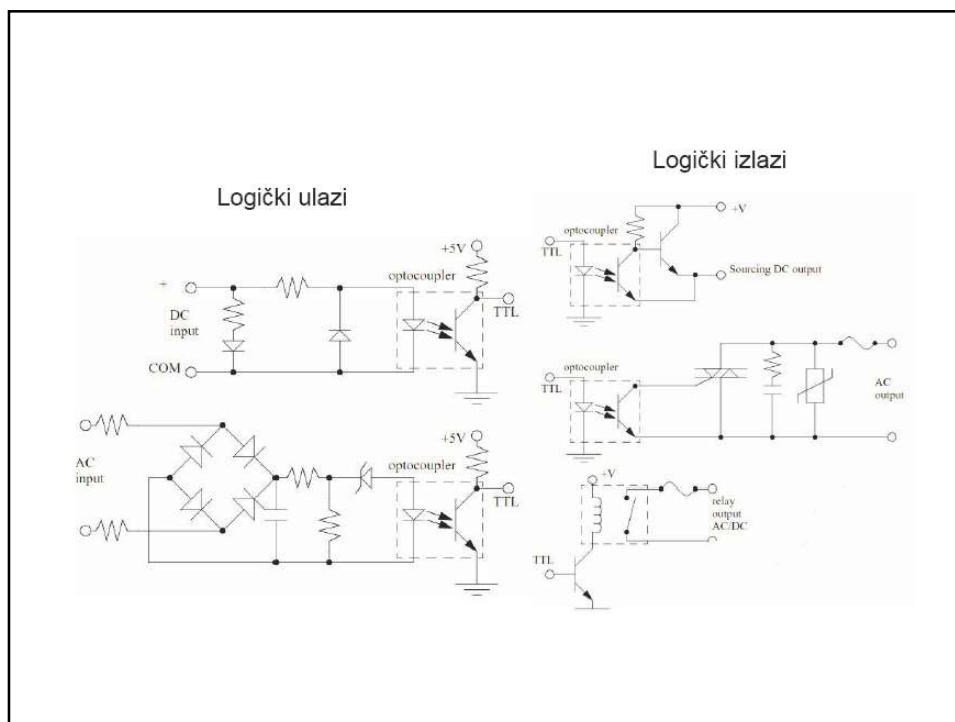
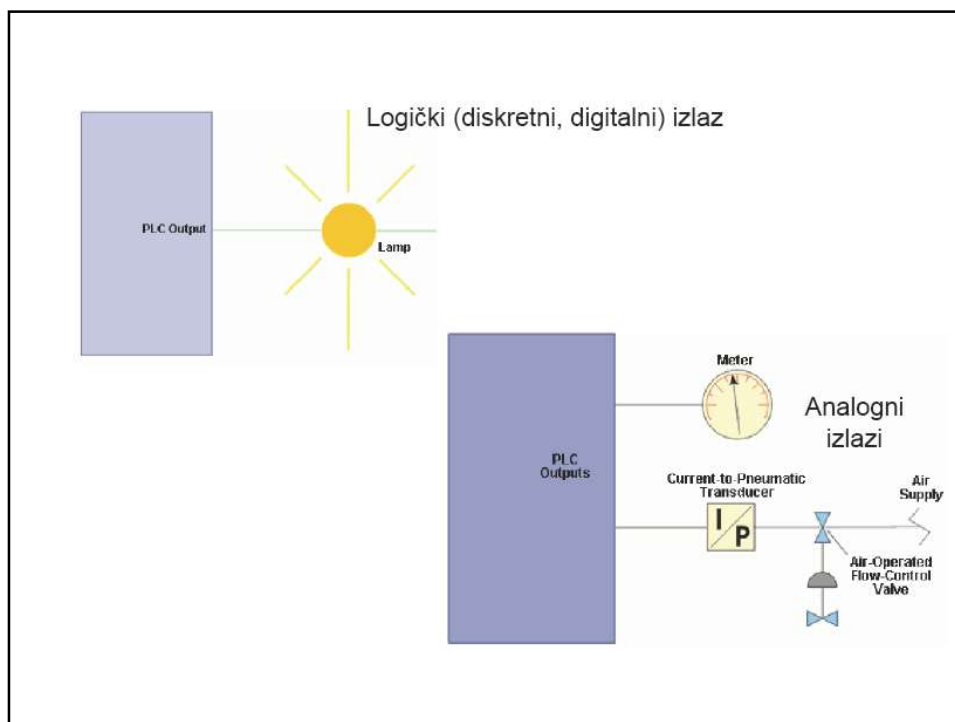


Ulazno/Izlazni (I/O) moduli – veza između PLC-a i upravljanog sistema

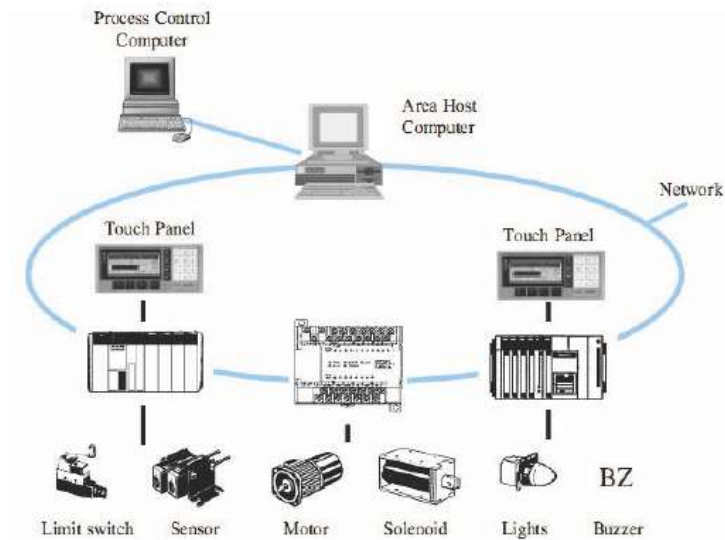
diskretni I/O – logički signali (graničnici, tasteri, sklopke, ...)

kontinualni I/O – analogni signali (mjerni pretvarači, senzori, ...)

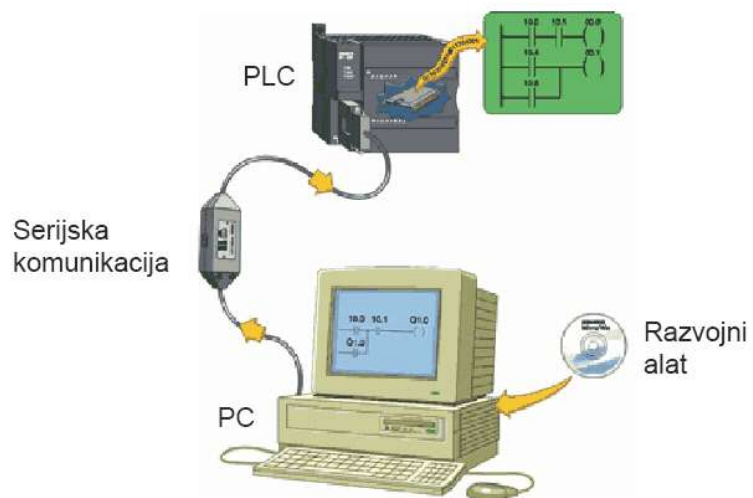




Komunikacioni (COM) moduli – veza između upravljačkih komponenti sistema (PLC <-> PLC, PLC <-> PC)

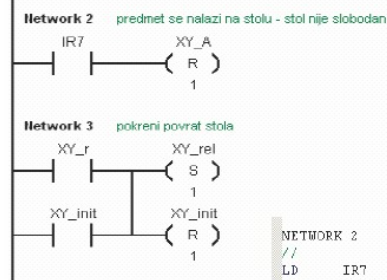


3. Programiranje

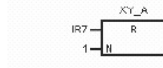


Osnovni razvojni jezici

Ladder diagram

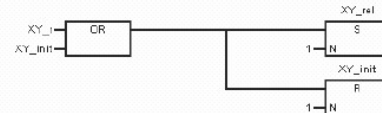


Network 2 predmet se nalazi na stolu - stol nije slobodan



FBD – function block diagram

Network 3 pokreni povrat stola



```

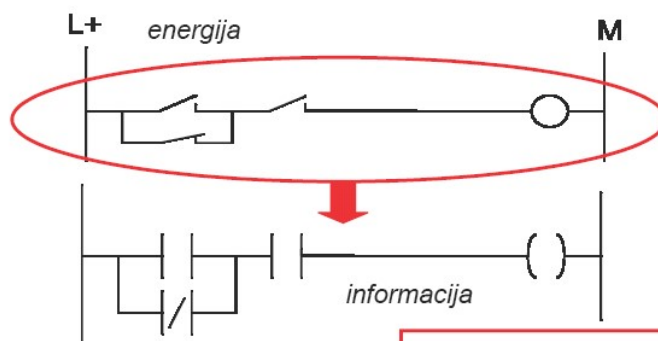
NETWORK 2      //predmet se nalazi na stolu - stol nije slobodan
//
LD      IR7
R      XY_A, 1

NETWORK 3      //pokreni povrat stola
//
LD      XY_r
O      XY_init
S      XY_rel, 1
R      XY_init, 1
    
```

STL – statement list

Ladder dijagram

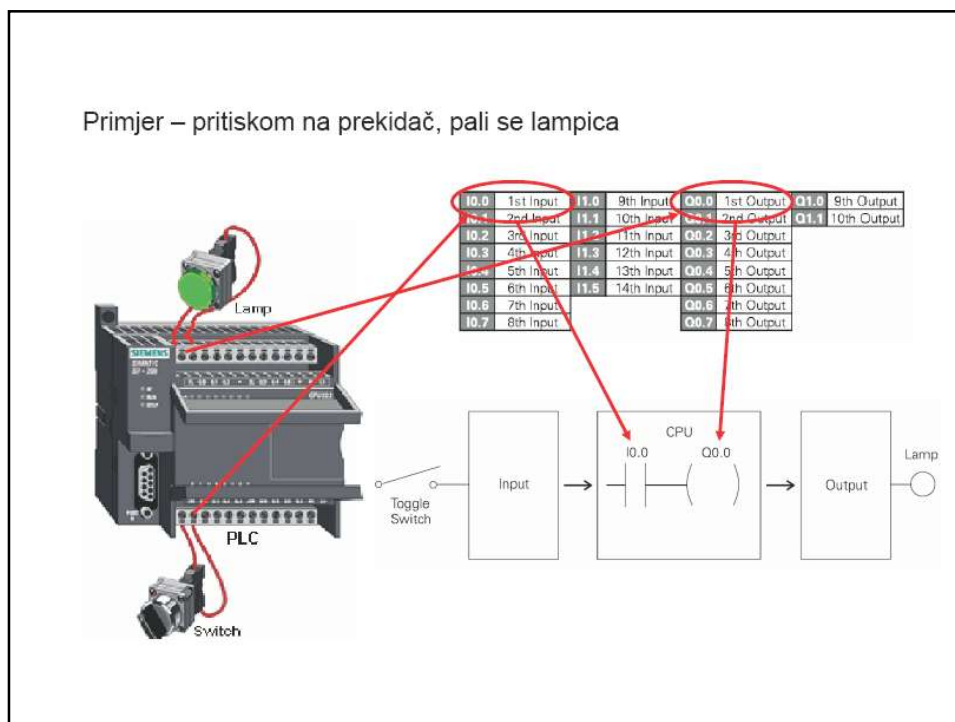
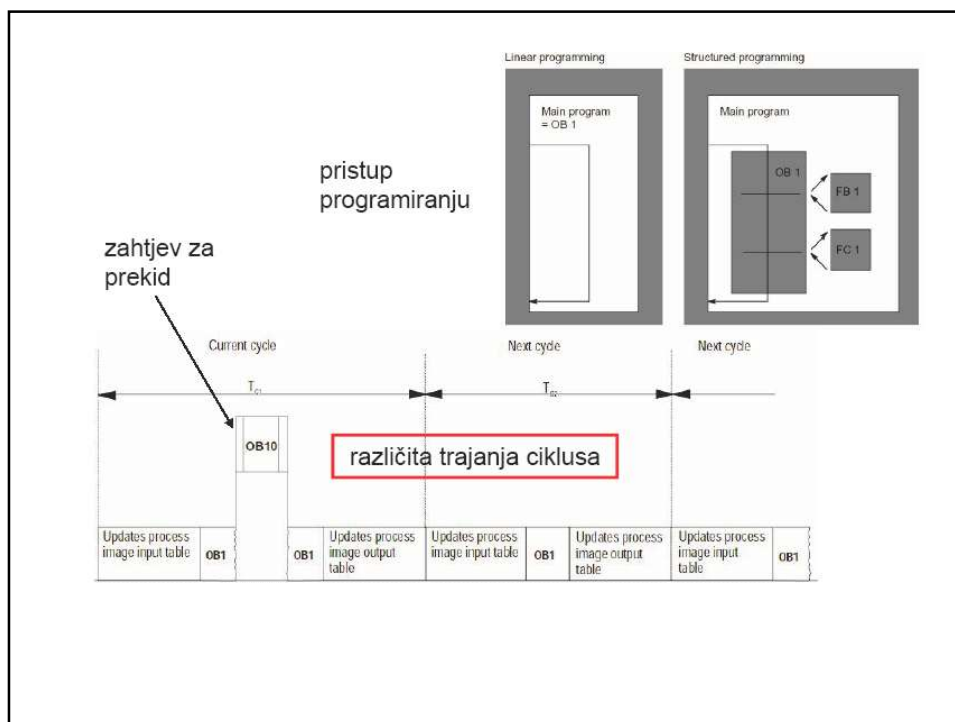
- osnovni entitet programa je mreža (*rung, network*)

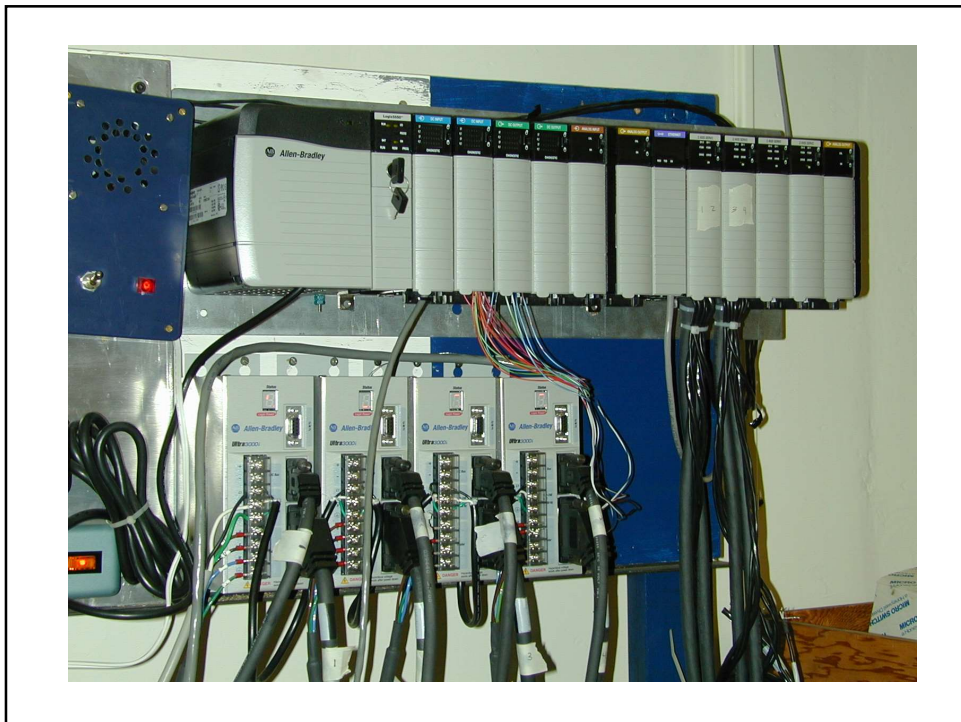


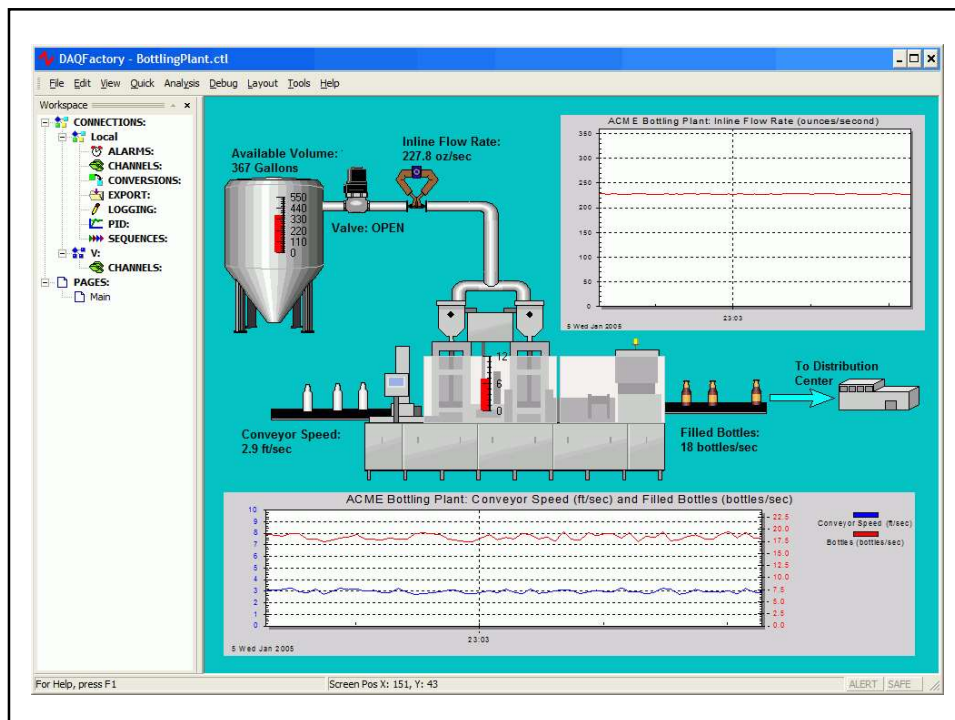
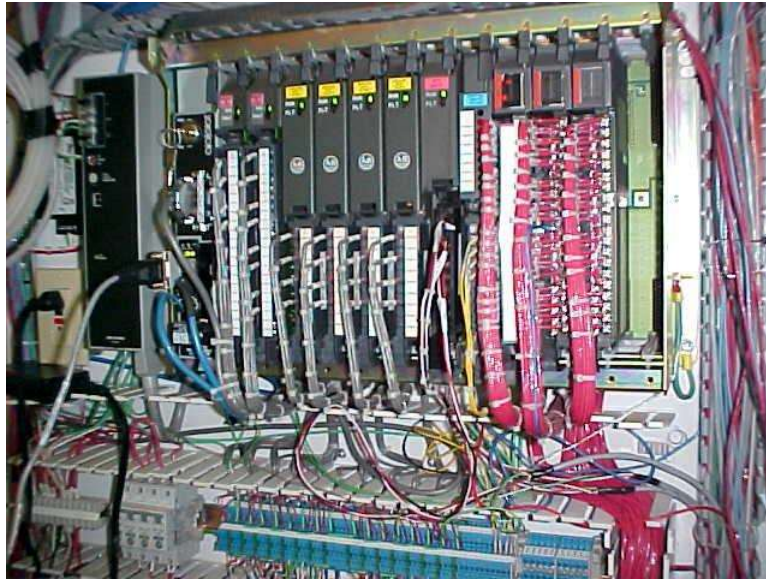
- mreža se sastoji od simbola koji predstavljaju instrukcije i upravljačke komponente

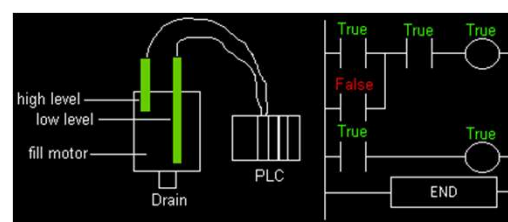
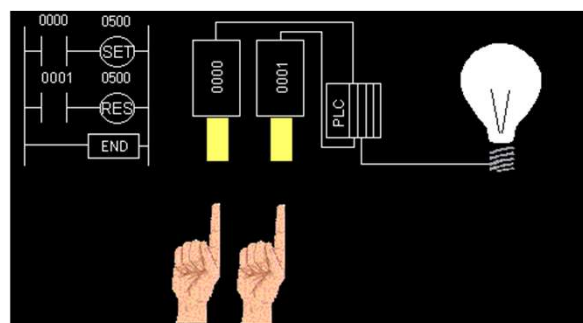
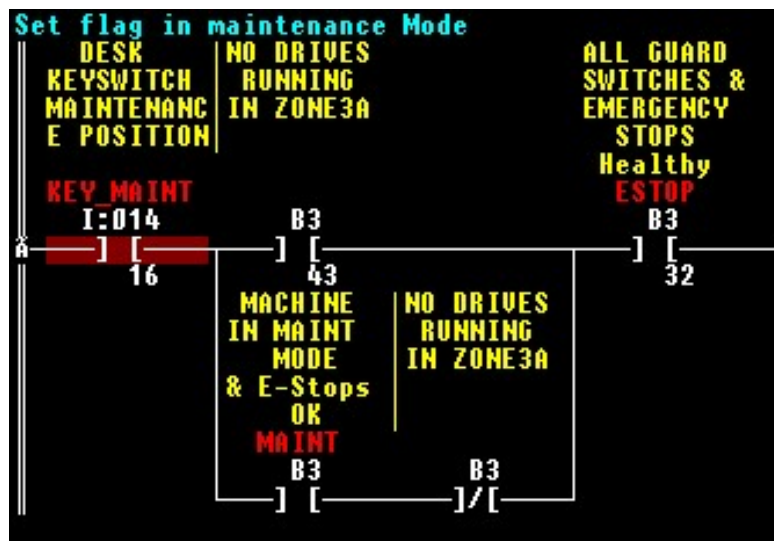
izvođenje programa:

1. odozgo prema dolje, mreža po mreža
2. odozgo prema dolje u mreži
3. slijeva na desno u mreži

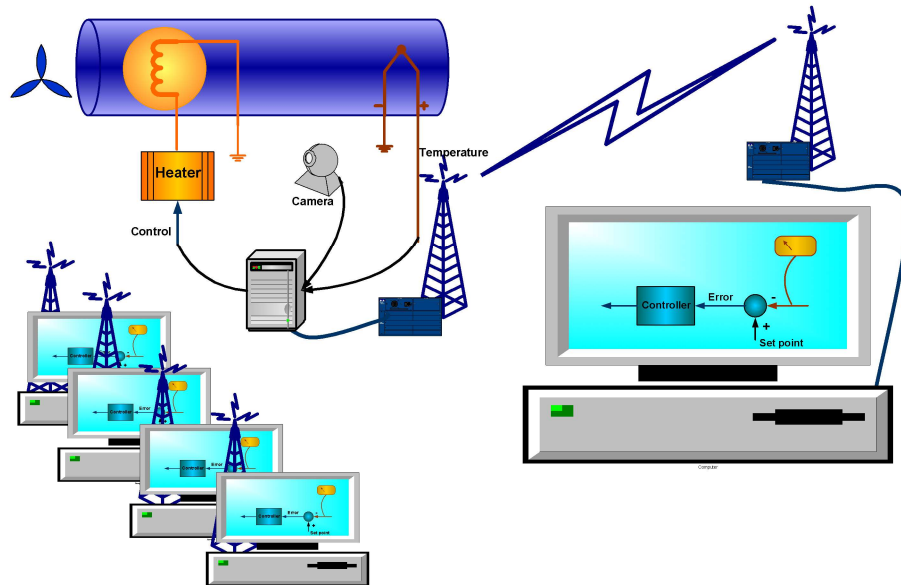








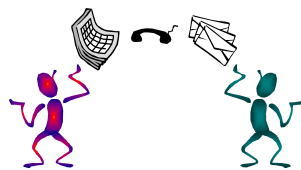
Upravljanje u realnom vremenu



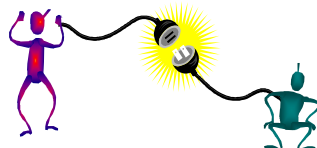
Proces (Sistem)

Razmjena

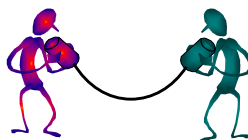
materijala



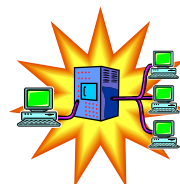
energije



podataka

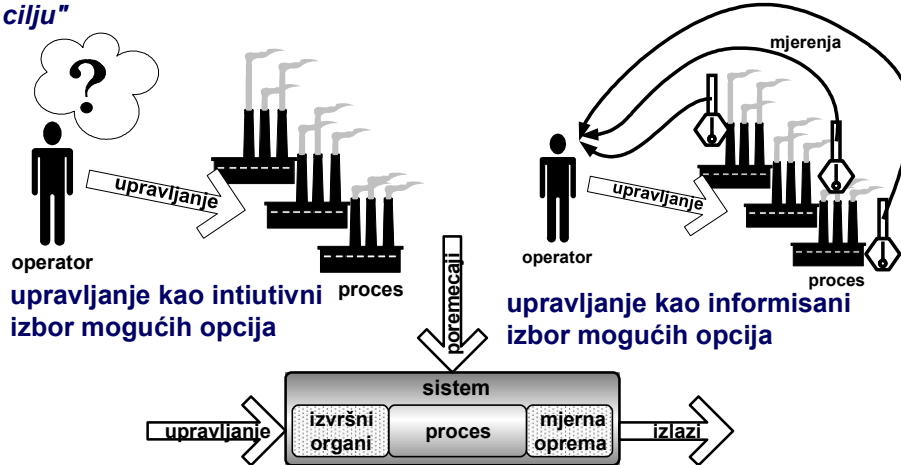


**industrijska
revolucija**



Upravljanje procesima

"proces je prirodna operacija koja se progresivno odvija kroz niz postepenih promjena koje slijede jedna drugu na relativno utvrđen način i vode ka određenom rezultatu ili ishodu, ili vještačka ili voljno izazvana progresivna operacija koja se sastoji od niza upravljanih akcija ili pokreta koji su sistematično usmjereni ka određenom rezultatu ili cilju"



Istorijat

Prve ideje – početak pedesetih
tržište – vojna industrija
autopilot & automatsko upravljanje oružjem
brzi procesi spori računari



Traži se odgovarajuće tržište – Hemijska industrija

• 1958 Elektrana, Louisiana (USA) –

Nadzor & supervajzorsko upravljanje

• 1959 (15. mart) Texaco comp. Port Arthur (USA) –

Zatvorena prva povratna sprega

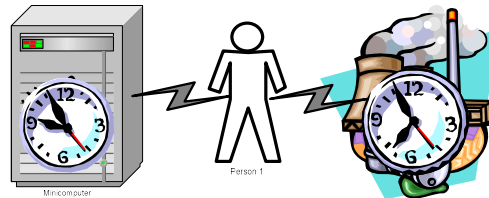
Šezdesete – projektovanje odgovarajućih računarskih sistema i softvera

mini (industrijski) računari

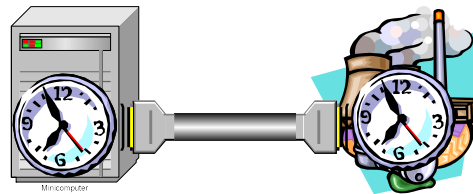
Sedamdesete i nadalje – **microracunari i jezik C**

"odgovor na sve naše molitve" – rađanje Upravljanja u realnom vremenu

Računarski sistem za rad u realnom vremenu

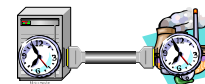


računar kao sredstvo za obradu podataka



računar koji radi u realnom vremenu

Pobude iz okruženja



vremenski događaj (protok vremena)



događaj na procesu

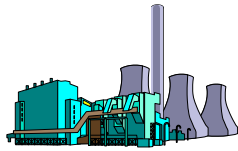
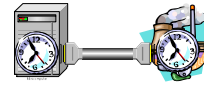


operatorski događaj (interaktivni rad)



sistemske i programske događaji

Dva tipa sistema za rad u realnom vremenu

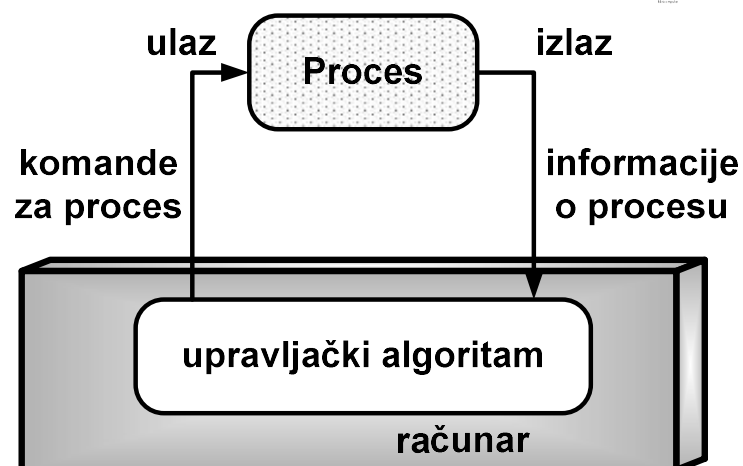
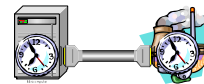


sistemi kod kojih se svaka grupa operacija mora završiti u okviru specificiranog maksimalnog vremena

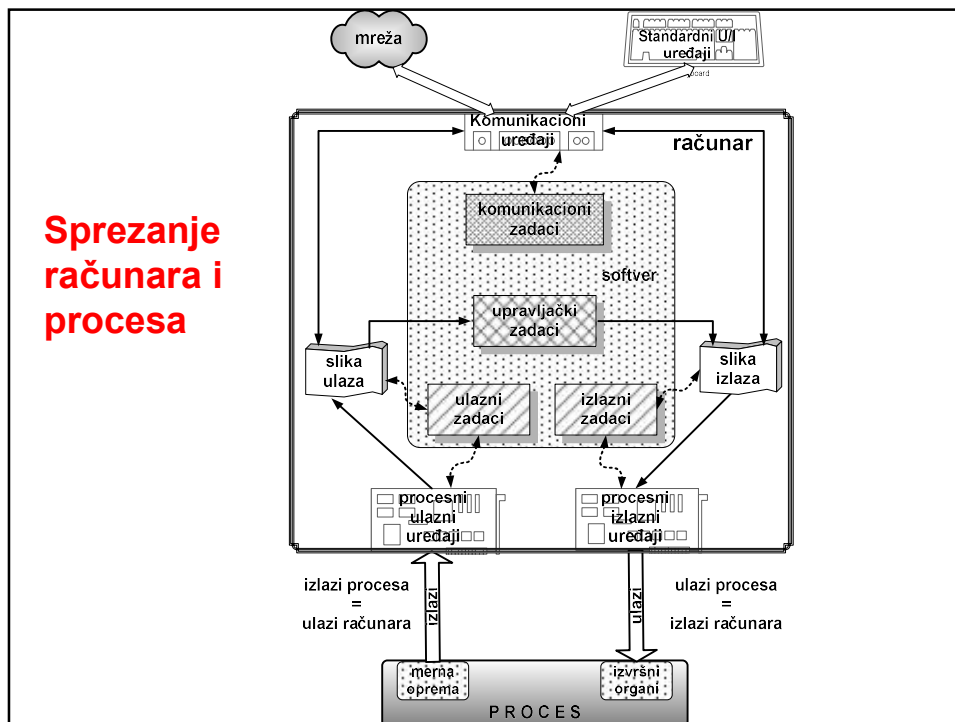


sistemi kod kojih je srednje vrijeme izvršavanja operacija, mjereno na nekom definisanom obimu posla, manje od zadatog maksimalnog vremena

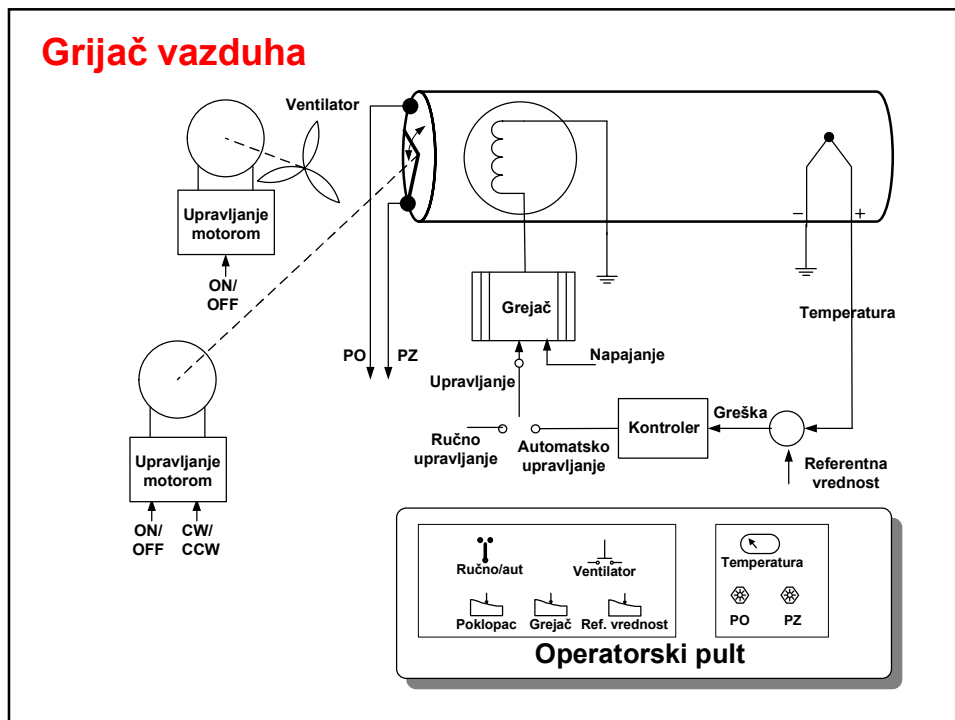
Računarski upravljani sistem

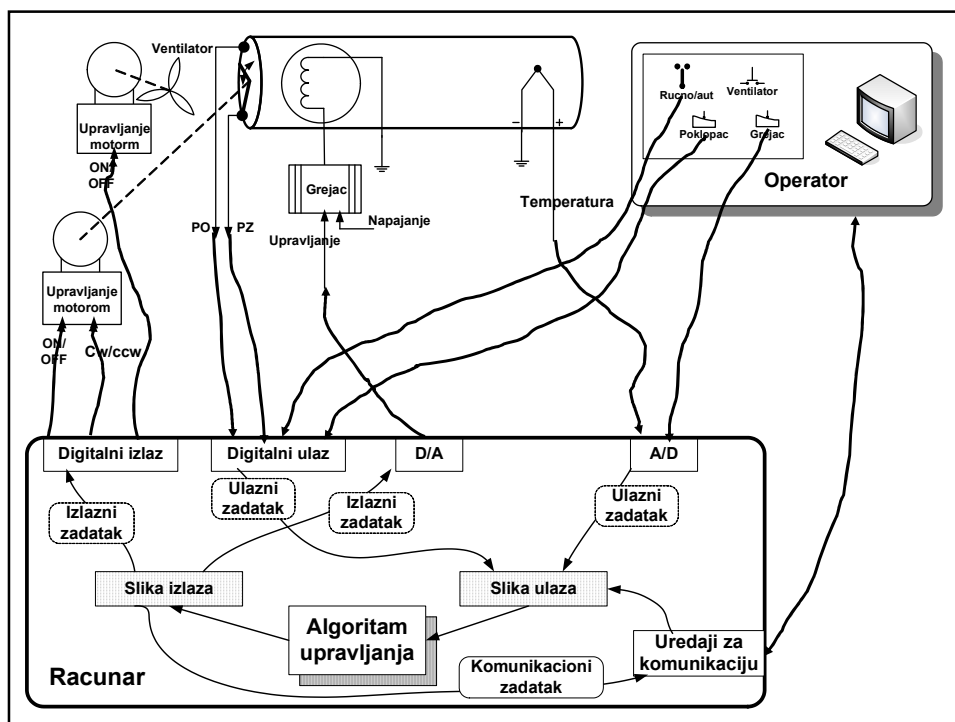


Sprezanje računara i procesa

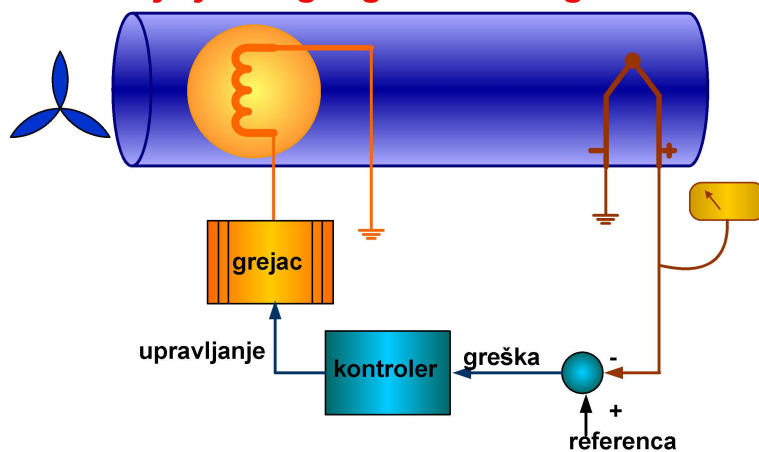


Grijač vazduha





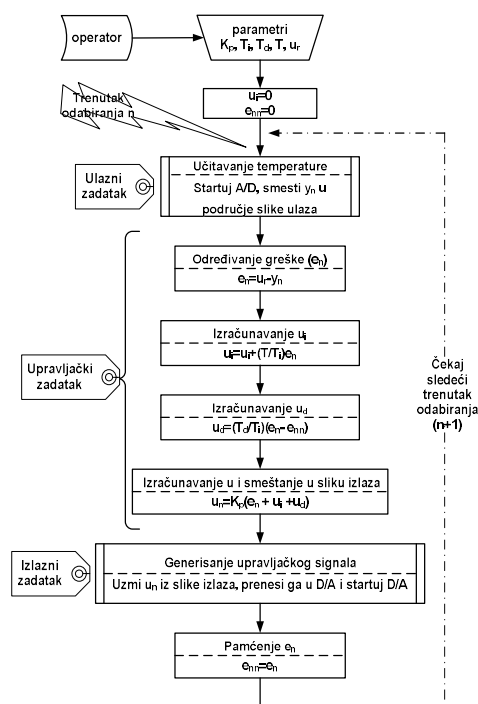
Realizacija jednog algoritamskog koraka



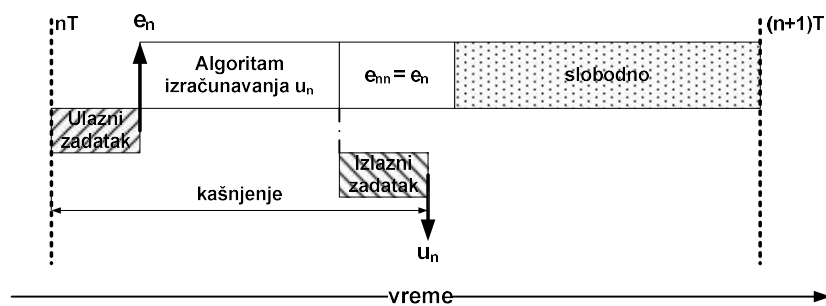
$$u(nT) = K_p e(nT) + u_i(nT) + K_p T_d \frac{e(nT) - e[(n-1)T]}{T}$$

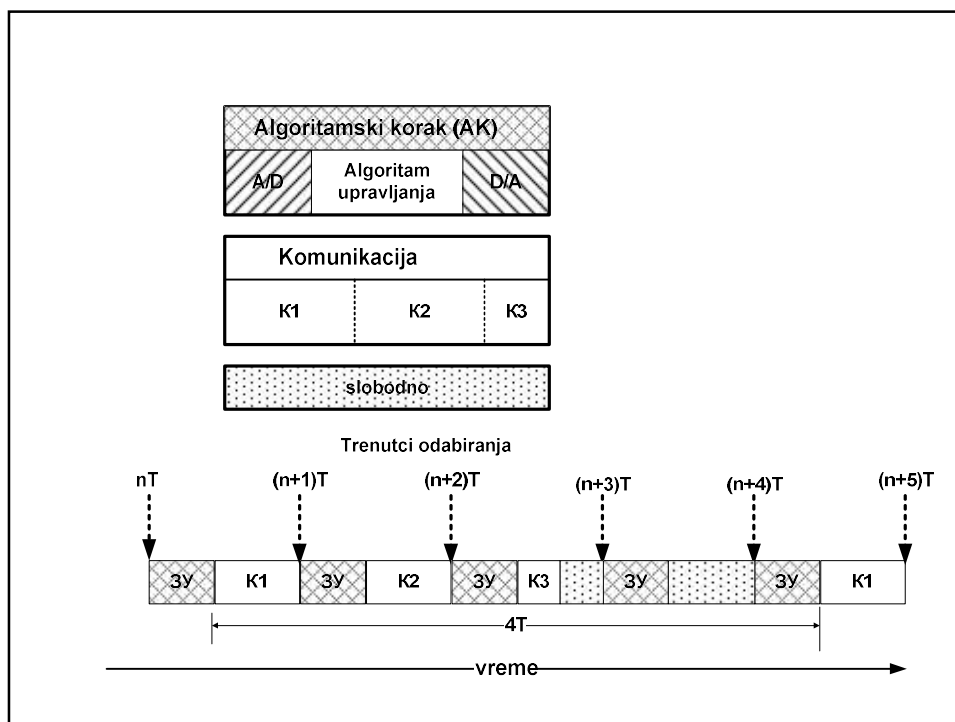
$$u_i(nT) = u_i[(n-1)T] + \frac{K_p}{T_i} T e[(n-1)T]$$

Grafička šema jednog algoritamskog koraka



Vremenski dijagram jednog algoritamskog koraka





Projektovanje računarske aplikacije

- *Apstraktni jezici za modeliranje računarskih aplikacija*
- *MASCOT (modular approach to software construction operation and test)* koristi apstraktne objekte za formiranje preliminarnog projekta za virtuelni računar

Aktivnost

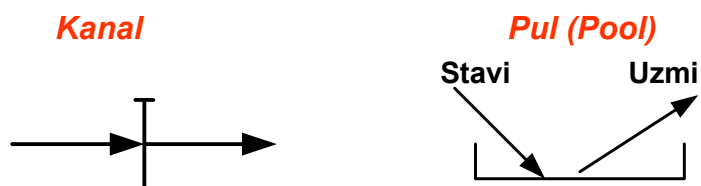


skup operacija koju virtuelna mašina može da realizuje kao zaseban zadatak

Komunikacija

Komunikacije između aktivnosti mogu da se podele u tri kategorije:

- direktna razmjena podataka između dvije aktivnosti
- deoba informacija između više aktivnosti
- sinhronizacioni signali

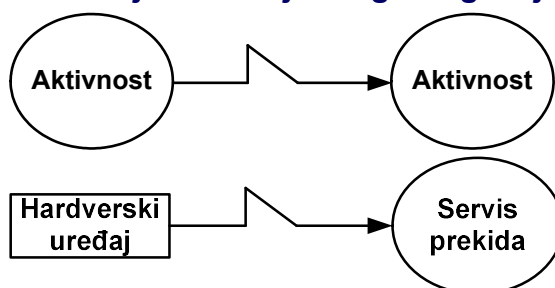


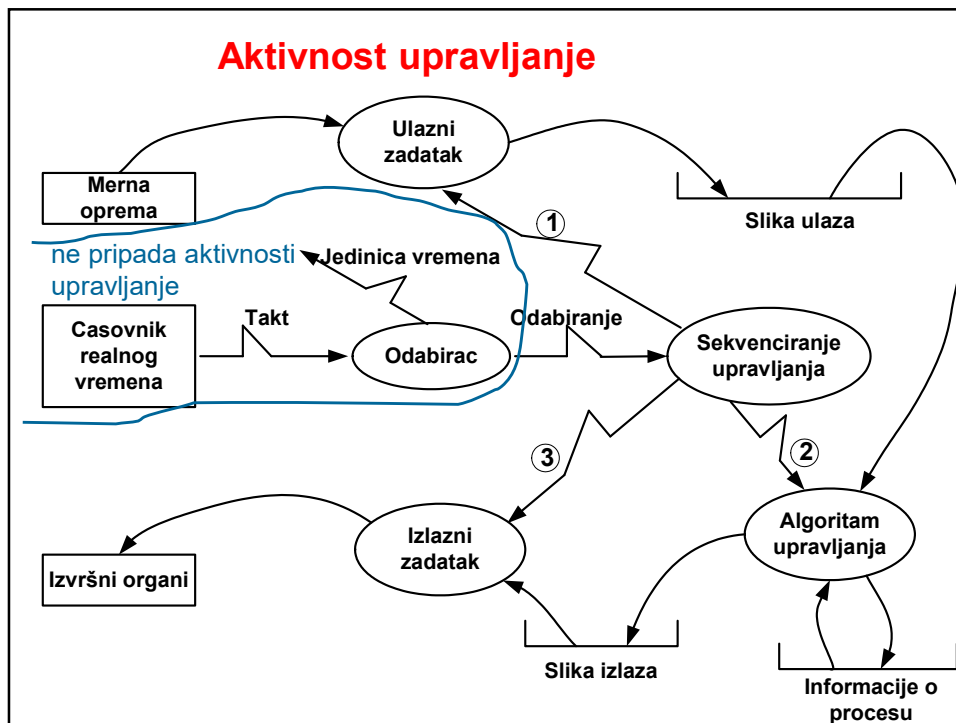
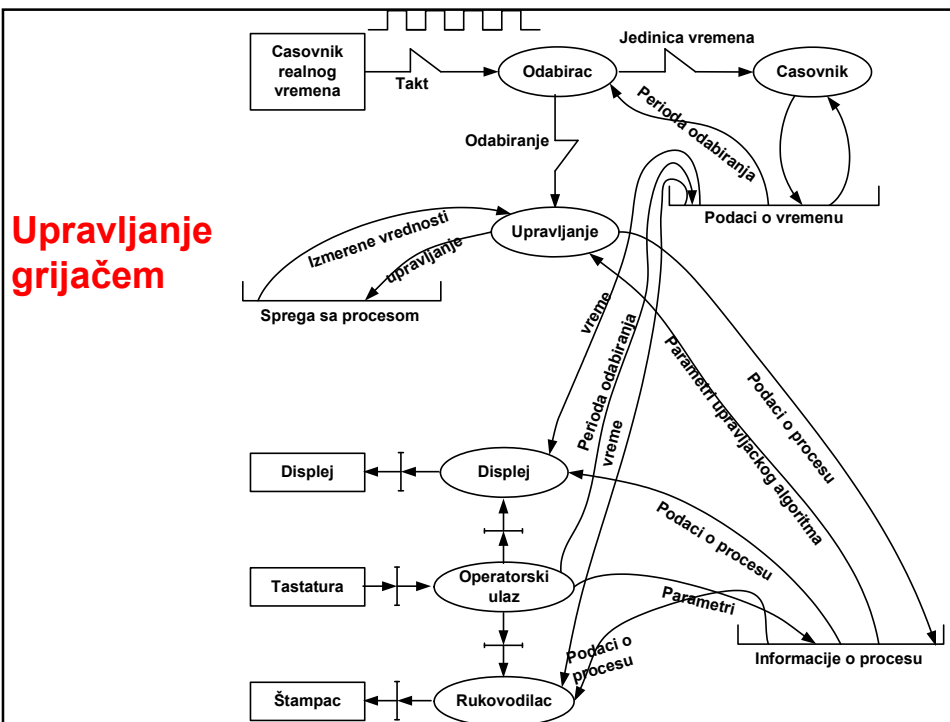
Sinhronizacija

Predstavlja se pomoću dvije procedure

- **WAIT (event)** – aktivnost se prekida (suspenduje) i čeka na događaj (event) koji je prouzrokovao suspenziju
- **SIGNAL (event)** – ova procedura "saopštava" (obznanjuje) da se odgovarajući događaj desio, sve aktivnosti koje su u stanju čekanja tog događaja nastavljaju rad

Spoljašni događaj sa procesa se može posmatrati kao SIGNAL





Klasifikacija programa

- *sekvencijalni programi*
- *multi-tasking programi* – uključuju sinhronizaciju zadataka i upravljanje računarskim resursima
- *programi za rad u realnom vremenu* – način izvođenja diktiran okruženjem (operativni sistem za rad u realnom vremenu)

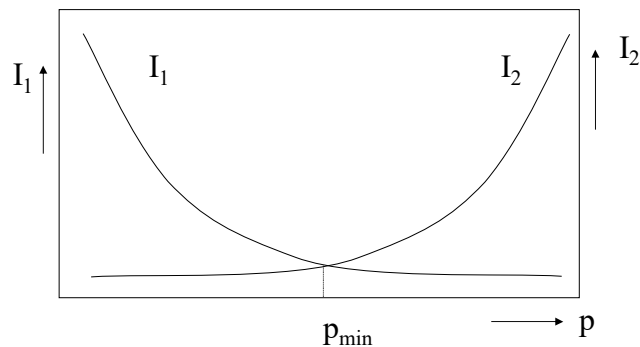
Performanse sistema automatskog upravljanja

Osnovna prednost SAU sa povratnom spregom je mogućnost podešavanja performansi prelaznog procesa i stacionarnog stanja.

Da bi mogao da se projektuje odgovarajući SAU uključujući i njegove komponente neophodno je definisati mjere performansi toga sistema.

Bazirajući se na performansama sistema parametri sistema moraju biti takvi da je moguće obezbijediti željeni odziv.

Budući da su sistemi upravljanja uobičajeno dinamički sistemi, performanse sistema obično se definišu parametrima vremenskog odziva na date ulaze i ponašanja u stacionarnom stanju.



Tehničke specifikacije na SAU uključuju razne zahtjeve koji nekada mogu da budu oprečni.

Mjera performanse naziva se obično indeks performanse.

Performanse sistema

Kod projektovanja SAU važno je da sistem ispuni zadate specifikacije sistema ili funkcionalno tehničke zahtjeve. Budući da su SAU dinamički sistemi, specifikacije performansi mogu biti date u parametrima odziva procesa na razne tipove ulaza, ili u vidu indeksa performanse.

Indeks performanse je broj koji pokazuje “dobrotu” performansi sistema, drugim riječima indeks performanse je numerička mjera performanse ili kriterijuma kvaliteta.

SAU je optimalan ako su vrijednosti parametara podešavanja izabrane tako da odabrani indeks performanse ima minimalnu ili maksimalnu vrijednost. Pri tome optimalna vrijednost parametara direktno zavisi od izabranog indeksa performanse.

Osnovni zahtjevi koji se postavljaju na izbor indeksa performanse su:

- da je selektivan, tj. da se podešavanje parametara direktno odražava na njegovu vrijednost,
- mora imati pozitivnu ili vrijednost jednaku nuli
- da ga je lako računati ili odrediti.

U tehničkoj praksi najčešće se susreću integralni kriterijumi bazirani na greški $e(t)$ ili nekoj funkciji od nje.

Kvantitativna mjera performanse sistema potrebna je za rad adaptivnih sistema upravljanja i optimalnih sistema upravljanja. Kada je cilj da se poboljša projektovanje sistema upravljanja ili adaptivnog sistema upravljanja, indeks performane mora biti odabran i numerički odredljiv ili mjerljiv.

$$I_1 = \int_0^T e^2(t) dt \quad I_2 = \int_0^T t e^2(t) dt \quad I_3 = \int_0^T |e(t)| dt \quad I_4 = \int_0^T t |e(t)| dt$$

Kod sinteze digitalnih SAU indeksi performansi dobijaju se diskretizacijom odgovarajućih analognih.

Za tehničku praksu tolerantna je degradacija indeksa performanse do $3I_{\min}$, kada se imaju još prihvatljivi ostali kriterijumi kvaliteta prelaznog procesa na step ulaz. Pri tome se opseg parametara koji tu degradaciju uzrokuje kreće u opsegu 15-20% za sistem drugog reda. To znači da za tu promjenu parametra p nije potrebno prepodešavanje upravljačke strukture.

Pouzdanost sistema automatskog upravljanja

Potreba za višim nivoom pouzdanosti i sa njom vezane raspoloživosti SAU raste iz godine u godinu radi različitih razloga, uključujući zahtjev za povećanim performansama i ekonomske konsekvence koje su rezultat otkaza rada sistema. Pouzdanost hardware-a sistema povezana je sa predviđanjem performansi sistema na datom vremenskom intervalu, uključujući i sposobnost sistema da apsorbira neki otkaz. Salgasno tome, postoje metode za računanje srednjeg vremena između otkaza za komponente, dijelove sistema, i sisteme, kao i vjerovatnoću rada čitavog sistema na datom intervalu vremena.

Pouzdanost sistema automatskog upravljanja

Pouzdanost sistema $^sR(t_0)$ sistema S u vremenu t_0 je vjerovatnoća da će sistem obaviti funkciju od interesa pod postavljanim uslovima i bez prekida od trenutka $t=0$ do $t=t_0$.

Sa otkazom sistem gubi jednu ili više izlaznih specifikacija. Postoje dva tipa otkaza sistema.

Kada je otkaz trenutan, potpun i ireverzibilan tada se radi o **katastrofalnom otkazu**. Takav otkaz ne može biti predviđen kada će se pojaviti i on se nekada zove slučajni otkaz.

Kada je otkaz rezultat drifta, ili degradacije u vremenu, jedne ili više komponenti sistema, onda se govori o **otkazu usled degradacije** ili drifta sistema (izmjena statičke karakteristike tokom vremena).

Teorijska osnova pouzdanosti

Pouzdanost je važan parametar kod planiranja, projektovanja i rada sistema. Izraz pouzdanost koristi se kao opšti kvalifikativ da se izrazi mogućnost sistema da obavlja predviđenu funkciju. Pouzdanost sistema može biti izražena kvantitativno korišćenjem teorije vjerovatnoće.

Sa otkazom sistem gubi jednu ili više izlaznih specifikacija. Postoje dva tipa otkaza sistema.

Kada je otkaz trenutni, potpun i ireverzibilan tada se radi o katastrofalnom otkazu. Takav otkaz ne može biti predviđen kada će se pojaviti i on se nekada zove slučajni otkaz.

Kada je otkaz rezultat drifta, ili degradacije u vremenu, jedne ili više komponenti sistema, onda se govori o otkazu usled degradacije ili drifta (izmjena statičke karakteristike tokom vremena) sistema.

Osnovni koncept pouzdanosti bazira na takozvanim misiono-orijentisanim sistemima. U tom slučaju pouzdanost može biti definisana kao vjerovatnoća da komponenta ili sistem obavlja svoju funkciju za period vremena koji se zahtijeva radnim uslovima.

Osnovne jednačine mogu biti dobijene razmatranjem situacije u kojoj velika grupa identičnih komponenti radi u test uslovima. Neka je N_0 broj komponenti pod testom, $N_s(t)$ broj preživjelih komponenti poslije testa, i $N_f(t)$ broj otkazalih komponenti u vremenu t . Vjerovatnoća broja komponenti koje nisu otkazale i još su u radnom stanju u vremenu t , može biti označena kao:

$$R(t) = \frac{N_s(t)}{N_0} = \frac{N_0 - N_f(t)}{N_0} = 1 - \frac{N_f(t)}{N_0}$$

$R(t)$ je pouzdanost komponente u misiji koja traje vrijeme t . Trenutna vrijednost funkcije raspodjele (učestanosti otkaza) $f(t)$ je:

$$f(t) = \frac{dN_f(t)}{dt} \frac{1}{N_0} = -\frac{dR(t)}{dt}$$

A intenzitet otkaza $\lambda(t)$ je

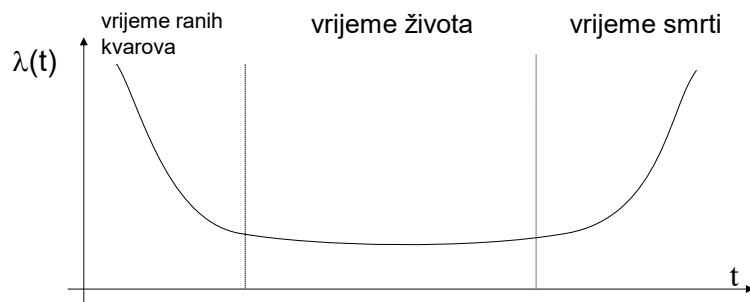
$$\lambda(t) = \frac{dN_f}{dt} \frac{1}{N_s(t)} = \frac{-dR(t)}{dt} \frac{1}{R(t)} = \frac{f(t)}{R(t)}$$

Na osnovu čega je $R(t)$:

$$R(t) = \exp \left[- \int_0^t \lambda(x) dx \right]$$

što daje vjerovatnoću opreme ili sistema, koji obavlja datu misiju u vremenu t , da preživi. Pri tome, $\lambda(t)$ je intenzitet otkaza i takodje je funkcija vremena. Da se primjeni poslednja relacija neophodno je napraviti određene pretpostavke.

Većina komponenti i sistema ima relativno standardni intenzitet otkaza u vremenu. Tipičan profil intenziteta otkaza $\lambda(t)$ koji se zove "kada" dat je na slici.



Praktična iskustva pokazuju da je produktivnije projektovati pouzdanost sistema nego tu pouzdanost određivati testiranjem. Konsekventno tome, razvijene su metode za matematičku predikciju ponašanja pouzdanosti. Postoje tri jasno definisana perioda u čitavom ciklusu komponenti ili sistema sastavljenog iz:

1. perioda ranih kvarova - region 1,
2. perioda upotrebljivog korištenja - region 2,
3. perioda intenzivne degradacije, "umiranja", sistema - region 3.

Najznačajniji period je period korišćenja sistema - region 2, gdje je intenzitet otkaza, konstantna veličina. U tom slučaju je funkcija gustine otkaza na osnovu

$$f(t) = \lambda \exp(-\lambda t)$$

Sistem može ostati dugo vremena u regionu 2 primjenom odgovarajućeg održavanja sistema. Najvažniji aspekt u ovom periodu je da je otkazivanje komponenti u toku misije nezavisno od početka misije u trenutku $t = T$, što nije slučaj sa regionima 1 i 3.

U ovom slučaju je:

$$R(t | T) = \frac{\int_{T+t}^{\infty} f(x) dx}{\int_T^{\infty} f(x) dx}$$

Može se lako pokazati da je $R(t)$ nezavisna od T . Matematičko očekivanje vremena do pojave greške komponente u sistemu je:

$$E(t) = \int_0^{\infty} t f(t) dt$$

U regionu 2 je:

$$E(t) = \int_0^{\infty} t \exp(-\lambda t) dt = 1 / \lambda$$

i ova očekivana vrijednost poznata je kao srednje vrijeme do pojave kvara, MTTF -(Mean Time To Failure).

Varijacija u intenzitetu otkaza $\lambda(t)$ može biti modelirana primjenom široke klase funkcija. Jedna od najraširenijih funkcija koja može biti korišćena u sva tri regiona na slici 11.1 je Weibull-ova raspodjela.

Vjerovatnoća funkcije gustine otkaza za Weibull-ovu raspodjelu je,

$$f(t) = \beta t^{\beta-1} \exp\left[-(t / \alpha)^{\beta}\right] / \alpha^{\beta}$$

Gdje je: α –parametar skaliranja
 β – parametar oblika

U ovom slučaju je:

$$R(t) = \int_T^{\infty} f(t)dt = \exp(-t/\alpha)^\beta$$

$$\lambda(t) = f(t) / R(t) = \beta t^{\beta-1} / \alpha^\beta$$

Posljednje tri relacije mogu se upotrijebiti da se sračuna pouzdanost u sva tri regiona sa slike odgovarajućim izborom veličina α i β . Sljedeća dva slučaja vrijedno je predstaviti.

Slučaj 1: $\beta=1$ u ovom slučaju je:

$$f(t) = (1/\alpha)^{-t/\alpha} \exp(-t/\alpha)$$

$$\lambda(t) = 1/\alpha$$

Slučaj 2: $\beta=2$ u ovom slučaju je:

$$f(t) = (2t/\alpha^2) \exp[-(t/\alpha)^2]$$

$$\lambda(t) = 2t/\alpha^2$$

Ovo je Rayleigh-eva raspodjela i predstavlja situaciju u kojoj intenzitet otkaza raste linearno s vremenom.

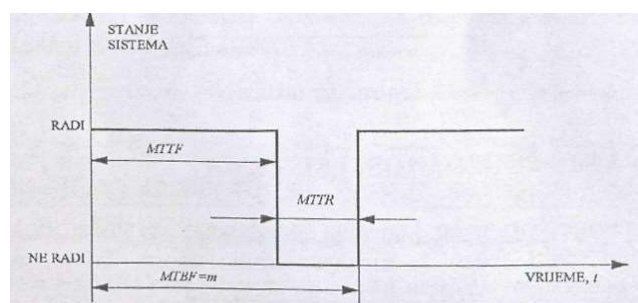
Tri regiona na slici mogu biti kategorisana kako slijedi:

Region 1: $\beta < 1$, opada intenzitet otkaza,

Region 2: $\beta = 1$, konstantan intenzitet otkaza,

Region 3: $\beta > 1$, raste intenzitet otkaza.

Ako se na ordinati nalaze dva stanja sistema RADI i NE RADI (U KVARU), a ne apscisi vrijeme t , tada se može definisati srednje vrijeme između otkaza, m , ili MTBF (Mean Time Between Failures). MTBF obično daje svaki proizvođač koji na osnovu drugih podataka o sistemu računa ovaj podatak. Očekivano srednje vrijeme da se otkaz popravi označeno je na slici sa MTTR.



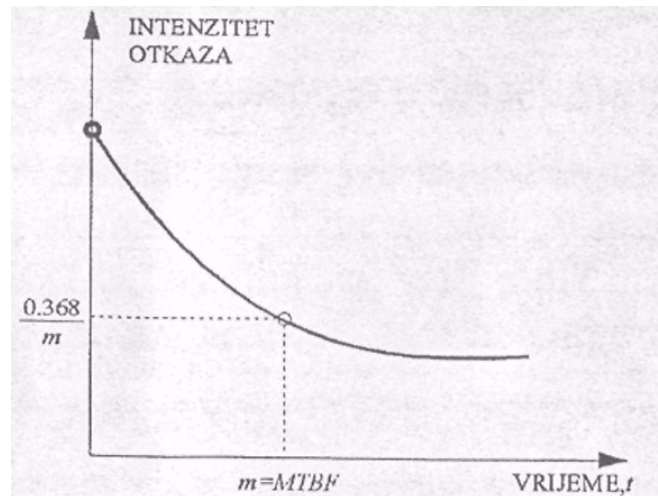
Rani otkazi normalno se eliminišu unutar prvih 20-200h procedurama razrade (burn-in) uređaja. Vrijeme razrade može biti određeno relacijom,

$$T_R = m_e \left(1 + \frac{1}{2} + \frac{1}{3} + \dots + \frac{1}{n_e} \right)$$

gdje je: T_R - vrijeme razrade uređaja, sistema,
 n_e - broj podstandardnih komponenti,
 m_e - srednje vrijeme između otkaza u periodu ranih kvarova.

Pri tome za vrijeme T_R pojavi se 63% ranih kvarova, a za vrijeme $5T_R$ 99% ranih kvarova. Broj članova u harmonijskom redu uzima se da je jednak n_e . Stvarno vrijeme razrade uzima se da je jednako $5T_R$.

Za komponente i sisteme u opštem slučaju, pouzdanost se obično podrazumijeva za vrijeme između završene razrade i početka "umiranja" sistema. Za vrijeme života sistema, region 2, otkazi su slučajni i ne mogu biti reducirani zamjenom komponenti. Mada ovi otkazi nisu poredivi, frekvencija njihove pojave konstantna je i definisana je kao recipročna vrijednost od MTBF, $1/m$. Saglasno tome, za vrijeme života sistema, pouzdanost će biti ista za jednake radne periode, bez obzira na startni trenutak. Nije teško pokazati da će otkazi ostati ispod 10% u intervalu od $1/10$ MTBF. Ako se u periodu života sistema ne vrši zamjena otkazalih komponenti do trenutka $t = m = \text{MTBF}$, samo 37% komponenti će biti u redu.



Na dijagramu 11.3 data je zavisnost $1/MTBF$ od vremena.

Određivanje pouzdanosti sistema

Najjednostavniji primjer primjene koncepta pouzdanosti je slučaj u kome sve komponente moraju raditi dobro da bi sistem radio dobro. To je, sa aspekta pouzdanosti, serijski sistem i pouzdanost sistema je data sa

$$R_s = R_1 \times R_2 \times \dots \times R_n = \prod_{i=1}^n R_i \quad R_s = \exp\left(-\sum_{i=1}^n \lambda_i t\right)$$

U slučaju da su dvije komponente spojene paralelno da formiraju redundantnu konfiguraciju, nepouzdanost sistema Q_s je

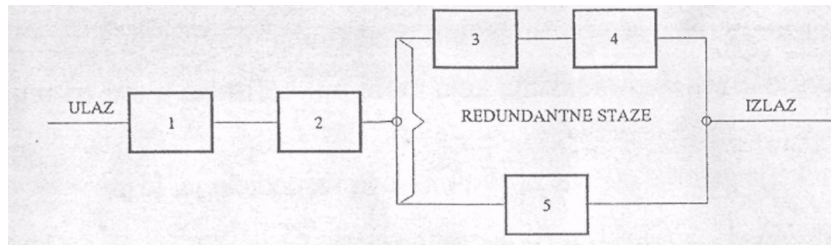
$$Q_s = Q_1 Q_2 \quad \text{a pouzdanost sistema je:}$$

$$R_s = 1 - Q_s = 1 - Q_1 Q_2 = 1 - (1 - R_1)(1 - R_2) = R_1 + R_2 - R_1 R_2$$

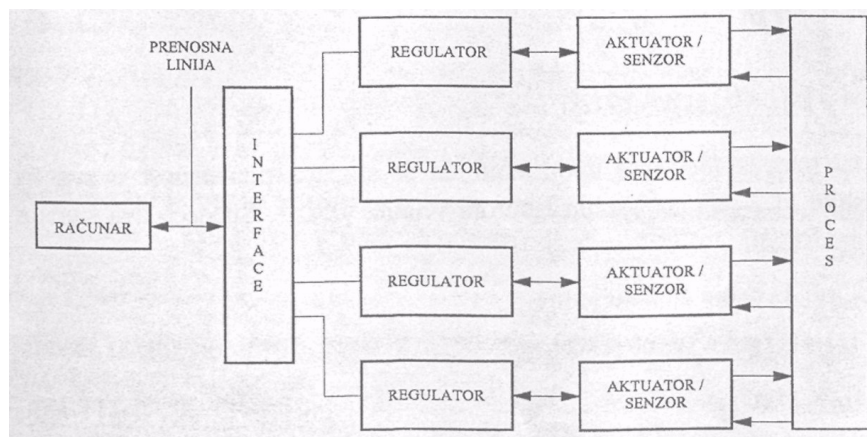
ili

$$R_s = \exp(-\lambda_1 t) + \exp(-\lambda_2 t) - \exp[-(\lambda_1 + \lambda_2)t]$$

Polazeći od ovih elementarnih relacija, moguće je izračunati pouzdanost za složenije sisteme. Neka je topologija sistema kao na sledećoj slici. Pouzdanost sistema je onda:



$$R_s = R_1 R_2 (R_3 R_4 + R_5 - R_3 R_4 R_5)$$



Pouzdanost se može razmatrati na nivou komponenti sistema, podsistema i sistema. Ovo će biti razmotreno na primjeru sistema upravljanja procesa čija je struktura data na slici. SAU sa slike sadrži računar koji nadgleda četiri kanala upravljanja preko odgovarajuće prenosne linije i interface-a.

Potrebno je odrediti pouzdanost sistema sa slike za vrijeme mjesečnog rada od 720 h/mjesec. Pretpostavlja se da se sistem nalazi u regionu 2. Podaci relevantni za pouzdanost dati su u TABELI. Pri tome, pouzdanost će biti definisana relacijom (11.5) gdje je $\lambda_1 = 1/MTBF$. Dalje, neka bude uvedena pretpostavka da je za sistem sa slike 11.5 moguće bilo koji kvar otkloniti za 8 h.

Oprema	MTBF (H)	Broj otkaza po mjesecu (720 H/MJ)
RAČUNAR	$m_R = 10^4$ 10^{-4}	0,072000
PRENOSNA LINIJA	$m_L = 10^6$ 10^{-6}	0,000720
INTERFACE	$m_I = 10^4$ 10^{-4}	0,072000
REGULATOR	$m_C = 10^5$ 10^{-5}	0,007200
AKTUATOR/SENZOR	$m_{AS} = 10^5$ 10^{-5}	0,007200

Očekivano srednje vrijeme između otkaza za čitav sistem MTBFS dato je izrazom

$$MTBF_s = \frac{1}{\frac{1}{m_R} + \frac{1}{m_L} + \frac{1}{m_I} + \frac{1}{m_C} + \frac{1}{m_{AS}}} = \frac{1}{\frac{1}{10^4} + \frac{1}{10^6} + \frac{1}{10^4} + \frac{4}{10^5} + \frac{4}{10^5}} = 3560h$$

Pouzdanost sistema u trenutku $t = 720h$ je onda

$$T(720) = e^{-\frac{720}{MTBF_s}} = e^{-\frac{720}{3560}} = 0.816$$

Raspoloživost sistema je sada

$$A = \frac{MTBF}{MTBF + MTTR_s} = \frac{3560}{3560 + 8} = 0.997$$

Degradacija pouzdanosti zahvaljujući serijskom spoju komponenti ilustrovana je u TABELI 2 za slučaj (a) i slučaj (b) kada svaka komponenta ima pouzdanost 0,9 i 0,99, respektivno.

Broj komponenti	Pouzdanost	
	Slučaj (a)	Slučaj (b)
1	0.9	0.99
2	0.81	0.9801
3	0.729	0.9703
5	0.59049	0.95099
10	0.348678	0.904382
20	0.121577	0.817902

U TABELI 3 daju se vrijednosti pouzdanosti paraleleno spregnutih komponenti, pri čemu svaka komponenta ima pouzdanost 0.9.

Broj komponenti	Pouzdanost
1	0.9
2	0.99
3	0.999
4	0.9999
5	0.99999