

APS - USMENI

Digitalni ulazi/izlazi

1. Usporedba PLC-a i mikrokontrolera

PLC:

- Robusno industrijsko ugradbeno „računalo“
- Modularne izvedbe – CPU, I/O moduli, funkcionalni moduli, komunikacijski moduli i dr.
- Namijenjeno za rad u industrijskim uvjetima/okolini (visoke/niske temperature, vlažna područja, itd.)
- Programske jezice određeni standardom IEC 61131
- Sadrži operativni sustav koji se brine za ispravan rad PLC-a
- Dugogodišnja podrška te novije verzije softvera i hardvera su kompatibilne sa prethodnim verzijama

Mikrokontroler:

- Kompaktni integrirani sklop koji se tipično ugrađuje u razne ugradbene sustave poput televizora, kućnih aparata, automobila, itd.
- Sadrži CPU, memoriju (podatkovnu i programsku), I/O jedinice za povezivanje s „vanjskim svijetom“ i dr.
- Dosta povoljan (početne cijene već od nekoliko dolara)
- Programske jezike ovisi o proizvođaču (najčešće podržavaju asembler, C/C++, itd.)
- Renomirani proizvođači mikrokontrolera izdaju novije verzije svakih par godina te ne garantiraju kompatibilnost s prethodnim verzijama



3

2. Pretvornik, sensor, izvršni član.

Pretvornik

- Pretvornik (*transducer*) – uređaj koji pretvara energiju iz jednog oblika u drugi

Senzor

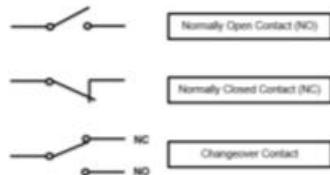
- Uredaj koji pretvara vrijednost mjerene fizikalne veličine u ekvivalentni (proporcionalni) električni signal

Izvršni član

- Uredaj koji prevara električnu energiju u mehaničku energiju

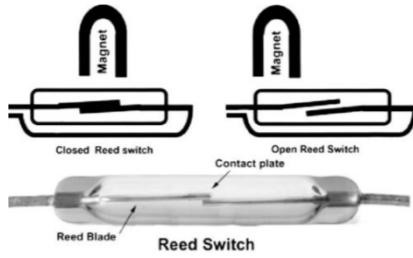
3. Mehanički logički senzori – princip rada, NO, NC, changeover

- Tipkala i sklopke
 - Preko mehaničke radnje kontakt(i) senzora se uključuje (ili isključuje)
- Vrste kontakata:
 - Radni kontakt (*Normally Open*, NO) – u slučaju kad senzor nije aktivan kontakt je otvoren (logička 0)
 - Mirni kontakt (*Normally Closed*, NC) – u slučaju kad senzor nije aktivan kontakt je zatvoren (logička 1)
 - Preklopni kontakt (*Changeover*) – kombinacija NO i NC kontakta



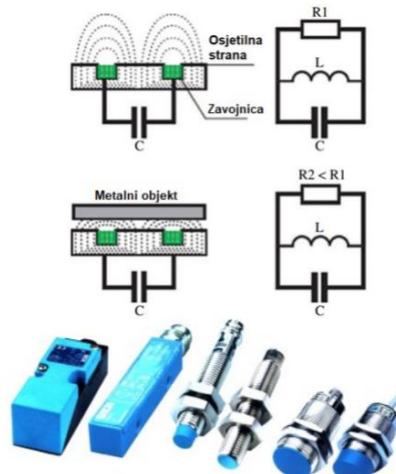
4. Elektromagnetski logički senzori.

- Sastoje se od magneta ili elektromagneta koji preko magnetskog polja mijenja stanje kontakata
- Položaj kontakata se mijenja kada magnetsko polje djeluje na njih
- Tipično se koriste za detekciju prisutnosti objekta
- Prednosti – robusni, bezkontaktna detekcija
- Mane – utjecaj drugog magnetskog polja može uzrokovati pogrešnu signalizaciju



5. Induktivni logički senzori.

- Rade na principu elektromagnetske indukcije
- Često se koriste za detekciju metalnih objekata
- Princip rada – oscilator (paralelni spoj zavojnice i kondenzatora) stvara magnetsko polje koje se mijenja prilikom dolaska metalnog predmeta; detekcijom promjene frekvencije oscilacija, elektronički sklop unutar senzora mijenja stanje kontakta
- Ne mogu se detektirati objekti koji nisu metalni



Pregled Induktivnih osjetnih članova blizine

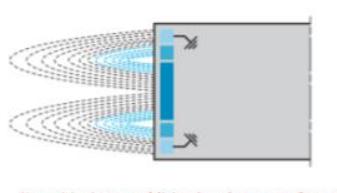
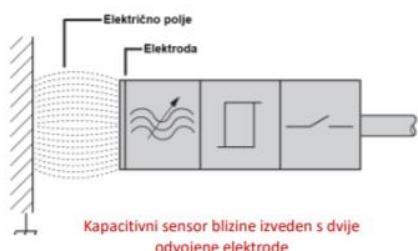
- Tipično se koriste za:
 - Mjerenje brzine vrtnje
 - Detekcija položaja (npr. detekcija krajnjih položaja).
 - Brojanje i detekcija metalnih elemenata
 - Detekcija klizanja (kod raznih vrsta prijenosa)

6. Kapacitivni logički senzori.

- Koriste fizikalni princip kapaciteta pločastog kondenzatora

$$C = \epsilon_0 \epsilon_r \frac{S}{d} [F]$$

- Promjenom iznosa dielektrične konstante ϵ_r , detektira se prisutnost predmeta između dvije elektrode



7. Optički logički senzori.

- Rade na principu detekcije odaslane svjetlosti
- Odaslana svjetlost najčešće je infracrvena (odabirom infracrvene valne duljine smanjuje se utjecaj smetnji – vidljive svjetlosti)
- Sastoji se od izvora svjetlosti (*Transmitter*) i prijemnika svjetlosti (*Receiver*)
- Dijele se na:
 - Optička vrata/brana (*Transmitting type*)
 - Difuzni senzor (*Reflecting type*)

8. Optička vrata.

- Izvor svjetlosti i prijemnik postavljaju se odvojeno jedno nasuprot drugom
- U trenutku dolaska objekta koji je potrebno detektirati između prijemnika i izvora, prekida se optički snop i objekt se detektira
- Prednosti - mogućnost velikog razmaka izvora svjetlosti i prijemnika (>10 m)
- Nedostaci - potrebno napajanje za obje jedinice

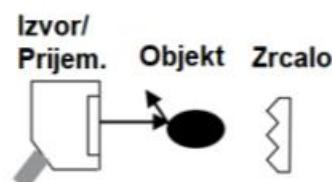


9. Difuzni senzor.

- Izvor svjetlosti i prijemnik nalaze se unutar jedne jedinice
- U trenutku prolaska objekta ispred difuznog osjetnika, svjetlost se odbija natrag u prijemnik i objekt se detektira
- Prednosti - jedna jedinica, jednostavna ugradnja
- Nedostaci - kraći doseg detekcije u usporedbi s ostalim optičkim osjetnicima, ovisnost o obliku i boji

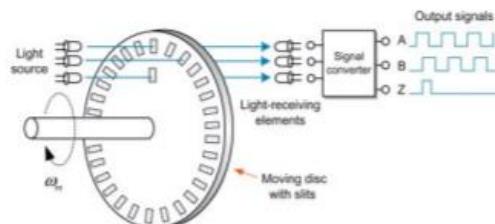
10. Retrorefleskni senzor.

- Izvor svjetlosti i osjetnik nalaze se unutar jedne jedinice, a nasuprot jedinice nalazi se zrcalo
- U trenutku prolaska objekta ispred jedinice i zrcala, snop svjetlosti se prekida i objekt se detektira
- Prednosti - jedna jedinica, jednostavna ugradnja, velik doseg detekcije (>10 m)
- Nedostaci - Ovisnost o obliku i boji, problem detekcije objekata s glatkim refleksnim površinama (staklo, polirani metal)



11. Inkrementalni enkoder.

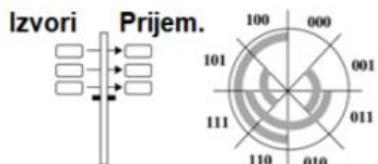
- Senzor za mjerjenje brzine vrtnje, odnosno položaja osovine (motora ili tereta)
- pomoću optičkih osjetnika generiraju slijed električnih impulsa čija je frekvencija proporcionalna brzini vrtnje
- posjeduje dva optička osjetna člana koji generiraju dva fazno pomaknuta pulsirajuća signala (signali A i B)
- Informacija o smjeru vrtnje sadržana je u faznom pomaku ovih dvaju signala, koji ovisno o smjeru vrtnje poprima vrijednosti 90° el.
- Postoje i izvedbe s trećim signalom, nulimpuls, za detekciju pređenog kruga



22

12. Apsolutni enkoder.

- Daje absolutni položaj/kut u obliku kombinacije binarnih signala
- Svaka kombinacija binarnog zapisa jedinstveno određuje položaj osovine
- Tipično se koristi GRAY-ev ciklički kod jer je 1 bit razlike između susjednih vrijednosti

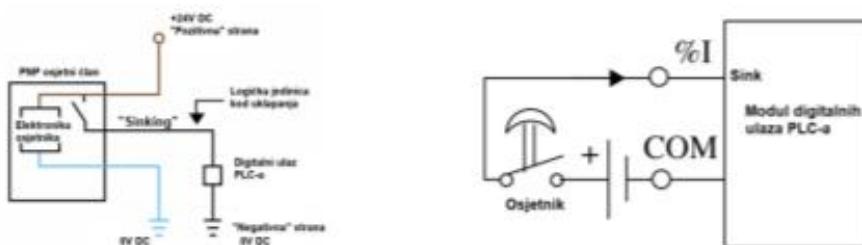


Položaj	0	1	2	3	4	5	6	7	8
Bit 1	0	1	1	0	0	1	1	0	0
Bit 2	0	0	1	1	1	1	0	0	0
Bit 3	0	0	0	0	1	1	1	1	1

13. Povezivanje logičkih senzora na PLC- sink način, shema.

Povezivanje logičkih senzora – Sink način (PNP)

- Smjer struje iz izlaza senzora u digitalni ulaz PLC-a
- Jedna strana digitalnog ulaza PLC-a unutar sebe je spojena na logičku nulu



Kod PNP i NPN izlaza senzora posebno je bitan pojam **polariteta** digitalnih ulaza PLC-a.

Postoje dvije opcije:

Pozitivna logika (PNP) – digitalni ulaz u *Sink* načinu rada.

Negativna logika (NPN) – digitalni ulaz u *Source* načinu rada.

Osnovna razlika između *Sink* i *Source* načina rada digitalnih ulaza jest smjer toka struje:

Sink (PNP) – Smjer struje iz izlaza senzora u digitalni ulaz PLC-a.

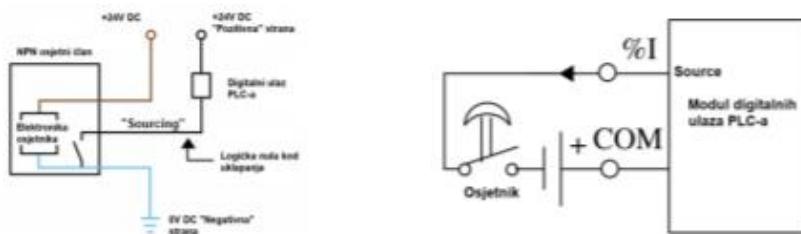
Jedna strana digitalnog ulaza PLC-a unutar sebe je zapravo spojena na logičku nulu.

Source (NPN) – Smjer struje iz digitalnog ulaza PLC-a u digitalni izlaz senzora. Jedna strana digitalnog ulaza PLC-a unutar sebe je zapravo spojena na logičku jedinicu.

14. Povezivanje logičkih senzora na PLC- source način, shema.

Povezivanje logičkih senzora – Source način (NPN)

- Smjer struje iz digitalnog ulaza PLC-a u digitalni izlaz senzora
- Jedna strana digitalnog ulaza PLC-a unutar sebe je spojena na logičku jedinicu



15. Načini povezivanja logičkih senzora.

- Izravnim fizičkim ožičenjem
- Upotrebom distribuiranih ulazno/izlaznih modula
- Upotrebom komunikacijske tehnologije IEC 61131-9 (IO-Link)

16. Navedite logičke izvršne članove.

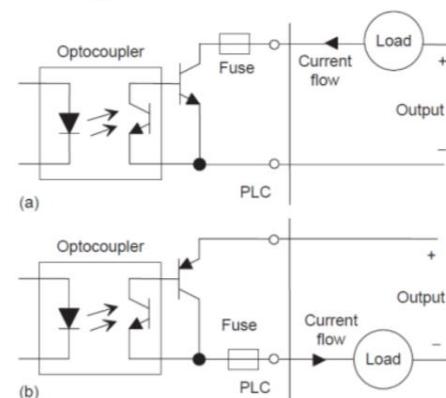
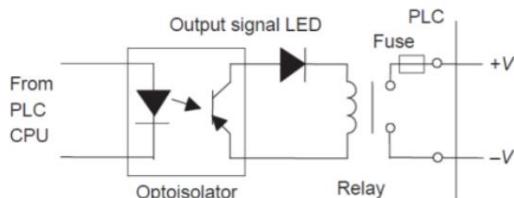
- Tipični logički izvršni članovi:
 - Releji
 - Sklopnići
 - Ventili
 - Svjetlosna i zvučna signalizacija

17. Razlika između releja i sklopnika. Koji se mogu spojiti na digitalni izlaz PLC-a?

- Releji (*Relay*) – ako je svitkom (zavojnicom) releja teče struja, magnetizira se jezgra te kontakti releja mijenjaju stanje
- Sklopnići (*Contactor*) – radi na sličnom principu kao relej, ali sadrži i energetske kontakte koji mogu podnijeti znatno veće iznose struje u odnosu na kontakte releja

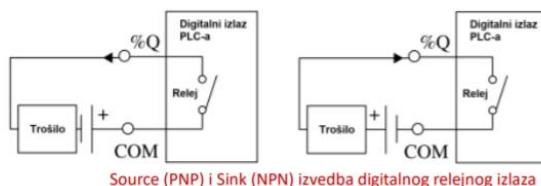
18. U kojoj izvedbi mogu biti digitalni izlazi PLC-a? Nacrtati shemu sa optičkim sprežnikom.

- Digitalni izlazi PLC-a, odnosno izlazi digitalnih modula, mogu biti:
 - Relejni
 - Tranzistorski

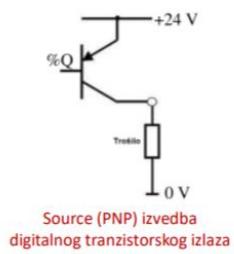


19. Sink i source izvedba povezivanja izvršnih članova.

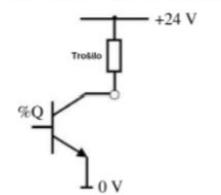
- Kod tranzistorske izvedbe digitalnih izlaza PLC-a treba paziti na način povezivanja s izvršnim članovima (slično kako i kod digitalnih ulaza):
 - *Sink (NPN)* izvedba – smjer toka struje je od izvršnog člana prema digitalnom izlazu PLC-a
 - *Source (PNP)* izvedba – smjer toka struje jest od digitalnog izlaza prema izvršnom članu



Source (NPN) i Sink (NPN) izvedba digitalnog relezjnog izlaza



Source (PNP) izvedba digitalnog tranzistorskog izlaza



Sink (NPN) izvedba digitalnog tranzistorskog izlaza

Analogni signali i senzori za mjerjenje

1. Razlika između analognog, digitalnog i binarnog signala

- Analogni signal je signal kojem se amplituda kontinuirano mijenja (kontinuirani signal) unutar definiranih granica
- Analogni signal ima definiranu vrijednost u svakom trenutku

2. Objasniti pretvorbu analognog u digitalni signal

- Proces analogne digitalne pretvorbe se sastoji od sljedećih koraka:
 1. Uzorkovanja signala u definiranim (konstantnim) vremenskim intervalima
 2. *Sample and hold*
 3. Kvantificiranje uzorka
 4. Kodiranje



3. Kad i zašto se filtriraju analogni signali

Potrebno je filtrirati mjereni signal kako bi se smanjio utjecaj šuma

4. Objasniti pojam rezolucije kod analogno-digitalne pretvorbe

Rezolucija analogno-digitalne pretvorbe je definirana brojem raspoloživih bitova za prikaz analogue vrijednost

5. Navesti tri standardna električna signala koji se primjenjuju kod senzora za mjerjenje analognih veličina

- Standardni električni signali senzora:
 - 4 – 20 mA (1 – 5 V, preko 250Ω otpornika)
 - 0 – 20 mA (0 – 5 V, preko 250Ω otpornika)
 - 0 – 10 V

6. Usporedba strujnih i naponskih analognih signala

- Najčešće se koristi strujni električni signal 4 – 20 mA
- Razlozi:
 - Naponski signali su više položni elektromagnetskoj semtrnji
 - Strujni signal se može prenijeti na većoj udaljenosti bez gubitaka
 - Donja granica od 4 mA omogućuje detekciju prekida u mjerenu

7. Razlika između dinamičke i statičke pogreške senzora.

Dinamička pogreška – pogreška tijekom tranzijenta mjerene veličine (odstupanje od stvarne vrijednosti tijekom tranzijenta)

Statička pogreška – pogreška mjerena u stacionarnom stanju (odstupanje od stvarne vrijednosti tijekom stacionarnog stanja)

8. Temperaturni senzori (navesti osnovne karakteristike svakog tipa senzora).

Termopar

- Baziran na činjenici kad se dvije žice različitog metala spoje, dođe do termalnog napona (*thermal voltage*) na mjestu spoja
- Tipično korišteni tipovi termoparova:

Type	α ($\mu\text{V}/^\circ\text{C}$)	Temperature range ($^\circ\text{C}$)	Metal (alloy)
J	50	0–750	Iron/Constantan
K	39	–200 to 1250	Chromel/Alumel
T	38	–200 to 350	Copper/Constantan



- Jednostavni, jeftini i dosta robusni
- Ne zahtijevaju zasebno napajanje pošto generiraju napon u ovisnosti o temperaturi
- Dosta brzo reagiraju na promjenu temperature
- Termalni napon je relativno nizak pa se zahtijeva pojačanje signala kako ne bi mjeri šum previše utjecao na mjerjenje
- Termopar se isporučuje sa žicama za povezivanje s PLC-om (ili drugim uređajem)

RTD – PT100 (NI1000)

- Svim metalima se mijenja otpor s temperaturom
- Na principu promjene otpora mjernog člana s temperaturom rade svi RTD mjeri senzori
- Najčešće korišteni su platina (PT) i nikal (NI)
- $\text{PT}100 = \text{PT}$ (platina) + 100Ω pri 0°C (pri -200°C otpor iznosi 18.53Ω , pri 100°C otpor iznosi 138.5Ω , a na 850°C otpor iznosi 390.38Ω)
- Iako se otpor ne mijenja linearno s promjenom temperature, promjena otpora s temperaturom u rasponu $0 - 100^\circ\text{C}$ se može aproksimirati s:

$$R[\Omega] = 100 + 3.85T[^\circ\text{C}]$$

- Često se koristi i NI1000 (= Ni (nikal) + 100 Ω pri 0°C)
- Za industrijsku primjenu PT100 se stavlja u kućište pa je vrijeme reakcije nešto sporije u odnosu na stvarnu promjenu temperature
- PT100 često uključuje i transmiter za pretvorbu u strujni signal od 0 do 20 mA



Termistor

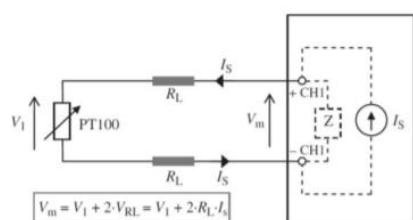
- Radi na sličnom principu kao i RTD (promjena otpora s temperaturom)
- Termistor se sastoji od poluvodičkih materijala, te se kod većine smanjuje otpor s povećanjem temperature (*negative temperature coefficient, NTC*)
- Postoje i PTC izvedbe (*positive temperature coefficient*)
- U odnosu na PT100, termistori su precizniji, ali i jako nelinearni
- Reagiraju brzo na promjenu temperature

9. Na što treba obratiti pažnju prilikom povezivanja analognih signala s PLC modulom (objasniti zašto).

- Smetnje u mernim signalima (šum) je neizbjeglan kod skoro svih industrijskih instrumenata
- Zbog toga potrebno je posebnu pažnju obratiti na:
 - Tip signalnih kabela
 - Oklop i uzemljenje signalnih kabela
 - Mjesto polaganja signalnih kabela

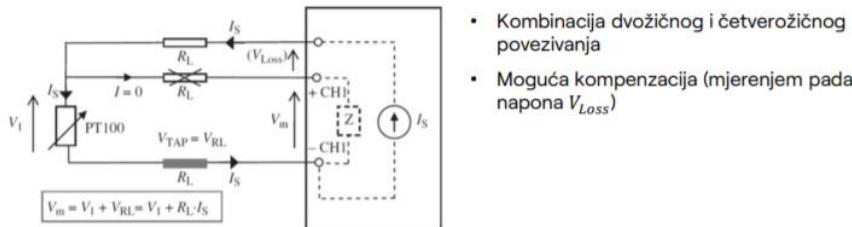
10. Načini povezivanja RTD mjernog člana.

Dvožično povezivanje RTD mjernih članova

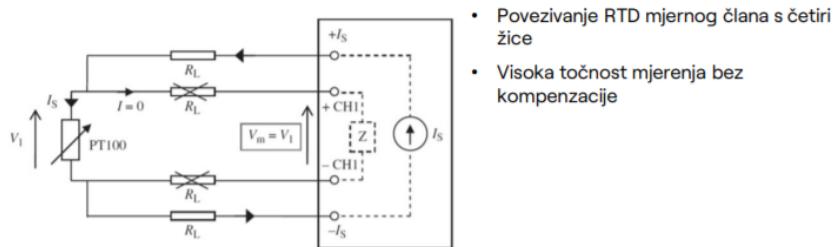


- Povezivanje RTD mjernog člana s dvije žice
- Zbog velike impedancije ulaza, ukupna struja I_s (internog strujnog izvora modula) prolazi kroz RTD
- Dužina signalnog kabela utječe na točnost mjerjenja (R_L)

Trožično povezivanje RTD mjernih članova



Četverožično povezivanje RTD mjernih članova



11. Standardni tipovi analognih izlaznih signala.

- Većina svojstva koja vrijede za ulazne analogne module vrijede i za izlazne analogne module
- Standardni tipovi signala analognih izlaznih modula:
 - 4 – 20 mA
 - 0 – 20 mA
 - 0 – 10 V
 - ± 10 V

12. Sink i source izvedba povezivanja izvršnih članova

?????

Brojevni sustavi, dijagrami toka i automati stanja

1. Navesti često korištene brojevne sustave u programiranju te objasniti razlike

Različiti zapisi digitalnih vrijednosti, odnosno brojevni sustavi se koriste u PLC-u. Brojevni sustavi koji se najčešće koriste u PLC-u su:

- binarni (BCD),
- heksadecimalni,
- oktalni i
- dekatski zapis.

2. Objasniti što je BCD

Dosta često se u digitalnim sustavima, poput PLC-a, binarni brojevi prikazuju kao cijelobrojne dekatske vrijednosti u formatu BCD-a (*Binary-Coded Decimal*). Način dobivanja BCD vrijednosti iz binarne vrijednosti je sličan načinu dobivanja heksadecimalne vrijednosti iz binarne vrijednosti. Binarni broj se podjeli u podgrupe od četiri znamenke (krenuvši od najmanje značajnog bita) te se svaka grupa od četiri znamenke zamjeni s ekvivalentnom dekatskom znamenkom (tablica 2). Za razliku od pretvorbe iz binarnog u heksadecimalni zapis, kod pretvorbe iz binarnog u BCD zapis postoje kombinacije četverobitnih vrijednosti koje nisu dozvoljene. To su kombinacije predstavljene heksadecimalnim znamenkama od A do F, odnosno dekatskim znamenkama od 10 do 15.

BCD kod	Decimalna vrijednost
0000	0
0001	1
0010	2
0011	3
0100	4
0101	5
0110	6
0111	7
1000	8
1001	9

3. Razlika između dijagrama toka i automata stanja

Procesi u kojima se radnje odvijaju određenim slijedom nazivaju se slijednim procesima (*sequential process*). Za prikaz slijeda radnji u slijednim procesima, često se koriste dijagrami toka (*flowchart diagram*). Osim za prikaz slijeda događaja u procesu, dijagrami toka se često koriste i za dokumentiranje načina rada slijednog procesa.

Dijagrami toka se mogu crtati na različite načine, ovisno o tome što se želi prikazati te koji se tipovi simbola koriste. Na slici 9 dani su neki od najčešće korištenih simbola u dijagramima stanja.

Automati, odnosno dijagram stanja prikazuju stanja sustava, veze između pojedinih stanja sustava te način i uvjete prelaska sustava iz jednog u drugo stanje. Prelasci između pojedinih stanja sustava definirani su trazicijama (*transitions*). Dobro je tijekom izrade dijagrama stanja sustava, pojedinim stanjima davati suvisla imena koja ukazuju na radnju koju sustav izvršava u tom stanju.

Sva stanja sustava, u dijagramu stanja, se označavaju s krugom u kojem se nalazi jedinstveno ime stanja. Oznaka stanja sustava može sadržavati i druge informacije (npr. stanja izlaza), ako to ne komplikira prikaz dijagrama stanja. Tipično se početno stanje sustava (*initial state*) označava s dvostrukim krugom. To je stanje u kojem sustav može postaviti odgovarajuće početne vrijednosti ili kalibrirati sustav te se čeka nalog za pokretanje. Svako stanje sustava ne treba izvršavati neku radnju nad sustavom. Može postojati stanje u kojem se ne odvija nikakva radnja, već se na primjer čeka da prođe specificirano vrijeme ili da aktivira senzor u sustavu.

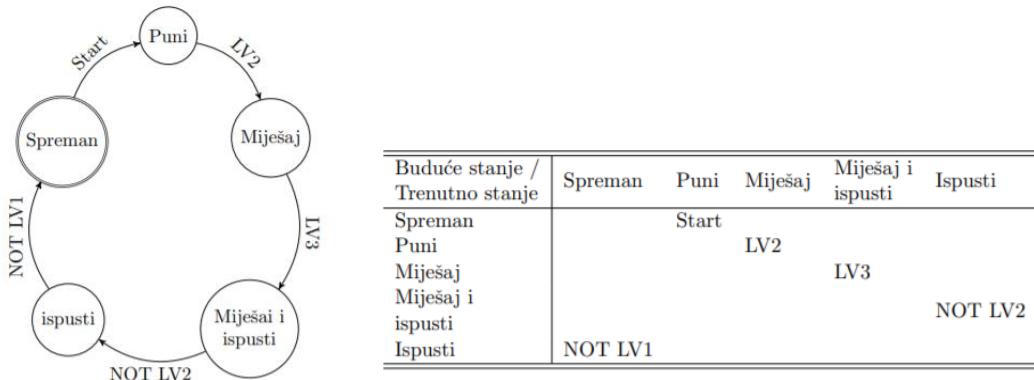
4. Nacrtati i objasniti najčešće korištene simbole u dijagramu toka



5. Navesti dvije najčešće korištene varijante automata stanja te objasniti razliku između istih

Algoritam upravljanja sustavom se može opisati pomoću skupa jedinstvenih stanja sustava bez postojanja definiranog slijeda radnji sustava. Za to se najčešće koriste automati stanja (*state diagram*), odnosno konačni automati (*finite-state diagram*). Postoje dvije varijante konačnih automata, *Moore-ov* i *Mealy-ov* model. *Moore-ov* model (koji se češće koristi) se definira radnje sustava, odnosno izlaze, na temelju stanja u kojem se nalazi sustav. Za razliku od *Moore-ovog* modela, *Mealy-ov* model definira izlaze sustava na temelju stanja sustava i ulaza u sustav. U nastavku su osnove rada s *Moore-ovim* modelom jer se on češće primjenjuje.

6. Iz zadanog dijagrama stanja napisati tablicu stanja (i obrnuto)



Umjesto dijagrama stanja proces se može opisati i preko tablice stanja (*state table*). U prvi redak i stupac tablice stanja postave se sva stanja, gdje prvi stupac predstavlja trenutno stanje sustava, a prvi redak buduće stanje sustava. Tranzicija između pojedinih stanja definirana je sjecištem odgovarajućeg retka (trenutno stanje) i stupca (buduće stanje). Tablica 3 predstavlja tablicu stanja za dijagram stanja dan na slici 14 (primjer 4).

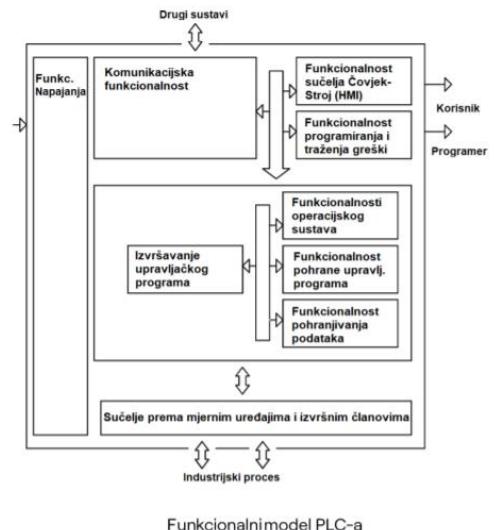
Norma IEC61131

1.Od kojih dijelova se sastoji norma IEC61131 i ukratko što predstavlja koji dio.

- **IEC 61131-1:2003 – Opće informacije** – Definicije, objašnjenja te identifikacija elemenata potrebnih za razumijevanje ostalih dijelova norme.
- **IEC 61131-2:2017 – Zahtjevi na opremu i ispitivanje** – Zahtjevi i ispitivanja PLC-ova te uređaja/sklopova usko vezanih uz PLC-ove kojima je cilj upravljanje procesima u industriji.
- **IEC 61131-3:2013 – Programske jezike** – Sintaksna i semantička specifikacija programskih jezika za PLC-ove.
- **IEC TR61131-4:2004 – Smjernice za korisnike** – Tehnički izvještaj (nema status norme, isključivo informativan) – Predstavlja smjernice za korisnike kod odabira i specifikacije PLC-ova te pripadajuće opreme.
- **IEC 61131-5:2001 – Komunikacije** – Daje prikaz te specifikaciju načina komunikacije PLC-ova s uređajima.
- **IEC 61131-6:2012 – Funkcionalna sigurnost** – Zahtjevi na PLC-ove te pripadajuće uređaje kod upotrebe unutar sigurnosnih instrumentacijskih sustava (SIS sustavi).
- **IEC 61131-7:2000 – Programiranje neizrazitog vođenja** – Definiranje programskog jezika za programiranje funkcija neizrazitog vođenja kod PLC-ova.
- **IEC TR61131-8:2017** – Preporuke za primjenu te implementaciju programskih jezika – Primjena norme IEC 61131-3
- **IEC 61131-9:2013 – Digitalno komunikacijsko sučelje jednospojnog voda za mala osjetila i aktuatore (SDCI)** – IO LINK tehničko komunikacijsko sučelje za povezivanje PLC-ova te pripadajuće opreme.
- **IEC 61131-10:2019 – PLC open XML format za razmjenu podataka** – Definicija eXtensible Markup Language (XML) formata za uvoz/izvoz programskih projekata napravljenih pomoću nekog od programskih jezika definiranih unutar IEC 61131-3 .

2. IEC61131-1 kratki opis, skicirati funkcionalni model PLC-a.

- **Norma IEC 61131-1:2003 – Opće informacije** predstavlja prvi dio norme.
- Unutar norme dan je opis i pregled temeljne funkcionalne strukture PLC-a, kao i osnovne funkcionalnosti:
 - Sučelje prema mjernim uređajima i izvršnim članovima – Pretvara ulazne/izlazne signale u oblike pogodne za PLC/mjerne uređaje i izlazne članove.
 - Komunikacijska funkcionalnost – Izmjena podataka između različitih sustava (PLC-ova, robota, računala i slično).
 - Sučelje čovjek-stroj (HMI) – Interakcija između korisnika te uređaja/procesa.
 - Funkcionalnosti programiranja, testiranja i dokumentiranja.
 - Funkcionalnosti vezane uz napajanje.

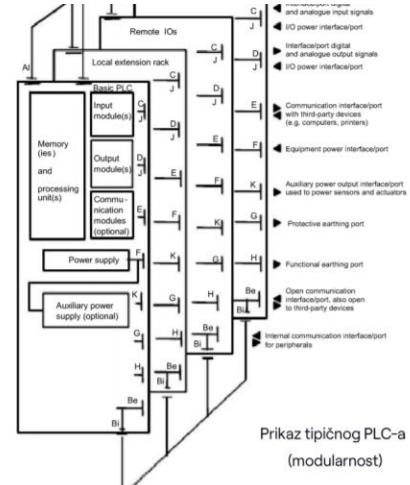


3. Ponovno pokretanje PLC-a u tri načina rada.

- Nakon isključenja napajanja PLC-a, moguće je ponovno pokretanje u tri načina rada:
 - **Hladno ponovno pokretanje** (engl. *Cold restart*) – Svi dinamički podatci (ulazni registri, timeri, brojači, zapisi stanja ulazno izlaznih modula) postavljaju se na predefinirano stanje.
 - **Toplo ponovno pokretanje** (engl. *Warm restart*) – Ponovno pokretanje kod kojeg ostaju podatci koje je korisnik odredio kao trajne, dok se svi ostali dinamički podatci postavljaju na predefinirano stanje.
 - **Vruće ponovno pokretanje** – Ponovno pokretanje kod kojeg stanja svih dinamičkih podataka ostaju nepromijenjena u odnosu na stanje prije nestanka napajanja PLC-a.

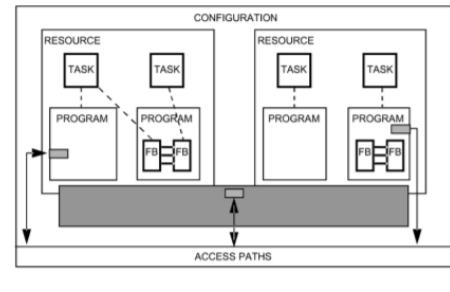
4. Koji zahtjevi se nalaze u normi IEC61131-2?

- **Norma IEC 61131-2:2017 – Zahtjevi na opremu i ispitivanje** predstavlja drugi dio norme.
- Unutar norme dan je pregled zahtjeva i načina ispitivanja PLC-ova te pripadajuće opreme u ovisnosti o mjestu i uvjetima uporabe.
 - Minimalni zahtjevi za uvjete okoline rada, prijenosa i pohranjivanja.
 - Funkcionalni zahtjevi na PLC-ove i pripadajuću opremu.
 - Zahtjevi vezani uz elektromagnetsku sukladnost.
 - Sigurnosni zahtjevi.
 - Nužne informacije koje proizvođač mora pružiti korisniku.
 - Metode ispitivanja verifikacije sukladnosti PLC-ova i pripadajuće opreme sa zahtjevima danim unutar norme.



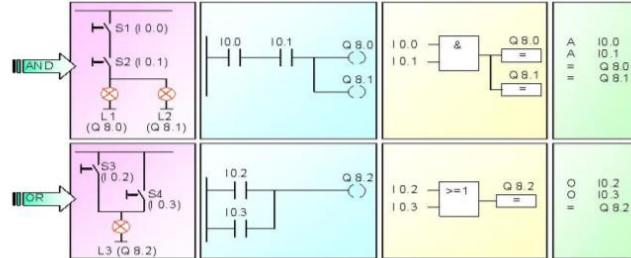
5. IEC61131-3, skicirati softverski model PLC-a.

- **Norma IEC 61131-3:2017 – Programske jezice** predstavlja treći dio norme.
- Unutar norme dana je specifikacija sintaksnih i semantičkih pravila programskega jezika za PLC-ove.
- Definiran je softverski model PLC-a (npr. Task-OB)
- Vrste podataka unutar PLC-a (binarni, cijelobrojni, realni podaci, polja podataka ...).
- Funkcije u ovisnosti o podacima (zbrajanje, oduzimanje, logičke operacije...)
- Više o dokumentu IEC 61131-3:2013 na sljedećem predavanju.



6. Koji programski jezici su definirani dijelom norme IEC61131-3?

- Definirana **dva tekstualna jezika**:
 - **Lista instrukcija** (engl. *Instruction List – IL*)
 - **Strukturirani tekst** (engl. *Structured Text – ST*)
- Definirana **dva grafička jezika**:
 - **Ljestvičasti dijagram** (engl. *Ladder Diagram – LD*)
 - **Funkcijski blokovski dijagram** (engl. *Function Block Diagram – FBD*)
- Dodatno definiran grafički jezik: **Sekvencijski funkcijalni dijagram** (engl. *Sequential Function Chart – SFC*)



7. Ukratko opisati IEC61131-5 i koji modeli su dani u tom dijelu?

- **Norma IEC 61131-5:2001 – Komunikacije** predstavlja peti dio norme.
- Unutar norme dana je specifikacija komunikacijskih aspekata PLC-a i pripadajuće opreme.
- Cilj norme jest specifikacija načina na koji bilo koji uređaj može komunicirati s PLC-om, kao i načina na koji PLC može komunicirati s bilo kojim uređajem.
- Normom se definira **komunikacijski i hardverski model PLC-a**, dok je kod funkcionalnog modela modela (definiran unutar norme IEC 61131-1) bitna komunikacijska funkcionalnost. Također, softverski model PLC-a definiran je unutar IEC 61131-3, a putem komunikacijske funkcionalnosti omogućuje se pristup programima, vrijednostima i konfiguracijskim datotekama.



8. Koja su dva parametra određena IEC61131-5 i ukratko ih opisati?

- Status PLC-a i pripadajućih sustava modelira se pomoću dva parametra:
- **Zdravlje** (engl. *Health*) – Parametar kojim se specificira stanje uređaja (postoje li problemi koji mogu onemogućiti izvršavanje ciljane funkcije/zadataka), a dijele se na tri moguća stanja:
 - **Dobro** (engl. *Good*) – PLC/Pripadajući uređaj nije detektirao neki problem koji bi onemogućio izvršavanje ciljane funkcionalnosti.
 - **Upozorenje** (engl. *Warning*) – PLC/Pripadajući uređaj nije detektirao neki problem koji bi onemogućio izvršavanje ciljane funkcionalnosti, ali je detektirao barem jedan problem koji bi mogao ograničiti funkcionalnost u nekom drugom slučaju. Ograničenje može biti vremensko, ograničenje mogućnosti i slično.
 - **Loše** (engl. *Bad*) – PLC/Pripadajući uređaj detektirao je neki problem koji onemogućava izvršavanje ciljane funkcije.
- **Stanje** (engl. *State*) – Lista binarnih parametara (1 ili 0) koji daju dodatne informacije o stanju PLC-a/pripadajućeg uređaja

9. IEC61131-6, što je rizik, koje su SIL razine, opis mjera za dostizanje razine.

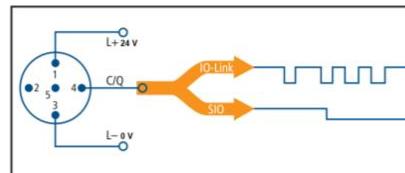
- **Norma IEC 61131-3:2017 – Funkcionalna sigurnost** predstavlja šesti dio norme
- Unutar norme dana je specifikacija PLC-ova i pripadajućih uređaja koji se koriste kao dio **sigurnosnih instrumentacijskih sustava (SIS)**.
- Sam dokument usko je vezan uz normu **IEC 61508** koja se bavi SIS sustavima, a namijenjen je proizvođačima PLC-ova s tom namjenom.
- SIS sustavi najčešće se sastoje od mjernih uređaja, izvršnih članova, softvera te upravljačkog sustava (unutar kojeg se nalazi PLC) čiji je zadatak dovođenje sustava u sigurno stanje (smanjenje rizika) u slučaju sigurnosne ugroze (greška, kvar opreme i slično).
- **Rizik** se definira kao vjerojatnost hazarda pomnožena s posljedicama takvog događaja.
- Posljedice hazardnog događaja moguće je mjeriti u različitim kategorijama (financijska, ugled, sigurnost, vrijeme popravka i slično).
- Količinu prihvatljivog rizika definira korisnik uz uzimanje u obzir niza faktora (zakonodavstvo, kritičnost sustava i slično).

- PLC unutar SIS sustava najčešće se naziva ***Fail Safety – FS-PLC***.
- Razina sigurnosti sustava definira se i opisuje pomoću četiri razine koje ovise o razini smanjenja rizika naziva ***Safety Integrity Level – SIL 1, 2, 3 i 4***.
- Dokument pokriva SIL razine do razine 3, dok je kod SIL-a razine 4 potrebno konzultirati normu **IEC 61508**.
- Cilj norme IEC 61131-6 je:
 - Uspostava i opis sigurnosnog životnog ciklusa FS-PLC-a u sukladnosti s IEC 61508.
 - Opis hardverskih i softverskih zahtjeva FS-PLC-a vezanih uz SIS sustave u skladu s ciljanom SIL razinom.
 - Specificiranje sigurnih stanja FS-PLC-a te ograničenja FS-PLC-a.
 - Opis svih mjera i tehnika za dostizanje određene razine SIL-a te uspostava evaluacijskih metoda za određivanje:
 - Sposobnosne razine SIL-a FS-PLC-a.
 - Određivanje vjerojatnosti kvara u slučaju potrebe za korištenjem (engl. *Probability of Failure on Demand – PFD*).
 - Određivanje vjerojatnosti kvara u vremenskom periodu od jedan sat (engl. *Probability of dangerous Failure per Hour – PFH*)
 - i ostalih veličina usko vezanih uz sigurnost.

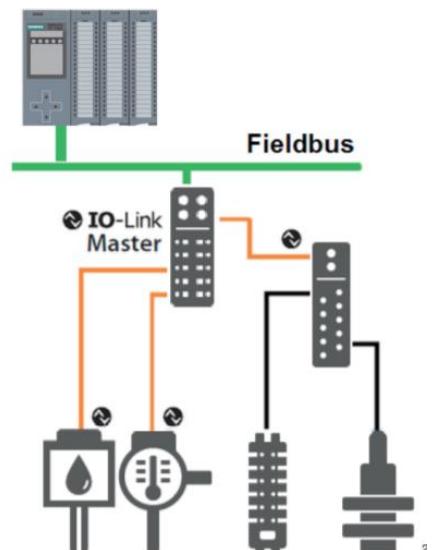
20

10. IEC61131-9, od čega se sastoji IO Link komunikacijski sustav, četiri osnovne vrste podataka i način pristupa.

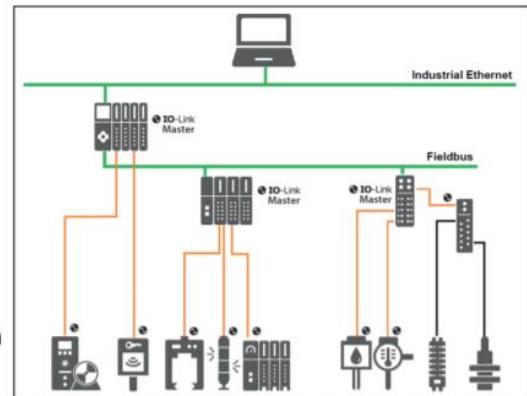
- **Norma IEC 61131-9:2013 – Digitalno komunikacijsko sučelje jednospojnog voda za mala osjetila i aktuator (SDCI)** predstavlja deveti dio norme.
- **SDCI** (komercijalni naziv **IO-Link**) predstavlja komunikacijsku tehnologiju za povezivanje mjernih uređaja te izvršnih članova preko pristupnika s nadređenim uređajem (PLC, HMI i slično).
- Prva komunikacijska tehnologija koja je prihvaćena kao međunarodna norma.
- **IO-Link – Serijska dvosmjerna** (engl. *Bi-directional*) komunikacijska tehnologija od **točke do točke** (engl. *Point-to-point*) **kratkog dosega**.
- IO-Link kabel – Trožilni ili peterožilni (slika).
- **Konektor** – Klasa A – Pin 2 i 5 nisu spojeni (4 pinski konektor).
Klasa B – Pin 2 i 5 standardno se koriste za napajanje uređaja (5 pinski konektor).



- IO-Link komunikacijski sustav se sastoji od sljedećih osnovnih komponenti:
 - **IO-Link master** – pristupnik (engl. *Gateway*) na koji se spajaju IO-Link uređaji i nadređeni uređaj/i. IO-Link uređaji priključuju se pojedinačno na IO-Link master putem IO-Link kabela, dok se nadređeni uređaj/i priključuju putem standardnih industrijskih komunikacijskih protokola (**Fieldbus** serijski protokoli – Profibus, Modbus serial, itd. te **Ethernet** – Profinet, Modbus TCP, itd.).
 - **IO-Link uređaj** (engl. *Device*) – mjerni uređaji, izvršni članovi, ulazno/izlazni moduli i slično.
 - **IO-Link kabel**
 - **IO-Link protokol** – 4 načina rada svakog priključka:
 - IO-Link – Standardna IO-Link dvosmjerna komunikacija.
 - DI – Priključak digitalnih ulaza.
 - DQ – Priključak digitalnih izlaza.
 - Deaktiviran – Priključak nije aktivan.



- Četiri osnovne vrste podataka:
 - Procesni podatci – Ciklički pristup
 - Stanja vrijednosti – Ciklički pristup
 - Podatci o uređaju – Ačiklički pristup
 - Događaji (alarmi i slično) – Ačiklički pristup
- Osim standardnog profila IO-Link komunikacijskog protokola, postoji i **IO-Link Wireless** profil kao proširenje IO-Linka na fizičkom sloju, a definiran je unutar standarda IEC 61139-2.
- Također, definiran je i **IO-Link Safety** kojim je moguća uporaba IO-Link tehnologije unutar SIS sustava.



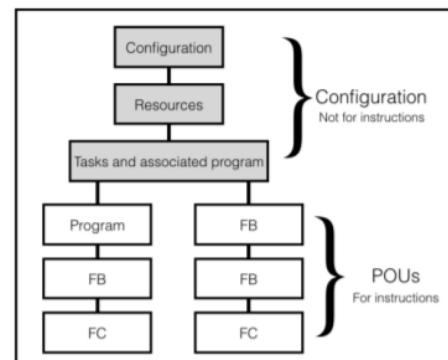
11. Ukratko opisati IEC61131-10.

- **Norma IEC 61131-10:2019 – PLC open XML format za razmjenu podataka** predstavlja deseti dio norme.
- Unutar norme predstavlja se definicija eXtensible Markup Language (XML) formata za uvoz/izvoz programske projekata napravljenih pomoću nekog od programskih jezika definiranih unutar IEC 61131-3.
- Namijenjen proizvođačima PLC-ova i razvoju programskih jezika.
- XML – opisni jezik za prijenos informacija (HTML za usporedbu služi za prikaz informacija).
- ```
<note>
 <to>osoba A</to>
 <from>osoba B</from>
 <heading>podsjetnik</heading>
 <body>ponesi PLC!</body>
</note>
```

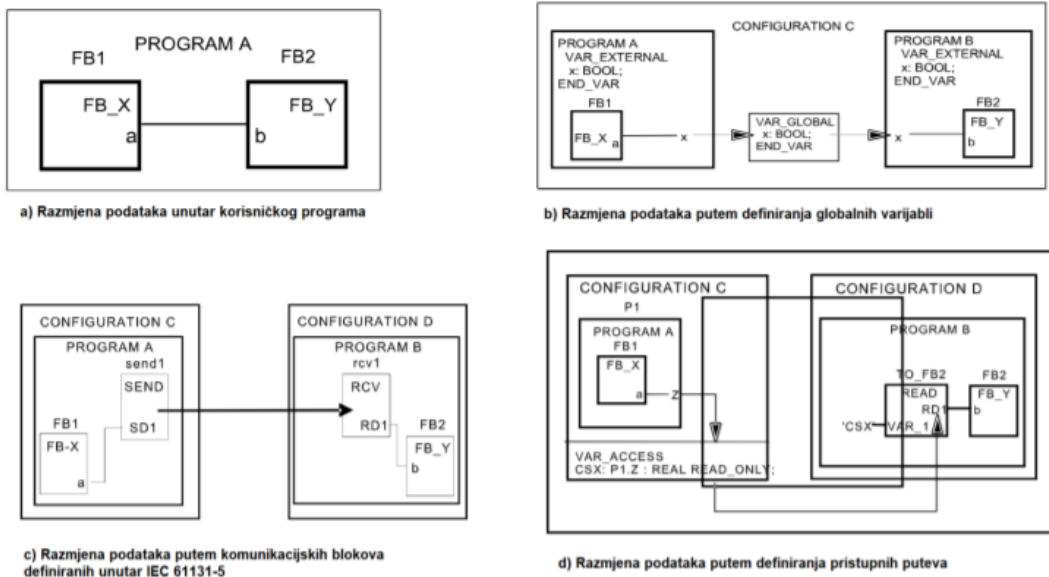
Note:  
To: osoba A  
From: osoba B  
Podsjetnik:  
ponesi PLC!
- Grafičkim i funkcionalnim opisom funkcija, funkcijskih blokova, elemenata, varijabli i svega povezanog uz programske projekte moguće je uvoziti/izvoziti projekte iz jednog programskog alata u drugi (univerzalnost programskih projekata).

## 12. IEC61131-3 softverski model.

- Configuration – struktura koja može imati više resursa
- Resource – signal processing function, obrada signala
- Task – zadatak, Siemens – OB
- FB – funkcijski blok
- FC – funkcija



### 13. IEC61131-3 komunikacijski model.



### Načini razmjene podataka između softverskih elemenata

#### 14. IEC61131-3 model programiranja.

- Model programiranja se sastoji od pravila za pozivanje i korištenje POU-ova u programskoj strukturi:
- 1. Programi moraju biti deklarirani. Unutar njih mogu se koristiti standardni tipovi ili korisnički tipovi (UDT) podataka, funkcije (FC), funkcionalni blokovi (FB) i klase.
- 2. Program se može izvesti u različitim konfiguracijama. Elementi se mogu koristiti kao globalne varijable, resursi, zadaci i pristupni putovi.
- 3. Mogu se deklarirati tipovi funkcionalnih blokova. Unutar njih se mogu koristiti standardni ili UDT podaci, funkcije ili drugi funkcionalni blokovi. Osim toga, objektno orijentirani FB i klase moraju biti definirane kako bi se koristile metode i sučelja.
- 4. Funkcije se mogu deklarirati. Unutar njih, mogu se koristiti standardni ili UDT tipovi podataka, standardne ili korisnički definirane funkcije.
- 5. Tipovi podataka moraju biti deklarirani.

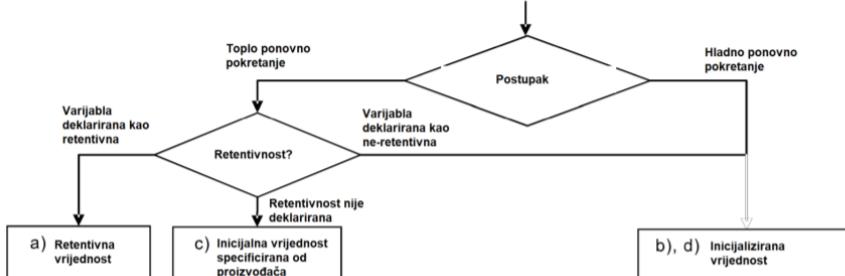
#### 15. IEC61131-3 varijable.

- Varijable predstavljaju veličine kojima se pridodaju vrijednosti.
- Kod PLC-ova, varijablama je potrebno dodjeliti i neki od elemenata vrste podataka.
- Deklaracija varijable može se ostvariti koristeći:
  - Elementarne vrste podataka.
  - Vrste podataka definirane od korisnika.
  - Pokazivače na ciljane podatke (ili funkcije/funkcijske blokove).
- Svakoj varijabli moguće je postaviti početnu vrijednost:
  - Definirane su standardne početne vrijednosti elementarnih vrsta podataka.
  - Null vrijednost kod pokazivača.
  - Vrijednost definiranu od korisnika.
- Korisnik može unutar korisničkog programa prikazivati varijable i direktno (pozivajući se na stvarnu logičku lokaciju unutar memoriskog prostora).

BR.	Opis	Primjer	Objašnjenje
<b>Memorijsko područje</b>			
1	Ulazi	I	Uzlazna riječ 215
2	Izlazi	Q	Izlazni bajt 7
3	Interna memorija	M	Dupla riječ na memoriskoj lokaciji 48
<b>Veličina</b>			
4a	Jedan bit	X	Uzlazna vrsta podataka BOOL
4b	Jedan bit	Nema označu	Uzlazna vrsta podataka BOOL
5	Bajt (8 bitova)	B	Uzlazna vrsta podataka BYTE
6	Riječ (16 bitova)	W	Uzlazna vrsta podataka WORD
7	Dupla riječ (32 bita)	D	Uzlazna vrsta podataka DWORD
8	Duga riječ (64 bita)	L	Uzlazna vrsta podataka LWORD

## 16. IEC61131-3 retentivnost

- Varijablama je kod deklaracije moguće odrediti **retentivnost** (engl. *Retentiveness*) – ponašanje vrijednosti varijabli kod ponovnog pokretanja PLC-a:
  - Hladno ponovno pokretanje** (engl. *Cold restart*) – Svi dinamički podatci (ulazni registri, timeri, brojači, zapisi stanja ulazno izlaznih modula) postavljaju se na početnu inicijaliziranu vrijednost.
  - Toplo ponovno pokretanje** (engl. *Warm restart*) – Ponovo pokretanje kod kojeg ostaju podaci koje je korisnik definirao kao retentivne, dok se svi ostali dinamički podatci postavljaju na početnu inicijaliziranu vrijednost.
  - Uz kvalifikator retentivnosti, varijablu je moguće deklarirati kao konstantu (nije joj moguće promijeniti vrijednost).



## 17. IEC61131-3 organizacijske jedinice programa.

- Organacijske jedinice programa** (engl. *Program organization units – POU*) predstavljaju najmanje neovisne dijelove korisničkog programske rješenja.
  - Svako programsko rješenje mora se sastojati od barem jednog POU-a.
  - POU** jedinice mogu biti stvorene od korisnika, ali i od proizvođača.
  - Svaka POU može pozvati drugu POU, kao i prenosi jednu ili više veličina prema pozvanoj organizacijskoj jedinici. Postoje tri osnovne vrste POU-a:
    - Funkcije (FC)** – Osnovna i najjednostavnija vrsta POU-a. Glavna značajka funkcija je da funkcija kod svakog poziva za jednake parametre daje jednaki izlaz (funkcijsku vrijednost). Navedena funkcionalnost funkcija implicira nepostojanje unutarnje memorije funkcija.
    - Funkcijski blokovi (FB)** – Druga POU po složenosti. U usporedbi sa funkcijama, funkcijski blok kod svakog poziva za jednake parametre ne mora davati jednaki izlaz – postoji unutarnja memorija kod koje je moguće spremati unutarnje vrijednosti.
    - Programi** – Najkompleksnija POU koja se sastoji od adresa, varijabli, konstanti, funkcija, funkcijskih blokova te upravljačkih algoritama koji kao jedinica upravljaju nekim tehničkim procesom.
- Iznad POU-ova postoje **zadaci** (engl. *Task*) – (Siemens – OB) koji mogu (periodično, ovisi o potrebi) pokrenuti izvršavanje seta programa i funkcijskih blokova.

## 18. IEC61131-3 koji su osnovni parametri funkcije?

**Parametri funkcije** – Variabile koje funkcija prima, odnosno daje kao rezultat izvršavanja. Funkcija dobiva isključivo „Sliku“ vrijednosti koje se dodjeljuju parametrima funkcije. Funkcija ne pristupa direktno memorijskim lokacijama dodijeljenih varijabli, već na početku zapisuje vrijednosti varijabli dodijeljenih na ulazu, odradjuje zadane radnje i zapisuje stanja na memorijskim lokacijama izlaznih varijabli. Nakon izvršavanja funkcije, sva unutarnja stanja funkcije se brišu (funkcija nema unutarnju memoriju). Osnovni parametri funkcije su:

- Input** (ulaz) – Parametar funkcije putem kojeg funkcija može primati vrijednosti pomoću kojih obavlja neku radnju (npr. funkcija za zbrajanje kao ulaz prima parametre koje je potrebno zbrojiti).
- Output** (izlaz) – Parametar funkcije putem kojeg funkcija predaje vrijednost (rezultat) određene radnje.
- InOut** (Ulaz/Izlaz) – Parametar funkcije koji po izvršavanju funkcije zapisuje rezultat funkcije na memorijsku lokaciju varijable dodijeljene na ulazu. Usporedba parametara Input, Output te InOut na sljedećoj prikaznici.
- Temp** (privremena vrijednost) – Interni parametar funkcije kojim se definiraju privremene vrijednosti kod izvršavanja funkcije (npr. međurezultati aritmetičkih operacija i slično).
- Constant** (konstante) – Interni parametar funkcije kojim se definiraju konstantne vrijednosti (npr. Konstanta Pi, e i slično). Doprinose preglednosti programske koda.
- Return** – Parametar koji funkcija vraća po izvršavanju. Moguće ga je koristiti za zapisivanje koda greške u slučaju neke logičke pogreške (npr. dodijeljeni parametri ulaza manji od 0, a funkcija se koristi za korjenovanje).

Izvršavanje_koraka_FC			
Name	Data type	Default value	Comment
1 <input>	Input		
2 <output>	Output		
3 <inout>	InOut		
4 <temp>	Temp		
5 <const>	Constant		
6 <return>	Return		

Parametri funkcije

## 19. Razlika između funkcije i funkcionskog bloka.

- **Funkcijski blokovi (FB)** – Druga POU po složenosti.
- Imaju unutarnju memoriju – Jednaki ulaz -> Jednaki izlaz (**ovisnost o prethodnim pozivima**)
- Uz standardne funkcione blokove, korisnik može napraviti i svoje funkcije uz deklaraciju ulaza, izlaza, unutarnje funkcionalnosti, unutarnje memorije i svih potrebnih parametara.
- **Standardni funkcijeski blokovi** (uobičajeni za svaki programske jezik, kao i za sve proizvođače PLC-ova):
  - **Funkcijski blokovi bistabila** (engl. *Flip-Flop*) (RS, SR bistabili) – Sklopovi koji pamte podatak veličine jednog bita.
    - Tablica stanja SR bistabila dana je slijekom:
    - SR bistabil – dominantan je S (set) bit
    - RS bistabil – dominantan je R (reset) bit

S	R	$Q_n$	$Q_{n+1}$
0	0	0	0
0	0	1	1
0	1	X	0
1	0	X	1
1	1	X	1
  - **Detekcija rastućeg i padajućeg brida signala** (engl. *Rising-Falling Edge*) (R\_TRIG, F\_TRIG)
  - **Funkcijski blokovi timera** (npr. TON, TOF, TP)
  - **Funkcijski blokovi brojača** (engl. *Counters*) (npr. CTU, CTD, CTUD)

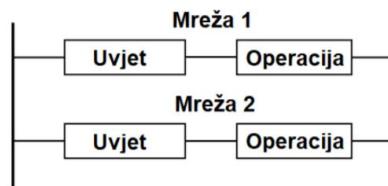
Funkcijski_blok				
	Name	Data type	Default value	Retain
1	Input			
2	Output		E	
3	InOut			
4	Static			
5	Temp			
6	Constant			

Parametri funkcijeskog bloka

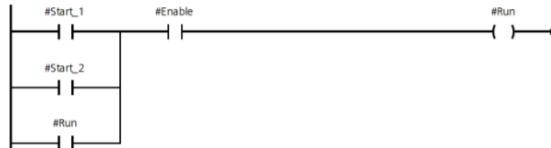
- **Parametri funkcijeskih blokova** – Varijable koje funkcijeski blok prima, odnosno daje kao rezultat izvršavanja. Funkcijeski blok dobiva isključivo „Sliku“ vrijednosti koje se dodjeljuju parametrima funkcijeskog bloka. Funkcijeski blok ne pristupa direktno memorijskim lokacijama dodijeljenih varijabli, već na početku zapisuje vrijednosti varijabli dodijeljenih na ulazu, odrađuje zadane radnje i zapisuje stanja na memorijskim lokacijama varijabli dodijeljenih na izlazu. Nakon izvršavanja funkcijeskog bloka, **unutarnja stanja parametara Input, Output, InOut te Static se pohranjuju** u unutarnjoj memoriji funkcijeskog bloka, dok se vrijednosti parametara **Temp brišu**.
- Kod ponovnog poziva funkcijeskog bloka, stanja dodijeljenih varijabli na ulazu zapisuju se na memorijске lokacije parametara Input te InOut. Nakon izvršavanja funkcijeskog bloka rezultat spremlijen u parametrima Output te InOut zapisuje se na memorijске lokacije dodijeljenih varijabli.
- U usporedbi s funkcijama, funkcijeski blokovi imaju jedan novi parametar:
  - **Static** – Parametar funkcijeskog bloka kod kojeg nakon izvršavanja funkcijeskog bloka ostanu zapisane vrijednosti. Parametar vrste Static posebno je koristan kod iterativnih postupaka te postupaka kod kojih je potrebno pamtitи prethodno stanje (npr. Brojači, timeri i slično).

## 20. Opiši programske jezik LAD.

- Norma definira mrežu (engl. *Network/Rung*) kao maksimalan set priključenih grafičkih elemenata (kontakata, zavojnica, blokova) između lijeve i desne „šine za napajanje“ (engl. *Power rails*)
- Stanje lijeve šine za napajanje – uvijek logička jedinica.
- „Tok energije“ (engl. *Power flow*) – Analogan toku energije u elektromehaničkim sustavima – kod ljestvičastog dijagrama definira se kao tok s lijeva na desno.
- U slučaju ispunjenja uvjeta, lijeva šina za napajanje priključuje se sa blokom Operacija i time se izvršava Operacija.



- Pravila za **izvršavanje i evaluaciju** pojedine mreže:
  - Niti jedan element (funkcija, funkcionalni blok) ne smije biti izvršen/evaluiran prije postavljanja svih njegovih ulaza (Stanja ulaza ne smiju biti nedefinirana).
  - Evaluacija segmenta mreže ne smije završiti sve dok se sva stanja izlaza ne evaluiraju/izvrše, čak i ako postoje elementi za upravljanje izvršavanjem (JUMP naredba kod grafičkih jezika) (nema prelaska na drugu mrežu sve dok se ne postave izlazi).
  - Redoslijed izvršavanja mreža kod ljestvičastog dijagrama ostvaruje se odozgo prema dolje.
- **Povratne veze** unutar petlji:
  - **Eksplisitne povratne veze** pojavljuju se isključivo u FBD programskom jeziku.
  - Redoslijed izvršavanja eksplisitnih povratnih veza ovisan je o proizvođaču PLC-ova (i razvojnih alata).
  - Potrebno je **inicijalizirati** sve varijable.



## 21. Opiši programski jezik FBD.

- Funkcijski blokovski dijagram (engl. *Function Block Diagram – FBD*)
- Jedan od grafičkih jezika za izradu upravljačkih programa PLC-a, a prikaz elemenata temelji se na normi **IEC 60617-12: Grafički simboli za dijagrame**.
- Sam jezik temelji se na konceptu „**toka informacija/signala**“ – toku analogan toku signala kod elektromehaničkih sustava između elemenata za obradu signala.
- Koncept „toka signala“ kod FBD-a drugačiji je od koncepta „toka energije“ kod ljestvičastog dijagrama – informacije koje teku putem spojnih linija kod FBD-a mogu biti različite (INT, REAL, BOOL, itd.), dok kod „toka energije“ mogu biti isključivo binarne (BOOL – ima energije, nema energije).
- Mreža se kod FBD jezika može sastojati od:
  - spojnih linija.
  - grafičkih elemenata za upravljanje izvršavanjem programa (naredbe JUMP i RETURN).
  - grafičkih elemenata za pozivanje funkcija i funkcionalnih blokova.
- Prema normi, tok signala te redoslijed izvršavanja POU mreža ovisi o proizvođaču.
- Najčešće vrijede sljedeća pravila:
  - **Tok signala – s lijeva na desno.**
  - **Izvršavanje POU mreža – Odozgo prema dolje.**
- Spojne linije mogu biti horizontalne i vertikalne.
- Unutar FBD-a ne postoje lijeve i desne „Šine za napajanje“.
- Unutar jezika FBD, funkcije i funkcionalni blokovi predstavljaju se **jednako** kao i kod ljestvičastog dijagrama.
- Korisnik jezika FBD ima pristup svim standardnim funkcijama/FB (timeri, brojači, itd.), kao i osnovnim logičkim funkcijama (Funkcija I, ILI, NE, SR bistabil, RS bistabil) u obliku funkcionalnih blokova/funkcija.
- Moguća je **direktna konverzija** između programskega jezika FBD i ljestvičastog dijagrama.

## 22. Opiši programski jezik IL.

- **Lista instrukcija** (engl. *Instruction List – IL*) predstavlja jedan od tekstualnih jezika definiranih unutar norme.
- Podsjeća na asemblerSKI jezik (jezik niske apstrakcije).
- Norma navodi kako je jezik **zastario** (engl. *Outdated*) te kako se u sljedećem izdanju norme IEC 61131-3 **neće više pojavljivati**.
- Lista instrukcija sastoji se od sekvencija naredbi.
- Svaka naredba započinje u novom retku, a sastoji se od operatora s opcionalnim modifikatorima te u slučaju potrebe određene operacije.
- Proizvođač PLC-ova Siemens naziva programski jezik Lista instrukcija kao **Statement List – STL**.

Labela	Operator	Operand	Komentar
START:	LD	%IX1	(* PUSH BUTTON *)
	ANDN	%MX5	(* NOT INHIBITED *)
	ST	%QX2	(* FAN ON *)

### 23. Opiši programski jezik ST

- **Strukturirani tekst** (engl. *Structured Text – ST*) predstavlja jedan od tekstualnih jezika definiranih unutar norme.
- Programski jezik više apstrakcije – temeljen na programskom jeziku PASCAL.
- Sadrži sve funkcije i funkcionalnosti modernog programskega jezika (FOR, WHILE i REPEAT za upravljanje izvođenjem te funkcije IF-THEN-ELSE za upravljanje grananjem).
- Posebno koristan kod procesiranja naprednih matematičkih i aritmetičkih izraza .
- Proizvođač PLC-ova Siemens naziva programski jezik Strukturirani tekst kao **Strukturirani upravljački jezik** (engl. *Structured Control Language – SCL*).

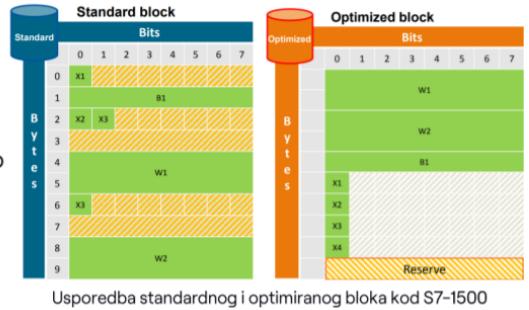
```
WHILE index1 < 100 DO
 index2 := 0;
 Value := Table1[index1];
 REPEAT
 Table2[index2] := Value + Table2[index2];
 IF Table2[index2] > 32767 THEN
 EXIT;
 END_IF;
 index2 := index2 + 1;
 UNTIL index2 >25
 END_REPEAT
 index1 := index1 + 1;
END WHILE
```

### 24. Opiši programski jezik SFC.

- **Sekvencijalni funkcijski dijagram** (engl. *Sequential Function Chart – SFC*) predstavlja jedan od tri grafičkih programskega jezika definiranih unutar IEC 61131-3.
- SFC nastao je specifikacijom **GRAFCET** jezika za funkcionalni opis sekvenčnih upravljačkih algoritama definiranog unutar **IEC 60848:2013** u oblik pogodan za PLC programiranje.
- SFC omogućuje podjelu POU-ova PLC-a na **set koraka i prijelaza** spojenih usmjerenim vezama.
- Svaki korak ima pridijeljen niz akcija, dok je prelazak između koraka definiran uvjetima prijelaza. Akcije koraka moguće je definirati putem ostalih programskega jezika (LD, FBD, ST, IL), kao i prelaska između koraka.
- Zbog potrebe za pamćenjem trenutnog stanja, u jeziku SFC moguće je programirati isključivo funkcijске blokove.
- Proizvođač PLC-ova Siemens naziva programski jezik Sekvencijalni funkcijski dijagram kao **S7 GRAPH**.
- **Sekvencijalni funkcijski dijagram** (engl. *Sequential Function Chart – SFC*) predstavlja jedan od tri grafičkih programskega jezika definiranih unutar IEC 61131-3.
- SFC nastao je specifikacijom **GRAFCET** jezika za funkcionalni opis sekvenčnih upravljačkih algoritama definiranog unutar **IEC 60848:2013** u oblik pogodan za PLC programiranje.
- SFC omogućuje podjelu POU-ova PLC-a na **set koraka i prijelaza** spojenih usmjerenim vezama.
- Svaki korak ima pridijeljen niz akcija, dok je prelazak između koraka definiran uvjetima prijelaza. Akcije koraka moguće je definirati putem ostalih programskega jezika (LD, FBD, ST, IL), kao i prelaska između koraka.
- Zbog potrebe za pamćenjem trenutnog stanja, u jeziku SFC moguće je programirati isključivo funkcijске blokove.
- Proizvođač PLC-ova Siemens naziva programski jezik Sekvencijalni funkcijski dijagram kao **S7 GRAPH**.

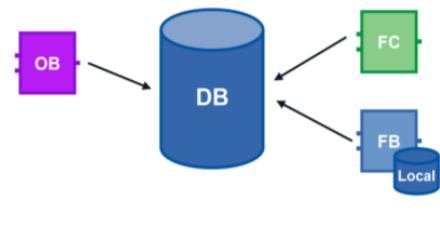
### 25. Vezano uz blok podataka opisati optimirano pohranjivanje podataka.

- S7-1200/1500 kontroleri podržavaju **optimirano pohranjivanje podataka** (engl. Optimized blocks) – Svi pohranjeni podaci sortiraju se obzirom na vrstu podataka.
- Sortiranje osigurava da su prazni prostori između različitih podataka minimalni, kao i da se podatci spremaju na način da se omogućava optimalan pristup procesoru podatcima u memoriji.
- Neoptimirano pohranjivanje podataka podržano je isključivo zbog **kompatibilnosti**.
- Usporedba blokova podataka kod PLC-a S7-1500 dana je slikom.
- Svojstva **optimiranih** blokova podataka S7-1500:
  - Smanjuju se praznine u memorijskom prostoru (veći blokovi podataka postavljaju se na početak).
  - CPU PLC-a S7-1500 istovremeno čita **4 bajta podataka**.
  - Bitovi podataka pohranjuju se **kao bajtovi** – omogućuje se brži pristup (nije potrebno maskiranje za pristup određenom bitu).
  - Pristup podatcima kod optimiranih blokova moguć je isključivo simbolički (putem naziva varijable, a ne absolutne adrese).
  - Kod PLC-a S7-1500 i optimiranih blokova, podaci se zapisuju u **Little Endian** formatu (standardni blokovi – **Big Endian** format).



## 26. Globalni blokovi podataka.

- Također, kod korištenja varijabli i podataka koje je potrebno koristiti u više različitih funkcija/funkcijskih blokova, preporuka je pohranjivanje vrijednosti u **globalne blokove podataka** (engl. Global Data Blocks – GlobalDB).
- Standardno, podatke je moguće pohranjivati izravno u memorijski prostor (npr. Bit %M0.0).
- Izravna pohrana podataka se **ne preporuča** – Preporuka jest **simboličko adresiranje** (pohrana putem globalnih blokova podataka).
- Prednosti globalnih blokova podataka u odnosu na čisti memorijski prostor PLC-a:
  - Preciznija struktura** programske rješenja.
  - Čitljivost koda** – simboličko adresiranje.
  - Brže čitanje i zapisivanje** (optimirani blok podataka).
  - Moguće ponovno korištenje funkcija, funkcijskih blokova i programskih rješenja na drugim PLC-ovima (programska rješenja neovisna su od hardvera na kojim se izvode).

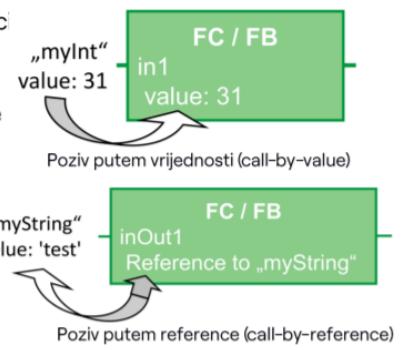


## 27. Lokalni blokovi podataka.

- Lokalni blokovi podataka** – blokovi podataka vezani uz određeni funkcijski blok.
- Sam poziv funkcijskog bloka naziva se instanca.
- Kod poziva funkcijskog bloka stvara se i lokalni blok podataka u koje funkcijski blok pohranjuje podatke.
- U slučaju poziva funkcijskog bloka unutar funkcije ili organizacijskog bloka (OB), automatski se stvara i posebni blok podataka za pripadajući funkcijski blok.
- U slučaju poziva funkcijskog bloka unutar nekog drugog funkcijskog bloka, korisnik ima tri mogućnosti:
- Single instance, Multi instance i Parameter instance** – Na OOP predavanju.

## 28. Prijenos vrijednosti u POU.

- U prethodnom predavanju dan je pregled mogućih ulaznih parametara funkcija i funkcijskih blokova.
- Parametri **In**, **Out** te **InOut** – Potrebne vrijednosti pružaju se POU-ovima putem parametara **In** te **InOut**, a rezultat se vraća putem parametara **Out** te **InOut**.
- Dvije osnovne opcije za prijenos podataka putem parametara:
  - **Poziv putem vrijednosti** (engl. *Call-by-value*) – vrijednost ulaznog parametra kopira se na formalni parametar unutar bloka (alocira se dodatni memorijski prostor).
  - **Poziv putem reference** (engl. *Call-by-reference*) – ulazni parametar prenosi referencu na memorijski prostor stvarne varijable (nema alociranja dodatnog memorijskog prostora).
- Zbog optimiranja memorijskog prostora, alat TIA Portal automatski koristi poziv putem reference kod određenih vrsta podataka.
- Strukture mogu biti iznimno velike – preporka je korištenje parametra **InOut** kod prijenosa **velikih struktura** funkcijskim blokovima (ne alocira se dodatno velik memorijski prostor).

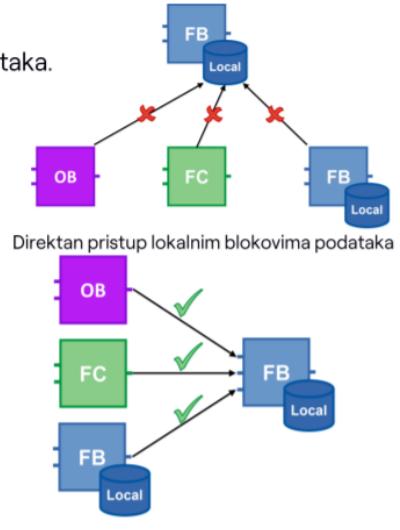


Programska organizacijska jedinica POU	Elementarni PLC podatci	Strukture
FC	Input	Call by value
	Output	Call by value
	InOut	Call by value
FB	Input	Call by value
	Output	Call by value
	InOut	Call by value

14

## 29. Pristup lokalnim blokovima podataka.

- Strogo **izbjegavati direktni pristup** drugim lokalnim blokovima podataka.
- Koncept memorijskog prostora unutar alata TIA Portal (lokalni blokovi podataka, globalni blokovi podataka) omogućuju modularnost, efikasno upravljanje i jednostavnije čitanje i testiranje programskih rješenja.
- Enkapsulacijom POU-ova omogućuje se **ponovna upotreba** programskih rješenja kod različitih programskih zadataka.
- Podatke koji su potrebni većem broju POU-ova pohranjivati u globalne blokove podataka.
- Kod potrebe za razmjrenom podatka između POU-ova, potrebno je pristupati podatcima putem **ulazno/izlaznih sučelja POU-a** (parametara **In**, **Out** te **InOut**), a ne direktno (lokalnom bloku podataka).
- Pristupanjem podatcima putem sučelja POU-a, osigurava se neovisnost blokova (brisanje jednog POU-a ne utječe na drugi POU).



## 30. Preporuke kod programskog jezika SCL.

- Implementacija složenih matematičkih funkcija i operacija može biti složena kod korištenja programskih jezika FBD i LAD – Alat TIA Portal dozvoljava **umetanje i izradu mreža** u programskom jeziku **SCL (ST)**, omogućujući na taj način **jednostavniju implementaciju** složenih matematičkih funkcija.
- Programski jezik SCL podržava jednostavno ubacivanje gotovih funkcija i funkcijskih blokova putem **povuci i spusti** (engl. *Drag and drop*) funkcionalnosti.
- Smanjuje se vrijeme potrebno za ispisivanje svih ulaznih i izlaznih parametara POU-a.

- FOR, WHILE i REPEAT petlje.
- Razlike u načinu izvođenja
- **FOR** petlja – izvodi se **stogo definirani** broj puta. Zbog optimiranja performansi PLC-a, broj izvođenja petlje definira se **u prvom pozivu petlje**. Naredbom EXIT moguće je prekinuti petlju.
- **WHILE** petlja – izvodi se do ostvarenja uvjeta za terminiranje petlje. Uvjet za terminiranje petlje provjerava se prije izvođenja petlje – U slučaju da uvjet u prvom pozivu nije ispunjen, petlja se **neće izvesti**.
- **REPEAT** petlja – izvodi se do ostvarenja uvjeta za terminiranje petlje. Uvjet za terminiranje petlje provjerava se nakon izvođenja petlje – Petlja će se izvesti **barem jednom** (čak i ako uvjet nije ispunjen).
- Korištenje **FOR** petlje preporuča se kad je broj koraka izvršavanja petlje **stogo definiran** (npr. Iteriranje svih članova polja, itd.).
- Korištenje **WHILE** i **REPEAT** petlje preporuča se kad je potrebno **mijenjati broj izvođenja** unutar petlje (broj izvođenja nije poznat u trenutku poziva).

```

FOR statCounter := statStartCount TO statEndCount DO
 // Statement section ;
END_FOR;
 FOR petlja

WHILE condition DO
 // Statement section ;
END_WHILE;
 WHILE petlja

REPEAT
 // Statement section ;
UNTIL condition
END_REPEAT;
 REPEAT petlja

```

- Izbjegavanje korištenja nepotrebnih IF funkcija.
- Programerski način razmišljanja – IF-THEN-ELSE.
- Preporuka je izbjegavati IF-THEN-ELSE konstrukcije ako je moguće ostvarivanje funkcija pomoću Booleove algebre.

```

IF (statOn1 = TRUE AND statOn2 = TRUE) THEN
 statMotor := TRUE;
ELSE
 statMotor := FALSE;
END_IF

```

||  
IF-THEN-ELSE konstrukcija logičke funkcije

```

statMotor := statOn1 AND statOn2;
Booleova konstrukcija logičke funkcije

```

### 31. Globalne i lokalne konstante.

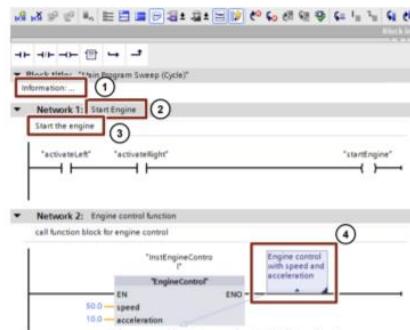
- Preporuča se korištenje **korisničkih konstanti**:
  - **Lokalne korisničke konstante** – Konstante koje se definiraju unutar lokalnih blokova podataka – dostupne isključivo unutar pripadajućeg funkcijskog bloka.
  - **Globalne korisničke konstante** – Konstante koje se definiraju unutar izbornika „PLC Tags“ – dostupne unutar cijelog programa PLC-a.
- Korisne korisničke konstante:
  - Prirodne konstante (Pi, e, korijen iz 2, itd.).
  - Kodovi grešaka.
  - Faktori za konverzije i slično.
- Konstante **nije moguće mijenjati** unutar programa (*read only*).

Name	Tag table	Data type	Value
1 PI	Default tag table	Real	3.1415
2 ERROR_TEMPERATURE	Default tag table	Int	13
3 ERROR_VOLTAGE	Default tag table	Int	17
4 ERROR_TORQUE	Default tag table	Int	23
5 KILO	Default tag table	Real	1000.0
6 MLU	Default tag table	Real	0.001
7 <Add new>			

Definiranje globalnih konstanti

## 32. Preporuke za komentare.

- Alat TIA Portal dozvoljava korištenje **komentara**.
- Komentari omogućuju jednostavnije razumijevanje i praćenje programskog rješenja.
- Preporučuje se obavezno korištenje komentara, kao i korištenje simboličkih naziva varijabli.
- Pregled mogućnosti komentiranja u programskim jezicima LAD i FBD dan je slikom:
  - 1. **Komentar bloka** – Komentar koji daje više informacija o trenutnom bloku (OB, FB, FC) unutar kojeg se nalazi.
  - 2. **Naslov mreže** – Komentar kojim je moguće definirati naslov mreže.
  - 3. **Komentar mreže** – Komentar kojim je moguće detaljno opisati funkciju programskog koda unutar mreže.
  - 4. **Komentar instrukcija** – Komentar kojim je moguće detaljno opisati funkcionalnost instrukcija unutar mreže.
- Također, moguće je komentiranje naredbi kod programskih jezika IL (STL) i ST (SCL) putem ključne riječi //



Komentari unutar TIA Portala

```
statFillingLevel := statRadius * statRadius * PI * statHeight;
// Calculating the filling level for medium tank
```

Komentari kod programskih jezika STL i SCL

## HMI i SCADA sustavi

### 1. Osnovne zadaće i razlike između HMI i SCADA sustava

HMI - Human Machine Interface

SCADA - Supervisory Control And Data Acquisition

#### HMI:

- Za prikaz trenutnih stanja i upravljanje manjih lokalnih (pod)sustava
- Ograničen broj tagova
- Ograničena „veličina prikaza“ (ovisi o tipu i proizvođaču)
- Jedan komunikacijski protokol za komunikaciju s podređenim jedinicama

#### SCADA sustav:

- Za centralni i distribuirani nadzor i upravljanje složenih distribuiranih sustava
- „Neograničen“ broj tagova
- „Veličina prikaza“ neograničena (ovisi o verziji i proizvođaču)
- Istovremeno koristi više komunikacijskih protokola za prikupljanje podataka s podređenih jedinica
- Podržava „naprednije“ funkcije nadzora i upravljanja (npr. skripte, logiranje, izrada izvještaja i dr.)
- ...

### 2. Piramida automatiziranog sustava te položaj SCADA sustava u istoj

- SCADA sustavi se nalaze u trećem sloju piramide automatiziranog sustava (eng. Automation Pyramid) – Nadzorna razina
- Osnovna zadaća komponenata Nadzorne razine je nadzor i upravljanje uređajima iz Upravljačke razine
- Komponente nadzorne razine često sadrže i grafički prikaz kako bi se korisnicima olakšao rad i vizualno prikazalo stanje sustava



### 3. Osnovne zadaće SCADA sustava

- Povezivanje s distribuiranim uređajima u sustavu:
  - prikupljanje podataka iz sustava (npr. industrijskog postrojenja) putem različitih industrijskih komunikacijskih protokola (npr. Profinet, Modbus TCP ili RTU, EtherCAT, itd.)
- Upravljanje i nadzor sustava:
  - grafički prikaz trenutnog stanja sustava
  - zadavanje postavnih veličina (npr. željena vrijednost tlaka, brzina vrtnje motora, temperature, uključenje/isključenje određenih komponenata u sustavu ili dijela sustava, itd.)
  - nadzor trenutnih vrijednosti mjerjenih veličina u sustavu (npr. težina materijala u spremniku, potrošnja električne energije pojedinih komponenata ili djela postrojenja, indikacija stanja dijelova postrojenja opasnih za rad, itd.)

- Definiranje i organizacija alarma sustava:
  - definiranje alarma i upozorenja u sustavu (digitalni i analogni; potrebna ili ne potvrda operatera; poruka koja opisuje alarm)
  - definiranje baze podataka arhive alarma
- Pohranjivanje mjerenih vrijednosti:
  - prikupljanje i pohranjivanje relevantnih vrijednosti u sustavu u svrhu analize rada sustava (npr. potrošnja električne energije pojedinog dijela i/ili cijelog postrojenja, sati rada pojedinih komponenata u sustavu radi lakšeg održavanja, itd.)
- Organizacija korisnika sustava:
  - definiranje ovlasti pojedinih korisnika sustava (npr. operater, nadzorni inženjer, administrator, voditelj postrojenja, itd.)
- Definiranje recepata:
  - recepti se tipično koriste za definiranje načina izvođenja procesa s obzirom na željeni proizvod (npr. recepti za proizvodnju različitih vrsta sokova)
- Generiranje izvještaja:
  - Automatsko ili na zahtjev generiranje predefiniranih izvještaja na temelju pohranjenih ili trenutno dostupnih podataka u SCADA sustavu

#### 4. Što su tagovi te čemu služe

- HMI tagovi (oznake):
  - tagovi predstavljaju rezervirana mjesta za podatke u SCADA sustavu
  - osnovna „karika“ za definiranje prikaza i načina rada SCADA sustava
  - memorijski prostor koji je potreban za pojedini tag ovisi o tipu podatka (npr. bool, integer, real, i dr.)

#### 5. Navesti i objasniti tipove HMI tagova

- HMI tagovi (oznake):
  - dijele se na eksterne, interne i sistemske tagove
  - **eksterni tagovi** se koriste za razmjenu vrijednosti podataka ili parametara s PLC-om, odnosno distribuiranim uređajem
  - **interni tagovi** se koriste samo unutar SCADA softvera (nisu povezani s drugim hardverom)
  - interne i eksterne tagove definira programer SCADA sustava
  - **sistemski tagovi** su predefinirani od strane proizvođača SCADA software-a, a koriste se za dohvaćanje sistemskih podataka SCADA sustava (npr. naziv trenutno logiranog korisnika, naziv računala ili servera na kojem se izvršava SCADA program, status redundantnog SCADA servera, i dr.)

#### 6. Navesti i objasniti tipove alarma (upozorenja) koji se nalaze u SCADA sustavima

- Alarmski i upozorenja:
  - glavna zadaća alarma je upozorenje korisniku o netipičnim ili nepoželjnim stanjima u sustavu
  - glavna razlika između alarma i upozorenja je što alarni zahtijevaju potvrdu korisnika kako bi automatizirani sustav dobio potvrdu da je korisnik svjestan kako postoji alarm u sustavu, dok su upozorenja samo informativne naravi i nije potrebna potvrda korisnika

- **Alarmi i upozorenja:**
  - većina SCADA sustava podržava:
    - diskretne alarne/upozorenja – aktiviraju se na promjenu diskretne vrijednosti signala (na rastući ili padajući brid; iz logičke nule u logičku jedinicu ili obrnuto)
    - analogni alarne/upozorenja – aktiviraju se kad je vrijednost taga veća (>), manja (<), jednaka (=) ili različita (!=) od definirane vrijednosti
    - sistemski upozorenja – upozorenja vezana uz stanje SCADA sustava (npr. prekid ili uspostava veze s PLC-om, backup baze podataka nije moguće, pristup bazi podataka nije moguće, i sl.)

## 7. Navesti i objasniti stanja moguća alarma u SCADA/HMI sustavima

- **Alarmi i upozorenja:**
  - alarmi i upozorenja mogu se grupirati u logičke skupine (npr. s obzirom na lokaciju u postrojenju ili s obzirom na određenu komponentu u postrojenju)
  - sve aktivne alarne iz iste grupe moguće je potvrditi odjednom
  - stanja alarma:
    - alarm aktivan i nepotvrđen,
    - alarm aktivan i potvrđen,
    - alarm nije aktivan, ali nije potvrđen (u slučaju da se otkloni uzrok alarma prije nego korisnik potvrdi alarm)
  - ovisno o zahtjevima SCADA sustava, pojedini ili svi alarmi (upozorenja) se mogu arhivirati, odnosno pohraniti u bazu podataka u svrhu analize uzroka kvara te automatske izrade izvještaja

## 8. Što je pojam logiranja, odnosno što predstavljaju logovi u SCADA sustavu?

- **Logiranje i vremenski prikazi (trendovi) vrijednosti:**
  - u svrhu analize rada sustava određene (diskretne i analogne) vrijednosti se mogu logirati (pohranjivati)
  - iz logiranih vrijednosti mogu se generirati automatski ili korisnički specifični izvještaji (npr. potrošnja električne energije za određeno vremensko razdoblje)

## 9. Što su skripte i kada se primjenjuju u SCADA sustavu?

- **Skripte:**
  - generički naziv za sve aktivnosti vezne uz korisnički definirane funkcije i lokalne skripte
  - funkcije se definiraju (programiraju) u slučaju kad SCADA programski paket nema predefiniranu sistemsku funkciju, a potrebno ju je izvršavati se često za različite ulazne parametre (slično kao funkcija u višem programskom jeziku)
  - skripte su lokalno izvođeni (pod)programi koji se aktiviraju ciklički ili na definirani događaj

Komponente SCADA sustava: opće postavke, grafički prikazi i povezivanje, HMI tagovi, alarmi i upozorenja, logiranje i vremenski prikazi vrijednosti, recepti, skripte, izvještaji, višejezičnost.

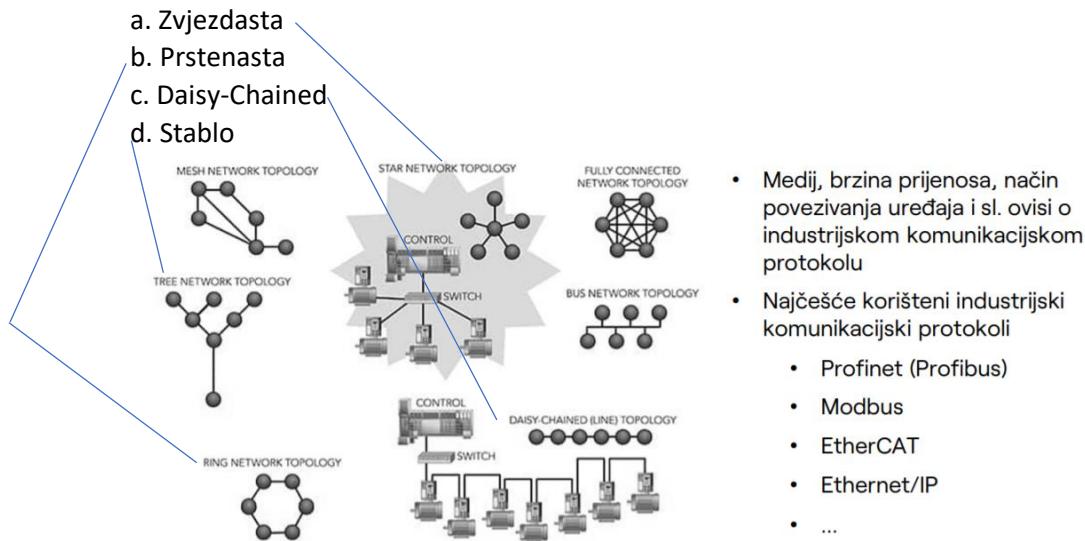
## Industrijski komunikacijski sustavi

### 1. Što je Fieldbus?

- Industrijski komunikacijski protokoli, *Fieldbus* – skup komunikacijski protokola koji se koriste u industrijskim komunikacijskim sustavima
- Standardizirani prema IEC 61158
- Neovisni o proizvođaču opreme
- Primarno za povezivanje podređenih uređaja (npr. senzori, distribuirani I/O moduli, frekvencijski pretvarači, itd.) i nadređenih upravljačkih uređaja (npr. PLC, PAC, PC, SCADA sustav, itd.)

### 2. Nacrtati i objasniti razlike između sljedećih mrežnih topologija:

- a. Zvjezdasta
- b. Prstenasta
- c. Daisy-Chained
- d. Stablo



- Medij, brzina prijenosa, način povezivanja uređaja i sl. ovisi o industrijskom komunikacijskom protokolu
- Najčešće korišteni industrijski komunikacijski protokoli
  - Profinet (Profibus)
  - Modbus
  - EtherCAT
  - Ethernet/IP
  - ...

### 3. Navesti tri najčešće korištena industrijska komunikacijska protokola

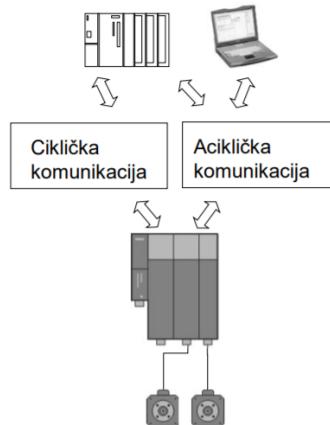
#### Modbus

- Jeden od najstarijih industrijskih komunikacijskih protokola (1979., Modicon; danas Schneider Electric)
- Spada u skupinu *Open protocol-a*
- Specifikacije su dostupne (<https://modbus.org/>)
- Modbus Organization se brine za razvoj protokola
- Tipovi:
  - Modbus RTU
  - Modbus ASCII
  - Modbus TCP
  - Modbus Plus

### 4. Koje sučelje koristi Profinet komunikacijski protokol te koje sve tipove komunikacije podržava (**objasniti razlike**)

## Profinet

- Koristi standardni Ethernet (*standard unmodified Ethernet*) – hardversko sučelje isto kao i za standardni Ethernet pa se može koristiti zajedno s drugim komunikacijskim protokolima (OPC UA, SNMP, i dr.)
- Podržava cikličku i acikličku komunikaciju



### 5. Razlika između Profinet i Profinet IRT-a

- Profinet RT ciklusi: od 512 ms do 250 µs
- Za zahtjevne sustave, gdje se zahtjeva kraće vrijeme ciklusa (*motion control*), koristiti se Profinet IRT (*isochronous Real Time*)
  - „Prilagođena“ pravila Ethernet prometa kako bi se osigurao brži ciklus
  - Ciklusi minimalno 31.25 µs uz kašnjenje (*jitter*) od 1 µs
  - Opcionalan i ne podržavaju ga svi Profinet uređaji

### 6. Razlika između Profibus-a i Profinet-a

#### Profinet:

- Standardni RJ 45 konektor
- 100 Mbit/s
- Dužina kabela do 100 m
- Može imati ciklus kraći od 1 ms
- Koristi standardni Ethernet; jedinstveno određen s:
  - IP adresa
  - MAC adresa
  - Device name

#### Profibus (PROcess Field Bus):

- Od 1993.
- Tipično se spaja preko DB 9 (9 pinski) konektora
- Koristi se kabel s uplenom paricom
- Serijska *daisy chain* topologija
- Jedinstvene adrese čvorova (od 1 do 127)
- Brzina prijenosa od 9600 bit/s do 12 Mbit/s (najčešće 1.5 Mbit/s)
- Dužina kabela od 50 do 1000 m

### 7. Princip rada EtherCAT komunikacijskog protokola

- **Ethernet for Control Automation Technology**
- Tehnologija temeljena na Ethernetu napravljeno isključivo za industrijsku automatizaciju
- Koristi standardni Ethernet okvir (*frame*) i fizički sloj kako je specificirano Ethernet standardom IEEE 802.3
- U odnosu na standardni Ethernet osigurava:
  - Minimalno vrijeme odziva
  - Minimalni broj podataka potreban za sigurnu komunikaciju
  - Mali troškovi za implementaciju
- Koristi ista prva dva sloja OSI modela kao Ethernet (Physical i Data Link), međutim protokoli se razlikuju u drugom dijelu
- Nadređena jedinica (*Master*) šalje poruku čvorovima (*Node*) te se poruka prosljeđuje od čvora do čvora dok se ne vratí nadređenoj jedinici
- Svaki čvor dodaje/cita svoj dio poruke te prosljeđuje dalje
- Nisi potrebni dodatni prespojnici (*switch*) jer skoro svi EtherCAT uređaji imaju dva Ethernet porta
- Koristi *Distributed Clock System* – svaki čvor doda *Timestamp* u poruku tako kad se poruka vrati nadređenoj jedinici zna se točno vrijeme kad je koji čvor primio poruku

8. Da li je moguća uspostaviti EtherCAT komunikaciju između uređaja ako je topologija mreže u Daisy-Cahined (**objasniti odgovor**)  
Zbog full-duplex tehnologije može!

## P&ID dijagrami

1. Što je i zašto se koristi P&ID(Piping And Instrumentation Diagram)?
  - P&ID je crtežna dokumentacija koja opisuje dio ili cijeli proces procesnog postrojenja
  - Moglo bi se reći da je to „procesno postrojenje na papiru“
  - P&ID je shematski prikaz cjevovoda (*pipes*), procesne opreme (*process equipment*) i upravljačkog sustava (*control system*) pomoću skupa unaprijed definiranih simbola
  - **Ne crta** se u skali niti ima orientaciju
  - Oprema procesnog postrojenja se tipično prikazuje pomoću simbola koji nalikuju stvarnoj opremi (bočni pogled)
  - Linijama se prikazuju cijevi i signali u procesu (ne prikazuje stvarni izgled, odnosno putanje cijevi i signala)
  - P&ID sadrži mnogo informacija vezanih uz:
    - proizvodnju opreme,
    - instalaciju,
    - pokretanju (*startup*) postrojenja,
    - puštanje u pogon i
    - rad postrojenja.
  - Također predstavlja kako procesno postrojenje treba raditi u izvanrednim (hitnim) situacijama

2. Koje blokove može sadržavati P&ID? Što pojedini blok sadrži?

- Svaki P&ID treba sadržavati barem sljedeće blokove:
  - Naslovni blok (*Title Block*) – što prikazuje P&ID
  - Vlasnički blok (*Ownership Block*) – tko je izradio i za koga je izrađen P&ID
- Dosta često P&ID sadrži i sljedeće blokove:
  - Blok referenci (*Reference Drawing Block*) – sadrži reference svih ostalih P&ID-a koji su povezani s prikazanim P&ID-om
  - Blok revizija (*Revision Block*) – sadrži popis svih revizija (izmjena) koje su napravljene od prve (nulte) revizije
  - Blok s komentarima (*Comments Block*) – sadrži sve što se ne može dati u grafičkom prikazu P&ID-a

Naslovni blok je tablica (često) u donjem desnom uglu , tipično sadrži: naslov crteža, broj crteža, broj zadnje revizije crteža, logo firme.

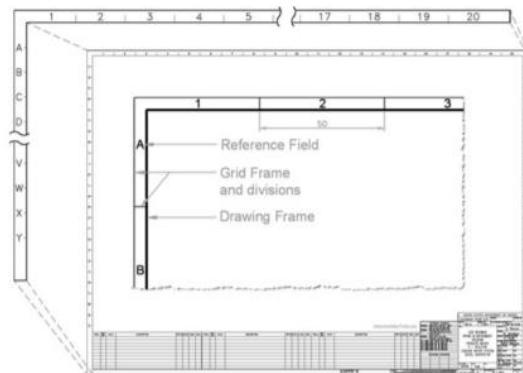
Vlasnički blok sadrži sljedeće informacije: tko je izradio crtež, tko je odobrio crtež, datum izrade odnosno odobrenja crteža, zakonske klauze.

Blok referenci je tablica koja sadrži sljedeće informacije: broj crteža, naslov (naziv) crteža, oznaku revizije koja je relevantna.

Blok revizija je tablica koja sadrži sljedeće informacije: broj revizije, datum revizije, kratki opis što je revidirano, tko je revidirao, odobrio reviziju.

Blok s komentarima sadrži dodatne informacije koje nije moguće prikazati u grafičkom dijelu P&ID-a.

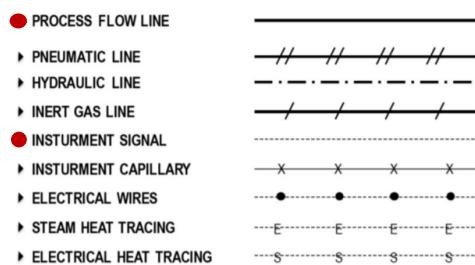
3. Kako izgleda i čemu služi mreža (grid) kod P&ID-a?



- P&ID se crta unutar „okvira“ koji sadrži mrežu (Grid) P&ID-a
  - Služi za referenciranje komponenata, odnosno dijelova sustava na P&ID-u
  - Sadrži:
    - Brojke,
    - Slova i/ili
    - Oboje
- pozicioniranih horizontalno i vertikalno

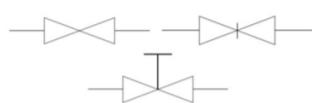
4. Nacrtati sljedeće P&ID simbole:

- Liniju za procesni cjevovod
- Liniju za instrumentacijske signale



c. Simbol ručno upravlјivog ventila

- Ventili se najčešće koriste kako bi se upravljalo:
  - smjerom,
  - protokom i
  - tlakom
 medija u cjevovodu
- Ručno upravlјiv ventil (Gate Valve)
  - simbol



- Simbol otvorenog ventila



- Simbol zatvorenog ventila



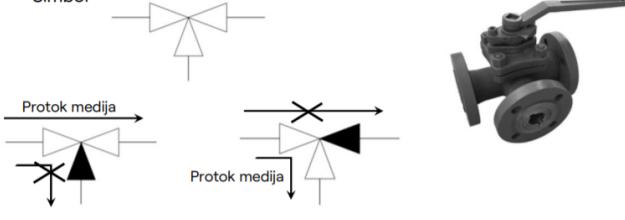
- Ručno upravlјiv ventil (Butterfly Valve)
- Simbol



- Iz simbola nije moguće zaključiti u kojem se položaju (otvoren ili zatvoren) nalazi ventil

d. Simbol troputnog ventila s naznačenim smjerom protoka

- Ručno upravlјiv troputni ventil (3-Way Valve)
- Simbol

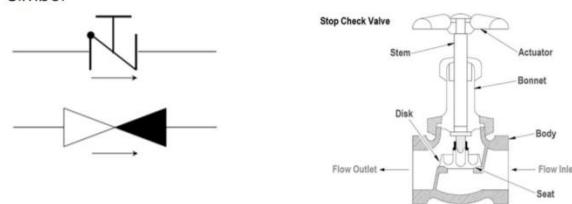


#### e. Simbol nepovratnog ventila

- Nepovratni (jednosmerni) ventil (Check Valve) – dozvoljava protok medija samo u jednom smjeru
- Simbol



- Ručno upravlјivi nepovratni ventil (Stop Check Valve) – dozvoljava protok medija samo u jednom smjeru
- Simbol



#### f. Simbol ventila s aktuatorom

- Za upravljanje ventilima koriste se različiti aktuatori:
  - S elektromotorom (Electric Motor Valve)



- Sa svitkom (Solenoid Valve)



#### g. Simbol pumpe/ventilatora

- Centrifugalne pumpe



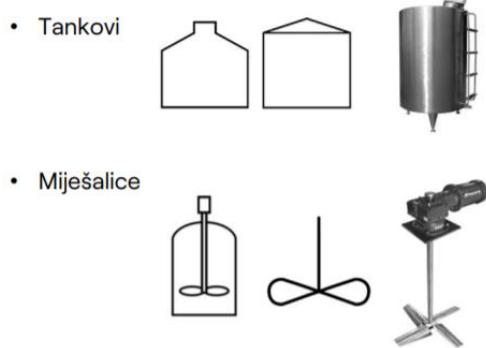
- Ventilator



- Izmjenjivač topline

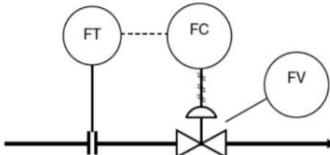


#### h. Simbol tanka i. Simbol mješalice



## 5. Označavanje komponenata, instrumenata i funkcija unutar P&ID-a

- Za prikaz elemenata (komponenata, instrumenata i funkcija) na P&ID-u često se koriste „krugovi“ s dodatnim oznakama koje opisuju isti



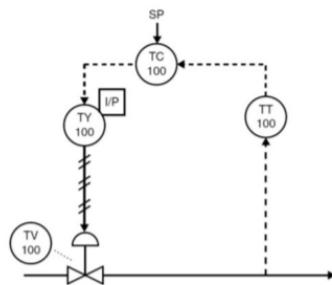
- Instrument dostupan u postrojenju (*in field*)
- Instrument dostupan u kontrolnoj sobi
- Instrument se nalazi u kontrolnoj sobi, ali nije dostupan (npr. potrebno otvoriti ormara)
- Instrument dostupan u upravljačkom ormaru u postrojenju
- Instrument u upravljačkom ormaru u postrojenju, ali nije dostupan (npr. potrebno otvoriti ormara)
- „Virtualni“ DCS instrument dostupan u postrojenju (npr. prikaz mjerenja na zaslonu)
- „Virtualni“ DCS instrument dostupan u kontrolnoj sobi
- „Virtualni“ DCS instrument dostupan u upravljačkom ormaru u postrojenju
- „Virtualni“ PLC instrument dostupan u postrojenju (npr. prikaz mjerenja na zaslonu)
- „Virtualni“ PLC instrument dostupan u kontrolnoj sobi
- „Virtualni“ PLC instrument dostupan u upravljačkom ormaru u postrojenju

Napomena: često se s rombovima označavaju sigurnosne funkcije na P&ID-u (potrebno proučiti legendu P&ID-a)

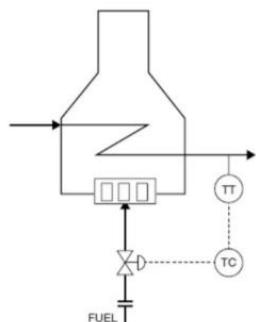
- Za identifikaciju pojedinih elemenata na P&ID-u koriste se standardizirane kratice od jednog do četiri slova
- Značenje slova

1. slovo – veličina koja se nadzire ( <i>sensed parameter</i> )	2. slovo – tip indikatora ( <i>type of indicator</i> )	3. slovo – tip komponente ( <i>type of component</i> )	4. slovo – tip signala ( <i>type of signal</i> )
F – protok (Flow)	I – indikator (Indicator)	T – transmpter	I – struja (Current)
P – tlak (Pressure)	R – zapis (Recorder)	(Transmitter)	V – napon (Voltage)
T – temperatūra (Temperature)	C – regulator (Controller)	M/Y – pretvornik	P – pneumatski
I – struja (Current)		(Modifier)	(Pneumatic)
V – napon (Voltage)			
L – nivo (Level)		E – element (Element)	
Z – pozicija (Position)			

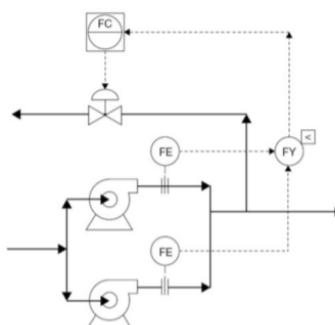
6. Objasniti funkciju P&ID-a sa slike (napomena: dan je samo primjer)



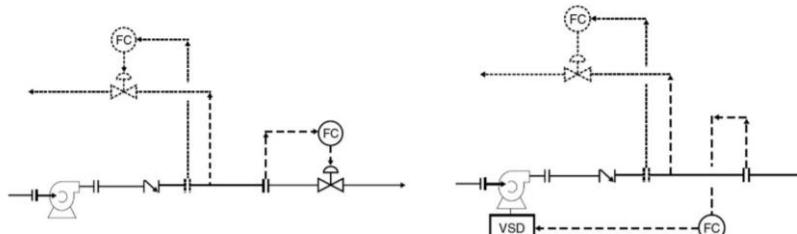
- TT 100 temperaturni transmiter mjeri trenutnu vrijednost temperature medija te istu proslijedi regulatoru temperature TC 100
- Regulator temperature na temelju zadane temperature (SP = setpoint) i trenutne vrijednosti postavlja upravljački signal za pretvornik TY 100 koji pretvara strujni signal u pneumatski (P/I) kako bi upravljao s ventilom TV 100



- TT temperaturni transmiter mjeri trenutnu vrijednost temperature medija te istu proslijedi regulatoru temperature TC
- Regulator temperature na upravlja s ventilom dovoda goriva te na taj način regulira temperaturu medija



- Element za usporedbu vrijednosti protoka FY na trenutnih vrijednosti elemenata protoka FE proslijedi signal regulatoru protoka FC
- Regulator protoka FC upravlja se ventilom



## Strukturno programiranje

- Neovisno o veličini sustava upravljanja (broj ulaza/izlaza, distribuiranih uređaja, upravljačkih struktura, itd.) PLC treba programirati tako da je:
  - Jasna struktura i način programiranja
  - Lako proširivi blokovi
  - Modularan program – koristit obrasce (*pattern design*)
  - Komentiran program
  - Program jednostavan za održavanje

### 1. Prednosti strukturnog programiranja

- Blokovi se mogu kreirati i testirati neovisno jedan o drugom
- Primjenom parametara, blokovi se mogu programirati tako da budu fleksibilni
- Blokovi se mogu više puta pozivati, na različitim mjestima programa, s različitim vrijednostima parametara
- Često korišteni blokovi se mogu pohraniti u biblioteku za korištenje u drugim programima

### 2. U kojem slučaju je preporučeno koristit tip podatka strukture (STRUCT)

Strukturiranje podataka pojavljuje se neposredno prilikom izrade funkcijskog bloka. Deklaracijom ulaznih, izlaznih i lokalnih varijabli stvaraju se strukturirani podaci (instanca) koji su vezani uz taj funkcijski blok, a pri pozivu funkcijskog bloka potrebno je definirati s kojom se strukturu podataka (instancom, DB-om) on izvršava. Na taj su način podaci grupirani i pridonose ispunjavanju preporuka norme IEC 61131.

### 3. U kojem slučaju je preporučeno koristit tip podatka polja (ARRAY)

Polje podataka se sastoji od niza istog tipa podatka (elementarnog ili korisničkog), a svakom članu niza pristupa se pozivom na njegov redni broj (engl. index) unutar niza. Polja se u programskim rješenjima PLC-a najčešće koriste za neke radnje koje je potrebno izvršiti nekoliko puta, ali s različitim podacima. Umjesto deklaracije više varijabli različitih imena te pozivanjem funkcije ili funkcijskog bloka za svaku radnju, deklarira se jedno polje varijabli, a funkcija ili funkcijski blok se poziva unutar petlje i pri tome se mijenja redni broj poziva. Iako će količina „posla“ koju PLC treba obaviti (izračunati) ostati ista, smanjenja je dužina programskog rješenja te su izbjegnuti potencijalni problemi prilikom kopiranja koda.

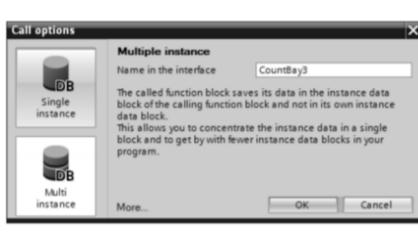
### 4. Što je UDT i kad se preporuča koristit?

UDT – User defined type

Korisnički tipovi podataka mogu se definirati unutar izbornika PLC data types, a jednom definirani mogu se koristiti za sve globalne varijable (globalni podatkovni blokovi) te kao ulazne, izlazne i lokalne varijable funkcija i funkcijskih blokova. Korisnički tipovi podataka mogu se sastojati od elementarnih tipova podataka (Int, Real, Bool...) i drugih korisničkih tipova podataka kojima je potrebno dodijeliti naziv. Tako se podaci grupiraju u smislene cjeline te se smanjuje potrebnii broj deklariranih varijabli.

### 5. Što je instanciranje i kad se primjenjuje?

- Slično kako inicijalizacija varijabli u višim programskim jezicima (prije inicijalizacije, varijabla se ne može koristiti)
- FB-u se može dodjeliti memorjsko područje, odnosno FB se može instancirati na sljedeće načine:
  - Specificiranjem podatkovnog bloka (DB) prilikom poziva FB-a
  - Deklariranjem instance funkcijskog bloka kako statičke varijable unutar nadređenog funkcijskog bloka



## Procesni upravljački sustavi

### 1. Navedite inženjerske alate u PCS7 procesnom upravljačkom sustavu.

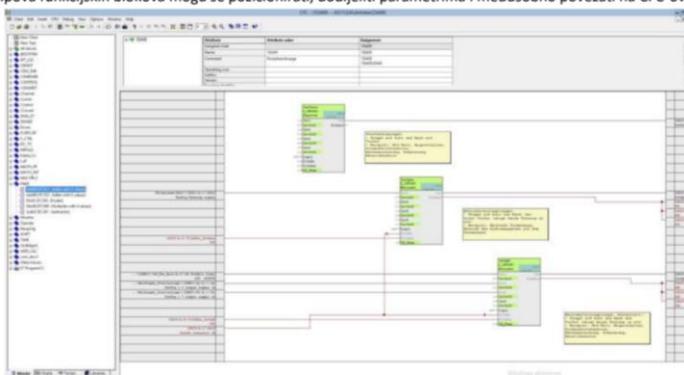
#### Process Object View

- Procesni objektni prikaz SIMATIC Manager-a podržava rad koji obavlja procesni inženjer pružajući univerzalni pogled na oznaku procesa
- Prikazuje hijerarhiju postrojenja predstavljenu u obliku stabla u kombinaciji s tabelarnim prikazom svih aspekata oznake procesa ili procesnog objekta (općenito, grafikoni, blokovi, parametri, signali, poruke, slikovni objekti, arhivske varijable, mape hijerarhije, svojstva opreme i globalne deklaracije) To omogućuje tehnologiju brzu orientaciju.
- Svi objekti u označenoj grani hijerarhije prikazani su u tablici tako da se mogu izravno obraditi korisničkim funkcijama za uređivanje, filtriranje, zamjenu, uvoz i izvoz.

### 2. Koja dva programska jezika se koriste za programiranje PCS7?

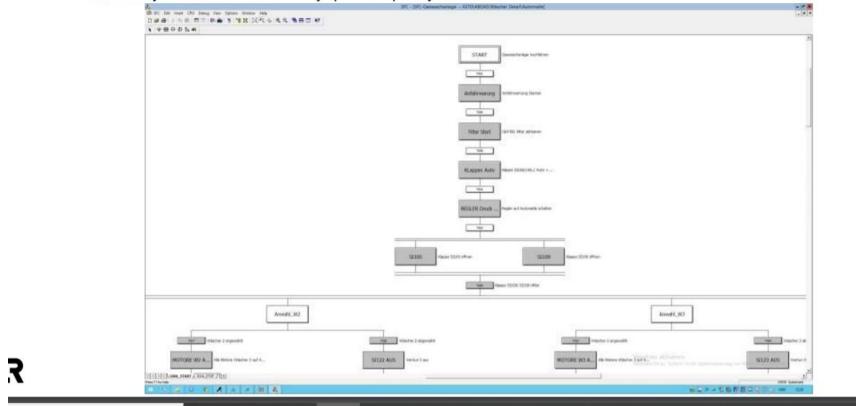
#### Continuos Function Chart - CFC

- **CFC editor** – koristi se za konfiguraciju funkcija kontinuirane automatizacije
  - Uključuje moćne funkcije testiranja i puštanja u rad, kao i individualno parametrirane funkcije za dokumentaciju.
  - Umetanje gotovih funkcionalnih blokova i njihovo međusobno povezivanje za stvaranje logičke cijeline
  - Funkcionalni blokovi – iz gotove Siemens biblioteke (Advanced Process Library – APL)
  - Instance tipova funkcionalnih blokova mogu se pozicionirati, dodjeliti parametrima i međusobno povezati na CFC-ovima.



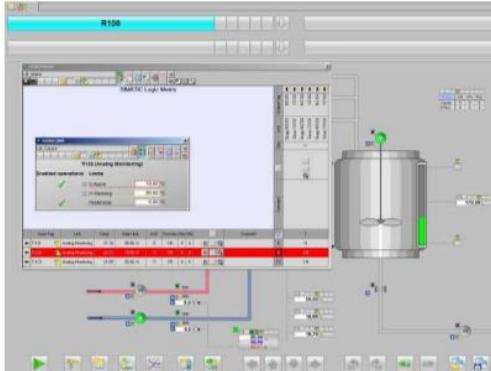
#### Sequential Function Chart - SFC

- **SFC editor** – koristi se za grafičku konfiguraciju i puštanje u rad sekvencijalnih kontrola za operacije serijske proizvodnje.
  - Integrirani grafički uredavač formula za aritmetičke operacije, Booleovu algebru i matematičke funkcije omogućuje izračune unutar SFC-a.
  - Korištenjem sekvencijalnog upravljanja, osnovne funkcije automatizacije koje se obično stvaraju pomoću CFC-a kontroliraju se i selektivno obrađuju pomoću promjena u načinu rada i statusu.



### 3. Što je logička matrica?

- SIMATIC PCS7 Logička matrica - omogućuje praktično stvaranje logike međusobnog zaključavanja između tehničkih funkcija (npr. upravljačkih modula ili modula opreme) projekta automatizacije pomoći matrice uzroka i posljedica.
- Inženjer povezuje potrebne ulazne signale u vodoravne redove matrične tablice usporedive s proračunskom tablicom.
- Logičke operacije, vremenski odgovor, alarmi itd. mogu se konfigurirati za svaki uzrok.
- Izlazi se mogu povezati s efektima u okomitim stupcima logičke matrice na sličan način.
- Uzroci i posljedice se zatim logički povezuju kroz sjecišta redaka i stupaca, omogućujući konfiguriranje različitih reakcija.



R

### 4. Koje su moguće arhitekture operatorskog sustava?

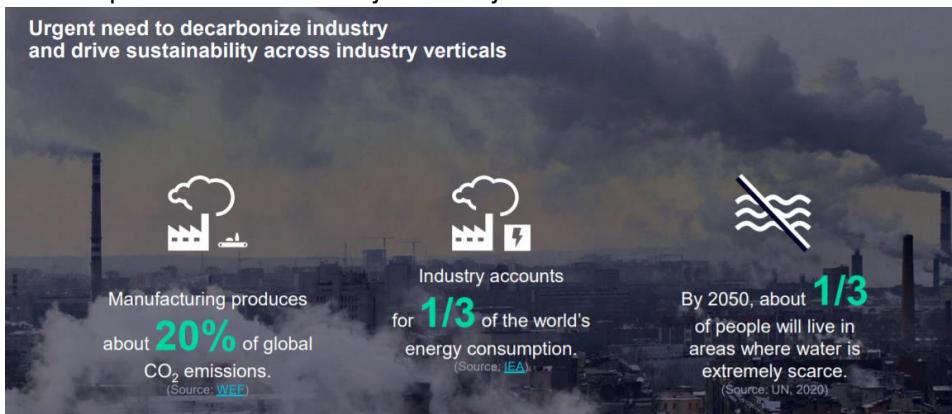
- Moguće arhitekture:
  - Sustav s jednom stanicom (OS single station) - s do 8 500 procesnih objekata
  - Sustav s redundantnim parom jedne stanice (Redundant OS single station) - Ravne konfiguracije sustava temeljene na redundantnom paru OS jedne stanice, proširivo s referentnim stanicama na do 8 OS pojedinačnih stanica.
  - Klijent-poslužitelj (Server/Client) - višestanični sustavi s do 18 OS poslužitelja/parova poslužitelja, svaki za 12 000 procesnih objekata i do 40 OS klijenata

### 5. Opišite šaržnu automatizaciju

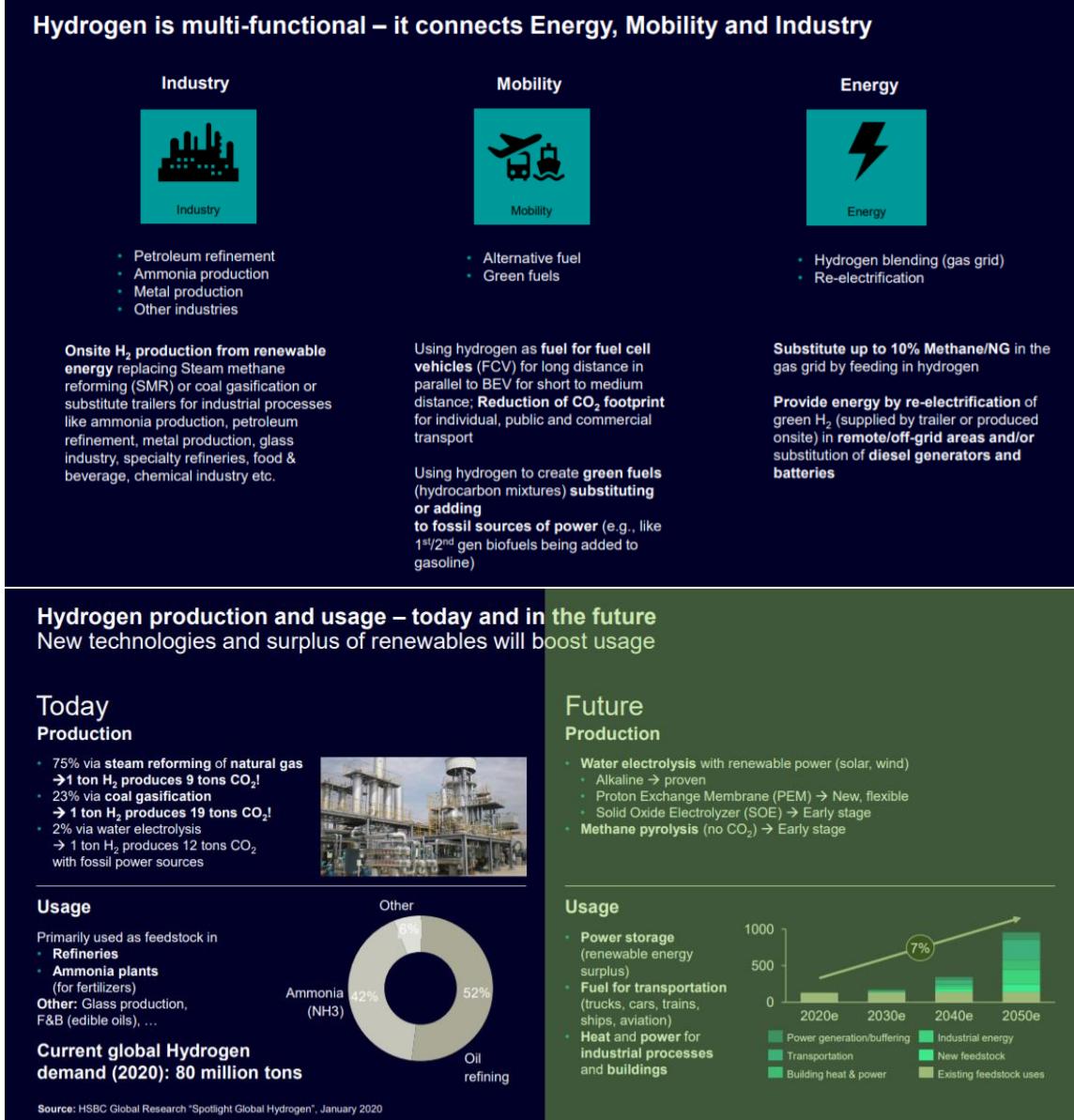
- **SIMATIC BATCH – Šaržna (batch) automatizacija**
  - Omogućuje učinkovito i fleksibilno izvođenje jednostavnih i složenih šaržnih procesa s promjenjivim kontrolnim sekvencama.
  - SIMATIC BATCH bez iznimke ispunjava sve povezane visoke zahtjeve:
    - Optimalna iskorištenost kapaciteta proizvodnih pogona
    - Jedinstvena kvaliteta proizvoda
    - Sljedivost
    - Usklađenost sa zakonskim standardima i direktivama
    - Brza reakcija na promjenjive tržišne uvjete
  - Konfiguracija kao sustav s jednom stanicom ili kao sustav klijent/poslužitelj.
  - U potpunosti integriran u SIMATIC PCS 7. Podaci o postrojenju mogu se u potpunosti konfigurirati pomoći inženjerskog sustava i svi podaci potrebeni za izradu recepta prenijeti u SIMATIC BATCH.
  - To omogućuje uređivanje recepata odvojeno od inženjerskog sustava.
  - Promjene konfiguracije na inženjerskom sustavu dostupne su SIMATIC BATCH-u pomoći funkcije ažuriranja.

## Tehnološke transformacije

### 1. Zašto se provodi dekarbonizacija industrije?

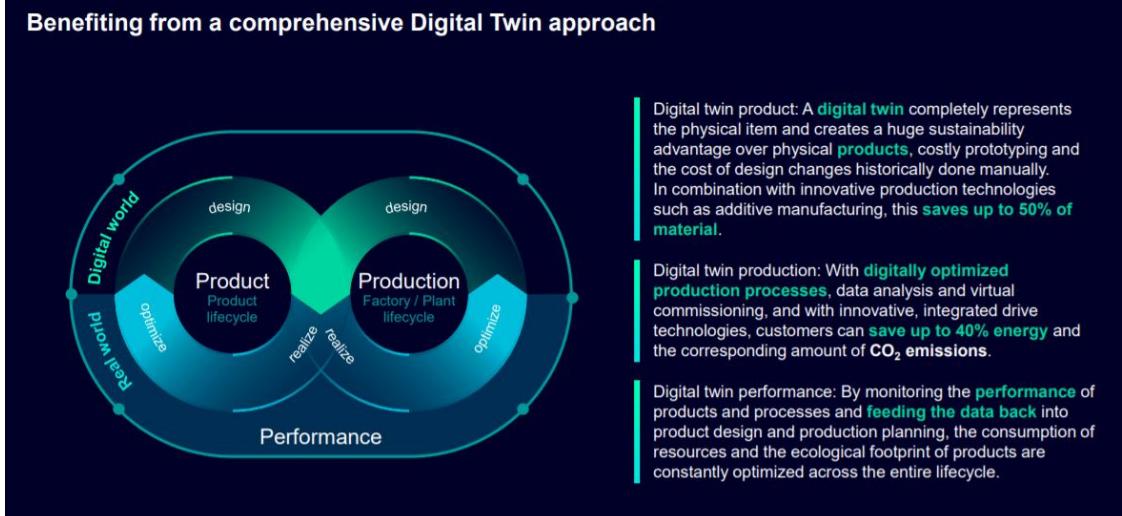


### 2. Gdje se može koristiti vodik, koje su prednosti, a koji nedostaci?



3. Za što se koriste digitalni blizanci i koje su prednosti?

Benefiting from a comprehensive Digital Twin approach



4. Koje su razine moderne tvornice od najniže do najviše?

From Sensor to Edge to Cloud

