

Viša tehnička škola – S u b o t i c a

Dr Zoltan Jegeš
Milan Adžić
Robert Marton

UPRAVLJANJE PRIMENOM *PLC* UREĐAJA

S u b o t i c a

oktobar 2005. god.

SADRŽAJ

	<i>Strana</i>
1. Primena i programiranje programabilnih kontrolera	1
2. Upravljanje sa PLC – om, rešavanje konkretnog upravljačkog zadatka	59
3. Primena PLC – a u elektromotornim pogonima	76
4. Primena analognih ulazno - izlaznih modula	103
5. Osnovne primene programabilnih terminala	138
6. Osnovne primena SCADA sistema za supervizijsku kontrolu i prikupljanje podataka	177

PRIMENA I PROGRAMIRANJE PROGRAMIBILNIH KONTROLERA

1. Uopšte o PLC kontrolerima

Uopšteno, sistem upravljanja u elektrotehnici čini skup elektronskih uređaja i opreme koji obezbeđuju stabilnost, tačnost i eliminaciju štetnih prelaznih stanja u proizvodnim procesima. Sistem upravljanja može biti različitog oblika i implementacije, od energetske postrojenja do mašina. Sistemi upravljanja su se razvijali tokom vremena. U ranom periodu razvoja sami ljudi su obavljali upravljačke zadatke. Krajem šezdesetih godina prošlog veka sistemi upravljanja su bili zasnovani na primeni relejne logike, zasnovane na relativno jednostavnim logičkim algoritmima. Glavna mana relejne logike je da se pri bilo kakvoj promeni u sistemu upravljanja ona mora menjati, promenom ožičenja ili čak ubacivanjem u potpunosti novih sklopova. Te promene izazivale su velike troškove ne samo za opremu već i dugotrajne zastoje potrebne za modifikaciju i testiranje. Napredak tehnologije u izradi mikroprocesora, u to vreme, doveo je do revolucije u sistemima upravljanja. Pojavila se ideja o izradi elektronsko-mikroprocesorskom upravljačkom uređaju koji bi se mogao jednostavno reprogramirati u slučaju izmene u upravljačkim zadacima. Izrađeni su i prvi takvi uređaji, koji su dobili naziv *programabilni logički kontroleri* (*Programmable logic controllers*) ili skraćeno *PLC*. Dalji razvoj ovih uređaja je bio vrlo brz, pošto su pokazali izuzetne prednosti u odnosu na logiku zasnovanu na primenu releja, jer nemaju mehaničkih pokretnih delova, fleksibilniji su zbog mogućnosti programiranja, manja je moguća pojava grešaka tokom ožičavanja, manjih su dimenzija, imaju manju sopstvenu potrošnju i pouzdanost rada im je velika.

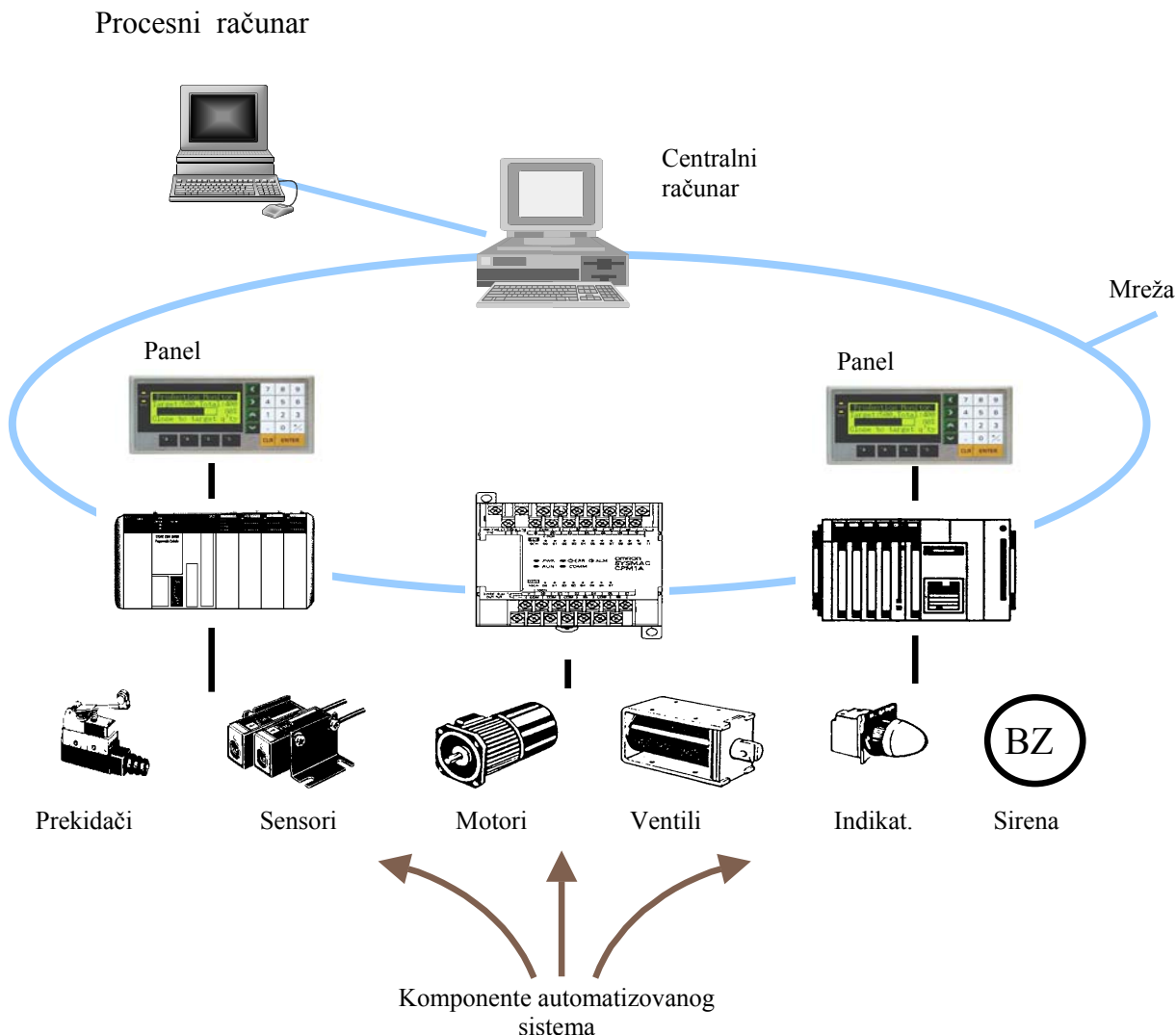
Prema standardima Udruženja proizvođača električne opreme (*The National Electrical Association – NEMA*) programabilni logički kontroler definisan je kao: “Digitalni elektronski uređaj koji koristi programabilnu memoriju za pamćenje naredbi kojima se zahteva izvođenje specifičnih funkcija, kao što su logičke funkcije, sekvenciranje, prebrojavanje, merenje vremena, izračunavanje, u cilju upravljanja različitim mašinama i procesima”.

PLC kao industrijski računar samim svojim dizajnom predviđen je za primenu u neposrednom okruženju procesa sa kojim upravlja, tako da je otporan na razne nepovoljne uticaje, prašina, vlaga, visoka temperatura, vibracije i elektromagnetne smetnje, tako da se obično primenjuje za rešavanje decentralizovanih upravljačkih zadataka, na samom mestu upravljanja, gde se povezuje preko ulaza i izlaza sa uređajima kao što su operatorski paneli, motori, senzori, prekidači, ventili i sličnim. *PLC* kao i svaki računar ima operativni sistem, koji svakako ima mnogo manje mogućnosti od operativnih sistema opšte namene, ali u današnje vreme opšte potrebe za komunikacijama, može u potpunosti da ih podrži. Stoga je moguće izvesti povezivanje programabilnih logičkih kontrolera (*PLC-a*) i eventualno centralnog računara ili drugih računara, radi rešavanja složenijih upravljačkih zadataka ili jednostavne akvizicije podataka i upravljanja sa daljine. Mogućnosti komunikacije među *PLC* uređajima su tako velike da omogućavaju visok stepen iskorišćenja i koordinacije procesa, kao i veliku fleksibilnost u realizaciji upravljačkog procesa, tako da mogućnost komunikacije kao i fleksibilnost predstavljaju glavne prednosti primene rešenja sa *PLC* uređajima.

***PLC* kontroler**

PLC kontroler je element automatizovanog sistema, koji na osnovu prihvaćenih ulaznih signala sa ulaznih uređaja, po određenom programu, formira izlazne signale sa kojima upravlja izlaznim uređajima. U automatizovanom sistemu, *PLC* kontroler je obično centar upravljanja. Izvršavanjem programa smeštenog u programskoj memoriji, *PLC* neprekidno posmatra stanje sistema preko ulaznih uređaja. Na osnovu logike implementirane u programu *PLC* određuje koje

akcije trebaju da se izvrše na izlaznim uređajima. Za upravljanje složenim procesima moguće je povezati više *PLC* kontrolera među sobom ili sa centralnim računarom.



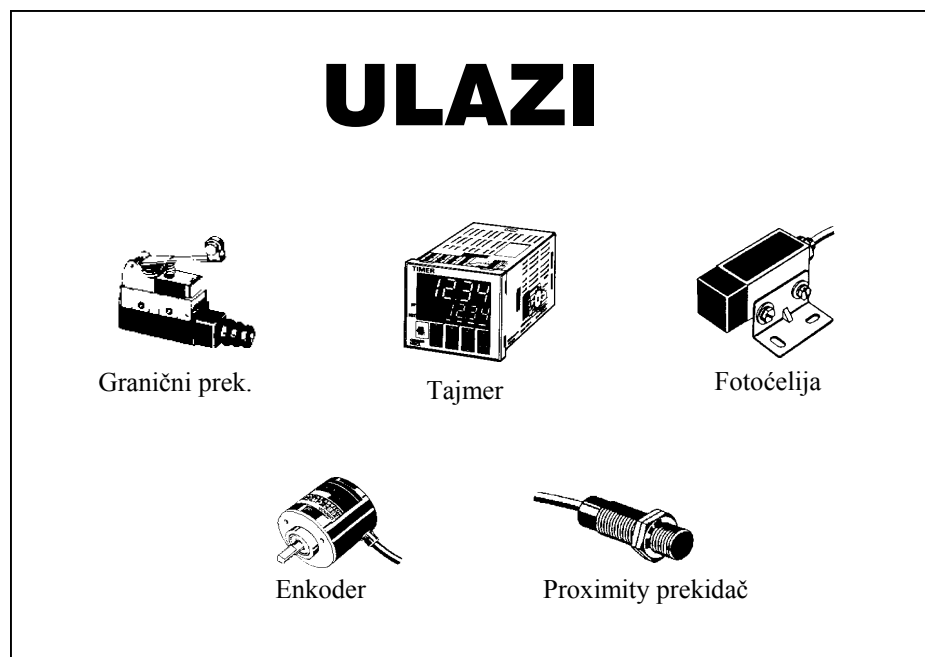
PLC kao element automatizovanog sistema

Prema broju ulazno/izlaznih priključaka *PLC* uređaji mogu se načelno podeliti na mikro sa maksimalno do 32, male do 256, srednje do 1024 i velike *PLC*-e preko 1024 ulazno/izlaznih priključaka. Sa povećanjem broja priključaka mora se povećati i brzina procesora kao i količina memorije a samim tim i složenost i cena samog uređaja raste.

Ulazni uređaji

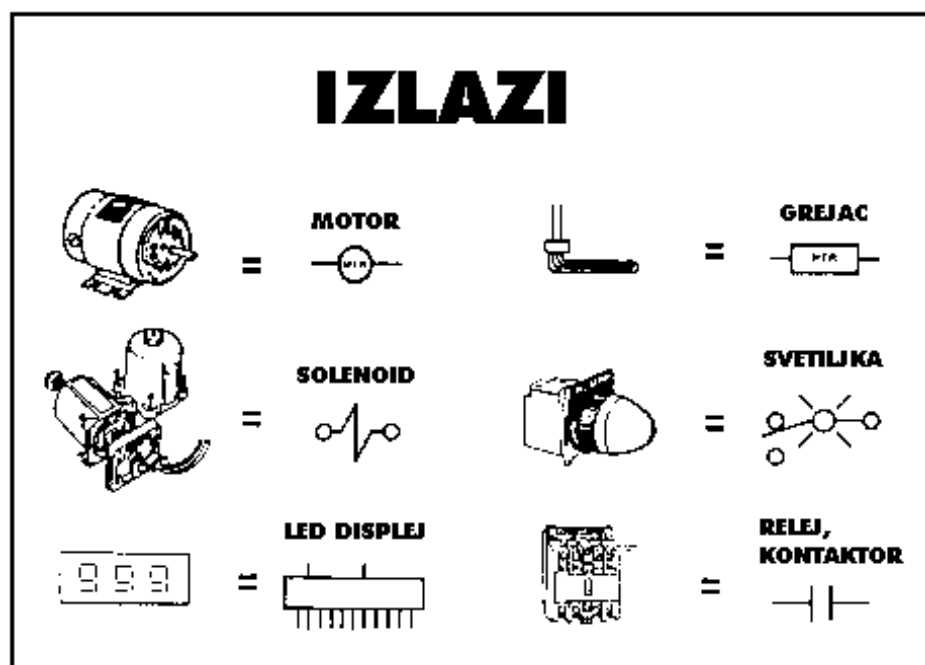
Ulazni uređaji čije signale prihvata *PLC*, mogu biti vrlo različiti. Po tipu signala koji daju na svom izlazu mogu se podeliti na digitalne (kontaktne prirode *ON / OFF*) i analogne. Karakteristični digitalni ulazni uređaji su tasteri, prekidači, krajnji prekidači, fotočelijske, presostati, termostati i drugi. Karakteristični analogni ulazni uređaji su termoelementi, otpornički termometri i drugi pretvarači električnih i neelektričnih veličina u standardne strujne i naponske signale. Ulazni signali pri tome se moraju prilagoditi sa odgovarajućim *PLC* ulaznim modulima. Ulazni moduli konstruišu se za prihvatanje

jednosmernih i naizmeničnih električnih signala naponskih nivoa od 10 do 250V, strujnih nivoa od 0(4) do 20mA, signala na TTL nivou, impulsnih ulaza sa brojačkim ili interapt prihvatom i slično.



Tipični ulazni uređaji

Izlazni uređaji



Tipični izlazni uređaji

Izlazni uređaji kojima upravlja *PLC* na osnovu programa i stanja na ulazima mogu biti releji, kontaktori, elektromagnetni ventili, elektromotori, step motori, pneumatski cilindri i drugi slični uređaji. Takođe kao ulazni uređaji mogu se podeliti na digitalne i analogne, pri čemu se izlazi *PLC* kontrolera moraju prilagoditi potrebnim naponskim i strujnim nivoima. Digitalni izlazi *PLC* kontrolera su obično galvansko izolovani kontakti, ili bez povećane galvanske izolovanosti sa triakom ili tranzistorom sa otvorenim kolektorom ili TTL izlazom.

1.1. Prednost upravljanja pomoću *PLC*-a u odnosu na upravljanje relejima

Početkom industrijske revolucije automatizovanim mašinama upravljali su pomoću releja, međusobno povezanih žicama unutar komandnog ormana. Za otkrivanje greške u sistemu bilo je potrebno mnogo vremena pogotovo kod složenih upravljačkih sistema. Vek trajanja kontakata releja je ograničen, pa se vremenom moraju zameniti. Prilikom zamene releja ili ostalih potrošnih delova, mašina se morala zaustaviti a time i proizvodnja. Komandni orman koristio se samo za jedan određeni proces i nije ga bilo jednostavno izmeniti prema potrebama novog sistema. Prema ovome izloženom do sada rtelejno upravljanje se pokazalo veoma neefikasnim. Ovi nedostaci su u velikoj meri otklonjeni uvođenjem *PLC*-a u sisteme upravljanja, što je još i doprinelo poboljšanju kvaliteta, uvećavanju produktivnosti i fleksibilnosti.

Prednost komandnog ormana urađenog na bazi *PLC* kontrolera u odnosu na komandne ormane napravljenih na bazi releja ogleda se unekoliko sledećih stavki:

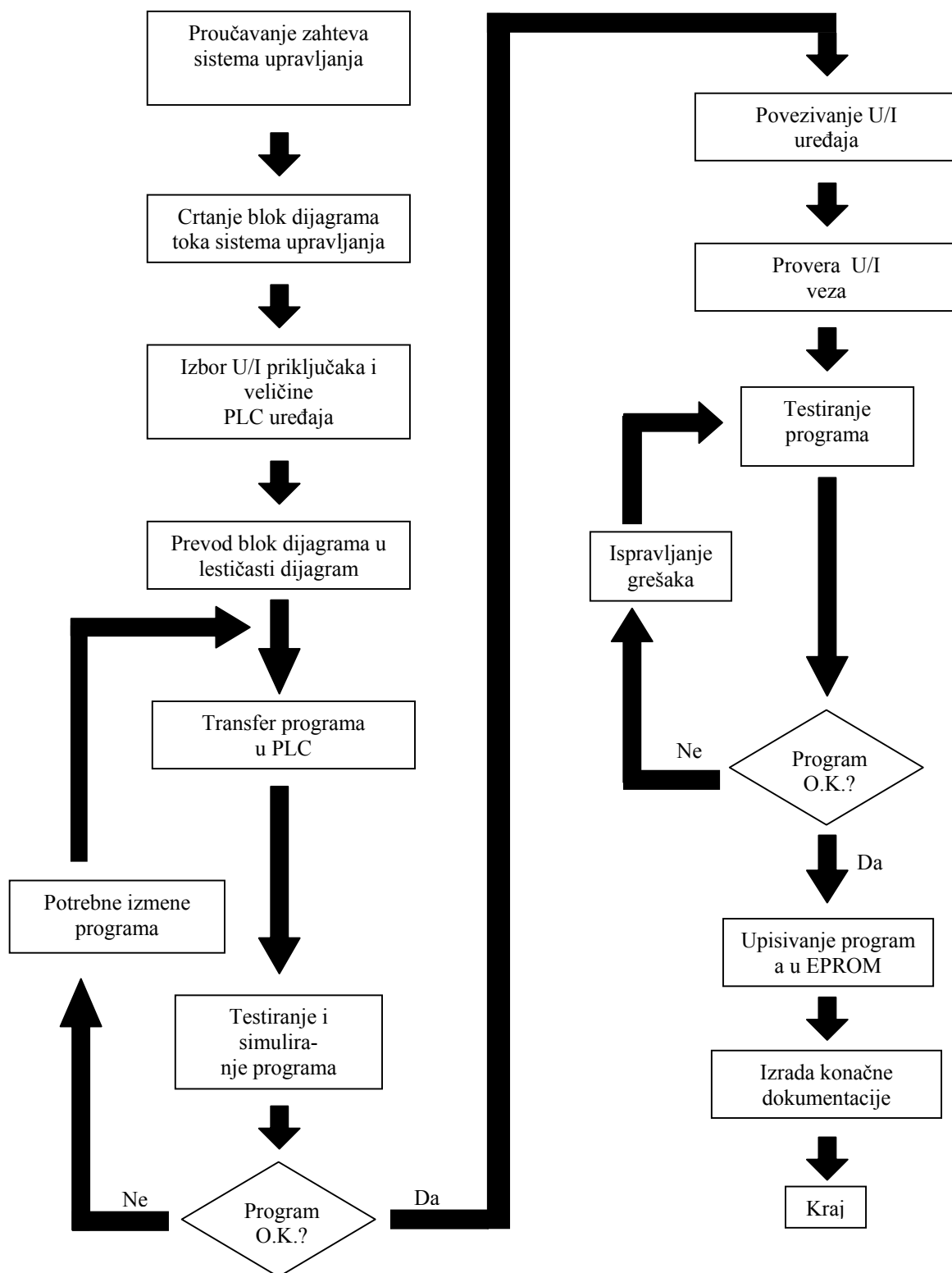
- Potrebno je 80% manje žica za povezivanje u poređenju sa konvencionalnim upravljačkim sistemom.
- Potrošnja je značajno smanjena jer *PLC* znatno manje troši od mnoštva releja.
- Dijagnostičke funkcije *PLC* kontrolera omogućavaju brzo i jednostavno otkrivanje grešaka.
- Izmena u sekvenciji upravljanja ili primena *PLC* uređaja na drugi proces, upravljanja može se jednostavno izvršiti izmenom programa preko konzole ili uz pomoć softvera na računaru (bez potrebe za izmenama u ožičenju, sem ukoliko se ne zahteva dodavanje nekog ulaznog ili izlaznog uređaja).
- Potreban je znatno manji broj rezervnih delova.
- Mnogo je jeftiniji u poređenju sa konvencionalnim sistemom, naročito u sistemima gde je potreban veliki broj *U/I* uređaja.
- Pouzdanost *PLC*-a je veća od pouzdanosti elektro-mehaničkih releja i tajmera.

1.2. Opis sistema upravljanja sa *PLC* uređajem

Sistem koji se automatizuje odnosno na koji se želi primeniti automatsko upravljanje naziva se objekat upravljanja. Rad objekta upravljanja se konstantno prati ulaznim uređajima (senzorima) koji daju informaciju *PLC* uređaju o zbivanju u sistemu. Kao odgovor na to *PLC* šalje signal spoljnim izvršnim elementima koji zapravo kontrolišu rad sistema na način kako je to programer programom odredio. Programer *PLC* uređaj programira na osnovu zahteva i postavljenih kriterijuma definisanih tehnološkim zadatkom. Program se piše u namenskom programskom jeziku, koji svaki proizvođač daje uz svoj *PLC*, a koji predstavlja kombinaciju programskog editora, kompajlera i komunikacionog softvera. U editoru se program piše prateći redosled operacija upravljanja, a zatim se proverava njegova sintaksa i vrši kompajliranje. Ako je sve u redu, komunikacionom vezom softver se šalje u memoriju *PLC*-a gde se smešta i pokreće.

Ulazni i izlazni uređaji, koji se povezuju sa *PLC* uređajem, optimalno se odabiraju na osnovu zahteva i postavljenih kriterijuma definisanih u tehnološkom zadatku koje treba da zadovolje. Ulazni uređaji su prekidači, senzori i davači. Izlazni uređaji mogu biti solenoidi, releji, elektromagnetni ventili, motori, kontaktori kao i uređaji za svetlosnu i zvučnu signalizaciju.

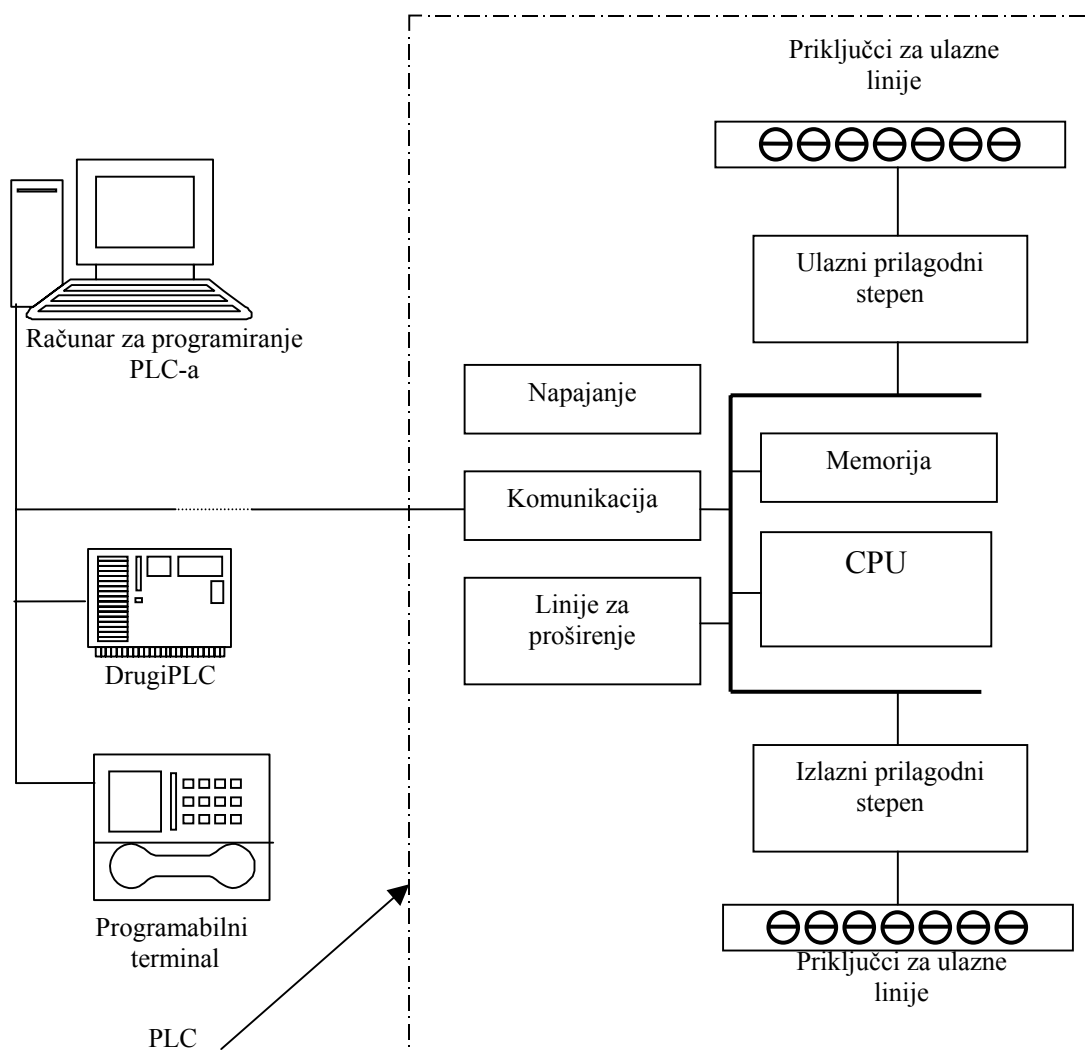
1.3. Sistemski pristup projektovanju sistema upravljanja pomoću PLC uređaja



1.4. Osnovna struktura PLC uređaja

Svi PLC kontroleri bez obzira na veličinu imaju istu hardversku strukturu, sličnu drugim računarskim sistemima, adaptiranu industrijskom okruženju, koja ima iste osnovne celine:

- CPU (centralna procesorska jedinica).
- Memorija za program i podatke.
- Komunikacioni deo.
- Mrežni deo za napajanje.
- Ulazni deo (digitalni, analogni).
- Izlazni deo (digitalni, analogni).
- Deo za proširenje.



Osnovni elementi PLC kontrolera

CPU centralna procesorska jedinica je mozak PLC kontrolera koji odlučuje šta da se radi: brine o komunikaciji, međusobnoj povezanosti ostalih delova ovog kontrolera, izvršavanju programa, upravljanju memorijom, nadgledanjem ulaza i postavljanjem izlaza. Centralna procesorska jedinica

izvodi se sa mikroprocesorom ili mikrokontrolerom kod uređaja sa manjim i srednjim brojem ulaza i izlaza ili kao multiprocesorska kod uređaja sa većim brojem ulaza i izlaza.

PLC kontroler komunicira sa upravljačkim procesom preko analognih, digitalnih i brojačkih ulaza i izlaza. Informacije o stanju ulaza primarno se obrađuju i smeštaju u memoriju stanja ulaza i izlaza.

Memorija se može podeliti na sistemsku i korisničku:

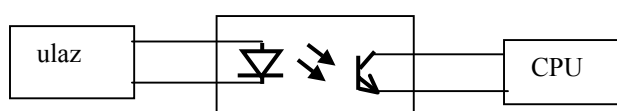
Sistemska memorija se koristi od strane *PLC*-a za operativni sistem. U njoj se pored operativnog sistema nalazi i korisnički program u binarnom obliku. Ova memorija je obično *EEPROM* i može se menjati samo kad se radi o menjanju korisničkog programa. Korisnički program sa algoritmom obrade ulaznih informacija unosi se preko odgovarajućeg programatora, danas obično *PC* računara. Dobra praksa je da se program smešta i u *RAM* memoriju podržanu baterijom, tako da se izvršava iz *RAM*-a, odnosno da se učitava u *RAM* iz *EEPROM*-a svaki put kad se uključuje *PLC*, ili u slučaju gubitka podataka iz *RAM*-a.

Korisnička memorija je podeljena u blokove koji imaju posebne funkcije. Jedan deo ove memorije se koristi za čuvanje stanja ulaza i izlaza, drugi deo se koristi za čuvanje vrednosti promenljivih kao što su vrednosti tajmera i brojača. Na osnovu stanja ulaza, stanja vremenskih članova, brojača i memorisanih međustanja, vrši se obrada koja formira stanja izlaza, koja se prenose u odgovarajuće memorijsko područje a odatle preko internog basa ka izlazu.

Komunikacioni deo obezbeđuju pre svega komunikaciju sa nadređenim programatorom ili *PC* računarom na kojem se piše upravljački program, šalje u *PLC* i zatim proverava njegova funkcionalnost. Ostale mogućnosti su komunikacija sa drugim *PLC* uređajima i raznim senzorima, komunikacija sa operatorskim panelima, nadređenim računarima i modemsom vezom. Gotovo svi *PLC*-i imaju ugrađen serijski port za komunikaciju (*RS232*), a komunikacija se vrši preko protokola koji zavisi od proizvođača (najčešće full duplex serijska veza).

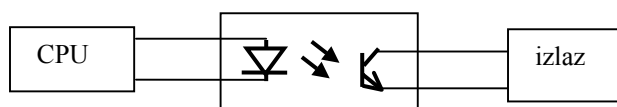
Moduo napajanja obezbeđuje napajanje i neosetljiv je na smetnje koje dolaze iz električne mreže kao i na kraće ispadne mrežnog napona u trajanju od 10 do 15ms. Standardni naponi napajanja su 120/230VAC i 24VDC.

Ulazni prilagodni stepen štiti *CPU* od mogućih prevelikih signala na ulazu. Ulazni prilagodni stepen pretvara nivo stvarne logike u nivo logike koji odgovara *CPU* jedinici. Ovo se najčešće obavlja pomoću optokoplera kod digitalnih ulaza.



Ulazni prilagodni stepen

Izlazni prilagodni stepen takođe mora biti galvanski odvojen. Kod digitalnih izlaza, odvajanje je slično kao i kod ulaza. *CPU* dovodi signal na *LED* i uključuje ga. Svetlost pobuđuje foto tranzistor koji aktivira izlazni uređaj, obično rele koji je sposoban da vrši prekidanje jačih naponskih i strujnih signala.

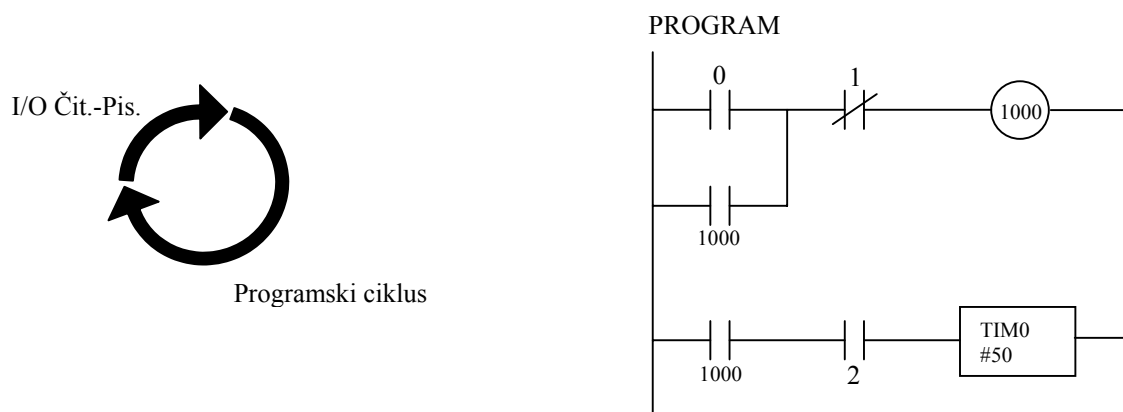


Izlazni prilagodni stepen

Mikro *PLC*-i izvedeni su mehanički obično u okviru jednog kućišta, dok se ostali sastoje iz šasije (*rack*) koji ima određeni broj slotova u koji se stavljaju pojedini funkcionalni moduli. Prva dva slotova u kućištu zauzimaju napajanje i procesorski modul, dok je raspored ostalih modula obično proizvoljan. U zavisnosti od broja modula, *PLC* može imati i više od jednog kućišta. Dodatni moduli i proširenja povezuju se preko dela odnosno *linija za proširenje*.

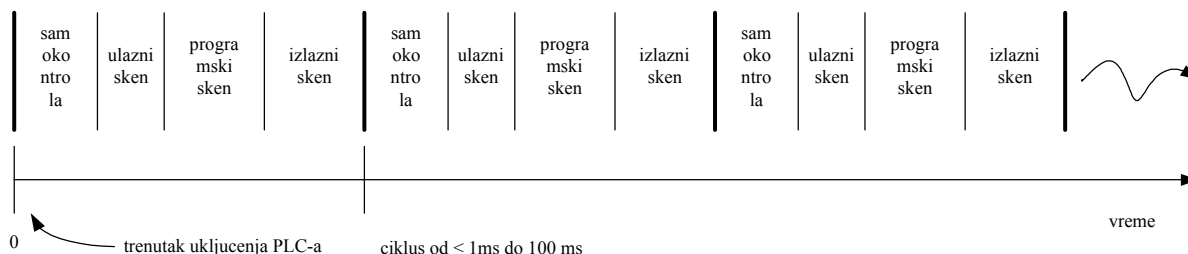
1.5. Princip rada *PLC* uređaja

Prihvatanje ulaza, obrada i ispis izlaza se ciklički ponavlja u skladu sa unesenim programom, određene dužine trajanja, zavisne od broja ulaza i izlaza i složenosti algoritma i vrste primenjenog procesora.



Programski ciklus obrade

Programski ciklus se sastoji od četiri faze. Pri inicijalizaciji, pri uključanju, *PLC* prvo proverava moguće greške u svom hardveru i softveru. Ako ih ne pronade, preuzima stanja ulaza (iz registara ulaza) i kopira njihove vrednosti u memoriju na zato predviđene lokacije. Taj postupak se naziva ulazni sken a podaci u memoriji se nazivaju slika ulaza. Koristeći ulazne podatke, odnosno njihovu sliku, procesor izvršava programske naredbe kojima su definisane odgovarajuće aritmetičko-logičke funkcije u fazi koja se naziva programski sken. Pri tom se rezultati obrade smeštaju u zato predviđeno memorijsko područje nazvano slika izlaza. Po završetku programskog skena u fazi nazvanoj izlazni sken podaci iz slike izlaza prenose se na izlaze (registre izlaza). *PLC* nakon izlaznog skena pokreće nanovo čitav ciklus, proverava greške itd.



*Ciklus rada *PLC*-a*

Provera da li se programski ciklus izvodi pravilno, izvodi se uz pomoć hardverskog *watch-dog* tajmera, koji se proziva u svakom skenu i pretstavlja osnovnu garanciju sigurnog rada. Ako se to ne dogodi signalizira se greška u samom programu ili kvar na opremi kontrolera. Na taj način štiti se sistem upravljanja, naprimer od ulaska u beskonačnu petlju. U zavisnosti od primenjenog tipa

procesora u *PLC*-u, ulazni i izlazni sken izvršavaju se u vremenu reda milisekundi (0.1 do 3ms), tako da se ciklus obrade ponavlja 10 do 100 puta u sekundi. Trajanje skena ciklusa obrade, posebno programskog dela zavisi od veličine programa.

2. Programiranje *PLC*-a

Proizvođači *PLC*-a uz njih isporučuju namenske programske jezike, koji su manje više u skladu sa standardom *IEC 61131-1* (*IEC = International Electrotechnical Commission*). Po tom standardu programski jezici za kodiranje dele se na tekstualne i grafičke. Tekstualni programski jezici su *IL – Instruction List* (klasa asemblerkih jezika) i *ST – Structured Text* (klasa proceduralnih jezika). Grafički programski jezici su *LD – Ladder Diagram* (lestvičast dijagram) i *FBD – Function Block Diagram* (funkcionalni blok dijagram). Neki proizvođači nude i mogućnost programiranja pomoću *BASIC* i *C* programskih jezika, ali ti jezici nemaju širu zastupljenost.

Najčešće upotrebljavan *PLC* programski jezik je kontakti lestvičast dijagram. Ovaj način programiranja ima za osnovu relejnu upravljačku šemu, odnosno njen grafički izgled, prilagođen principima rada *PLC* kontrolera. Ovaj način programiranja korišćen je već kod prvih primena *PLC*-a, kako bi korisnici navikli za izradu šema u relejnoj tehnici, bezbolno prešli na primenu *PLC*-a. Kako je ovaj grafički način programiranja lako shvatljiv i onima koji se nisu bavili relejnim upravljanjem, on se široko odomaćio.

Programski jezici se obično instaliraju na *PC* računar pod *WINDOWS* ili *DOS* platformom, tako da se dobija pristupačna platforma programatora za editovanje, kompajliranje i prenos programa na *PLC*. Komunikacija programatora sa *PLC*-om može biti aktivna i tokom izvođenja programa u njemu. Na taj način na ekranu programatora možemo pratiti stanje ulaza i izlaza tokom rada i zadavati eventualno nove naredbe na jednostavan način.

PLC se takođe može programirati i preko namenskih programatora, obično ručnih koji poseduju mali *LCD* ekran i tastaturu. Takvi uređaji se direktno spajaju na *PLC* i koriste se za kraće programe ili za manje izmene programa, kada se to vrši u pogonu. Za neke jednostavnije primene postoje čak i *PLC* kontroleri koji na sebi poseduju displej i nekoliko funkcijskih tastera, čime se obezbeđuje njihovo programiranje na mestu ugradnje. Neki *PLC*-i su opremljeni izmenljivim *EEPROM* memorijskim karticama, što olakšava programiranje odnosno izmene programa tokom rada. Dovoljno je ugasiti *PLC*, izmeniti memorijsku karticu unapred napunjenu sa novim programom i ponovo uključiti *PLC* koji automatski prihvata novi program.

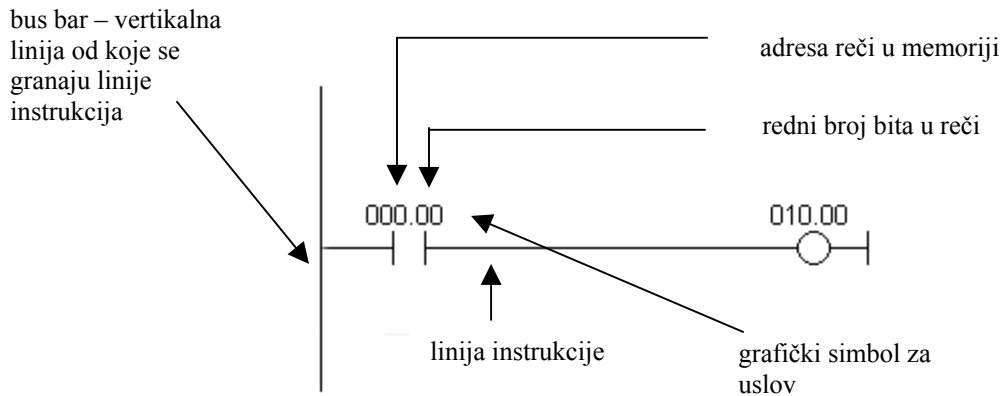
2.1. Više o lestvičastom dijagramu

Programabilni kontroleri su uglavnom i pre svega programiraju u takozvanom lestvičastom dijagramu (pored ovog prisutan je i naziv *ladder* ili relejni dijagram) koji nije ništa drugo do grafičko simbolično predstavljanje električnih kola. Izabrani simboli zapravo izgledaju slično šematskim simbolima električnih uređaja, što je olakšalo prelazak korisnika na programiranje *PLC* kontrolera. Korisnik koji nikad nije video *PLC* može da razume lestvičasti dijagram.

Lestvičasti dijagram sastoji se od jedne vertikalne linije, koja se nalazi na levoj strani, i linija koje se granaju prema desnom delu. Linija sa leve strane naziva se *bus bar* a linije koje se granaju na desno su linije instrukcija. Duž linija instrukcija smešteni su uslovi koji vode do instrukcija pozicioniranih na desnom kraju dijagrama. Logička kombinacija ovih uslova određuje kada i na koji način se instrukcija na desnoj strani izvršava. Osnovni elementi lestvičastog dijagrama vide se na sledećoj slici, na primeru programske linije napisane u programskom jeziku *SYSWIN* za kontroler *CPM1A*.

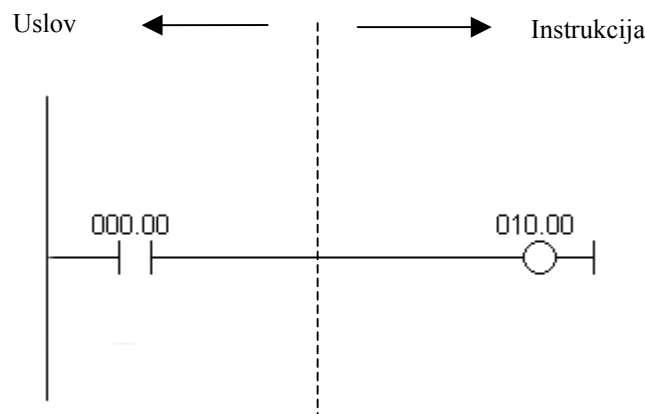
Najveći broj instrukcija zahteva korišćenje najmanje jednog operanda, a često i više njih. Operand može biti neka memorijska lokacija, jedan bit memorijske lokacije ili neka numerička vrednost - broj. U gornjem primeru operand je bit 0 memorijske lokacije *IR000*. U slučaju kada se za

operand želi proglasiti konstanta, koristi se oznaka # ispred numeričkog zapisa (da bi kompajler znao daje u pitanju konstanta a ne adresa).



Osnovni elementi lestvičastog dijagrama

Na osnovu gornje slike treba primetiti da se lestvičasti dijagram sastoji iz dva osnovna dela. Levi deo koji se naziva uslovni i desni koji sadrži instrukcije. Logika je u sledećem, kada se ispuni uslov instrukcija biva izvršena.



Uslov i instrukcija u lestvičastom dijagramu

Gornja slika predstavlja primer lestvičastog dijagrama u kome se aktivira relej u *PLC* kontroleru kada se pojavi signal na ulaznoj liniji 00. Parovi vertikalnih linija nazivaju se uslovi. Svaki uslov u lestvičastom dijagramu ima vrednost *ON* ili *OFF*, zavisno od statusa bita koji mu je dodeljen. U ovom slučaju taj bit je i fizički prisutan kao ulazna linija (klemu) u *PLC* kontroler. Ukoliko se priključi taster na klemu koja mu odgovara, moguće je menjati stanje bita iz stanja logičke jedinice u stanje logičke nule i obratno. Stanje logičke jedinice se najčešće označava kao *ON* a stanje logičke nule kao *OFF* (po engleskim rečuma *on* i *off* koje bi u bukvalnom prevodu značile *uključeno* i *isključeno*).

Desni deo lestvičastog dijagrama je instrukcija koja se izvršava u slučaju da je levi uslov ispunjen. Postoji više vrsta instrukcija koje bi se najlakše mogle podeliti na jednostavne i složene. Primer jednostavne instrukcije je aktiviranje nekog bita u memorijskoj lokaciji. U gornjem primeru taj bit ima i fizičko značenje jer je povezan na relej unutar *PLC* kontrolera. Kada *CPU* aktivira neki

od prva četiri bita u reči IR010 kontakti releja se pomeraju i vrše spajanje linija koje su povezane na njega. U ovom slučaju to su linije spojene na klemu obeleženu sa 00 i jednu od COM klema.

2.2. Normalno otvoreni i normalno zatvoreni kontakti

Pojmove *normalno otvoren* i *normalno zatvoren* veoma je važno razumeti jer se često sreću u industrijskoj praksi. Oba pojma se primenjuju na reči kao što su kontakti, ulaz, izlaz itd (sve kombinacije imaju isto značenje bez obzira da li se radi o ulazu, izlazu, kontaktu ili nečemu drugom).

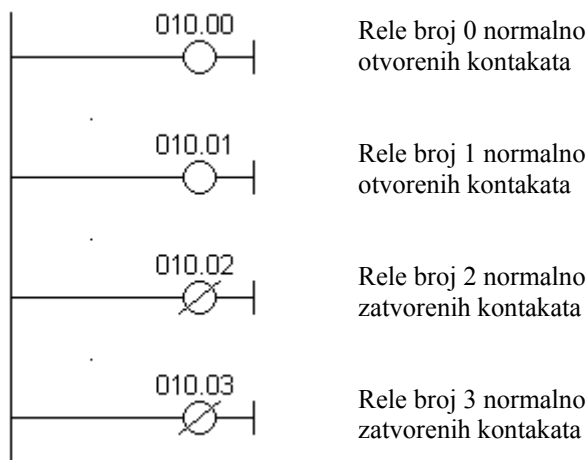
Sušтина je veoma jednostavna, *normalno otvoren* prekidač neće provesti struju dok nije pritisnut a *normalno zatvoren* prekidač će provoditi sve dok nije pritisnut. Dobri primeri za oba slučaja su zvono na ulaznim vratima i alarm za kuću.

Ako se izabere *normalno zatvoren* prekidač, zvono bi stalno radilo sve dok neko ne bi pritisnuo prekidač. Pritiskom na prekidač, otvaraju se kontakti i zaustavlja se protok stuje do zvona. Naravno, tako koncipiran sistem ne bi nikako odgovarao vlasniku kuće. Bolji izbor bi svakako bio *normalno otvoren* prekidač, njegovim korišćenjem zvono neće raditi sve dok neko ne pritisne dugme na prekidaču i time označi svoje prisustvo pred vratima.

Kućni sistem bezbednosti (kućni alarmni sistem) je primer upotrebe *normalno zatvorenog* prekidača. Predpostavimo da je alarmni sistem namenjen nadgledanju ulaznih vrata u kuću. Jedan od načina da se ožiči kuća bi bio da se sprovede jedan normalno otvoren prekidač od svakih vrata do alarma (upravo kao i prekidač za zvono). Tada, ako bi se vrata otvorila, to bi zatvorilo prekidač i alarm bi se aktivirao. Ovako izveden sistem bi radio ali bi bilo problema. Neka se pretpostavi da prekidač ne radi, da je žica slučajno u prekidu ili se recimo prekidač polomi, itd (ima mnogo načina na koje bi sistem mogao da postane nefunkcionalan). Problem je što domaćin nikad ne bi znao da sistem ne radi. Provalnik bi mogao da otvori vrata, prekidač ne bi radio i alarm se ne bi aktivirao. Očigledno ovo nije dobar način kako napraviti sistem. Sistem treba da se postavi tako da se alarm aktivira od strane provalnika ali i sam od sebe ako neka od komponenti ne funkcioniše (domaćin svakako želi da zna ako sistem ne radi). Obzirom na ove nove okolnosti bolje je koristiti prekidač sa *normalno zatvorenim* kontaktima koji će detektovati neovlašćen ulaz (otvaranje vrata prekida tok struje i taj signal se koristi za aktiviranje zvučnog signala) ili kvar na sistemu kao što je prekid žice. Razmatranja kao što su ova su još značajnija u industrijskom okruženju gde bi kvar mogao da prouzrokuje povredu nekog radnika. Jedan od takvih primera gde se koriste izlazi sa normalno zatvorenim kontaktima je sigurnosna ograda kod mašina za sečenje. Ukoliko se vrata ograde otvore prekidač deluje na izlaz sa *normalno zatvorenim kontaktima* i prekida kolo za napajanje čime mašina staje i time sprečava povređivanje radnika.

Pojmovi normalno otvoren i normalno zatvoren se mogu primeniti i na senzore. Senzori se koriste da bi se osetilo prisustvo fizičkih objekata, izmerila neka količina ili veličina. Na primer, jedna vrsta senzora može da se koristi da bi se detektovalo prisustvo kutije na pokretnoj traci, druga vrsta može da se koristi za merenje fizičke veličine kao što je toplota itd. Ipak, većina senzora je tipa prekidača. Njihov izlaz je u stanju *ON* ili *OFF* u zavisnosti od toga šta senzor oseća. Neka se kao primer uzme senzor koji je napravljen da oseti metal kada metalni deo prolazi kraj senzora. Za tu namenu mogao bi se upotrebiti senzor sa *normalno otvorenim* ili sa *normalno zatvorenim* kontaktom na izlazu. Ako bi bilo potrebno obavestiti *PLC* svaki put kada deo prođe kraj senzora, trebalo bi izabrati senzor sa *normalno otvorenim* izlazom. Izlaz senzora bi se aktivirao samo ako bi metalni deo bio ispred senzora i odmah isključio kad bi deo prošao. *PLC* bi onda mogao da izračuna broj puta koliko se *normalno otvoren* kontakt na izlazu senzora aktivirao i time znao koliko je metalnih delova prošlo kraj senzora.

Pojmove *normalno otvoren* i *normalno zatvoren* kontakt treba i konkretno pojasniti na primeru ulaza i izlaza samog *PLC* kontrolera. Najlakše ih je objasniti baš na primeru releja.



Primer normalno otvorenih i zatvorenih kontakata na izlazu PLC kontrolera

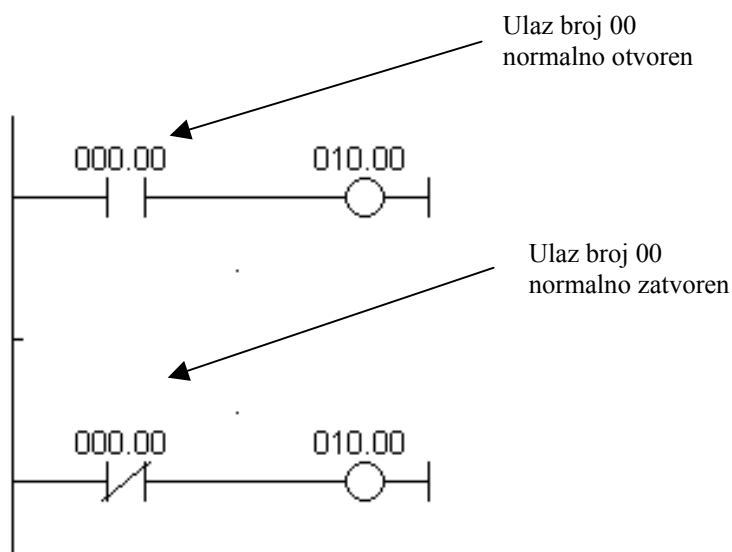
Normalno otvoreni kontakti bi predstavljali kontakte releja koji će po dovođenju signala izvršiti spoj. Za razliku od njih kod *normalno zatvorenih* kontakata signal će prekinuti kontakt tj. isključiti relej. Na prethodnoj slici se vidi kako to izgleda u praksi. Prva dva releja su definisani kao *normalno otvoreni* a druga dva kao *normalno zatvoreni*. Svi releji reaguju na pojavu signala! Prvi rele (00) ima signal i zatvara svoje kontakte. Drugi relej (01) nema signal i ostaje otvoren. Treći relej (02) ima signal i otvara svoje kontakte s obzirom da je definisan kao zatvoreni kontakt. Četvrti relej (03) nema signal i ostaje zatvoren jer je tako i definisan.

Pojmovi *normalno otvoren* i *normalno zatvoren* se mogu odnositi i na ulaze *PLC* kontrolera. Neka se kao primer ulaza u *PLC* kontroler iskoristi taster. Ulaz na koji je taster priključen se može definisati kao ulaz sa *otvorenim* ili *zatvorenim* kontaktima. Ako je definisan kao ulaz sa *normalno otvorenim* kontaktom pritisak na taster će aktivirati instrukciju koja se nalazi iza uslova. U ovom slučaju to će biti aktiviranje releja 00.

Ako je ulaz definisan kao ulaz sa *normalno zatvorenim* kontaktom pritisak na taster će prekinuti izvršenje instrukcije koja se nalazi iza uslova. U ovom slučaju to će prouzrokovati deaktiviranje releja 00 (rele je aktivan sve dok se ne pritisne taster). Na slici ispod je prikazan način povezivanja tastera i relejni dijagrami za oba slučaja.

Normalno otvoreni / zatvoreni uslovi se u lestvičastom dijagramu razlikuju po dijagonalnoj liniji preko simbola. Ono što određuje uslov izvršenja (eng. *execution condition*) za instrukciju jeste status bita naznačenog ispod svakog uslova na liniji instrukcije. Normalno otvoreni uslov je *ON* ako njegov operand bit ima status *ON*, odnosno, njegov je status *OFF*, ako je takav i status njegovog operand bita. Normalno zatvoreni uslov je *ON* kada je njegov operand bit *OFF*, odnosno on ima status *OFF* kada je status njegovog operand bita *ON*.

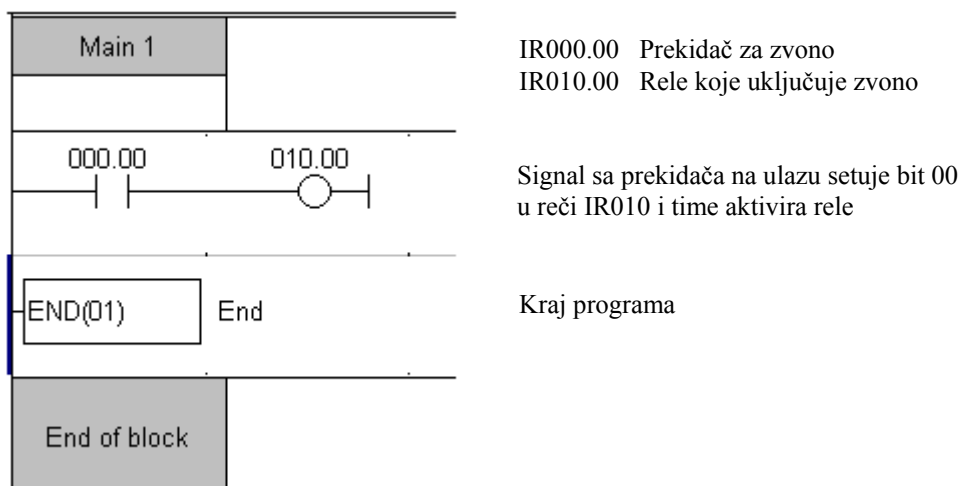
U programiranju sa lestvičastim dijagramom, logička kombinacija *ON* i *OFF* uslova postavljenih ispred instrukcije određuje konačni uslov pod kojim će instrukcija biti izvršena, ili ne. Ovaj uslov, koji može imati samo vrednosti *ON* ili *OFF*, naziva se uslov izvršenja instrukcije. Sve instrukcije izuzev *LOAD* instrukcije imaju uslov izvršenja. Operand dodeljen bilo kojoj instrukciji u lestvičastom dijagramu može biti određen statusom I/O bitova, flegova, radnih bitova, tajmera/brojača, itd (bilo koji bit iz IR, SR, HR, AR, LR ili TC sektora memorijske mape *Omron*-ovih kontrolera).



Primer normalno otvorenih i zatvorenih kontakata na ulazu PLC kontrolera

2.3. Kratak primer

Donji primer predstavlja elementaran program. Primer se sastoji od jednog ulaznog uređaja i jednog izlaznog vezanog na izlaz *PLC* kontrolera. Ulazni uređaj je taster a izlazni zvonice koje se napaja preko kontakta releja 00 na izlazu *PLC* kontrolera. Ulaz 000.00 predstavlja uslov za izvršenje instrukcije nad bitom 010.00. Pritiskom na taster aktivira se bit 000.00 i ispunjava uslov za aktiviranje bita 010.00, čime zvono počinje sa radom. Za ispravan rad programa potrebna je još jedna linija programa sa *END* instrukcijom i time je program završen.

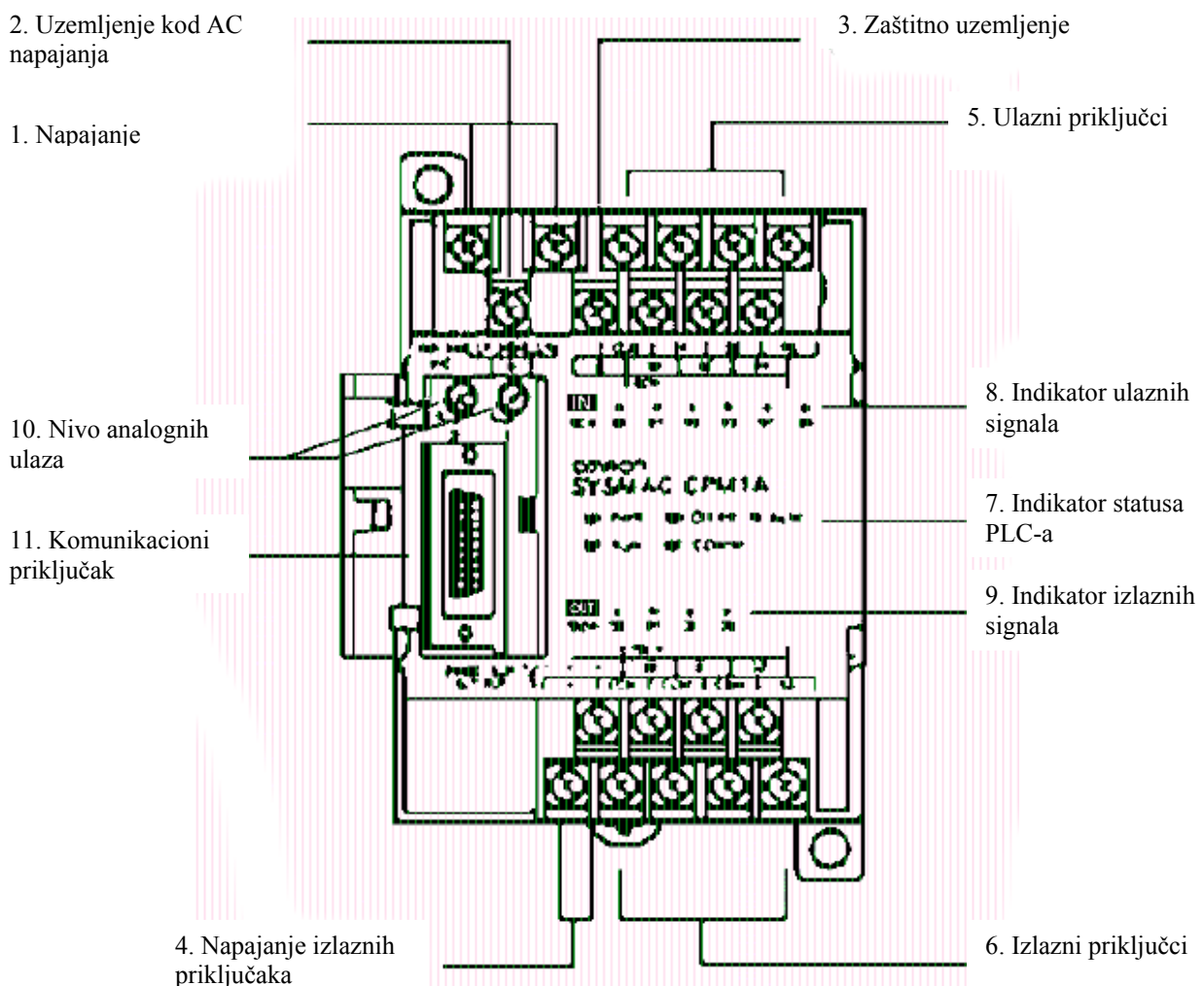


Lestvičasti dijagram kratkog primera

3. PLC kontroler CPM1A

CPM1A pripada grupi malih kontrolera serije C programabilnih logičkih kontrolera kompanije *Omron*. Serija poseduje veliku fleksibilnost u pogledu proširenja sa novim ulazno-izlaznim linijama, male dimenzije i velik izbor kontrolera i proširenja. Ukoliko se u procesu pojavi potreba za dodatnim U/I linijama postojeći sistem se lako dograđuje analogno-digitalnim proširenjima. Njegove male dimenzije su kao stvorene za ugradnju u prostore koji su ograničeni po pitanju mesta za montažu. Broj linija kojim se može upravljati kreće se od 10 do 100 ulaza-izlaza. Odnos ulaza i izlaza je 60:40% u korist ulaznih linija. Svaka od izlaznih linija može imati relejni ili tranzistorski izlaz. Uz opcioni komunikacioni adapter CPM1A se lako može vezati na programabilni terminal, programsku konzolu, drugi PLC ili PC.

Konkretni izgled PLC kontrolera tipa CPM1A (6 ulaza + 4 izlaza), vidi se na sledećoj slici.



Spoljni izgled PLC kontrolera CPM1A

U daljem tekstu dati su samo osnovni podaci za kontroler dovoljni za izvođenje jednostavnijih zadataka objašnjenih u ovom uputstvu. Detaljniji podaci mogu se naći u opisu proizvođača.

Alati za programiranje: CPM1A je podržan istim programskim konzolama i SYSWIN programskim alatom kao veći kontroleri iz serije SYSMAC C. SYSWIN je programski alat za

Windows 3.1 i Windows 95 koji omogućava programiranje, debugging i monitoring sa personalnog računara.

Modeli: CPM1A CPU se izrađuje u varijantama po 10, 20, 30 i 40 priključnih tačaka.

Napajanje: Svi moduli mogu biti napajani sa jednosmernog (24 VDC) ili naizmeničnog izvora (100 do 240 VAC, 50/60Hz), što se označava sa karakterom "A" ili "D" na kraju tipske oznake.

Potrošnja: Pri jednosmernom napajanju CPU moduli od 10 i 20 I/O linija troše 6 W a moduli od 30 i 40 I/O linija 20W maksimalno. Na naizmeničnom napajanju potrošnja iznosi 30 VAC i 60VAC respektivno.

Proširenje: Na CPU module sa 30 i 40 priključnih linija se mogu dodati analogna i digitalna proširenja do ukupno 100 U/I linija.

Memorija: CPM1A poseduje data memoriju od 1024 reči i memoriju za program veličine 2048 reči.

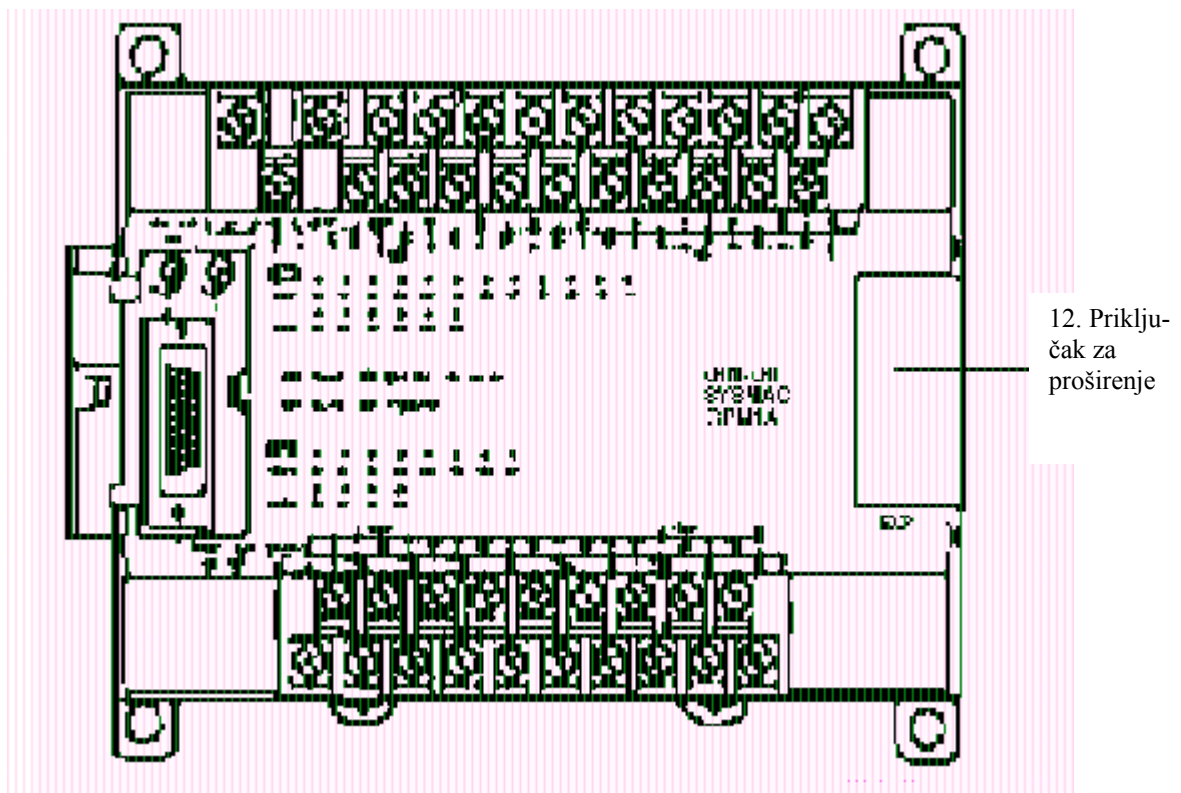
Set instrukcija: Postoji 14 osnovnih i 77 tipova specijalnih instrukcija. Brzina izvršavanja osnovnih instrukcija iznosi 1.72 ms a specijalnih 16.3 ms.

Tajmeri/Brojači: Ukupno postoji 128 tajmera (TIM/CNT 000 do TIM/CNT 127) sa rezolucijom od 10 ms

Analogno zadavanje vrednosti: CPM1A poseduje dva analogna potencimetra koja se mogu koristiti za podešavanja analognih tajmera ili brojača. Okretanjem osovine potencimetra u SR deo se smešta BCD vrednost u intervalu od 0-200.

Brzi brojači: Brojači velike brzine se mogu koristiti u inkrementalnom i "up/down" modu što omogućava jednostavno priključivanje inkrementalnih enkodera.

Brzi ulazi: Na CPU modulima sa 10 priključnih tačaka postoje dva a na CPU modulima sa 20, 30 i 40 priključnih tačaka četiri brza ulaza.



Priključak za proširenje PLC kontrolera CPM1A

3.1. Memorijska mapa PLC kontrolera CPM1A

Pod memorijskom mapom podrazumeva se organizacija memorije PLC kontrolera. Memorija je podeljena na delove, kojima je pridodata određena uloga. Memorija CPM1A kontrolera organizovana je u 16-bitnim rečima. Skup više reči istih funkcija čini oblast, a skup oblasti čini memoriju PLC kontrolera. Memorija PLC-a skoro u potpunosti je adresabilna po bitima, što znači da je dovoljno napisati adresu memorijske lokacije i iza nje broj bita da bi se manipuliralo sa njim (Naprimera $201.07 = 201$ reč i njen sedmi bit).

Oblast podataka		Reči	Bit(ovi)	Funkcija
IR oblast	Ulazi	IR 000 to IR 009 (10 reči)	IR 00000 do IR 00915 (160 bita)	Ovi bitovi dodeljuju se spoljašnjim U/I priključcima. Neki od njih imaju direktan izlaz na klemne. Kod CPM1A modela ulazi su od IR000.00 do IR000.05 a izlazi od IR010.00 do IR010.03.
	Izlazi	IR010 to IR019 (10 reči)	IR 01000 do IR 01915 (160 bita)	
	Radna oblast	IR 200 to IR 231 (32 reči)	IR 20000 do IR 23115 (512 bita)	
SR		SR 232 to SR 255 (24 reči)	SR 23200 do SR 25515 (384 bita)	Ovi bitovi imaju specijalne funkcije kao što su pokazivači i kontrolni bitovi.
TR oblast		--	TR 0 do TR 7 (8 bita)	Ovi bitovi se koriste za privremeno memorisanje ON/OFF stanja prilikom skokova u programu.
HR oblast ²		HR 00 to HR 19 (20 reči)	HR 0000 do HR 1915 (320 bita)	Ovi bitovi se koriste za memorisanje ON/OFF stanja kod nestanka napajanja..
AR oblast		AR00 to AR 15 (16 reči)	AR 0000 do AR 1515 (256 bita)	Ovi bitovi imaju specijalne funkcije kao što su pokazivači i kontrolni bitovi.
LR oblast ¹		LR 00 to LR 15 (16 reči)	LR 00000 do LR 1515 (256 bita)	Koriste se kod 1:1 povezivanja sa drugim PLC-om.
Tajmer /brojačka oblast		TC 000 to TC 127 (brojevi tajmera/brojača) ³		Isti brojevi se koriste i za tajmere i za brojače.
DM oblast	Čitanje/pisanje ²	DM 0000 to DM0999 (1,002 reči)	---	Podacima iz DM oblasti može se pristupiti samo u rečima. Sadržaj reči se čuva i pri nestanku napajanja.
	Zapis grešaka ⁴	DM1000 to DM 1021 (22 reči)	---	Deo memorije u kom se čuvaju informacije o vremenu i kodu nastale greške. Kada se ne koriste za tu namenu, mogu se koristiti kao obične DM reči za čitanje i pisanje.
	Samo čitanje ⁴	DM6144 to DM 6599	---	Programom se ne mogu menjati.
	PC Setup ⁴	DM 6600 to DM 6655 (56 reči)	---	Koriste se za čuvanje raznih parametara koji kontrolišu rad PLC kontrolera.

- Primedbe:**
1. IR and LR bitovi koji se ne koriste za njihove funkcije, mogu se koristiti kao radni bitovi.
 2. Sadržaj HR oblasti, LR oblasti, brojačke oblasti, i DM oblast za čitanje i pisanje čuva se sa *backup* kondenzatorom. Na temperaturi 25 °C, kondenzator čuva sadržaj memorije 20 dana.
 3. Prilikom pristupa PV, TC brojevi koriste se za podatke u obliku reči; kada se pristupa Completing Flag-ovima, koriste se kao bit podaci.
 4. Podaci od DM6144 to DM6655 ne mogu se menjati iz programa, ali ih može promeniti periferni uređaj.

4. SYSWIN programski paket za programiranje PLC kontrolera

4.1. Šta je to SYSWIN?

SYSWIN je programski paket za rad sa *Omron*-ovim programabilnim kontrolerima C i CV klase. Namenjen je kreiranju i održavanju programa, kao i testiranju rada PLC kontrolera, kako u simulacionom tako i u radnom režimu kontrolera.

SYSWIN 3.4 je programski paket koji radi u Windows 98 okruženju. Osnovna namena paketa je uređivačka, odnosno editorska, što znači da je podržano grafičko okruženje za unošenje programa, u obliku lestvičastog dijagrama. Organizaciju programa pisanog korišćenjem SYSWIN-a olakšava njegova blokovska struktura. Korišćenjem dva nivoa obima ovih blokova, uz mogućnost pisanja komentara, moguće je program podeliti u logičke celine. Standardne Windows "clip-board" procedure manipulacija ovim blokovima, kao i kompletnim programima olakšavaju rad nudeći mogućnost pravljenja standardnih blokova za pojedine segmente koji se često koriste (inicijalizacija kontrolera, programiranje reakcije na sistemske alarme, i sl.). Korisnički interfejs SYSWIN-a je u potpunosti prilagođen Windows okruženju. Alatima za editovanje programa, kao i ostale operacije, moguće je pristupiti iz padajućih menija, a postoje i prečice sa tasterima, pomoću kojih se na brz i lak način mogu koristiti ugrađene funkcije softvera.

Sistemska je podržana komunikacija sa *Omron*-ovim programabilnim kontrolerima. Tip kontrolera potrebno je specificirati na početku rada, jer paket automatski ograničava set instrukcija koji korisniku stoji na raspolaganju na one koji podržava konkretan tip kontrolera. Pored toga, korisniku će biti onemogućeno i korišćenje memorijskih lokacija koje ne postoje u memoriji kontrolera za koji se piše program. Pozivanje naprednih funkcija je jednostavno, kao i unošenje njihovih parametara. SYSWIN ne dozvoljava unošenje vrednosti parametara izvan dozvoljenog opsega za konkretnu instrukciju.

Komunikaciju paketa sa kontrolerom moguće je ostvariti u nekoliko modova. Kao prvo u programskom modu, moguće je kreirati program prebaciti u kontroler, ili izvršiti proces u suprotnom smeru, odnosno izčitati već postojeći program iz kontrolera. Ako u toku rada kontrolera imamo potrebu da promenimo program, paket će zatražiti dozvolu da pređe u programski mod, a posle izmene, sam će predložiti povratak u radni režim. Nadzor rada kontrolera izvodi se u režimu za nadzor, takozvanom monitor modu. U njemu je moguće nadgledati izvršenje programa, uz ručno podešavanje odnosno postavljanje uslova. U ovom modu podržava se niz se komandi za korisničku intervenciju, kao što su forsirano setovanje / resetovanje bitova, postavljanje vrednosti na pojedinačne memorijske adrese i drugo. Pored pojedinačnih forsiranja, moguće je napraviti listu adresa nad čijim sadržajima će se izvršiti određena operacija. Uz istovremeni monitoring većeg broja bitova i reči, ova funkcija može predstavljati efikasan alat za otklanjanje grešaka u programu. Konačno, paket može raditi i u radnom režimu, takozvanom RUN modu.

Na početku rada na nekom projektu, pored izbora vrste kontrolera, paket nudi i opciju za izbor kontrolera za koji se piše program i podešavanje parametara u različitim tipovima mreža (naprimer serijsku komunikaciju i Ethernet). Podržani su protokoli za komunikaciju kontrolera sa *Omron*-ovim uređajima za automatizaciju svih nivoa. Ove mogućnosti čine integraciju pojedinačnih kontrolera u mrežu veoma jednostavnom.

Pored alata za editovanje programa i komunikaciju sa kontrolerom postoji i velik broj dodatnih funkcija za otkrivanje grešaka u programu, kao što su skokovi na nepostojeće lokacije, neispravno programirane petlje, nepravilno napisane potprograme ili interapt rutine i drugo.

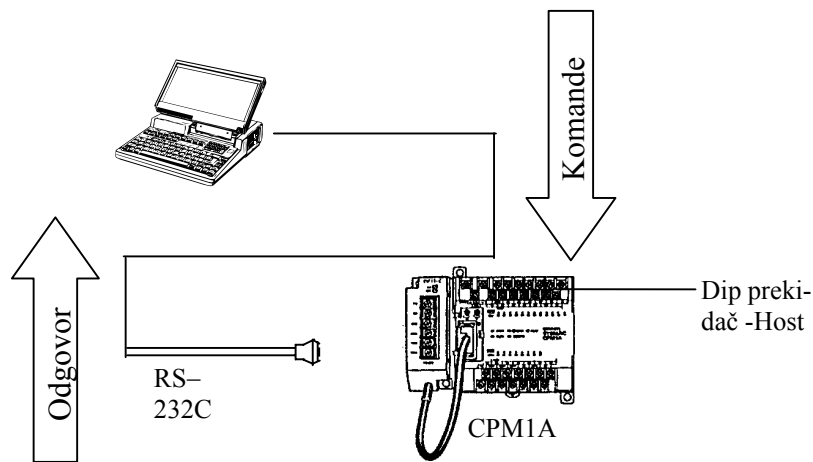
SYSWIN nudi i mogućnosti za dokumentovanje programa, u različitom obliku od lestvičastog dijagrama sa komentarima, liste korišćenih adresa do mnemoničkog zapisa.

4.2. Hardverski i softverski preduslovi

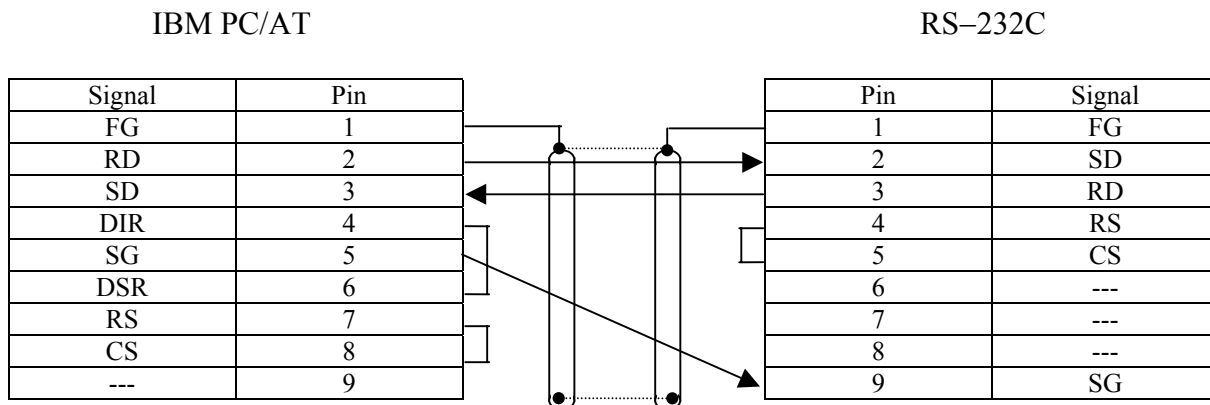
Neophodni uslovi za pokretanje SYSWIN-a su Microsoft Windows okruženje na standardnom IBM ili kompatibilnom 386/486 ili Pentium računar, sa najmanje 8MB RAM-a i 10MB slobodnog prostora na disku.

4.3. Povezivanje PLC kontrolera i PC računara

PLC kontroler se povezuje sa PC računarom preko RS-232 kabla. Jedan kraj kabla se povezuje na serijski port PC-a (9-pinski ili 25-pinski konektor), dok se drugi kraj povezuje sa RS-232C konektorom na RS232 modulu kontrolera CPM1A. Da bi se uspostavila veza sa računarom DIP 1 konektora mora biti postavljen na poziciju *Host*.



Povezivanje PLC kontrolera i PC računara



Šema kabla za povezivanje PLC kontrolera i PC računara

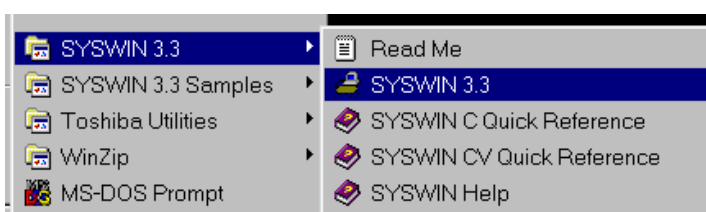
4.4. Instalacija programskog paketa SYSWIN

Paket za obuku za rad sa CPM1A ide na tri instalacione diskete SYSWIN-a. Može se instalirati pod operativnim sistemom Windows ver. 3.1, 3.11, 95, 98 ili NT ver. 4.0. Instalacija se pokreće odabiranjem opcije *RUN* iz *START* menija, posle čega se pojavljuje dijaloški box u koji treba upisati putanju do datoteke *setup.exe*. Pomenuta datoteka može se naći u instalacionim

direktorijumima programa SYSWIN. Nakon kraće procedure instalacije dobija se programska grupa SYSWIN kao na slici.

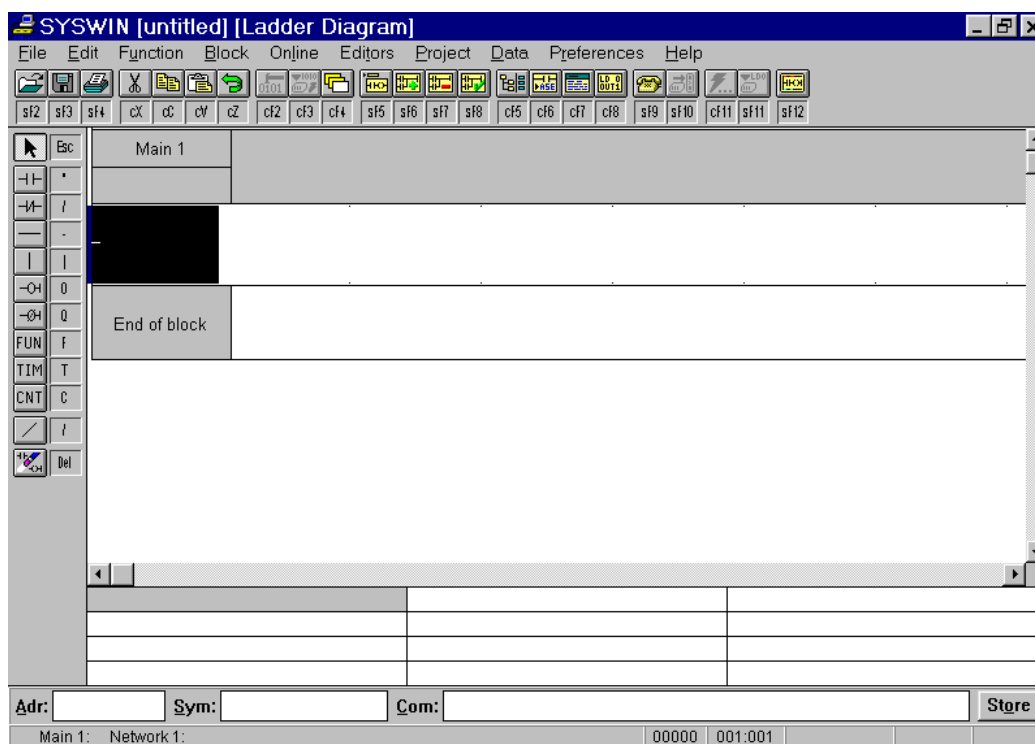


Dijaloški box za pokretanje SYSWIN instalacije



Grupa SYSWIN koja se dobija nakon instalacije

Duplim klikom na ikonu SYSWIN startujemo program koji otvara radnu površinu kao na sledećoj slici.



Radna površina SYSWIN koja se dobija nakon pokretanja programa

4.5. Pisanje prvog programa

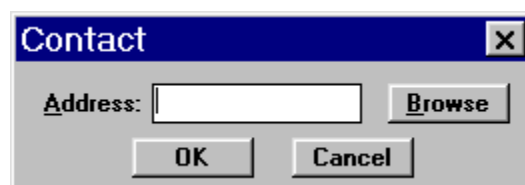
Pisanje programa počinje odabiranjem opcije *New Project* iz *File* menija. U dijalogu, koji se pojavljuje treba izabrati opcije kao na slici.



Podешavanje opcija projekta

Klikom na dugme *OK* odabira se *PLC* kontroler i program je spreman za upotrebu. Na početku potrebno je (preporučuje se) u zaglavlju programa napisati ime programa, autorovo ime i ulaze/izlaze koji se koriste. Ovo može izgledati kao gubljenje vremena ali nije, jer se pisanje komentara u budućnosti isplaćuje.

Program koji ćemo napisati na početku je elentaran i namenjen upoznavanju SYSWIN-a. Program detektuje pritisak na taster i aktivira relej na izlazu *PLC* kontrolera. Dokle god je taster pritisnut i rele je aktivan. Rad releja i tastera može se pratiti preko LED dioda na kućištu *PLC* kontrolera. Pisanje programa počinje klikom na prvu ikonicu sa leve strane koja se prepoznaje po dve vertikalne crte. Ispod nje je druga ikonica slična prvoj ali sa kosom crtom. Te dve ikonice odgovaraju pojmovima normalno otvoren i normalno zatvoren kontakt sa kojima počinju sve linije instrukcija. Klikom na prvu ikonicu bira se opcija sa otvorenim kontaktom a klikom na crni pravougaonik odmah desno pojavljuje se mali prozor u koga treba upisati adresu bita na koga se kontakt odnosi.

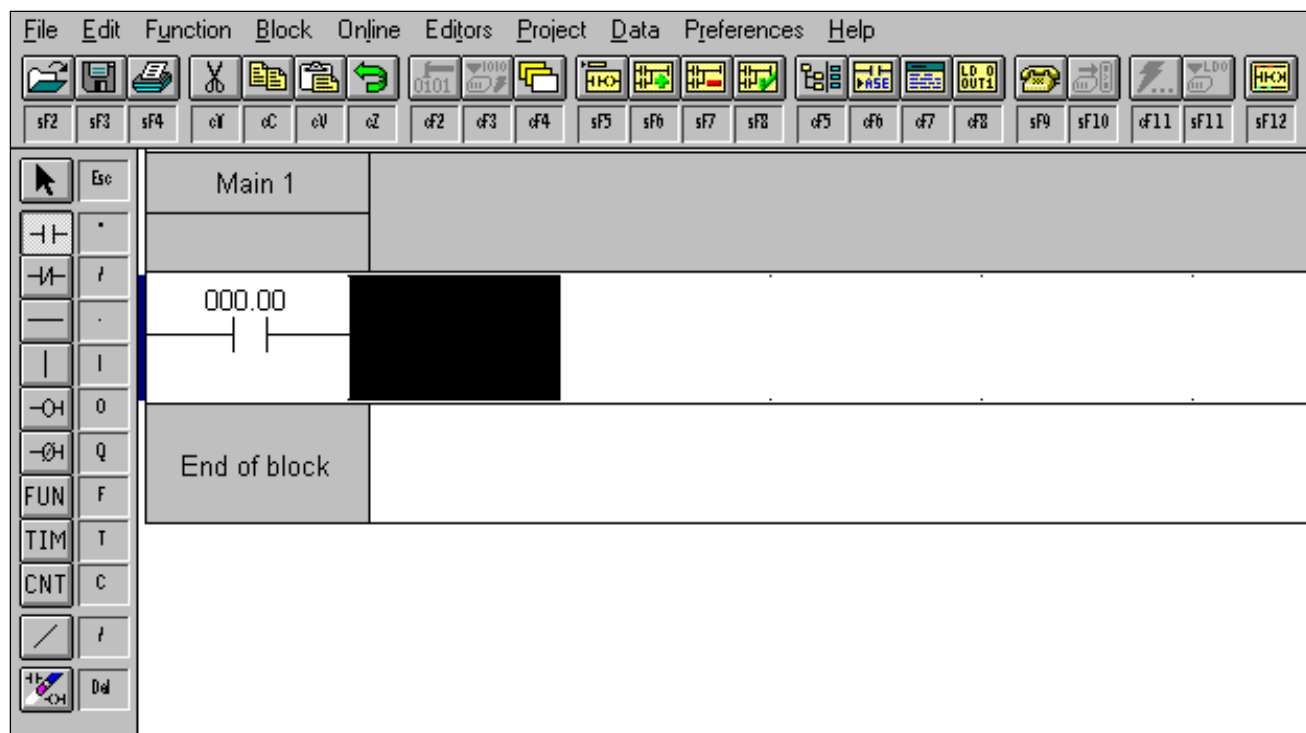


Dijaloški box za upis adrese bita na koga se odnosi kontakt

Veoma je važno da se adrese u programiranju pomoću SYSWIN-a koriste na regularan način. Adrese mogu imati dva dela, prvi koji se odnosi na adresu reči i drugi koji se odnosi na adresu bita u

toj reči (oba broja moraju biti odvojena tačkom). Na primer, ako se koristi adresa 200, SYSWIN će to protumačiti kao 2.00 i biće pozvan nulti bit čija je adresa reči 2. Ako se želi pristupiti reči 200, odnosno njenom nultom bitu, mora se koristiti poziv 20000 ili bolje 200.00. U ovom primeru za adresu ulaza (taster) zadaje se adresa 000.00 koja predstavlja nulti bit reči 000 iz memorijske oblasti IR. Jednostavnije rečeno to je klema na ulazu označena kao 00 ulaz. Spajanjem tastera na nju i jednu od COMM klemu ostvaruje se potrebna veza PLC kontrolera i tastera.

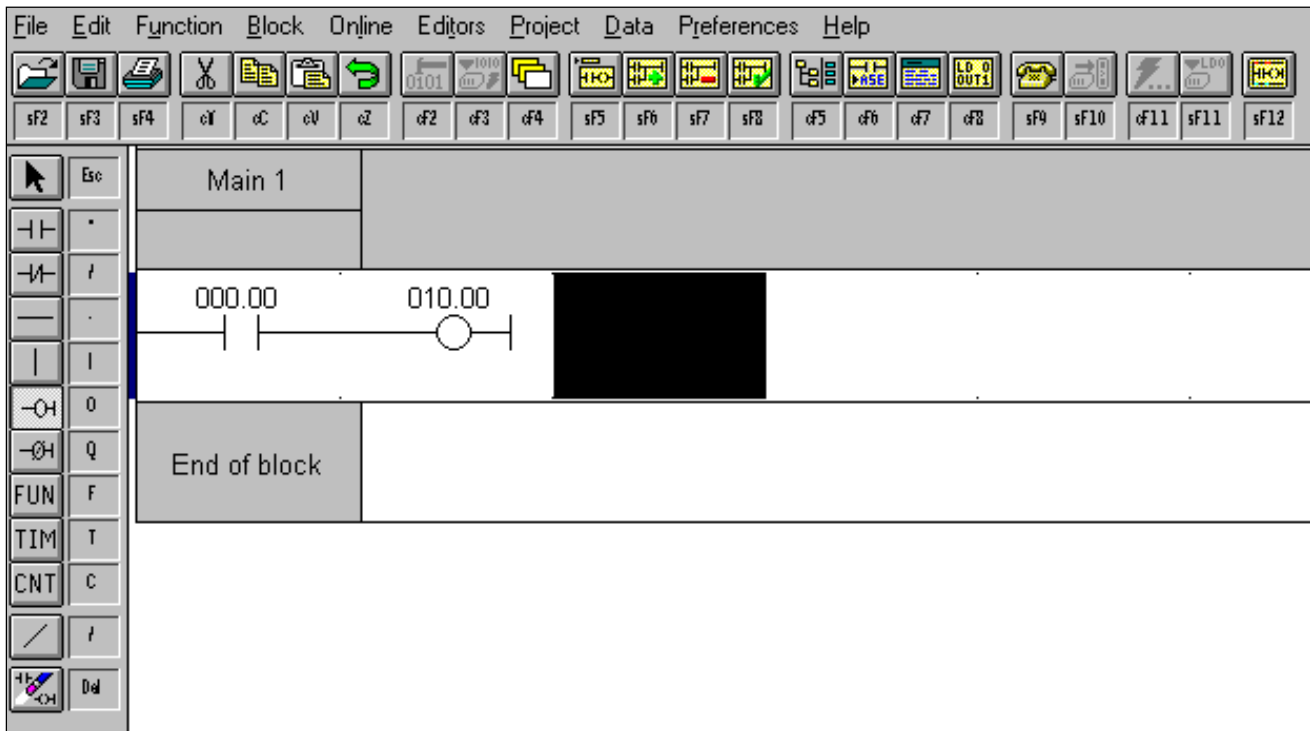
Nakon upisa 000.00 pritiskom na taster *OK* pojavljuje se prvi segment programa. Iznad simbola sa dve vertikalne crte pojavljuje se adresa bita na koga se simbol odnosi a crni pravougaonik se pomera za jedno polje desno.



Postavljen prvi element programa proba.swp

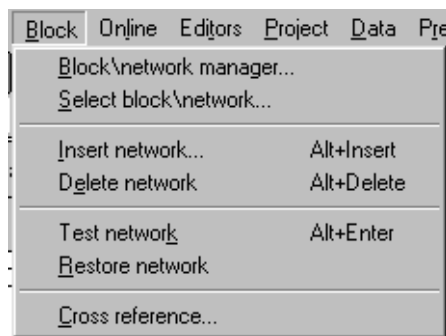
Prve instrukcije do bus bar-a nazivaju se *uslovi* jer se njihovim ispunjenjem aktiviraju instrukcije koje se nalaze desno od uslovnih instrukcija. Kad je unešen uslov potrebno je uneti i odgovarajuću instrukciju koja se aktivira ispunjenjem uslova. U ovom primeru to je rele koga kontroliše bit 00 u reči 010 memorijske oblasti IR. Izlazne instrukcije predstavljene su krugom ili krugom sa crtom ako je reč o normalno zatvorenom kontaktu. Klik na ikonicu sa krugom bira se opcija izlaza sa normalno otvorenim kontaktima. Novim klikom na crni pravougaonik pojavljuje se *contact* prozor u koji treba upisati adresu izlaznog bita 010.00. Izlazni deo IR oblasti nalazi se na IR010 a prva četiri bita te reči predstavljaju releje unutar PLC kontrolera (ako se radi o modelu CPMIA sa relejnim izlazima). Do sada urađen program izgleda kao na sledećoj slici.

Osnovna funkcionalna celina jednog programa jeste *network*. Program se sastoji od više *network*-a koji se nalaze jedan ispod drugog. Operacije sa njima nalaze se u opciji *Block* u meniju. Od svih opcija najviše se koriste dve elementarne, *Insert network* i *Delete network*. U dokumentaciji drugih proizvođača PLC kontrolera umesto pojma *network* u upotrebi su neki drugi termini kao na primer *rung* što bi u prevodu bilo prečka ili deo lestvice.

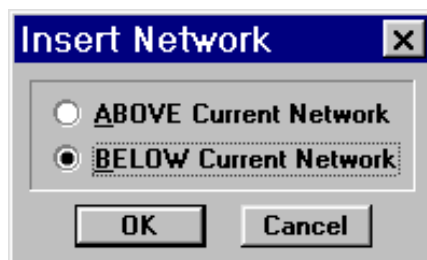


Postavljen drugi element programa proba.swp

Jednostavnije rečeno radi se o sekvenci *PLC* programa koja sadrži jednu ili više izvršnih instrukcija i koja zajedno sa *END* instrukcijom može da čini jedan ispravan *PLC* program. Kako je prvi *network* u programu iskorišćen potrebno je dodati sledeći. Dodavanje *network*-a vrši se komandom *Insert network* iz menija *Block*.

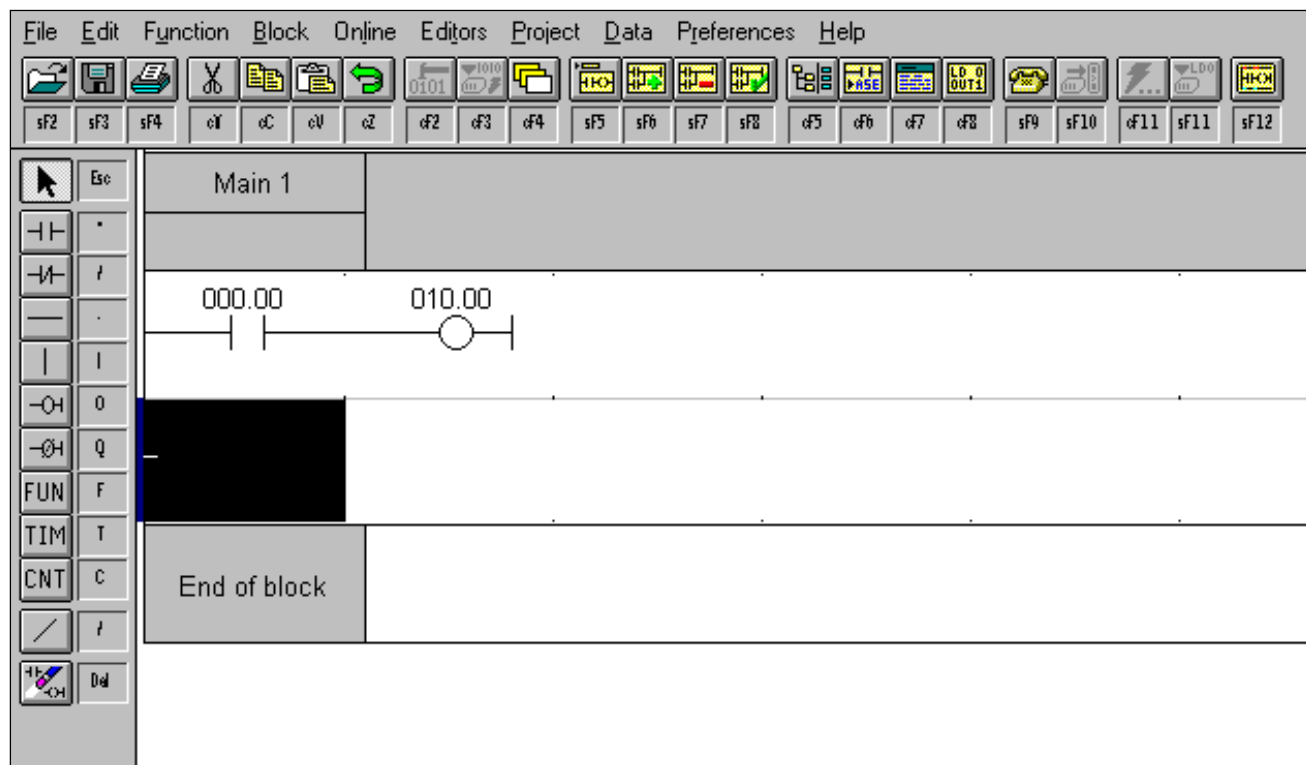


Opcija Blok i njene podopcije



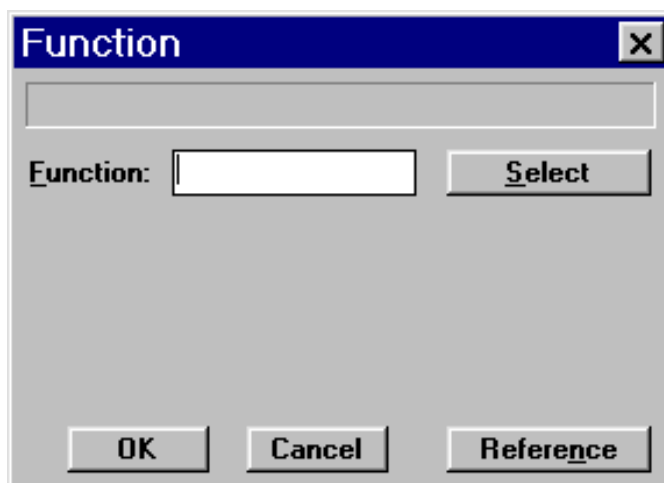
Dijaloški box za dodavanje network-a

Po selektovanju te opcije pojavljuje se mali prozor u kome treba odabrati da li će se novi *network* pojaviti iznad ili ispod postojećeg (*Above* = iznad a *Below* = ispod). U našem slučaju treba odabrati drugu opciju i kliknuti na dugme *OK*. Nakon toga pojavljuje se novi *network* kao na slici.

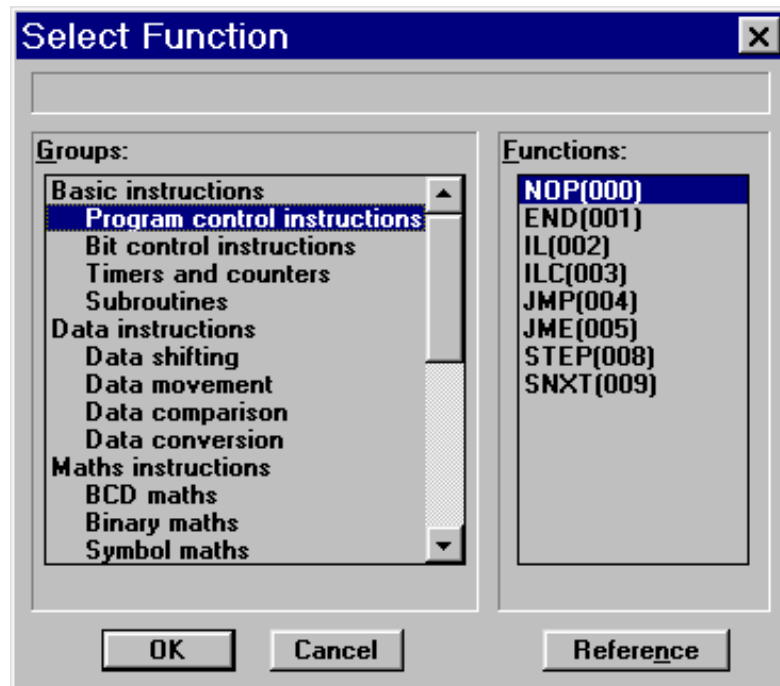


Dodat novi network u programu proba.swp

Poslednji *network* u svakom programu mora da sadrži instrukciju *END*. Kako je ovo jednostavan primer drugi *network* je ujedno i poslednji. Instrukcija *END* nalazi se među funkcijama. Da bi se do nje došlo potrebno je kliknuti na ikonicu obeleženu sa *FUN* nakon čega se dobija prozor kao slici.



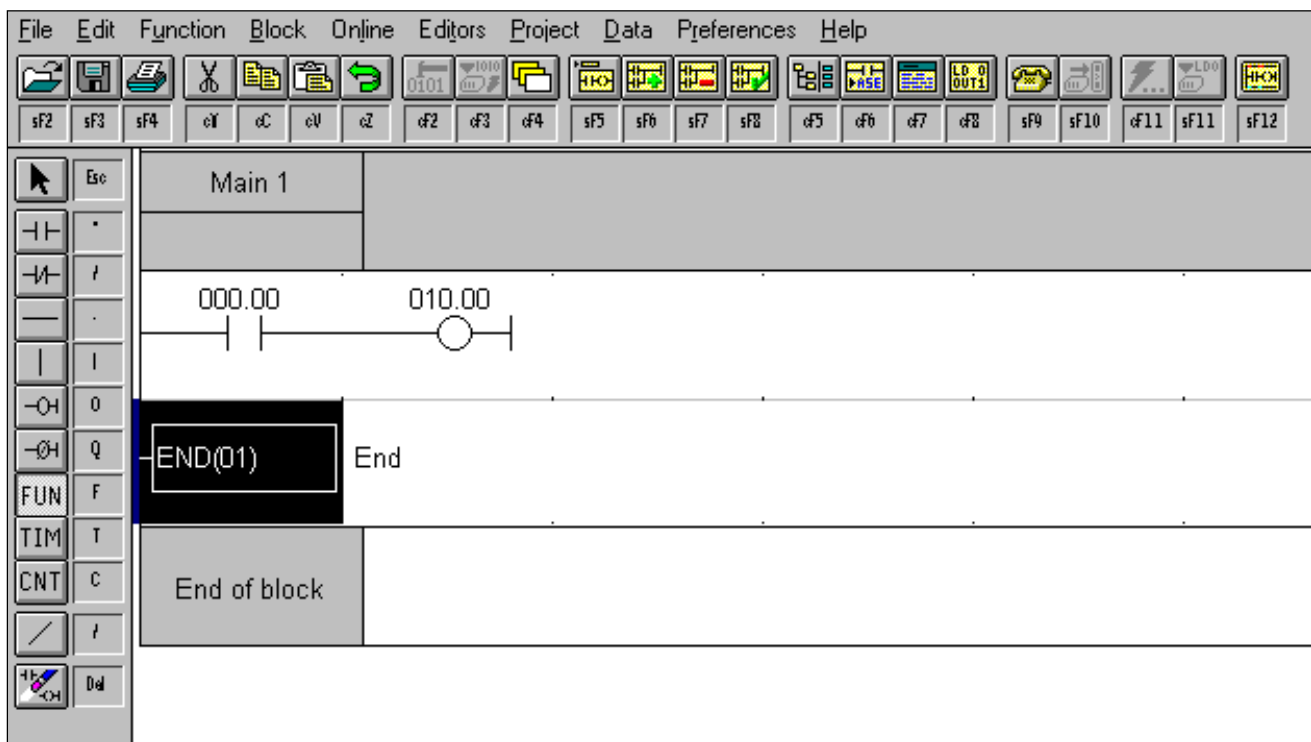
Biranje funkcija klikom na ikonicu FUN



Biranje instrukcije END iz seta instrukcija sortiranih po oblastima

Instrukcija *END* može se dobiti ili upisivanjem *END* u novo dobijenom prozoru ili klikom na dugme *Select* koje daje sve instrukcije PLC kontrolera sortirane po oblastima kao na slici gore.

Unošenjem instrukcije *END* pisanje programa se završava. Gotov program izgleda kao na slici.

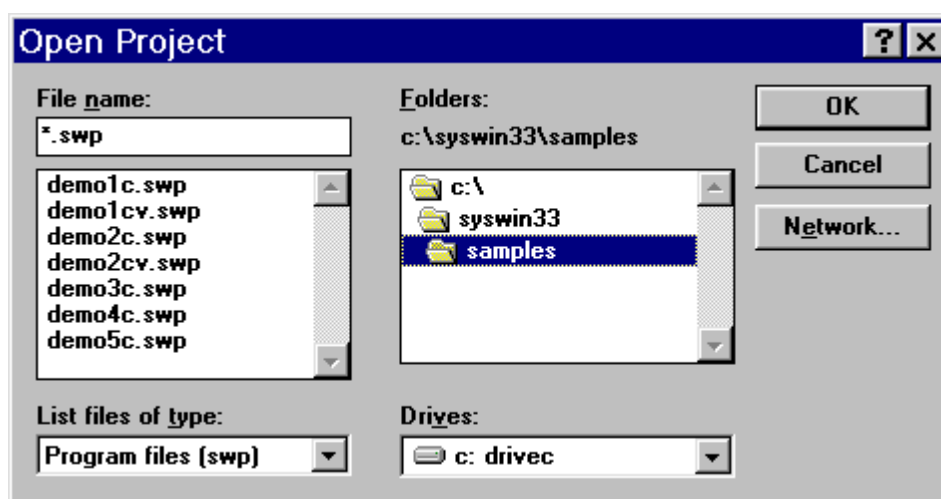


Izgled gotovog programa proba.swp

4.6. Snimanje projekta

Pošto je pisanje programa završeno, potrebno je snimiti projekat. U meniju *File* bira se opcija *Save Project* a u dobijenom dijalogu upisuje se ime datoteke (u našem slučaju *proba.swp*). Posle klika na dugme *OK* projekat je snimljen. Sadržaju SYSWIN datoteka može se pristupiti isključivo iz SYSWIN-a, tip datoteke identifikuje se sledećim ekstenzijama:

- *Project.swp* - SYSWIN program
- *Project.swl* - SYSWIN biblioteka
- *Project.swt* - SYSWIN šablon
- *Project.swb* - SYSWIN back-up datoteka
- *Project.prg* - PMD program.



Snimanje programa *proba.swp*

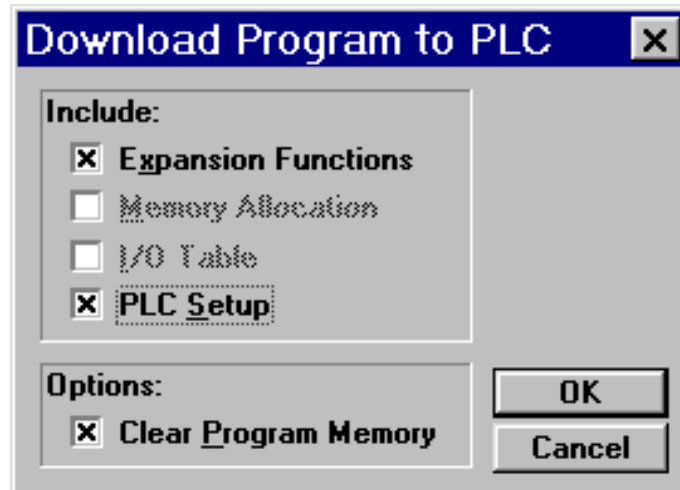
4.7. Prenos programa u PLC kontroler

Prvo treba proveriti da li je *PLC* dobro povezan sa računarom, odnosno proveriti fizičku vezu preko serijskog kabla. Nakon toga treba odabrati u meniju *Project* opciju *Communication* radi podešavanja parametara serijske komunikacije. Pri tome najbitnije je odabrati pravilno serijski port računara na koji je *PLC* povezan. Za CPMIA default podešavanja su: *COM1*, *9600 Baud*, *Unit 00*, protokol *ASCII 7 bit Even Parity 2 stop* i ne treba ih menjati. Provera rada komunikacije obavlja se se klikanjem na dugme *Test PLC*, čime se inicira provera veze sa *PLC* kontrolerom.



Podešavanje parametara komunikacije između *PLC* kontrolera i *PC* računara

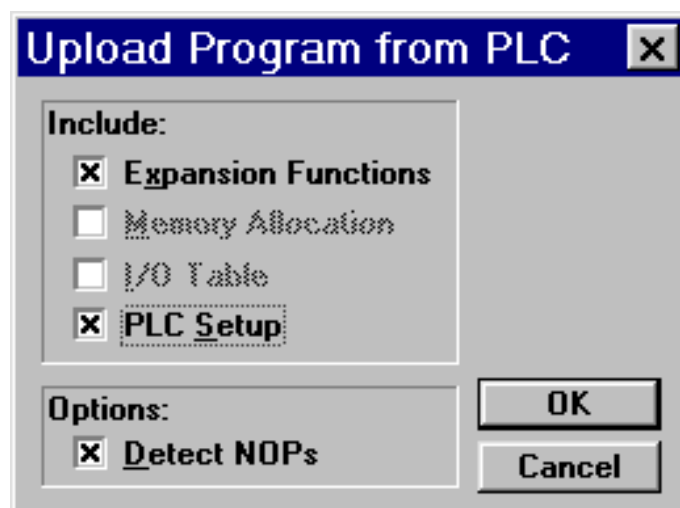
Posle uspostavljanja veze, prenos programa u *PLC* započinje se odabiranjem dijaloškog box-a *download* iz menija *Online*. Pri tome treba odabrati opcije *expansion function* ili *memory allocation*. Poželjno je da se pre programiranja *PLC*-a obriše sadržaj njegove programske memorije tako da se preporučuje i selektovanje i te opciju *Clear Program Memory* kao na slici.



Podešavanje parametara prenosa programa u PLC kontroler

Na kraju, po prenosu programa u *PLC* pojaviće se dijalog koji daje poruku o uspešnosti prenosa.

Napomenimo da se na sličan način ostvaruje i prenos programa iz *PLC* kontrolera u PC. Pri tome se preporučuje selektovanje opcije NOP (no operation).



Podešavanje parametara prenosa programa iz PLC kontrolera

4.8. Provera ispravnosti programa

Opcija *Program check* u meniju *Project* omogućava proveru ispravnosti programa. Dijalog koji se dobija posle aktiviranja komande ima nekoliko opcija koje mogu biti izabrane pre izvršenja provere. Posle odabiranja opcije, potrebno je kliknuti na dugme *Execute*. Na ekranu posle izvršenja provere ispravnosti, prikazuje se izveštaj o proveri i nađenim greškama. Posebno, može se

proveravati jedna po jedna greška, pri čemu pomoć obezbeđuje dugme *Go to Network*, koje nas prebacuje do segmenta u kojem je nađena greška .

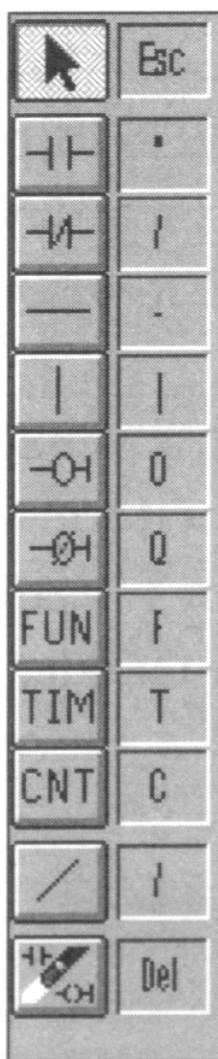
SYSWIN poseduje i klasične editorske mogućnosti, kao što su *Edit/Find* ili *Edit/Replace* komande. Pretraga po programu za zadatim vrednostima ili simbolima je brza i nudi veliki broj opcionih filtera. Možemo pretraživati kompletan program, ili njegove segmente, što se definiše pri pozivu opcije. Takođe, ostavljena je mogućnost definisanja pravca pretraživanja, kao i razne akcije pri pronalaženju traženog elementa kao što su *Replace* sa ili bez potvrde.

Osim ovoga, SYSWIN obezbeđuje razne prednosti u situacijama kada nam treba trajno arhiviranje korisničkog programa. Obezbeđene su i opcije za jednostavno i brzo dokumentovanje projekta na papiru, odnosno štampanje projekta. Projekti mogu biti štampani u velikom broju različitih formata, a takođe štampanje može uključiti pojedine delove projekta.

4.9. Opis ikona u paleti alata

SYSWIN poseduje nekoliko tipova editora od kojih je najviše u upotrebi relejni dijagram ili *ladder editor* koji je prvi editor koji nas čeka nakon startovanja programa SYSWIN.

Osnovni alati sadržani su u paleti *Drawing Tools*. Do pojedinih alata iz palete osim uobičajenog klika mišem na izabranu ikonu, može se doći i preko tastature odgovarajućim prečicama. Kraj svake ikone nalazi se i legenda koja prikazuje odgovarajući taster prečice.



Paleta alata sa prečicama na tastaturi

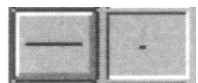
Klikom na ikonu samo se odabira željeni alat a tek klikom na deo *networka* u program smešta se simbol. U daljem tekstu posebno je dato objašnjenje svake od ikona iz palete *Drawing Tools*.



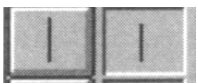
Ikona otvoreni kontakt. Klikom na ovu ikonu (ili tasterom <">) unosimo otvoreni kontakt u *Network*. Element koji unosimo pozicioniramo na označeno mesto (crno polje). Posle ovoga, automatski se aktivira dijalog gde se unose informacije (adresa otvorenog kontakta-broj reči, pozicija bita).



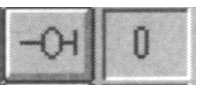
Ikona zatvoreni kontakt. Klikom na ovu ikonu (ili </>pomoću tastature) unosi se zatvoreni kontakt odnosno invertovani uslov u *network*.



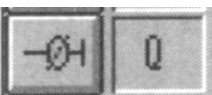
Horizontalna linija. Klikom na ovu ikonu (ili <-> pomoću tastature) produžava se horizontalnih linija s leva na desno. SYSWIN, međutim, zadržava pravo da iscrtane linije optimizuje u pogledu dužine ili ukaže na eventualne greške. Ova opcija koristi se u slučaju kada treba dodati još neki uslov pre instrukcije koja zavisi od njega ili naprosto nešto ne može da stane.



Vertikalna linija. Klikom na ovu ikonu ili pomoću tastera <|>, iscrtava se vertikalna linija, odzgo na dole. Ovaj alat neophodan je kod crtanja paralelnih veza između kontakata.



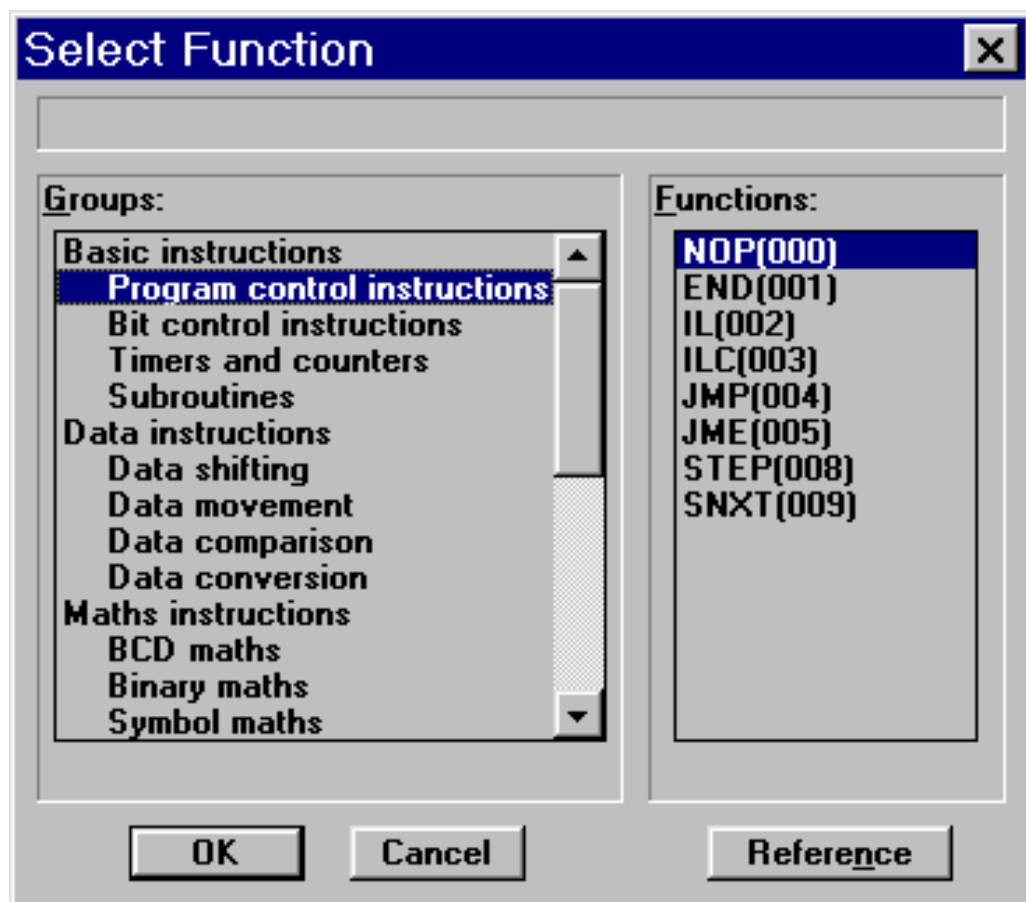
Izlazna instrukcija. Predstavlja instrukciju koja se izvršava ako je izvršena instrukcija (instrukcije) uslova koja joj prethodi. Pomoću ove instrukcije vršimo prosleđivanje rezultata logičkog izraza izlaznim varijablama (bitovima). Do ove instrukcije može se doći i uz pomoć tastature (taster <O>).



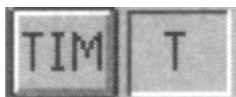
Invertovana izlazna instrukcija (prečica-taster <Q>). Slično kao u predhodnom slučaju, ovom izvršnom instrukcijom prosleđujemo rezultat logičkog izraza izlaznom bitu s tom razlikom da je ovaj bit uključen ako uslov nije uspunjen i obrnuto.



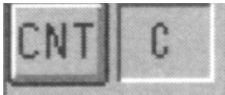
PLC funkcije (prečica-taster <F>). Klikom na ovu ikonu stičemo mogućnost ugradnje kompleksijih PLC instrukcija u program. Prozor koji se pojavljuje nakon klika na ikonu sadrži sve instrukcije sortirane po oblastima. Neke od tih instrukcija su date i posebno kao ikone a nekima je moguće pristupiti samo preko ove funkcije. Jedna od takvih instrukcija je i *END* instrukcija koja se koristi u svakom program. Dijaloški prozor koji se pri tome pojavljuje prikazan je na sledećoj slici. Po dobijanju ovog prozora potrebno je odabrati željenu funkciju i potvrditi je tasterom *OK*.



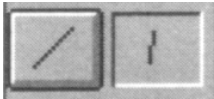
Dijaloški prozor PLC instrukcija dobijen po odabiranju ikone FUN



Klikom na ovu ikonu (ili pritiskom na taster <T>) dobija se mogućnost unošenja tajmera u program. Klikom mišem u osvetljeno polje ekrana pojavljuje se dijaloški box, u koji unosimo potrebne podatke vezane za tajmer (oznaku tajmera i dužinu trajanja u milisekundama). Na ovaj način dobijamo klasičan tajmer odnosno tajmer sa kašnjenjem pri uključenju. Ukoliko je potrebna neka druga verzija tajmera treba koristiti prethodnu ikonu *FUN* i u listi odabrati stavku *Timers and counters* (videti predhodnu sliku).



Ikona brojač. Klikom na ovu ikonu (ili taster <L>) u *PLC* program ugrađuje se klasičan brojač. Predhodno u dijaloškom box-u koji se pojavi unosimo potrebne podatke: oznaku brojača (na primer CNT001) i vrednost brojača. Promena stanja (dekrementiranje, odnosno smanjivanje za 1) brojača obavlja se kada ulazni signal (CP) prelazi sa *OFF* na *ON* stanje.



Pomoću ove ikone vršimo invertovanje ranije unetog kontakta, izlaza ili ulaza. Inverzija se obavlja tako da prvo kliknemo na ovu ikonu a zatim na varijablu čiju inverziju želimo da obavimo.

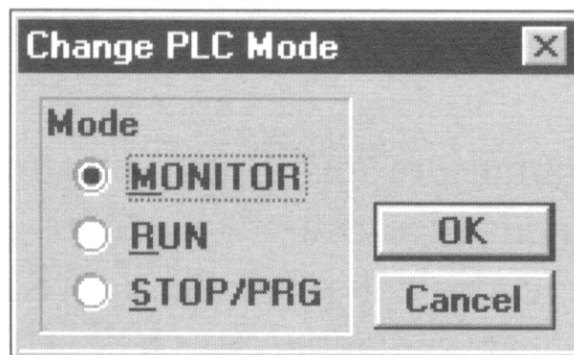


Ikona za brisanje. Klikom na nju i na zatamnjeni deo u *network*-u briše se zatamnjeni deo programa.

Važnu ulogu u SYSWIN programu igra miš. Svaki dvostruki klik po nekoj *PLC* instrukciji rezultuje javljanjem odgovarajućeg editora gde se mogu uneti potrebne izmene. Ovo pravilo dosledno je ugrađeno u SYSWIN, pa tako dvostruki klik na blok ili *network* zaglavlje (*BLOCK HEADER BAR* ili *NETWORK HEADER BAR*) dovodi do istog efekta.

4.10. Režimi rada *PLC* kontrolera

Trenutni režim rada (mod rada) može se saznati na nekoliko načina, na primer iz menija *Online Mode* ili istoimenog dugmeta iz *Toolbar*-a. Ova opcija je dostupna ukoliko je ispravno uspostavljena komunikacija sa *PLC* kontrolerom. Posle njenog aktiviranja pojavljuje se sledeći dijaloški box:



Biranje režima rada *PLC* kontrolera

Ako izaberemo mod koji se razlikuje od trenutnog, promena moda biće trenutna. Da ne bi došlo do slučajne promene moda *PLC* kontrolera, postoji opcija kojom se računar obavezuje da pre svake promene moda postavi pitanje da li je to ono što korisnik zaista želi (ova opcija je uključena po *default*-u). Postoje tri moda *PLC* kontrolera C klase, *MONITOR*, *RUN* i *PROGRAM/STOP* mod.

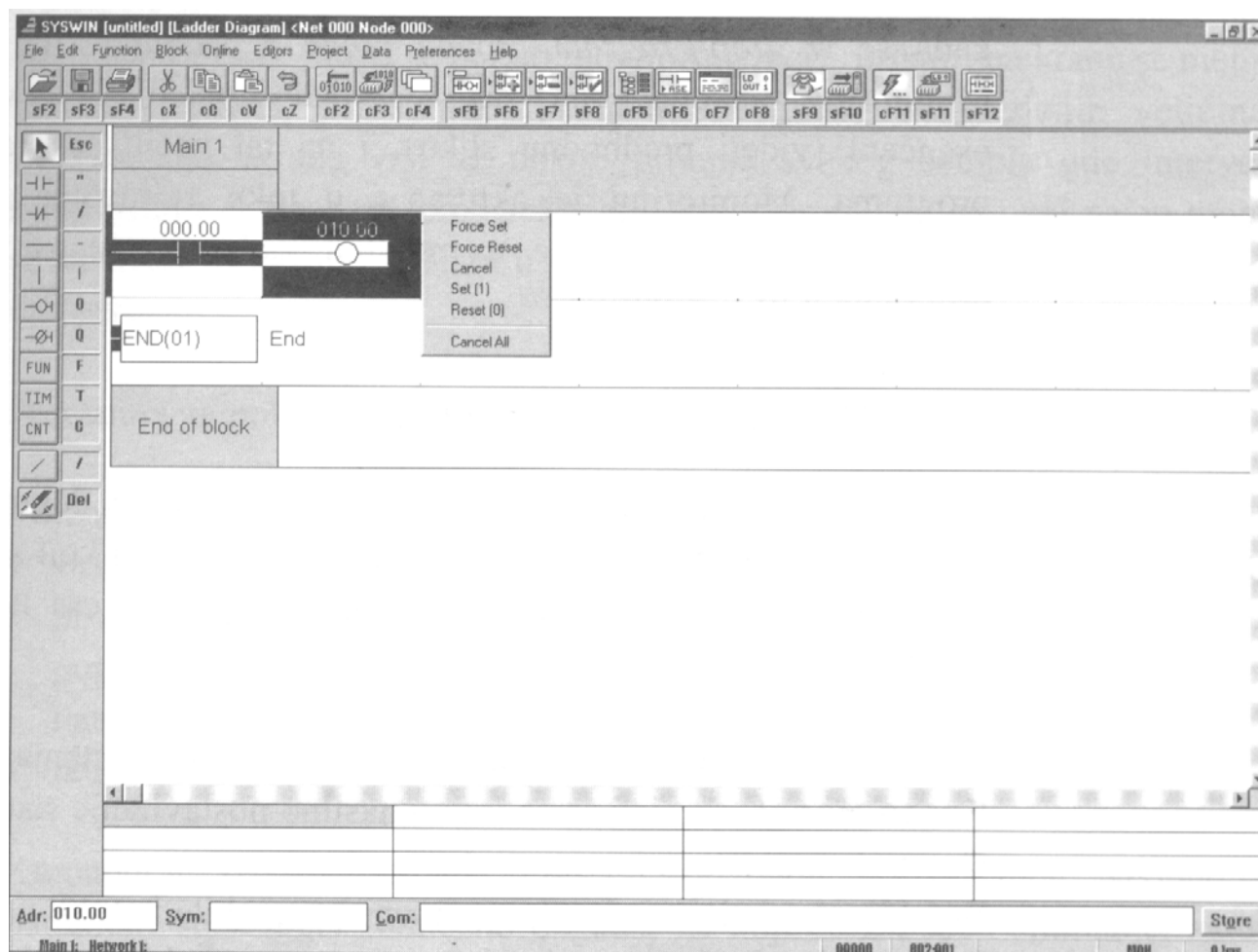
4.11. Režim *RUN*

Ovaj mod *PLC*-a omogućava izvršenje programa kao osnovne operacije. Koristi se za finalno testiranje, nakon što je program detaljno testiran i greške otklonjene. *SYSWIN* ne može menjati sadržaj memorije *PLC* kontrolera u ovom modu, niti je moguća promena programa koji se izvršava. Naravno, kada se završi sa izradom i testiranjem programa, *PLC* započinje svoj novi život u komandnom ormanu, predhodno postavljen u *RUN* mod.

4.12. Režim *MONITOR*

U ovom modu omogućeno je izvršenje programa, ali je pored toga omogućeno i editovanje i prikazivanje (monitoring) tokom rada. Ovo je najčešće korišćen mod u fazi razvoja programa. Kada je izabran ovaj mod, kontroler ima obavezu da osim izvršenja korisničkog programa, računar snabdeva informacijama vezanim za sam program, tačnije za stanje promenljivih u programu. Ako još dodatno potvrdimo u meniju *Online* opciju *Monitoring* dobijamo mogućnost praćenja trenutnih vrednosti promenljivih na samom ekranu, u realnom vremenu, što je ilustrovano i na narednoj slici.

Sve promene ulaza i izlaza vide se na ekranu a stanja promenljivih i programskih lokacija korišćenih u programu se registruju i pamte.



Izgled ekrana u monitorskom režimu rada PLC kontrolera

4.13. Režim *PROGRAM-STOP*

Izbor ovog režima jednostavno zaustavlja rad *PLC* kontrolera ukoliko je *PLC* bio u režimu *RUN* ili *MONITOR*. Koristi se za prenos programa i podataka prema *PLC* kontroleru.

4.14. Izvršenje i prikazivanje programa

Program koji je prenet sa računara na *PLC* počinje da se izvršava u trenutku kada se iz *Stop/Program* moda pređe u *Monitor* ili *Run* mod. Kada *Monitoring* funkcija počne da se izvršava, pojedini delovi ekrana biće osenčeni (videti predhodnu sliku), i na taj način se prati tok izvršenja programa. *Monitoring* je aktivan i u toku editovanja nekog segmenta programa, a zaustavlja se u trenutku kada se prenosi promenjeni deo programa u *PLC* kontroler.

4.15. Uticaj na program tokom prikazivanja

Kada je monitoring u toku, može se koristiti desno dugme na mišu da se pozove meni nekih elemenata leđer dijagrama. Meni koji se pojavljuje kada kliknemo na lokaciju na kojoj je pozicionirana adresa nekog bita, sadrži sledeće elemente:

- Force Set* - koristi se za trajno nasilno postavljanje stanja bita u *ON*.
- Force Reset* - koristi se za trajno nasilno postavljanje stanja bita u *OFF*.
- Cancel* - poništava forsirano stanje.
- Set (1)* - koristi se za kratku promenu stanja bita iz *OFF* u *ON* stanje.
- Reset (0)* - koristi se za kratku promenu stanja bita iz *ON* u *OFF* stanje.
- Cancel All* - poništava forsirana stanja svih bitova.

Uz pomoć ovih opcija može se menjati stanje bitova, sadržaj reči u memoriji kontrolera, kao i poništiti sva ili neka predhodna forsiranja. Pojam forsirati podrazumeva nasilno postavljanje nekog ulaza/izlaza u *ON* ili *OFF* stanje radi provere programa. U trenutku kada *PLC* izađe iz monitoring režima, podaci o forsiranim bitovima i rečima se gube. Moguće je istovremeno forsiranje i proveravanje sadržaja većeg broja veličina, i za to se koristi *Data Set Bar*, koji se obično nalazi na dnu ekrana (videti predhodnu sliku). Editovanje kao i definisanje polja u *Data Set Bar*-u obavljamo dvostrukim klikom posle čega se pojavljuje odgovarajući dijaloški prozor u koji upisujemo adresu bita čije se stanje prati.

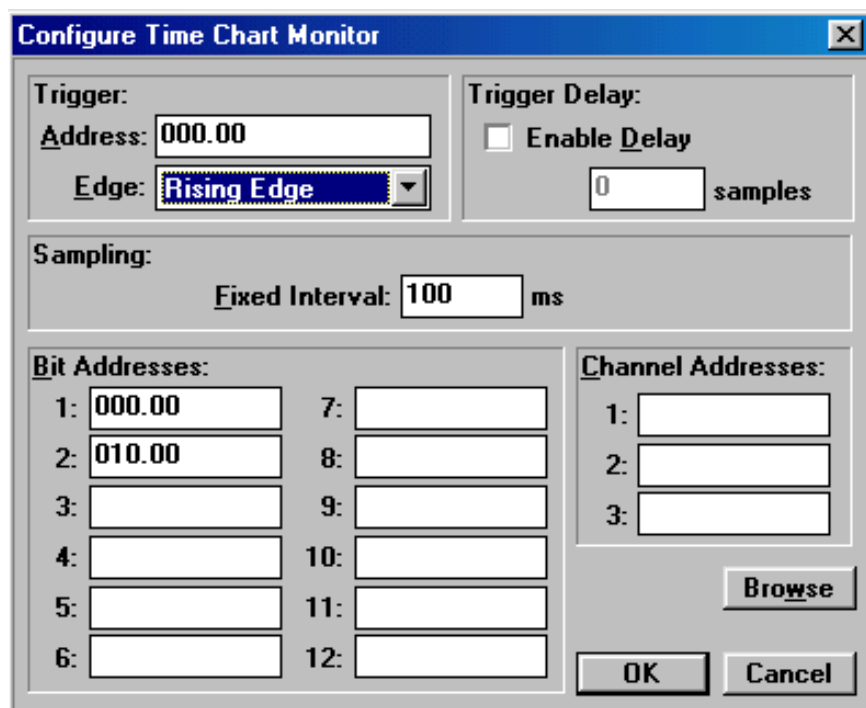
4.16. Grafički prikaz promene veličina u programu

SYSWIN omogućava i grafičko predstavljanje veličina sa vremenom kao apscisom (x-osa). Kada se koristi monitoring mod, prikaz na ekranu se menja tokom vremena, prikazujući promenu vrednosti nadgledanih veličina. Osvežavanje ekrana vrši se nakon prijema svakog odbirka gde intervali odabiranja iznose 0,1 - 65,5 sekundi. Grafici snimljeni na ovaj način mogu se sačuvati za kasniju analizu u obliku datoteke, kao i učitati već prethodno snimljenu datoteku.

Procedura za pokretanje grafičkog prikazivanja je sledeća:

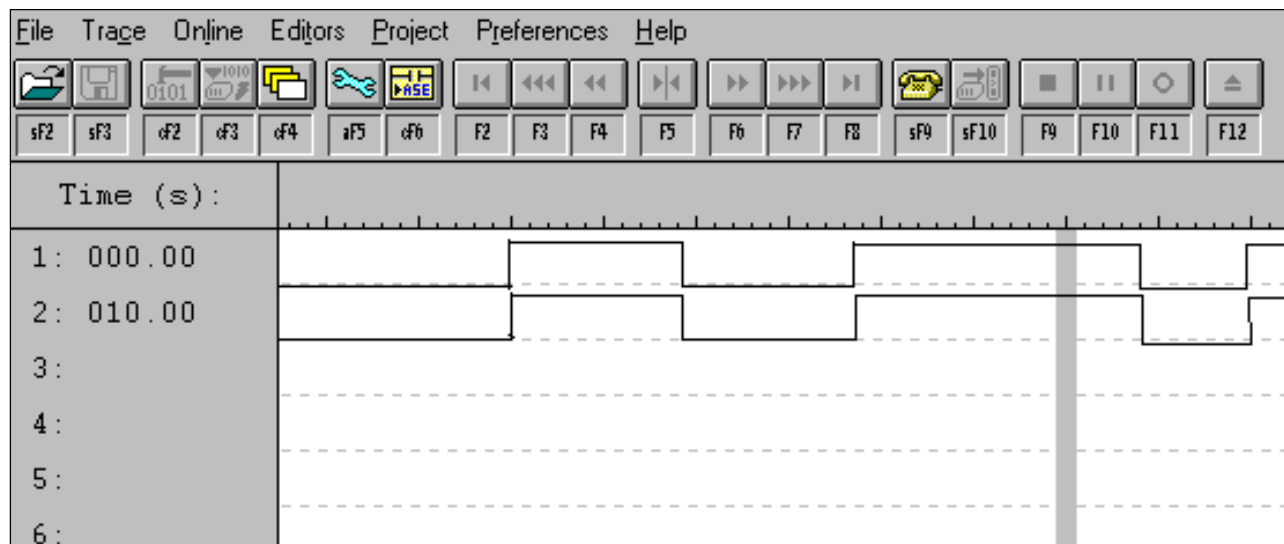
- U meniju *Editors* biramo opciju *Time chart monitoring*.
- U novonastaloj paleti alata biramo dugme *Trace Configure* (prepoznaje se po nacrtanom ključu).
- Popunjavamo polja dijaloškog box-a *Configure Time Chart Monitor* (videti narednu sliku).
- U meniju *Online* biramo *Tracing*.

Komandom *Trace/Configure* podešavamo parametre praćenja. Neophodni parametri su *Trigger*, odnosno događaj na koji će započeti snimanje, zatim perioda odabiranja kao i bitovi i/ili reči čiju vrednost pratimo.



Biranje bit adresa koje će se prikazivati u grafičkom obliku

Nakon zadnje komande, počinje iscrtavanje vremenskog dijagrama veličina koje su prethodno odabrane.



Izgled ekrana sa grafičkim prikazom bit adresa 000.00 i 010.00

Zaustavljanje prikaza vrši se klikom na ikonicu sa crnim kvadratićem a ponovno pokretanje klikom na ikonu sa crvenim krugom. Povratak na editor sa lestvičastim dijagramom vrši se klikom na meni *Editors* i podmeni *Program editor*.

5. PLC kontroler CQM1H-CPU51

Programabilni logički kontroleri familije CQM1 firme *Omron* spadaju u mini kontrolere. Svoju primenu nalaze u upravljanju jednostavnijim objektima upravljanja, ako se za merilo složenosti objekta uzme broj ulazno/izlaznih tačaka koje upravljački sistem je u stanju da obradi. Taj broj je u slučaju kontrolera CQM1 klase 256.

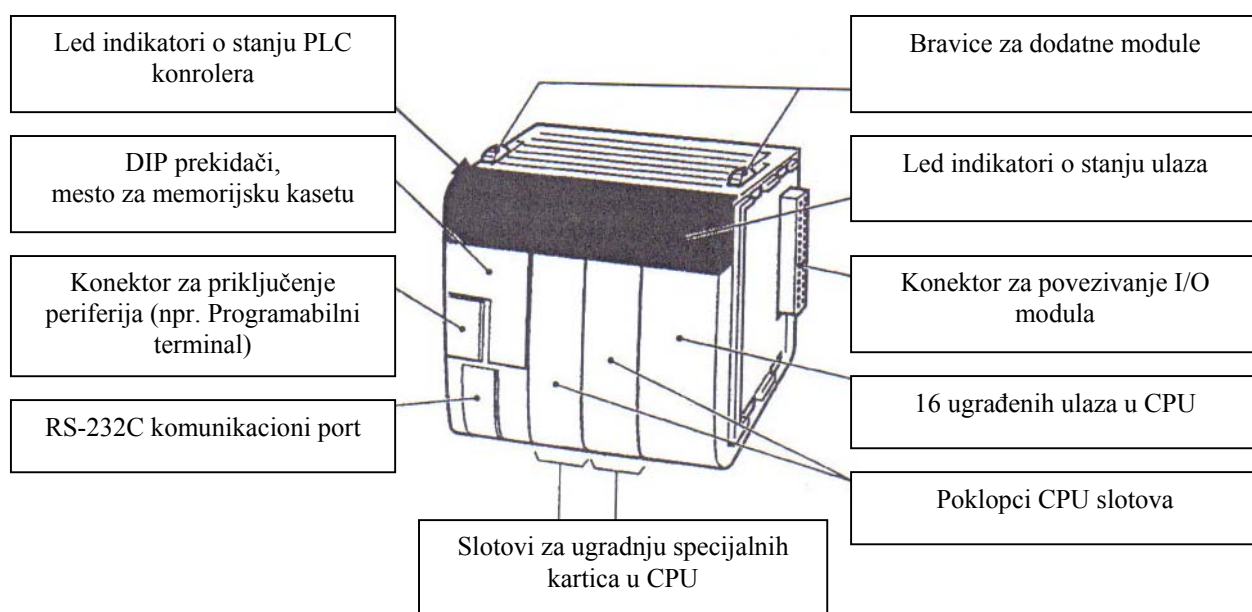
Sistemi upravljanja zasnovani na primeni ovog kontrolera imaju modularnu strukturu, a to znači da ne postoji jedna konkretna celina koja predstavlja programabilni kontroler, već se on dobija kombinovanjem funkcionalno različitih blokova u celinu koja je u stanju da realizuje zahtevani algoritam upravljanja realnim sistemom i specifična je samo za njega. Takođe, jedno od merila uspešnosti nalaženja rešenja za konkretan problem predstavlja i mera u kojoj implementirana konfiguracija prevazilazi zahteve koje pred nju postavlja sistem kojim treba upravljati. Optimalni odnos poštovanja pomenuta dva zahteva moguće je ostvariti upravo korišćenjem prednosti modularne strukture kontrolera CQM1. Sama centralna procesorska jedinica ima veoma male mogućnosti komunikacije sa stvarnim sistemom (16 I/O terminala je ugrađeno), a većina linija komunikacije ostvaruje se dodavanjem specijalnih modula za komunikaciju sa objektom. Ovi specijalni moduli mogu na sebe preuzeti i deo upravljanja, što omogućava decentralizaciju funkcija upravljačkog sistema. Ovo rešenje rezultira smanjenjem broja zahteva koji se postavljaju pred procesorski modul, što opet dovodi do povećanja brzine rada i jednostavnijeg korišćenja.



Spoljni izgled PLC kontrolera CQM1

Postoje specijalni I/O moduli namenjeni obradi različitih tipova signala, kao što su temperaturni modul ili moduli za obradu signala koji na kontroler stižu sa raznih vrsta senzora. Posebna klasa I/O modula predstavljaju jedinice za komunikaciju sa uređajima koji se fizički nalaze udaljeni od mesta gde je montiran kontroler, npr. na različitim delovima sistema upravljanja. Ovi moduli pružaju mogućnost da se veliki broj signala prenese preko redukovanog broja provodnika, što smanjuje troškove instalacije i olakšava održavanje. U ovu grupu specijalnih jedinica spada i master za DeviceNet komunikaciju, standardni network protokol za komunikaciju na nivou uređaja. Pomoću ovog modula moguće je povezati uređaje različitih proizvođača u jedinstven sistem sa *Omron*-ovim kontrolerom kao centralnim upravljačkim modulom.

Modularni koncept pruža još jednu pogodnost, a to je laka nadgradnja postojećeg sistema. U slučaju da zahtevi koje postavlja upravljani sistem u pogledu broja signala za obradu porastu, npr. u slučaju porasta proizvodnje, na postojeći kontroler moguće je montirati dodatne module, i tako povećati kapacitet upravljačkog sistema bez zaustavljanja proizvodnje.



Elementi PLC kontrolera CQM1

Značenje statusa led indikatora o stanju *PLC* kontrolera:

Indikator	Boja	Status	Značenje
RUN	zelena	ON	<i>PLC</i> normalno funkcioniše u MONITOR ili RUN režimu
		OFF	<i>PLC</i> je u PROGRAM režimu i ne izvršava se program, ili izvršavanje programa je prekinuto zbog fatalne greške
ERR/ALM	crvena	ON	Došlo je do pojave fatalne greške
		Blinka	Došlo je do greške koja nije fatalna
		OFF	<i>PLC</i> normalno funkcioniše
INH	žuta	ON	Uključen bit za resetovanje izlaza (SR 252.12)
		OFF	Isključen bit za resetovanje izlaza (SR 252.12)
PRPHL	žuta	ON	Prenos podataka preko perifernog porta
		OFF	Nema prenosa podataka preko perifernog porta
COMM	žuta	ON	Prenos podataka preko RS-232C porta
		OFF	Nema prenosa podataka preko RS-232C porta

5.1. Memorijska mapa PLC kontrolera CQM1H-CPU51

Oblast za podatke		Reči	Bitovi	Funkcija
I/O reči	Ulazna oblast	IR 000 do IR 015	IR 00000 do IR 01515	Bitovi koji se dodeljuju spoljašnjim ulazno/izlaznim priključcima
	Izlazna oblast	IR 100 do IR 115	IR 10000 do IR 11515	
Radna oblast		IR 016 do IR 089	IR 01600 do IR 08915	Radni bitovi koji se mogu slobodno koristiti u programu pošto ne poseduju nikakve specijalne funkcije
		IR 116 do IR 189	IR 11600 do IR 18915	
		IR 216 do IR 219	IR 21600 do IR 21915	
		IR 224 do IR 229	IR 22400 do IR 22915	
Controller Link status oblast		IR 090 do IR 095	IR 09011 do IR 09515	Bitovi koji se koriste u komunikaciji preko Controller Link mreže
		IR 190 do IR 195	IR 19000 do IR 19515	
Makro operand	Ulazna oblast	IR 096 do IR 099	IR 09600 do IR 09915	Bitovi koje koristi instrukcija MCRO(99)
	Izlazna oblast	IR 196 do IR 199	IR 19600 do IR 19915	
Ulazno/izlazna oblast 1. CPU slota		IR 200 do IR 215	IR 20000 do IR 21515	Bitovi rezervisani za specijalne ulazno/izlazne kartice
Trenutna vrednost brzog brojača		IR 230, IR 231		Na ovim adresama se nalazi trenutna vrednost brzog brojača
Ulazno/izlazna oblast 2. CPU slota		IR 232 do IR 243	IR 23200 do IR 24315	Bitovi rezervisani za specijalne ulazno/izlazne kartice
SR oblast		SR 244 do SR 255	SR 24400 do SR 25507	Oblast kontrolnih bitova i flagova
TR oblast			TR 0 – TR 7	Koristi se za privremeno skladištenje podataka prilikom skokova u programu
HR oblast		HR 00 do HR 99	HR 0000 do HR 9915	Oblast za skladištenje podataka koja zadržavaju stanje prilikom nestanka napajanja
AR oblast		AR 00 do AR 27	AR 0000 do AR 2715	Oblast kontrolnih bitova i flagova
LR oblast		LR 00 do LR 63	LR 0000 do LR 6315	Oblast koji se koristi prilikom komunikacije dva PLC kontrolera
TC oblast tajmera/brojača		TIM/CNT 000 do TIM/CNT 511		Oblast tajmera i brojača
DM oblast	Čitanje/ pisanje	DM 0000 do DM 6143		Podacima iz DM oblasti može se pristupiti samo u rečima
	Samo čitanje	DM 6144 do DM 6568		Programom se ne mogu menjati. Sadrži parametre za komunikaciju.
	Zapis grešaka	DM 6569 do DM 6599		Oblast koji se koristi za memorisanje koda i vremena nastale greške
	Samo čitanje	DM 6600 do DM 6655		Koristi se za čuvanje parametara koji kontrolišu rad PLC kontrolera

6. CX-PROGRAMMER programski paket za programiranje PLC kontrolera

6.1. Osnovne osobine CX Programmer-a

CX Programmer programski paket spada u aplikacije koje su namenjene za rad sa *Omron*-ovim PLC kontrolerima i to počev od onih najmanjih, tzv. mikro PLC-ova (CPM1, SRM1), do CS kontrolera visokih performansi. CX Programmer projektantima pruža alat neophodan za programiranje kompleksnih sistema sa distribuiranom inteligencijom korišćenjem već standardnih lersstvičastih dijagrama i mnemoničkih listi. Pored kompletnog programskog okruženja CX Programmer pruža i alate za projektovanje, testiranje i otkrivanje grešaka bilo kog sistema automatizacije. U online režimu rada moguće je izvršiti učitavanje i izčitavanje programa, monitoring, kao i izmenu pojedinih programskih linija i praćenje efekata promene. CX Programmer je kompatibilan sa prethodnim *Omron*-ovim programskim paketima: LSS, SSS, CVSS, SYSMAC-CPT i SYSWIN.

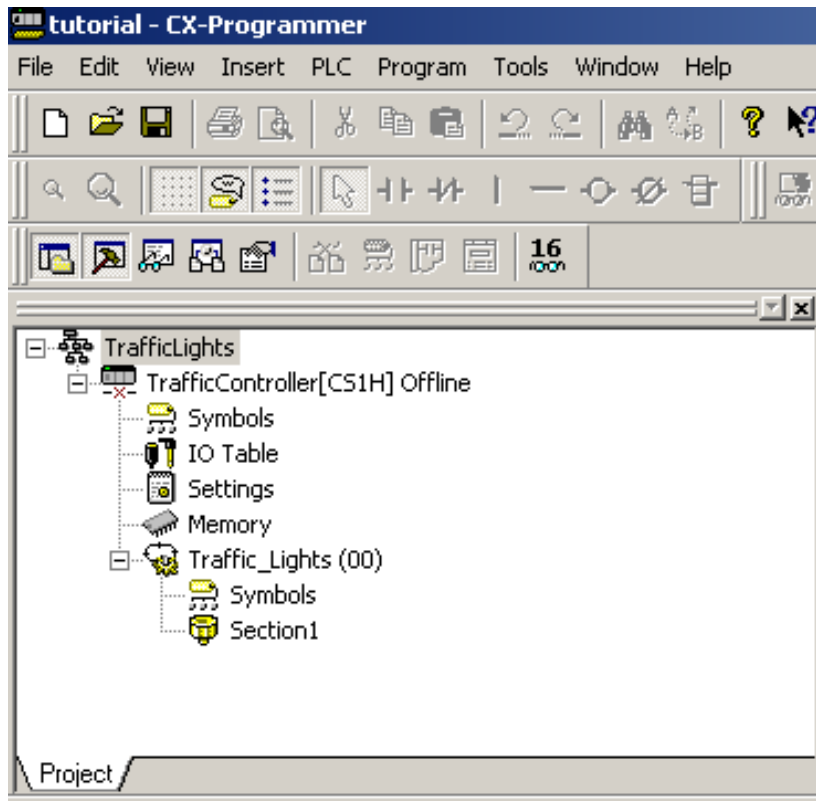
CX Programmer je deo šireg *Omron*-ovog programskog paketa CX Automation Suite koji u sebi integriše i sisteme za kontrolu procesa i sisteme za prenos poslovnih informacija pomoću jedne iste mrežne arhitekture. Za korisnika to znači da se podaci o parametrima procesa, i podaci o poslovnim efektima mogu zajedno prikazati u formi tabela koje se dalje mogu uređivati nekim od poznatih spreadsheet procesora (Excel, Delphi i sl.). Pomoću notebook PC računara možete se priključiti bilo gde u mreži, prikupiti željene informacije ili vizuelizovati proces i menjati parametre. Za ovakve poslove razvijen je modul CX Server Lite, koji intenzivno koristi interfejsse ActiveX i COM.

Kako bi u potpunosti iskoristio mogućnosti naprednih mreža, *Omron* je standardizovao i način na koji pojedini uređaji komuniciraju sa mrežom. To znači da je za svaki uređaj, npr. *Omron PLC*, predviđen i softver kojim se taj uređaj standardizuje i pretvara u mrežnu komponentu. Uključenjem raznih komponenti na ovaj način u mrežu i ona sama postaje transparentna, što znači da se određena informacija može preneti između totalno različitih uređaja spojenih u mrežu. Na taj način se dobija mreža koja se može adaptirati i širiti u kojem god pravcu poželite. CX Automation Suite poseduje idealnu modularnu strukturu: u sebi uključuje drajvere za sve značajne mreže, što obezbeđuje transparentnost u priključenju uređaja bez obzira na njihov tip, kao i tip mreže koja ih povezuje.

Sam softverski paket CX Programmer je zasnovan na Windows korisničkom interfejsu, tako da se intenzivno koriste tehnike rada sa klipbodom (cut, copy, paste), drag&drop, desni taster miša (context senisitive pop-up meni), čak su i mnogi prozori slični standardnim Explorer-ovim prozorima. Dovoljno jak Help, zajedno sa pomenutim tehnikama garantuje programeru udoban rad čak i ako se prvi put sreće sa ovim paketom.

CX Programmer, kao jedan od modula CX Automation Suite paketa, namenjen je projektovanju upravljačke aplikacije sistema zasnovanog na PLC-u kao upravljačkom modulu (obratite pažnju na razliku: ne govori se samo o programiranju PLC-a, već o definisanju kompletnog projekta automatizacije). Pri pokretanju CX Programmer-a moguće je prikazati više prozora i pokrenuti više softverskih alata:

Project Workspace - radni prostor za definisanje projekta. U savremenom konceptu upravljanja centralno mesto ne zauzima samo PLC, odnosno program koji on izvršava, već se definiše kompletna konfiguracija koju čine hardverski i softverski resursi upravljačkog sistema (PLC, CPU, memorija, mreža, softver, protokol komunikacije sa čvorovima mreže). Za opis jedne takve konfiguracije koristi se Project Workspace, čiji je prozor prikazan na sledećoj slici, na kojoj se vodi da je prozor organizovan slično Explorer-u, samo što su umesto fajlova i foldera prikazani resursi PLC kontrolera. Jednom projektu može se pridružiti više PLC-a, i za svaki od njih treba da se definišu njegovi resursi:



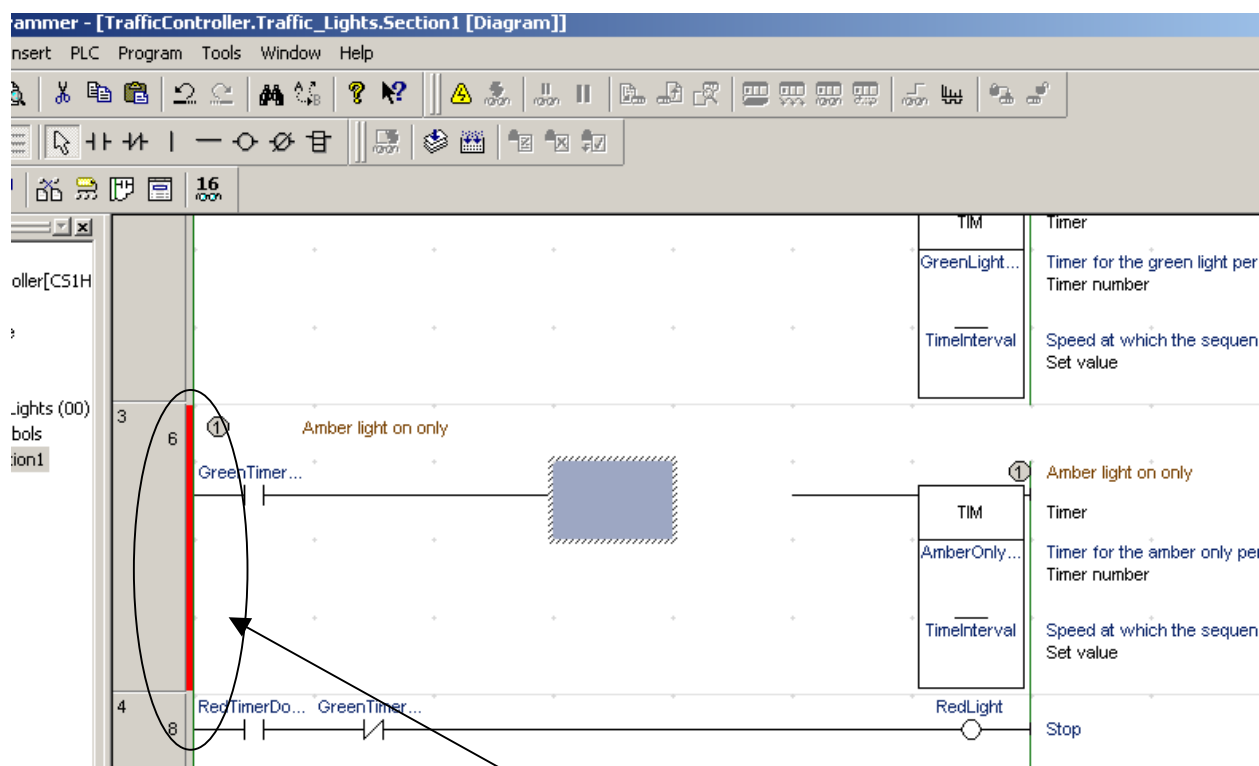
Prozor Project Workspace

1. Lista globalnih i lokalnih simbola (važi samo kod CX Server familije PLC-a); globalne promenljive su one čija je vidljivost u svim programima pridruženim jednom PLC-u, dok lokalne važe samo u programu u kome su deklarirane. Lista simbola se može kreirati i u Excel-u, i tehnikama klipborda iskopirati u tabelu simbola
2. I/O tabela predstavlja svojevrsnu mapu rekovala i modula priključenih na određeni PLC
3. PLC Settings – definisanje parametara koji opisuju PLC
4. Memory - omogućen je prikaz memorijskih oblasti PLC-a i pojedinih lokacija u njima
5. New Program - CX Server i CV familije PLC-a poseduju sposobnost istovremenog izvršavanja više programa (multitasking), bilo da se oni izvršavaju ciklično, ili su inicirani interaptom. C familija dozvoljava samo jedan program po PLC-u

Output Window predstavlja prozor u kome se prikazuju poruke kompajlera ili programa tokom kompajliranja ili izvršavanja.

Watch Window je prozor koji omogućuje nadgledanje statusa ulaza ili izlaza tokom izvršavanja programa; moguće je i “forsiranje” pojedinih bitova pomoću menija ovog prozora kako bi se uvideo efekat na tok programa.

Ladder Design je prozor u kome je moguće kreiranje lestvičastog (ladder) programa, i svojim korisničkim interfejsom i bogatom paletom alatki pojednostavljuje rad u ovom, inače već standardnom, programskom jeziku. Svaka linija ladder-a (rung) je označena posebnim zaglavljem sa brojem linije i opcionalnim komentarom šta ta linija radi; odmah do zaglavlja stoji zelena ili crvena vertikalna traka, zavisno od ispravnosti sintakse trenutne programske linije (odnosno da li su “kontakti”, “releji” i funkcije na njoj korektno definisani).



Primer nekorektno definisane programske linije
(nedostaje ulaz u funkcijski blok)

Prozor Ladder Design

Pri obavljanju standardnih operacija tokom programiranja (unos i definisanje ulazno/izlaznih veličina i funkcija) korisniku se nudi dodatan komfor dijalog-okvirima koji ga interaktivno vode kroz sve faze definisanja jedne programske linije.

Kreiran program se kompajlira, i izvrši se download u programsku memoriju PLC-a. U online modu rada PLC-a moguće je pratiti “tok energije” (Power flow monitoring) kroz sve linije ladder programa.

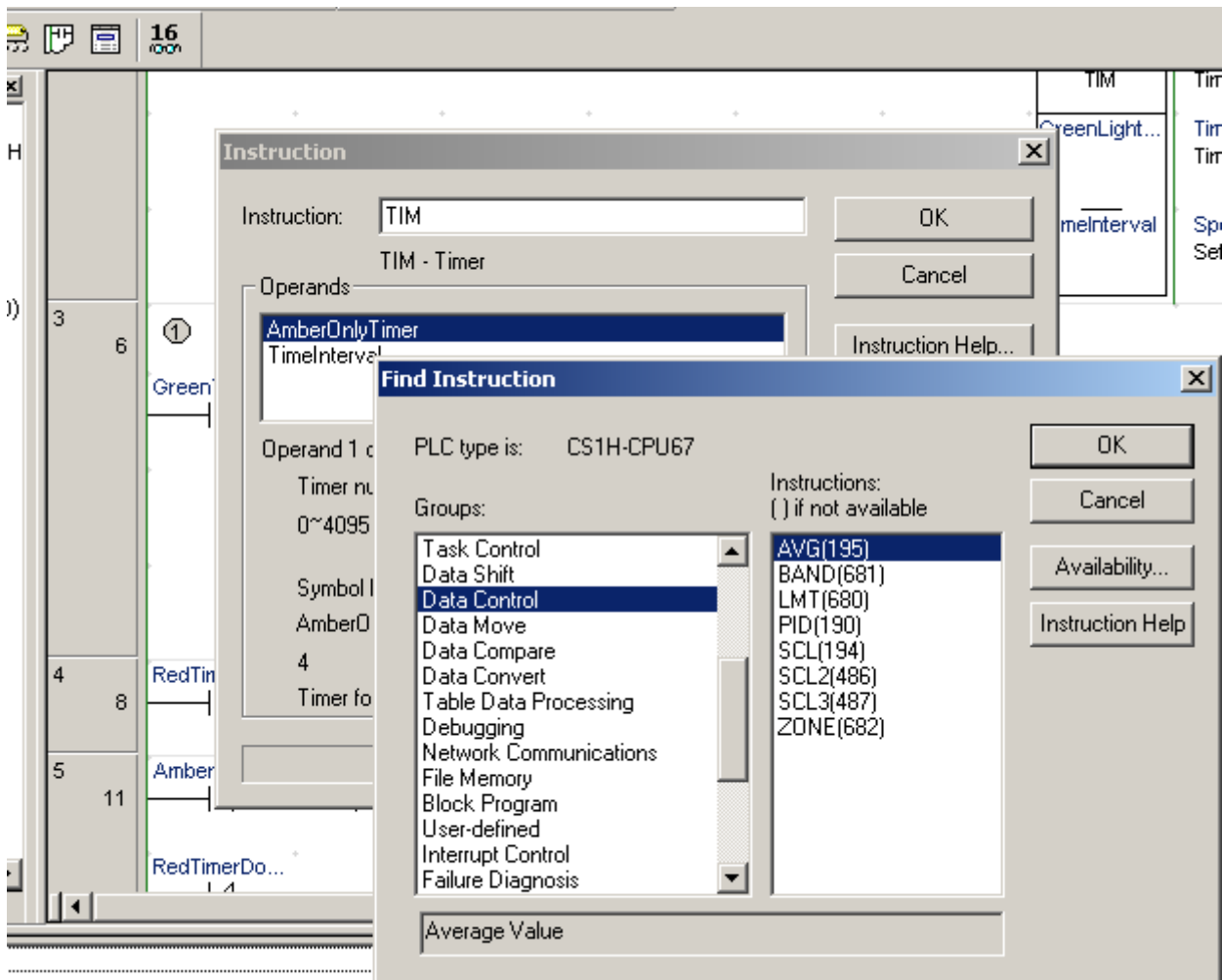
Kontakti koji su aktivni dobijaju npr. zelenu boju (čak i boje mogu da se definišu); moguće je i “forsirati” pojedine kontakte i pratiti kakve su posledice na tok programa.

Tokom online režima rada moguće je izvršiti upload izvršnog koda iz programske memorije PLC-a u radnu memoriju CX Programmer-a, editovanje jedne ili više programskih linija, i ponovo ih smestiti u programsku memoriju PLC-a.

Windows tehnike drag&drop, cut/paste, context-sensitive popup meniji se intenzivno koriste tokom razvoja aplikacije: npr. moguće je neku promenljivu definisanu simboličkom adresom u tabeli simbola iskopirati u ladder dijagram, gde će se ta promenljiva pojaviti u vidu “kontakta” ili “relea”.

Data Monitor Window prikazuje memorijske lokacije PLC-a u tabelarnom formatu. Ukoliko je PLC u online modu, tada sadržaj pojedinih ćelija tabele odražava trenutnu vrednost promenljivih; moguće je da korisnik i promeni sadržaj pojedinih lokacija, i da na taj način prati efekat na izvršavanje programa.

Input/Output tabela PLC-a daje svojevrsnu mapu svih reka i pojedinih I/O modula aktivnog PLC-a. Svaki slot reka ili modul imaju pridruženu adresu, kao i I/O string koji se koristi za opis ulazno/izlaznih karakteristika te jedinice. Sam prozor je urađen u formi Explorer-a, samo što su umesto fajlova i foldera prikazani raspoloživi I/O resursi.



Dijalog okviri CX Programmer-a

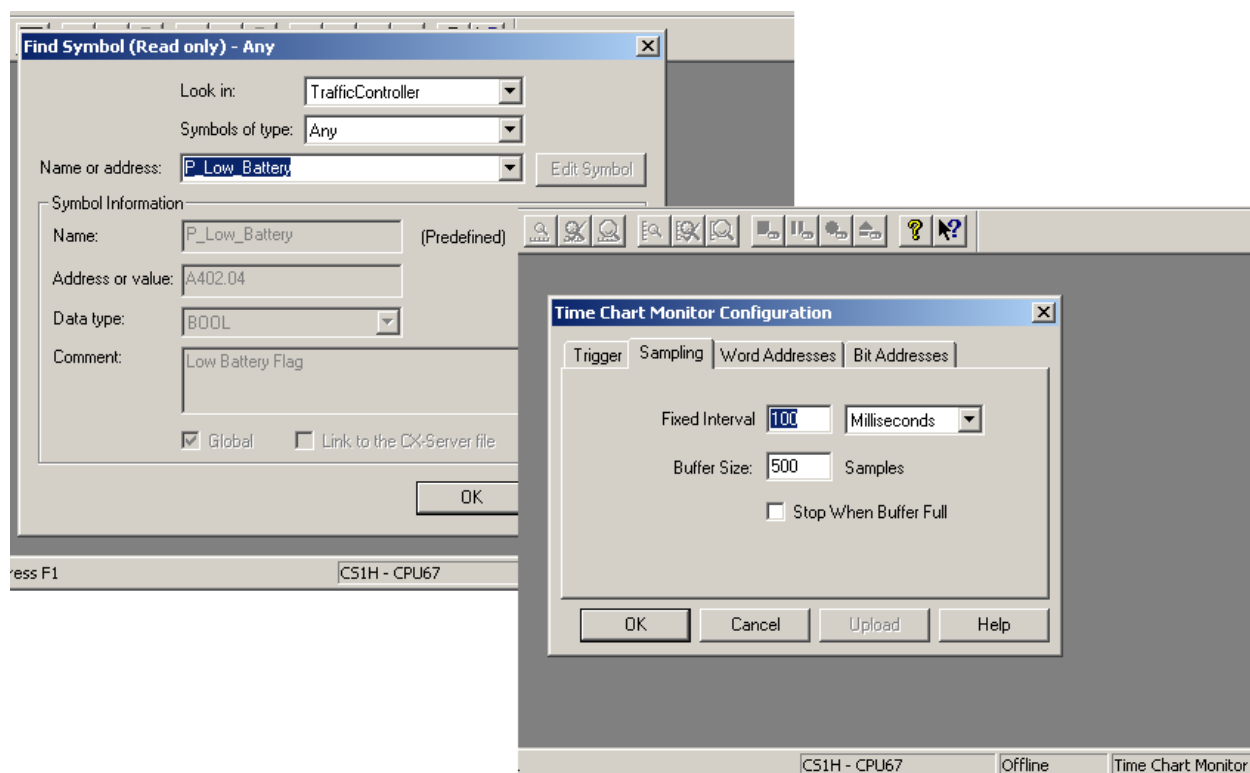
Data Trace Editor se koristi za grafički prikaz podataka iz određenih memorijskih lokacija PLC-a radi određivanja trenda njihove promene tokom izvršavanja programa, podešavanja same aplikacije i slično. Projektant odredi memorijske lokacije koje želi da nadgleda, snimi njihov sadržaj u online režimu rada i kasnije ih prati preko vremenskih dijagrama. Definisanjem triger uslova projektant određuje trenutak od kada počinje snimanje podataka; triger uslov može da bude trenutak kada neka promenljiva u određenoj memorijskoj lokaciji dostigne opadajuća ivica određene bitpromenljive, i slično. Projektant može da definiše i veličinu bafera, te kada broj uzoraka snimljene promenljive dostigne tu vrednost, snimanje se prekida.

6.2. Simboli

CX Programmer omogućuje strukturno programiranje. Naime, program može da se podeli na blokove i sekcije, koje zatim mogu se da koriste u različitim projektima kao podprogrami ili funkcije. Takođe, i sam program sastavljen od blokova je pregledniji i jednostavniji za čitanje i održavanje: moguće je npr. da se izvrši upload pojedinog bloka ili sekcije, editovanje u online režimu rada i ponovni download u memoriju PLC-a.

Memorijskim lokacijama PLC-a koje se koriste kao operandi u PLC programu mogu se pridružiti simbolička imena i komentari radi čitljivosti programa. Adresa memorijske lokacije,

zajedno sa imenom i komentatom jednim imenom se naziva Simbol. Tabela simbola je dinamička lista definicija simbola.



Data Trace Editor

Pored toga, ta lista daje podatke o načinu korištenja te adrese; na primer, ako je ta adresa pripala I/O mapi memorijskog prostora PLC-a, tada uz takvu adresu stoji i hardverski opis; Input ili Output. Takođe, moguće je specificirati i format podataka koji se nalaze na toj adresi, što može biti korisna informacija kako bi se tokom pisanja ili izvršavanja programa proveravalo da li se podatak konzistentno koristi.

Postoje sledeći tipovi podataka:

Tip	Veličina	Format	Opis
BOOL	1 bit	binarni	adresa logičke binarne vrednosti (bit)
CHANNEL	1 ili više reči	bilo koji	adresa bilo koje vrednosti veće od 1 bit
DINT	2 reči	binarni	adresa double integer-a
INT	1 reč	binarni	adresa integer-a
LINT	4 reči	binarni	adresa long integer-a
NUMBER	-	decimalni	slovna vrednost - ne adresa; NUMBER simboli se koriste kod operanada sa prefiksom: #, &, + ili -. Koriste se kod BCD ili binarnih instrukcija; kod BCD instrukcija tretiraju se kao hekso-vrednosti.
REAL	2 reči	IEEE	adresa floating-point broja
UDINT	2 reči	binarni	adresa unsigned double integer-a
UDINT BCD	2 reči	BCD	adresa unsigned double BCD
slično i UINT, ULINT			

Name	Type	Address / Value	Rack Location	Usage	Comment
AmberLight	BOOL	10.01	Main Rack : ...	Out	Prepare to go/stop
AmberLightTimer	NUMBER	2			Timer for the amber light period
AmberOnlyTimer	NUMBER	4			Timer for the amber only period
AmberOnlyTimerDone	BOOL	T0004		Work	
AmberTimerDone	BOOL	T0002		Work	
GreenLight	BOOL	10.02	Main Rack : ...	Out	Go
GreenLightTimer	NUMBER	3			Timer for the green light period
GreenTimerDone	BOOL	T0003		Work	
RedLight	BOOL	10.00	Main Rack : ...	Out	Stop
RedLightTimer	NUMBER	1			Timer for the red light period
RedTimerDone	BOOL	T0001		Work	
TimeInterval	NUMBER	48			Speed at which the sequence works ...

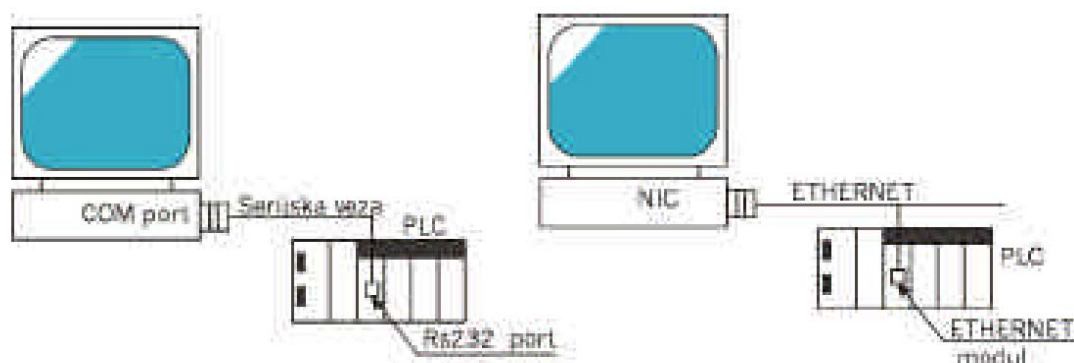
Dinamička lista definicija simbola

6.3. Komunikacija

Postoji više načina za povezivanje PLC-a i kompjutera; CX programer podržava sledeće:

Sysmac-Way je jednostavan način povezivanja PC-a na PLC u cilju nadzora ili dijagnosticanja; veza može biti preko RS232 interfejsa (1:1 veza) ili preko RS422 interfejsa (1:N veza).

Ethernet je danas postao industrijski standard, jer omogućava povezivanje procesnih uređaja i office aplikacija (procesori baza podataka, SCADA) i to na nivou čitavih postrojenja. U Network Setting dijalogu CX Programmer-a projektant može zadati IP adresu, broj mreže, broj Fins čvora host kompjutera i PLC-a.



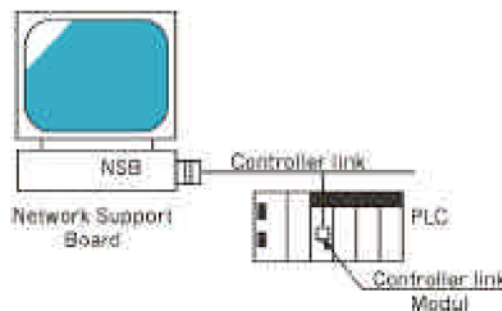
Sysmac-Way i Ethernet komunikacija

ToolBus je veza slična Sysmac-Way, sa razlikom što se podaci šalju u binarnom formatu. Veza sa PLCom ide preko perifernog porta.

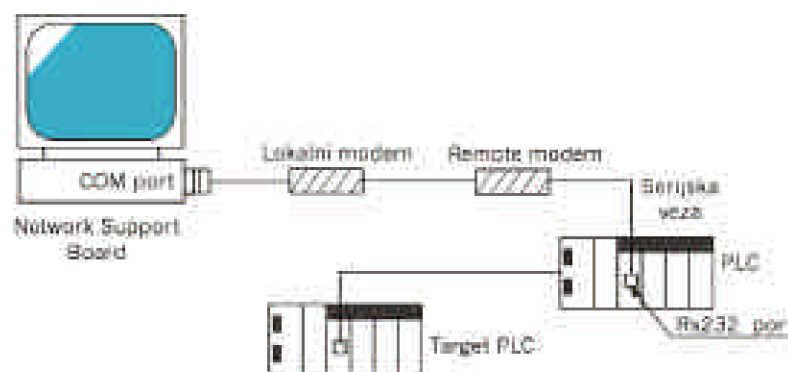
Controller Link Network je mreža koja može da fleksibilno šalje i prima velike pakete podataka između Omron PLC-a i PC računara. Controller Link podržava linkove podataka koji obezbeđuju data sharing, kao i servisne poruke pri iniciranju slanje i primanja paketa poruka. Da bi se PC povezo sa PLC-om preko ove mreže, potrebna je posebna kartica. U Network Setting dijalogu treba navesti broj mreže i broj čvora na kome se nalazi PLC.

Most (Sysmac Way/Modem) CS i CV familije OMronovih PLC-a omogućuju Bridging/Gateway funkcije, gde host spojen serijskom vezom sa bridge/gateway PLC-om ima mogućnost da pristupa i drugim čvorovima mreže.

Podešavanje načina pristupa mreži, izbor mreže i parametara komunikacije obavlja se jednostavno preko Network Setting dijaloga, koji je sastavni deo alata Network Configuration Tool.



Controller Link Network komunikacija



Modem komunikacija

Network Configuration Tool obezbeđuje alate za online postavljanje parametara, kao i offline uređivanje routing tabela. Pri tom su podržani različiti tipovi mreža.

Pomoću ovog interfejsa mogu se obavljati sledeće funkcije:

1. priključenje uređaja (npr. PLC-a) na mrežu
2. analiza uređaja priključenih na mrežu
3. dijagnosticiranje stanja mreže i echo-back test mreže
4. kreiranje i transfer linkova podataka i/ili routing tabela ka PLC-u
5. upravljanje i nadgledanje linkova podataka

Data Link Editor omogućuje postavljanje linkova podataka, koji automatizuju prenos i pristup mrežnim informacijama. Kreiranje i uređivanje linkova je olakšano postojanjem data link tabela, i moćnim Help Wizard-om koji svojim "savetima" vodi projektanta kroz proces postavljanja linkova.

6.4. Hardverski i softverski preduslovi

Neophodni uslovi za pokretanje instalacije CX Programmer-a su Microsoft Windows okruženje na standardnom IBM ili kompatibilnom Pentium računaru.

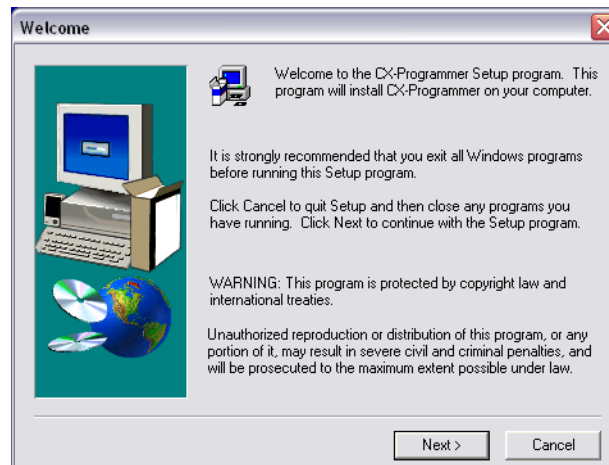
6.5. Instalacija programskog paketa

Instalacija CX Programera počinje duplim klikom na ikonicu "Setup" u direktorijumu CX/Disk1 na pratećem disku. Odmah nakon pokretanja treba odabrati jezik (ostaviti ponuđeni izbor za Engleski jezik kao na slici ispod).



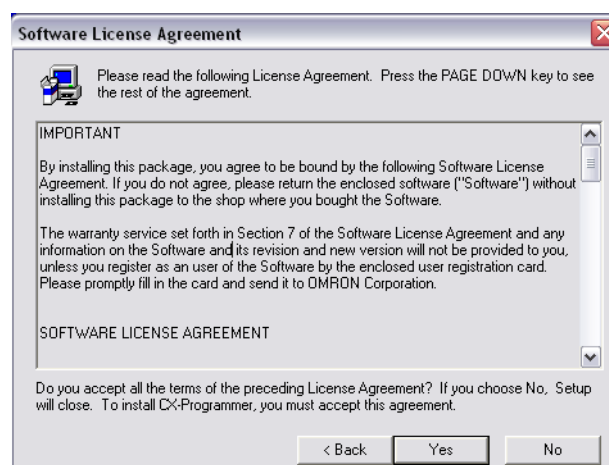
Izbor jezika

Naredni prozor obaveštava da instalacija kreće i da je to zaštićen program korporacije "Omron".



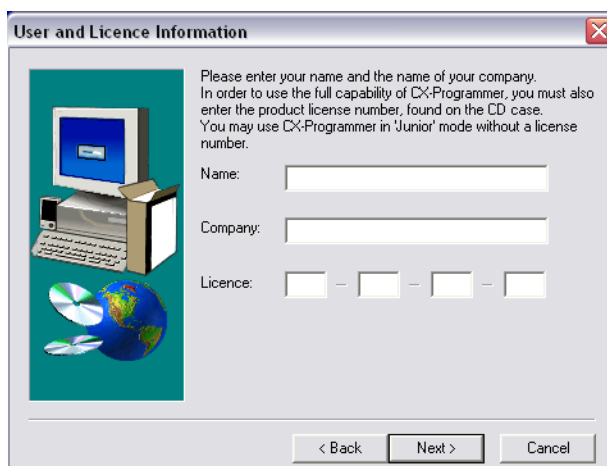
Upozorenje o zaštiti softvera

Klikom na "Next" prihvataju se uslovi pod kojima se program distribuira i instalacija ide dalje.



Upozorenje o zaštiti softvera

Kako se koristi *PLC* koji nije obuhvaćen licencom dovoljno je kliknuti na "*Next*" da se instalacija nastavi dalje.



Serijski kod je potreban samo za veće modele PLC kontrolera

Na ovo pitanje treba odgovoriti sa "*Yes*" jer se radi o *PLC* kontroleru za koji licenca nije potrebna.

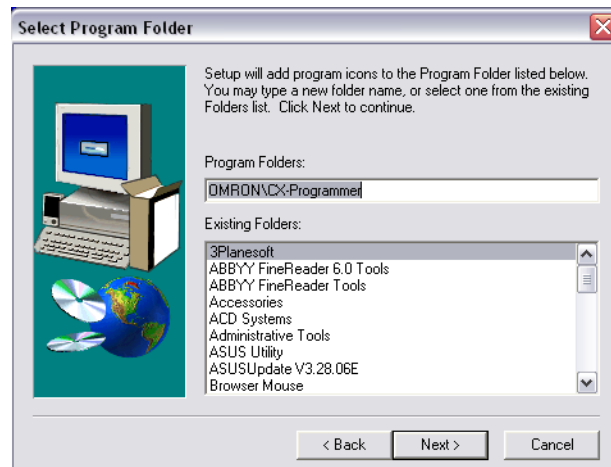


Sledeće pitanje se odnosi na direktorijum gde će se smestiti CX Programmer (poželjno je ne menjati direktorijum), klikom na "*Next*" instalacija se nastavlja.



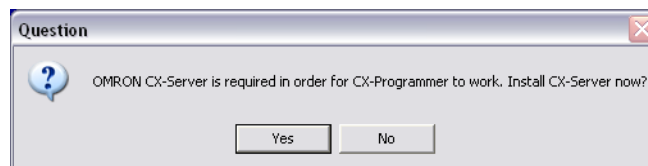
Izbor direktorijuma u koji se smešta CX Programmer

Sledeći korak je kreiranje programske grupe. Klikom na "Next" programska grupa je kreirana i instalacija teče dalje.



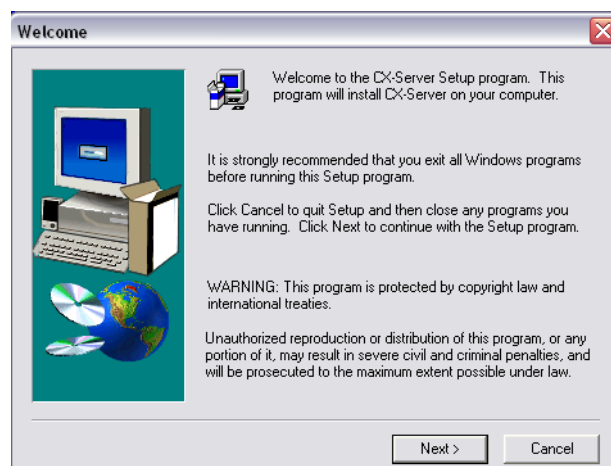
Kreiranje programske grupe

Naredni prozor je dosta važan jer obaveštava o instalaciji ključnog dela CX programskog paketa. Radi se o CX serveru koji drži sve pojedinačne aplikacije u vezi. Klikom na "Yes" instaliraće se CX Server. Ukoliko se izabere "No" CX programer neće moći da se pokrene.



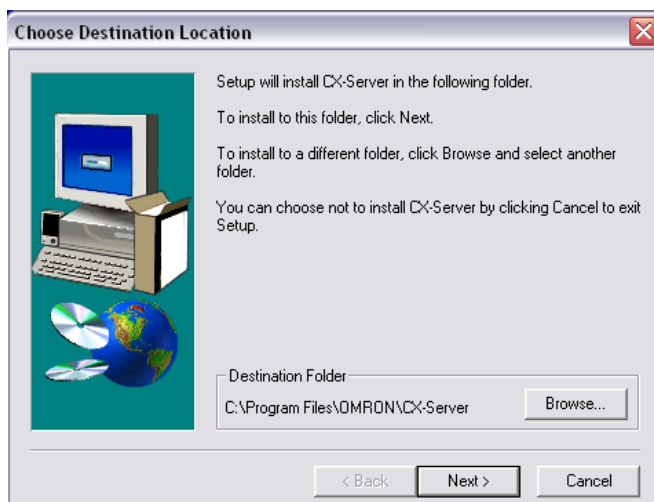
Obavezno instalirati CX Server

Sledećih nekoliko prozora su slični kao na početku. Klikom na "Next" ide se dalje.



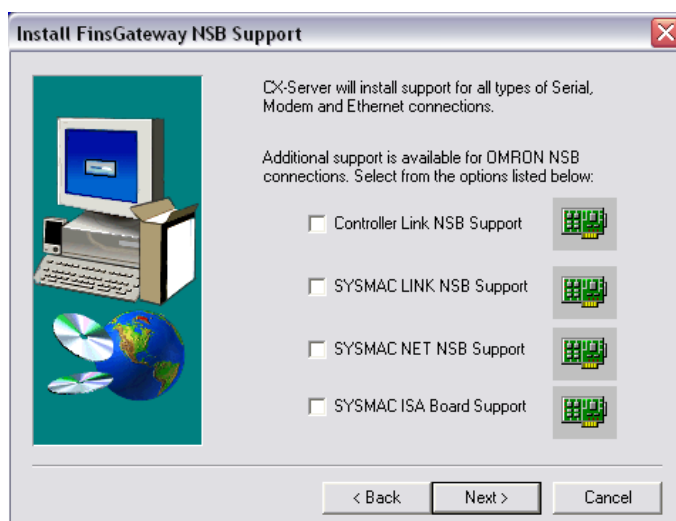
Upozorenje o zaštiti softvera

Naredni korak je biranje direktorijuma za CX Server. Poželjno je ostaviti predloženi direktorijum.



Izbor direktorijuma u koji se smešta CX Server

Sledeći korak se odnosi na razne komunikacije koje CX Server podržava. Jednostavnijim PLC kontrolerima ne treba instalirati nijednu od ponuđenih komunikacija.

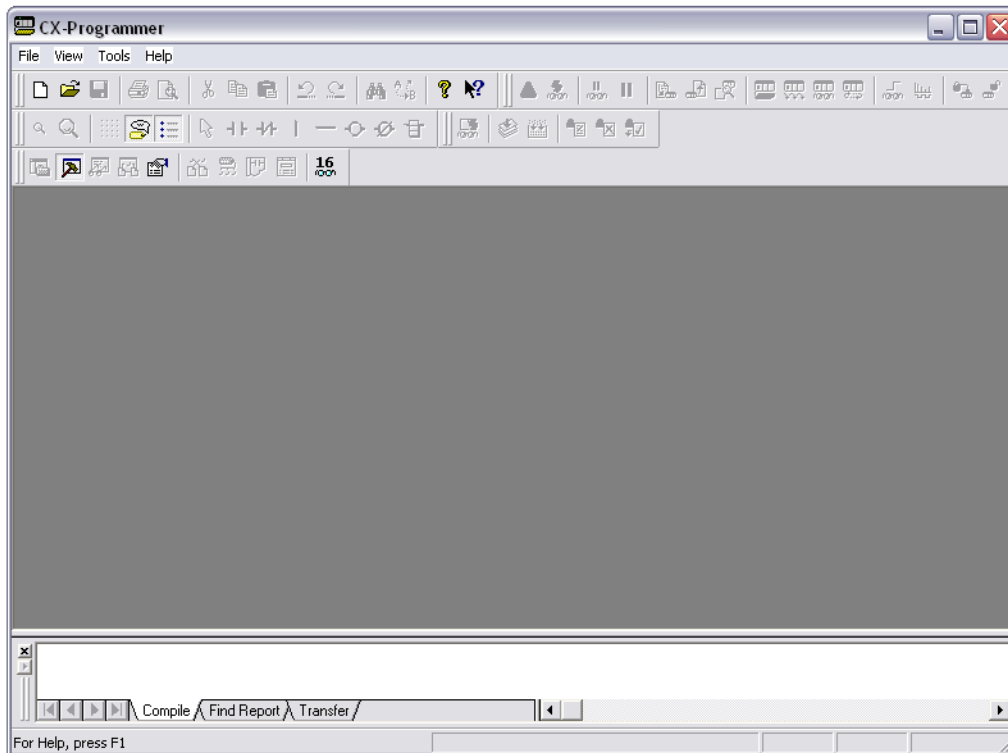


Napredne opcije CX Servera

Klikom na "Next" CX Server se instalira a samim tim se završava i instalacija CX Programmer-a.

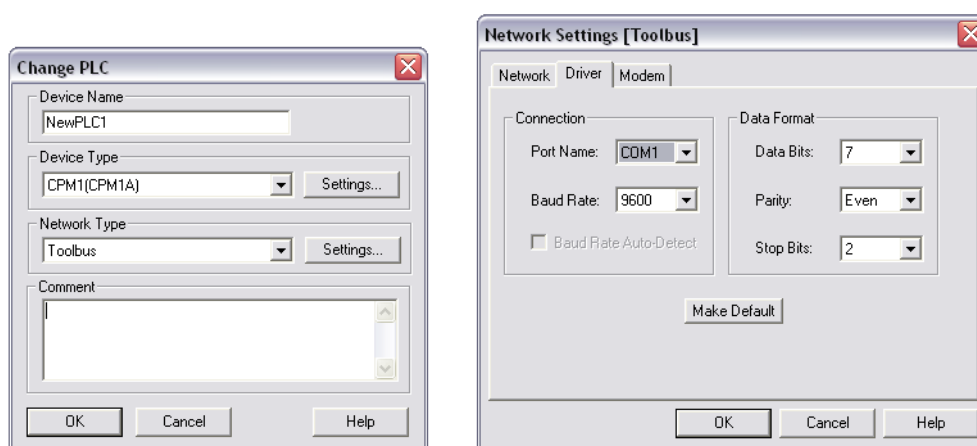
6.6. Pokretanje CX Programera

CX programer se poziva kao i sve druge Windows aplikacije iz "Start" menija. Po pokretanju pojavljuje se prozor kao na slici ispod.



Prvi prozor nakon pokretanja CX Programmera

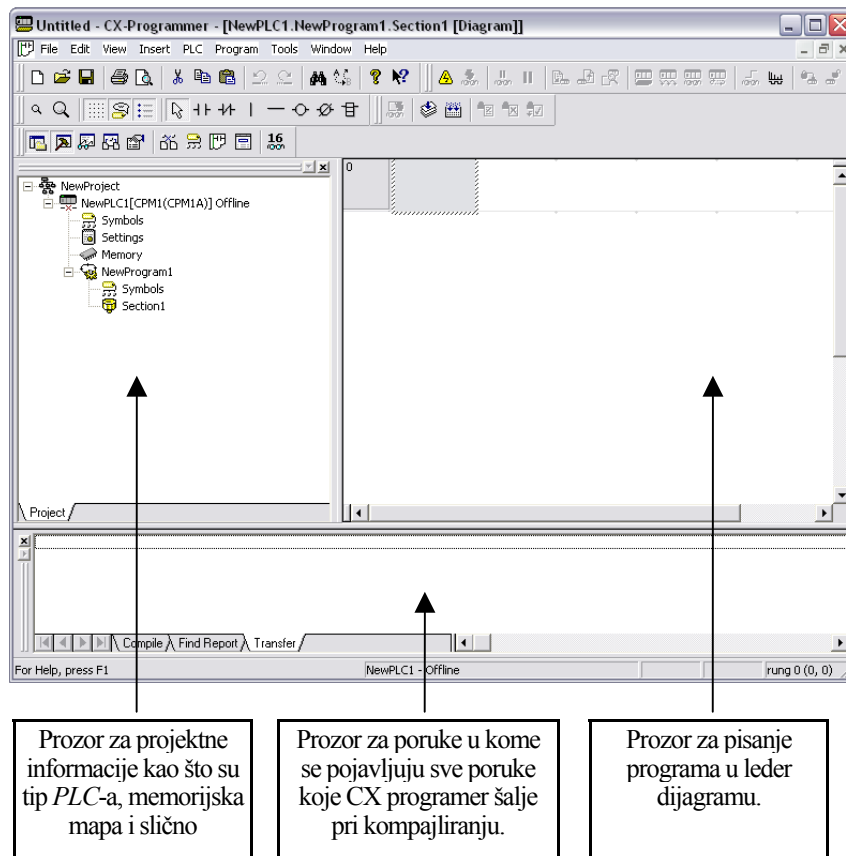
Klikom na prvu ikonicu sa leve strane (ili na *File - New*) pojavljuje se novi prozor u kome treba izabrati tip *PLC* kontrolera. U slučaju da se umesto COM1 koristi COM2 mora se u *Network Type* ⇒ *Settings* promeniti komunikacioni port za CX programer. U suprotnom CX programer će prijaviti da *PLC* nije konektovan.



Podešavanje PLC kontrolera i komunikacionog porta

6.7. Pisanje prvog programa

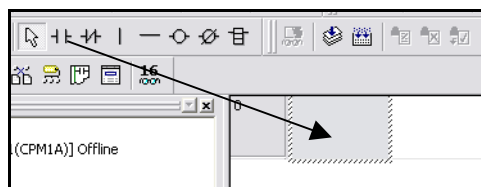
Selektovanjem *PLC*-a i klikom na *OK* pojavljuje se prozor spreman za pisanje programa kao na slici ispod.



Program koji se u ovom poglavlju piše je elementaran i namenjen upoznavanju CX programera. Namena programa je da detektuje pritisak na taster i aktivira relej na izlazu *PLC* kontrolera. Dokle god je taster pritisnut i rele je aktivan, otpuštanjem tastera i rele se otvara. (rad releja i tastera se može lako pratiti preko LED dioda na kućištu *PLC* kontrolera). Ovaj jednostavan program se može napisati u svega nekoliko koraka.

1. korak

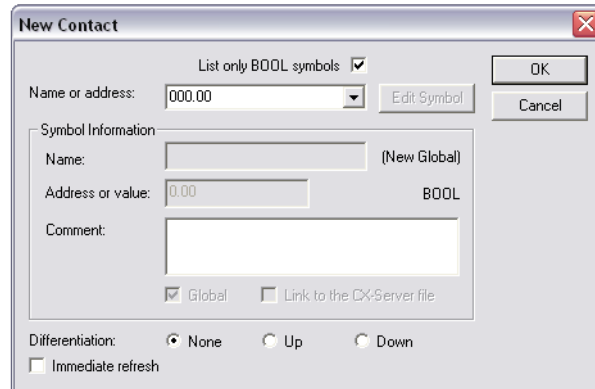
Pisanje programa počinje klikom na sedmu ikonicu u drugom redu koja simbolizuje uslov sa kojim skoro sve instrukcije počinju (prepoznaje se po dve vertikalne crte). Nakon klika kursor poprima oblik kao na crtežu sa ikonice. Klikom na zatamljeni pravougaonik u programskom delu ekrana pojaviće se prozor kao na narednoj slici.



2. korak

Ono što treba uraditi sledeće je upisivanje adrese kontakta. Veoma je važno da se adrese u programiranju pomoću CX-a koriste na regularan način. Adrese mogu imati dva dela, prvi koji se odnosi na

adresu reči i drugi koji se odnosi na adresu bita u toj reči (oba broja moraju biti odvojena tačkom). Na primer, ako se koristi adresa 200, CX će to protumačiti kao 2.00 i bice pozvan nulti bit čija je adresa reči 2. Ako se želi pristupiti reči 200, odnosno njenom nultom bitu, mora se koristiti poziv 20000 ili bolje 200.00. U ovom primeru za adresu ulaza (taster) zadaje se adresa 000.00 koja predstavlja nulti bit reči 000 iz memorijske oblasti IR.

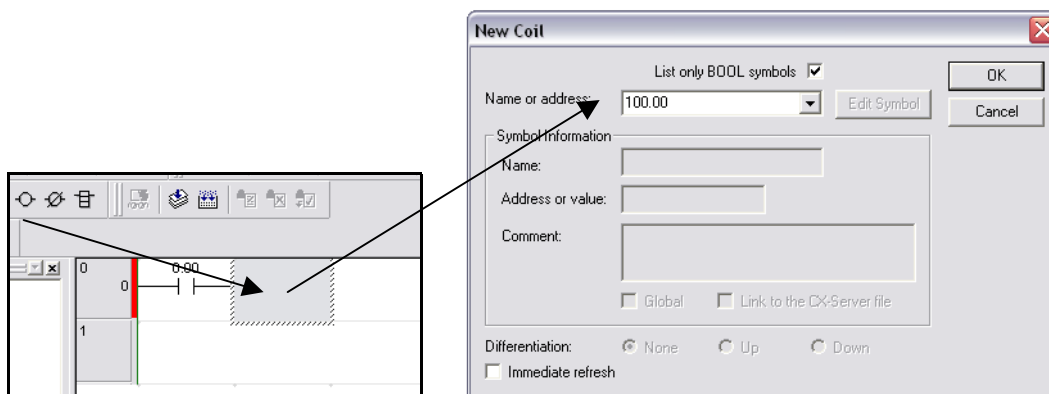


Dijalog box za upis adrese bita na koga se odnosi kontakt

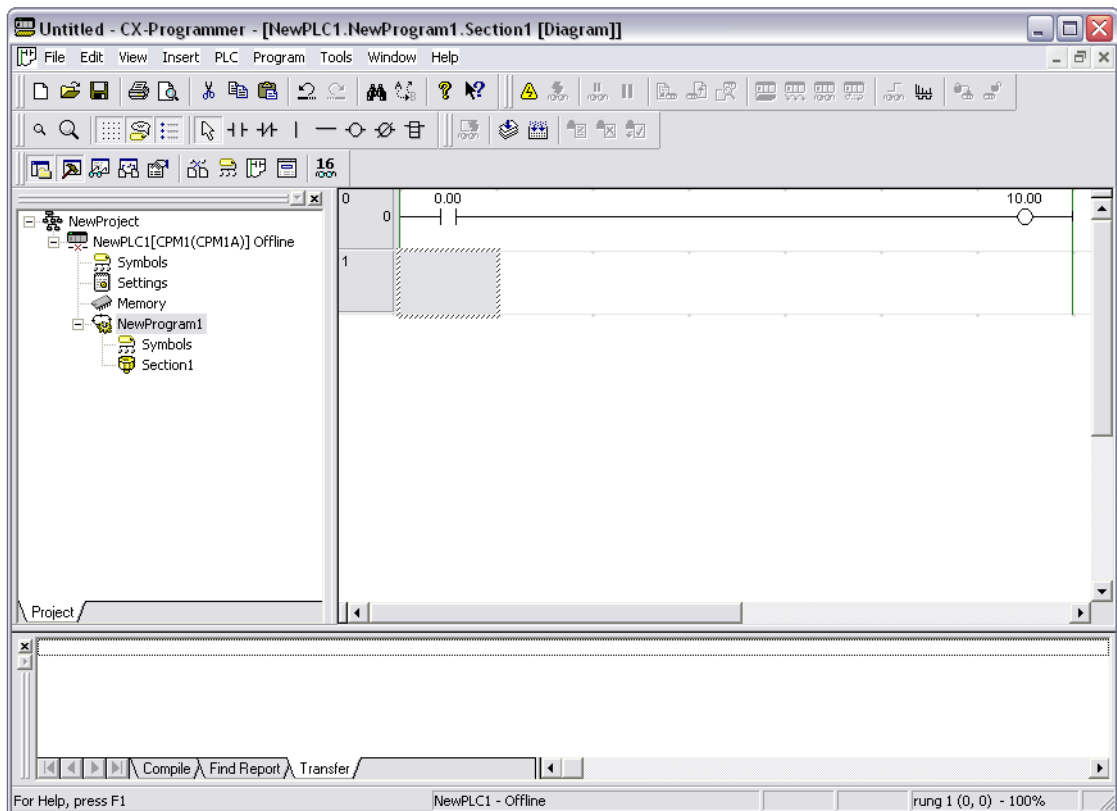
Jednostavnije rečeno to je klemna na ulazu označena kao 00 ulaz. Spajanjem tastera na nju i jednu od COMM klemna ostvaruje se potrebna veza PLC kontrolera i tastera. Nakon upisa 000.00 pritiskom na taster OK pojavljuje se prvi segment programa. Iznad simbola sa dve vertikalne crte pojavljuje se adresa bita na koga se simbol odnosi a crni pravougaonik se pomera za jedno polje desno.

3. korak

Prve instrukcije do *bus bar*-a se nazivaju uslovi jer se njihovim ispunjenjem aktiviraju instrukcije koje se nalaze desno od uslovnih instrukcija. Kad je unesen uslov potrebno je uneti i odgovarajuću instrukciju koja se aktivira ispunjenjem uslova. U ovom primeru to je rele koga kontrolise bit 00 u reči 010 memorijske oblasti IR. Izlazne instrukcije su predstavljene krugom (ili •- krugom sa crtom ako je reč o normalno zatvorenom kontaktu). Klikom na ikonicu sa krugom bira se opcija izlaza sa normalno otvorenim kontaktima. Novim klikom na crni pravougaonik pojavljuje se prozor u koga treba upisati adresu izlaznog bita 100.00. Izlazni deo IR oblasti se nalazi na adresi IR100.



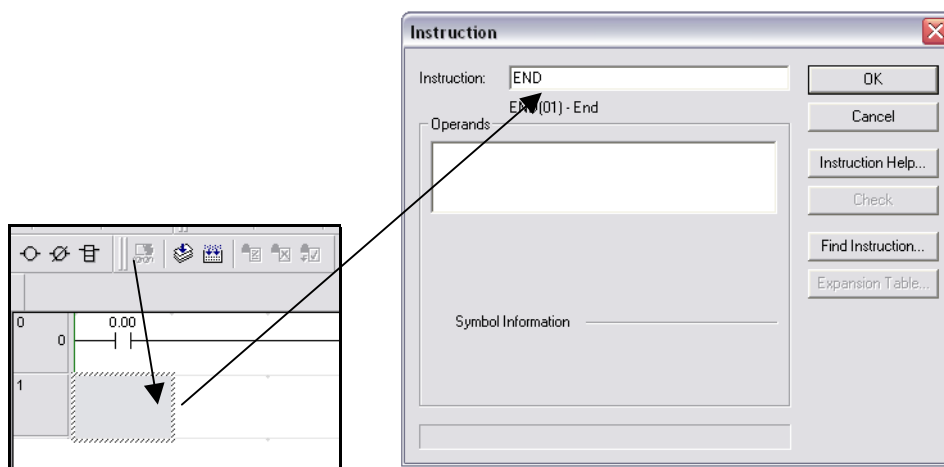
Do sada urađen program izgleda kao na narednoj slici. Treba primetiti da se sa leve strane nalazi uslov (taster) a sa desne akcija (bit u reči kojim se kontrolise rele 0)



Postavljena prva linija programa

4. korak

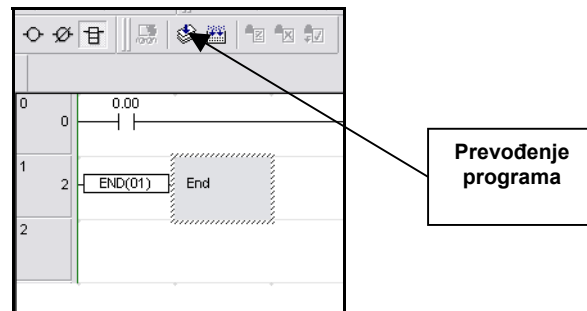
Ono što još fali da bi se program završio je END funkcija na kraju programa. Klikom na trinaestu ikonicu u drugom redu i zatim na zatamljeni deo dobija se prozor u koga treba ukucati ime funkcije koja se traži, u ovom slučaju to je END fnkcija. Postavljanjem END funkcije pisanje programa je završeno. Poslednji *network* u svakom programu mora da sadrži instrukciju END. Kako je ovo jednostavan primer drugi *network* je ujedno i poslednji.



Biranje funkcije

5. korak

Prevođenje programa se vrši klikom na petnaestu ikonicu u drugom redu. Ako je sve u redu u prozoru za poruke će se pojaviti poruka da je prevođenje uspelo i da nema grešaka.

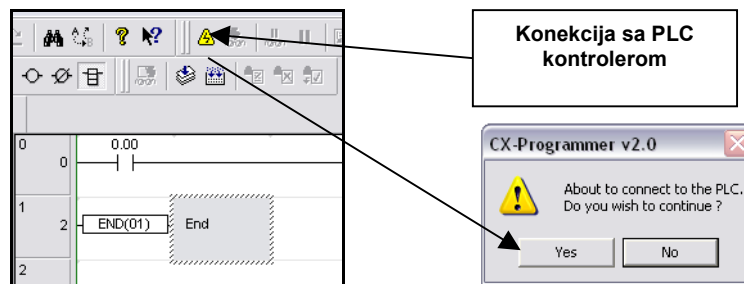


6. korak

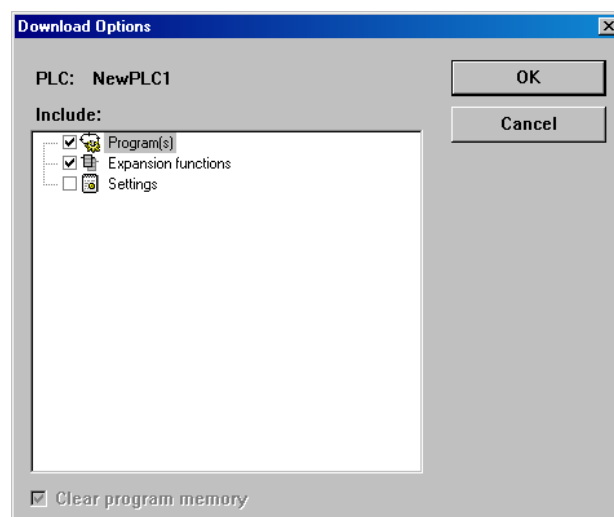
Pošto je pisanje programa završeno, potrebno je snimiti projekat. Iz *File* menija se bira opcija *File - Save* a u dobijenom dijalogu se upisuje ime datoteke. Posle klika na dugme *Save* projekat je snimljen.

7. korak

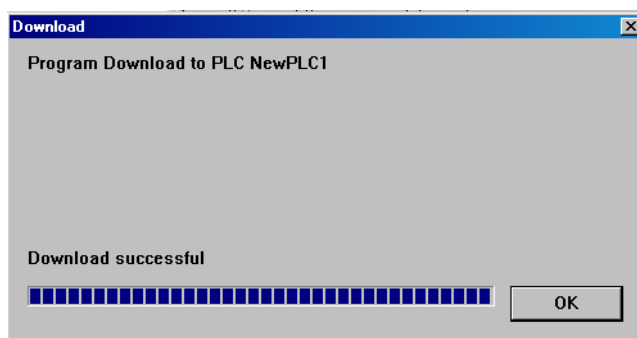
Prebacivanje programa u *PLC* kontroler. Prvo treba proveriti da li je *PLC* dobro povezan sa računarom što se svodi na proveru fizičke veze preko serijskog kabla. Klikom na ikonicu kao na slici ispod vrši se konekcija sa *PLC* kontrolerom. Klikom na ikonicu radna površina postaje siva.



Kada je veza uspostavljena, prenos programa u *PLC* započinje se klikom na devetnaestu ikonicu u prvom redu. Nakon toga pojaviće se prozor sa ponuđenim memorijskim oblastima koje treba uprogramirati u *PLC* (nije potrebno ništa selektovati na tom prozoru što već nije selektovano). Klikom na *OK* dobija se još jedan prozor na kome treba kliknuti na "Yes" da bi prenos programa počeo.



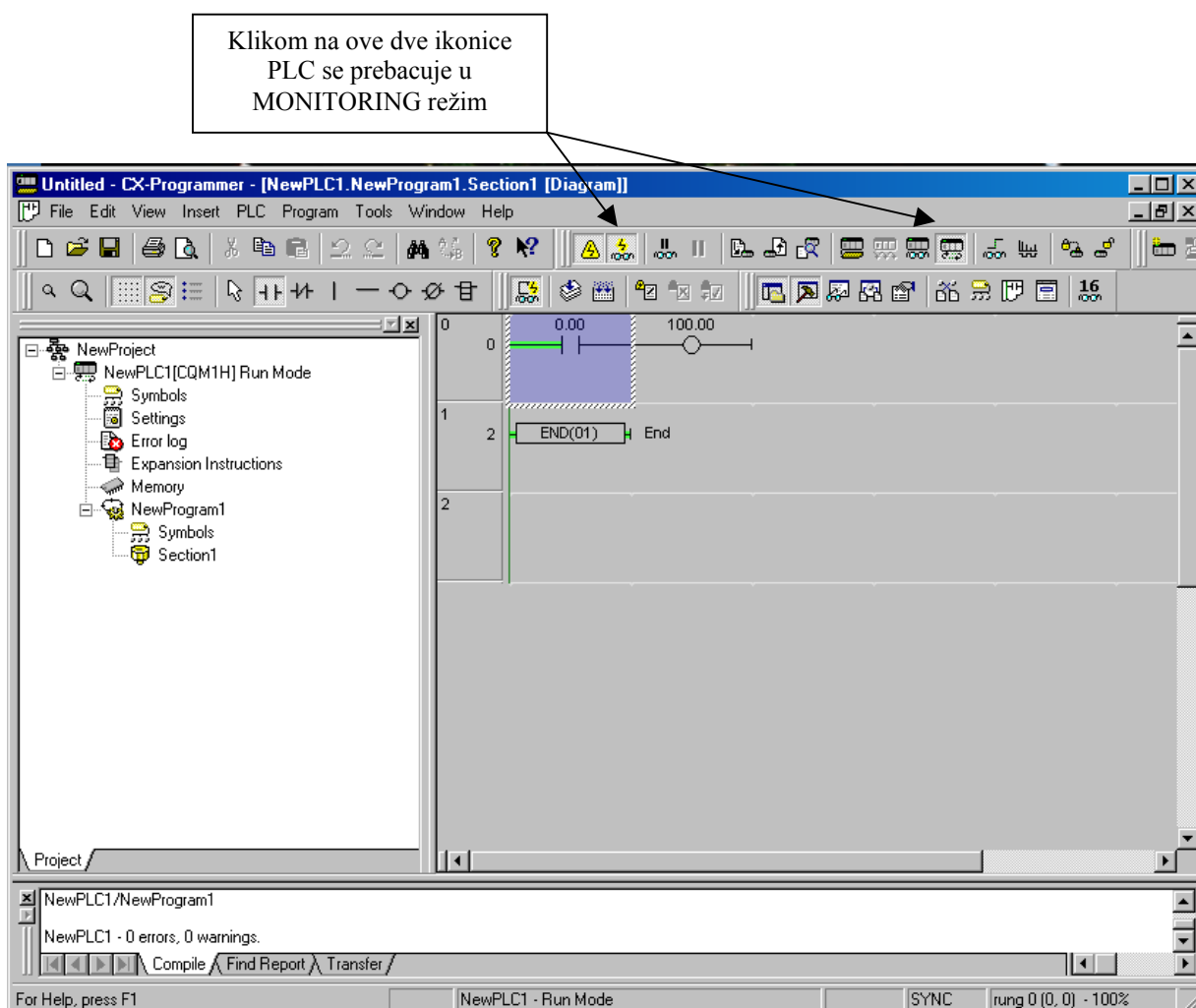
Na kraju, po uspešnom prenosu programa u *PLC* pojaviće se dijalog koji nas obaveštava o tome.



Sa 7. korakom *PLC* je isprogramiran. Ostaje još da se proverí program u praksi.

6.8. Provera rada programa

Najjednostavniji način je da se *PLC* prebaci u *RUN* mod i program isproba na samom uređaju ili u nekom sistemu automatizacije. Ipak, bolji i bezbedniji način je da se *PLC* prebaci u monitoring mod i da se izvrši simulacija rada uređaja ili procesa za čiju kontrolu se piše program.

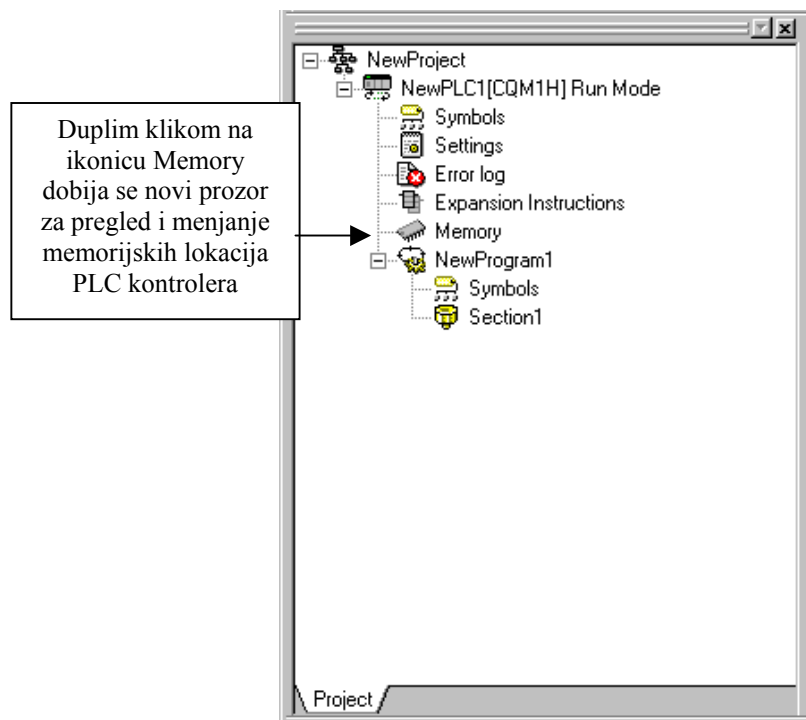


U slučaju da se program izvršava na realnom *PLC* kontroleru biće potrebno aktivirati ulaz IR000.00 da bi se aktivirao izlaz IR100.00. Obzirom da se radi o bitu na koji se može delovati preko ulazne klemne to neće biti problem. Problem nastaje kada se želi delovati na neki bit unutar *PLC* kontrolera. CX programer ima mogućnost da sam sebi zadaje stanja bitova u reči, ovaj oblik setovanja bitova se naziva "*Forsiranje bitova*" (*Force On* ili *Force Off*).

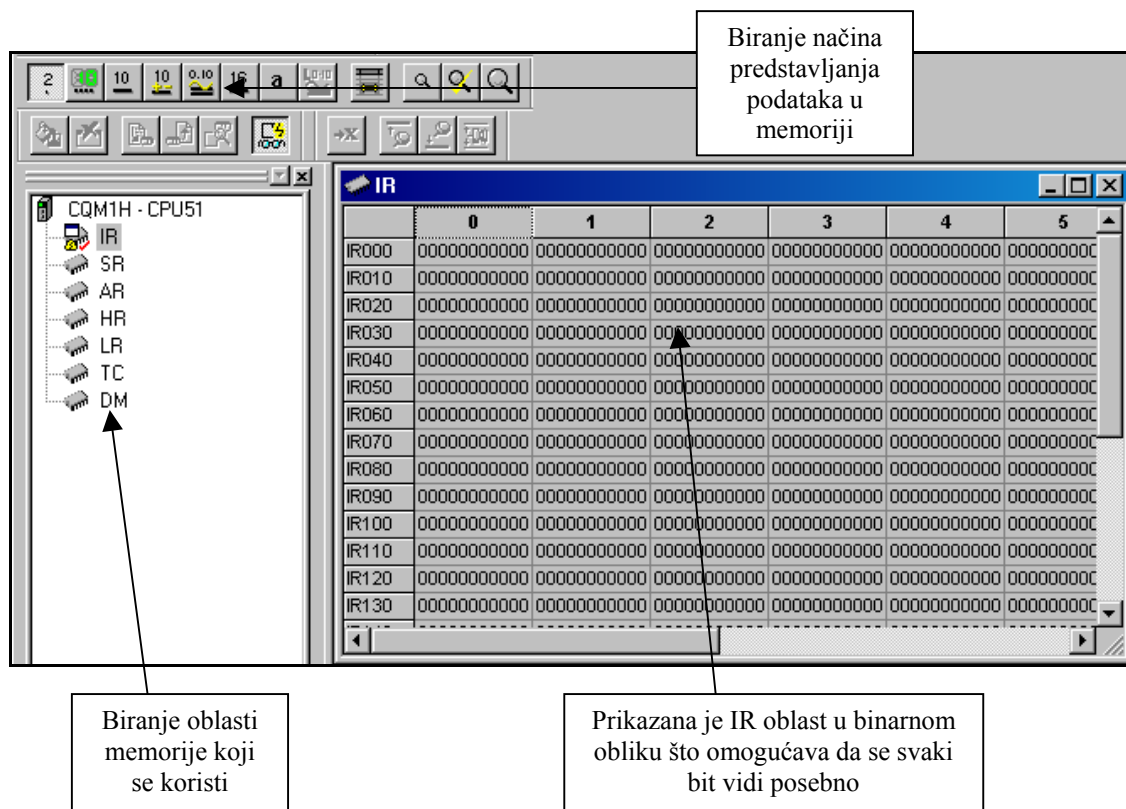
Koristeći ovu opciju (koja se dobija kada se na uslov koji se želi forsirati klikne desnim tasterom miša) moguće je simulirati spoljne uticaje na *PLC* kontroler. U slučaju ovog jednostavnog programa moguće je aktivirati rele na adresi IR100.00 i bez aktiviranja prekidača na adresi IR000.00 forsiranjem uslova na adresi IR000.00. Drugim rečima rele na adresi IR100.00 će "*kliknuti*" iako prekidač spojen na IR000.00 nije aktiviran!

6.9. Pristup svim memorijskim oblastima *PLC* kontrolera

Ova opcija bi se još mogla nazvati i "*memorija na dlanu*" što dovoljno govori šta se sve može uraditi. Prozor za rad sa memorijom se dobija duplim klikom na ikonicu "*Memory*" u projektnom delu CX programera.



Klikom na ikonicu dobija se prozor kao na narednoj slici. Ovo je bitan deo CX programera jer omogućuje da se vide unutrašnja stanja registara i drugim memorijskih lokacija. Podaci u memoriji se mogu videti na nekoliko načina počev od binarnog do heksadecimalnog ili ASCII. Binarni se koristi kada se radi o nekoj lokaciji koja kao celina nema neko značenje. Primer takve lokacije su ulazne i izlazne linije koje su spojene na klemne *PLC* kontrolera. Ako se ove lokacije (IR000 i IR100) predstavljaju binarno, onda će se svako aktiviranje ulaza ili izlaza videti kao promena stanja sa "0" na "1". ASCII način predstavljanja se koristi kada se u memoriji nalazi neka tekstualna poruka jer će na taj način poruka biti čitljiva. Heksadecimalni i decimalni zapis se koristi kada celokupna reč u memoriji nosi informaciju. Takav primer je recimo broj taktova koje je brojač registrovao i slično.



Jedna od važnih opcija je i direkto povezivanje sa realnim *PLC* kontrolerom gde će svaka promena biti odmah prikazana u prozoru željene memorijske oblasti. Na ovaj način dobijen je vrlo važan alat za nalaženje grešaka u programu ali i za bolje razumevanje rada *PLC* kontrolera.

PLC memory prozor ima još namena. Jedna od njih je i menjanje ili zadavanje vrednosti koje se nalaze na određenim adresama. Postupak ide na sledeći način:

1. korak: Priključiti *PLC* i prebaciti ga u "*on-line*" režim.
2. korak: Duplim klikom na ikonicu *Memory* aktivirati *PLC memory* prozor.
3. korak: Izabrati neku od memorijskih oblasti duplim klikom na nju.
4. korak: Klikom na četvrtu ikonicu u trećem redu sve vrednosti koje se nalaze na adresama te memorijske oblasti se prenose u *PLC memory* prozor.
5. korak: Promeniti vrednost koja se nalazi na adresi od interesa klikom na tu adresu i unosom nove vrednosti preko tastature.
6. korak: Klikom na treću ikonicu u trećem redu sve vrednosti koje se nalaze na adresama te memorijske oblasti se prenose u *PLC* kontroler.

Nakon poslednjeg koraka u *PLC* memoriji se nalazi željena vrednost. Ovaj način je dosta zgodan za proveru nekih računskih operacija, ali i za neke procese za koje ne treba čekati određeno vreme da se izvrše. Jedan takav primer je i tajmer koji recimo treba da reaguje tek nakon 100 minuta. Ubacivanjem vrednosti 98 u odgovarajuću memorijsku lokaciju programer treba da sačeka samo 2 umesto 100 minuta da bi video kako se program ponaša u realnim uslovima.

6.10. Značenje ikonica u paleti sa alatima

CX programer poseduje velik broj opcija i mogućnosti. Za nekoga ko se prvi put susreće sa ovim programom to može biti i problem. Na narednim stranama su date sve ikonice sa značenjima. Samo manji deo od svih njih se koristi u svakodnevnom radu tako da programer može sam izabrati koje želi da vidi a koje ne. Klikom na *View* ⇒ *Toolbars...* mogu se skinuti sve ikonice koje nisu potrebne.

	Novi projekat		
	Otvaranje postojećeg projekta		
	Snimanje projekta		
	Štampanje projekta		Pauza pre trigeru
	Izgled odštampanog projekta		Pauza
	Kopiranje sa brisanjem – cut		Prenos programa u PLC
	Kopiranje bez brisanja – copy		Čitanje programa iz PLC-a
	Lepljenje kopiranog dela programa – paste		Poređenje programa sa onim u PLC-u
	Korak unazad – undo		Programski režim
	Korak unapred – redo		Debug režim
	Pretraživanje programa – search		Monitor režim
	Pretraži i zameni		Run režim
	Help		Diferencijal monitor za praćenje promene određenog bita
	Help o onom delu programa na koji se klikne		Data trace
	ON-LINE povezivanje sa PLC kontrolerom		Postavljanje lozinke za pristup programu u PLC-u
	Prebacivanje PLC kontrolera u Monitoring režim		Skidanje lozinke za pristup programu unutar PLC-a

Ikonice u gornjem redu

	Umanji program - <i>Zoom out</i>		
	Uvećaj program - <i>Zoom in</i>		
	Prikaži raster – <i>Grid</i>		Invertovana izlazna instrukcija. Slično kao u predhodnom slučaju, ovom izvršnom instrukcijom prosleđujemo rezultat logičkog izraza izlaznom bitu s torn razlikom da je ovaj bit uključen ako uslov nije uspunjen i obrnuto.
	Prikaži komentare u programu		
	Prikazuje dodatne informacije o liniji instrukcija		
	Selekcija određenog dela programa		PLC funkcije. Klikom na ovu ikonicu dobija se mogućnost ugradnje kompleksnijih PLC instrukcija u program. Prozor koji se pojavljuje nakon klika na ikonicu sadrži i dugme " <i>Find instruction</i> " gde se mogu videti sve instrukcije sortirane po oblastima. Po dobijanju ovog prozora potrebno je izabrati instrukciji i kliknuti na taster OK.
	Ikonica otvoreni kontakt. Klikom na ovu ikonicu unosi se uslov na početku linije instrukcija. Element koji se unosi pozicionira se na označeno mesto (zatamljeno polje). Posle ovoga, automatski se aktivira dijalog gde se unose informacije (adresa otvorenog kontakta-broj reči, pozicija bita).		
	Ikonica zatvoreni kontakt. Klikom na ovu ikonicu unosi se zatvoreni kontakt odnosno invertovani uslov u liniju instrukcija.		Monitoring prozori
	Vertikalna linija. Klikom na ovu ikonicu, vrši se iscrtavanje vertikalnih linija, odozgo na dole, Ova opcija je neophodna kod realizovanja paralelnih veza između kontakata.		Prevođenje programa – <i>Compile</i>
	Horizontalna linija. Klikom na ovu ikonicu produžava se horizontalna linija s leva na desno.		Prevođenje svih programa - <i>Compile all</i>
	Izlazna instrukcija. Predstavlja instrukciju koja se izvršava ako je izvršena instrukcija uslova koja joj prethodi. Pomoću ove instrukcije vrši se prosleđivanje rezultata logičkog izraza izlaznim varijablama (bitovima).		On-line editovanje, promena dela programa u PLC kontroleru
			Prestanak on-line editovanja
			Slanje on-line promena u PLC kontkontroler

Ikonice u srednjem redu

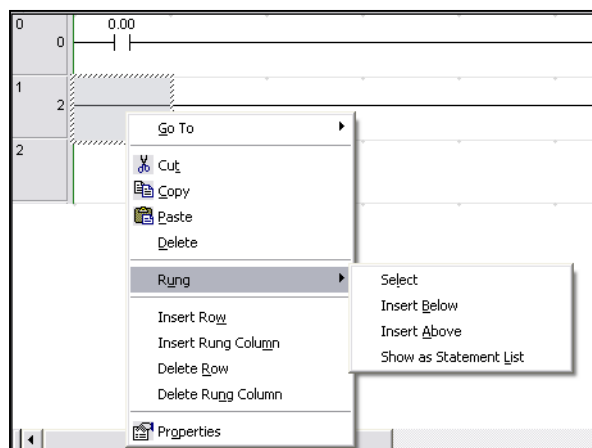
	Uključuje i isključuje projektni prozor sa desne strane.		Pravi tabelu korišćenih adresa u programu
	Uključuje i isključuje prozor za poruke na dnu ekrana		Tabela korišćenih lokalnih simbola
	Uključuje i isključuje prozor za praćenje promenljivih		Prikazuje program u vidu leder dijagrama
	Uključuje i isključuje prozor za adrese promenljivih		Prikazuje program u obliku mnemonika
	Prikazuje prozor sa karakteristikama programa.		Definiše heksadecimalni format vrednosti na adresama pri on-line monitoringu

Ikonice u donjem redu

6.11. Rad sa linijama instrukcija

Osnovna funkcionalna celina jednog programa jeste *Rung* (što bi u prevodu bilo prečka ili deo lestvice) ili jednostavnije "*linija instrukcija*". Program se sastoji od više linija instrukcija koje se nalaze jedan ispod druge. Operacije sa njima se dobijaju kada se na neku postojeću liniju instrukcija klikne desnim tasterom miša i izabere opcija *Rung* a zatim već po potrebi *Insert Above* (iznad) ili *Insert Below* (ispod). Pored toga tu se nalaze i opcije brisanja i selektovanja.

Samu liniju instrukcija je nekada potrebno proširiti (kako bi se npr. stavili paralelni uslovi) što se radi sa opcijama *Insert row* i *insert rung column*.



Dodavanje nove linije instrukcije iznad ili ispod selektovane se vrši desnim klikom na liniju instrukcije

Samo programiranje se svodi na izbor uslova, izbor akcije ili izbor neke funkcije kao npr. tajmera ili brojača i klikom na željeno mesto u liniji instrukcija.

UPRAVLJANJE SA PLC-om, REŠAVANJE KONKRETNOG UPRAVLJAČKOG ZADATAKA

1. Laboratorijska vežba se sastoji od:

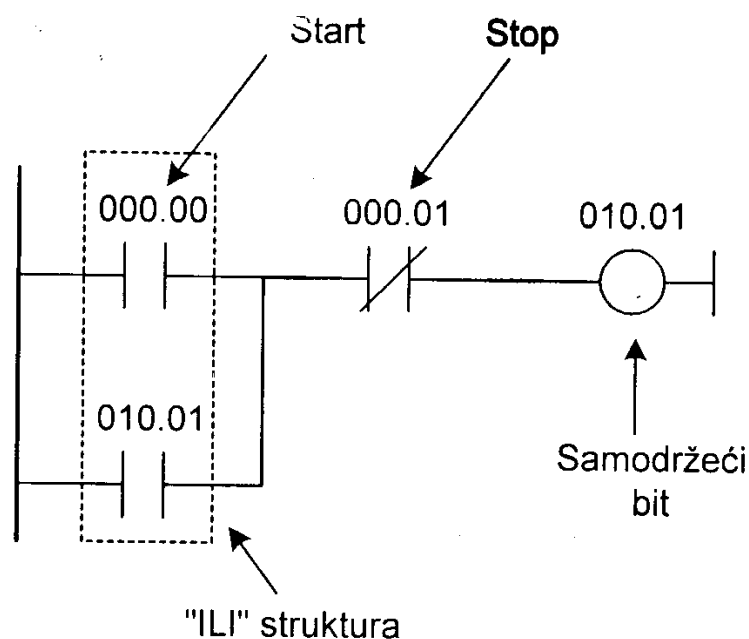
- PC računara sa instaliranim softverom SYSWIN za upravljanje PLC kontrolerima firme OMRON;
- Tabla sa PLC kontrolerom CPM1;
- Simulator ulaza i izlaza PLC kontrolera.

2. Primeri upravljačkih zadataka

Prvu grupu primera čine primeri vezani za samo programiranje. Dati su kao zasebni mali programi koji se kasnije mogu uključiti u veće. Drugu grupu primera čine primeri koji se mogu primeniti na neke realne probleme.

2.1. Samodržanje

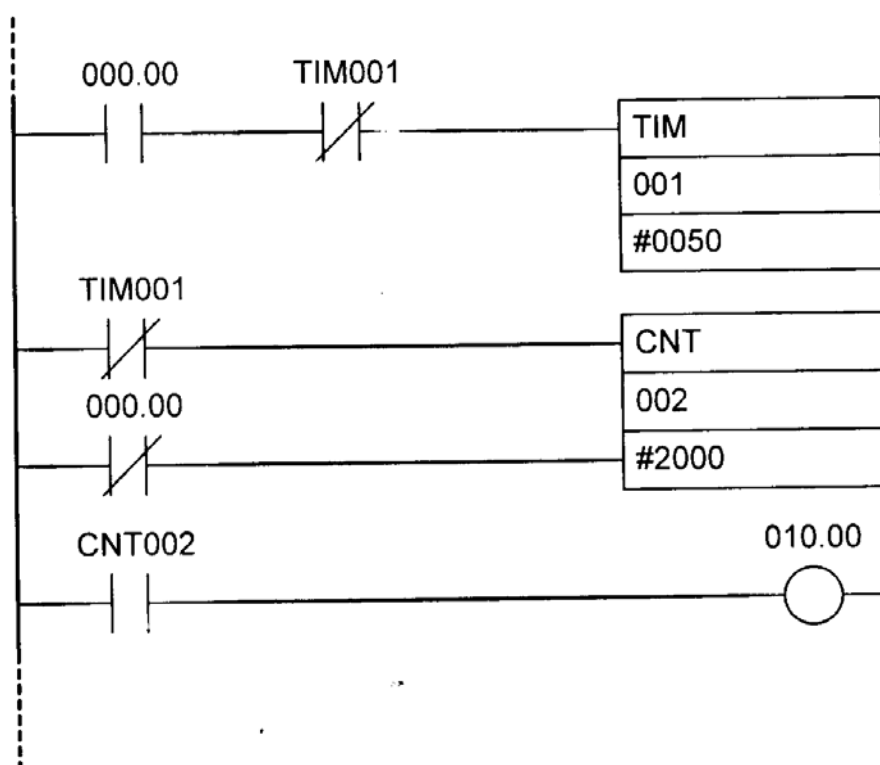
Program omogućava da izlaz ostane u stanju ON i po prestanku uslova koji ga je u to stanje doveo. Primer na slici ispod ilustruje kako se pritiskom na koji je povezan na ulaz IR000.00 menja stanje izlaza IR010.01 u ON. Otpuštanjem tastera izlaz IR010.01 se ne resetuje jer preko ILI kola (koga šine sa IR000.00) sam izlaz IR010.01 drži sebe u stanju ON u kome ostaje sve dok se ne pritisne taster na ulazu IR000.01 koji se nalazi u I vezi sa izlaznim pinom IR010.01 čime se raskida uslov i bit IR010.01 resetuje. Primer samodržanja je jako čest u konkretnim primenama. Ako bi na izlaz IR010 bio vezan neki potrošač onda bi sa dva tastera (znači bez upotreba prekidača) mogla da se ostvari START i STOP funkcija. Konkretno ulaz IR000.00 bi bio START taster a IR000.01



Lestvičasti dijagram primera sa samodržanjem

2.2. Veliki vremenski interval

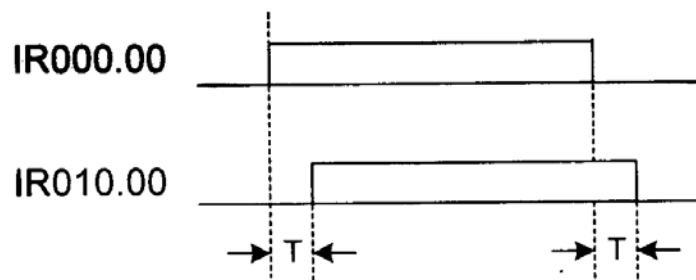
Ukoliko je potrebno napraviti veći vremenski interval od 999,9 sekundi ($9999 \times 0,1s$) mogu se koristiti dva vezana tajmera ili tajmer i brojač kao u ovom primeru. Brojač je setovan na broji do 2000 a tajmer je postavljen na 5 sekundi što daje vremenski interval od 10.000 sekundi ili 2.77 sata. Ispunjavanjem uslova na ulazu IR000.00 tajmer počinje da odbrojava. Kada dođe do kraja, setuje fleg TIM001 koji raskine vezu i ujedno resetuje tajmer. Po isteku 5 sekundi fleg TIM001 menja stanje u ON i ispunjava uslov na ulazu u brojač CNT002. Kada brojač izbroji 2000 takvih promena stanja flega tajmera TIM001 setuje svoj fleg CNT002 čime se ispuni uslov da izlaz IR010.00 promeni stanje u ON. Vreme koje je proteklo između promene stanja ulaza IR000.00 u ON i i promene stanja izlaza IR010.00 u ON iznosi 10.000 sekundi.



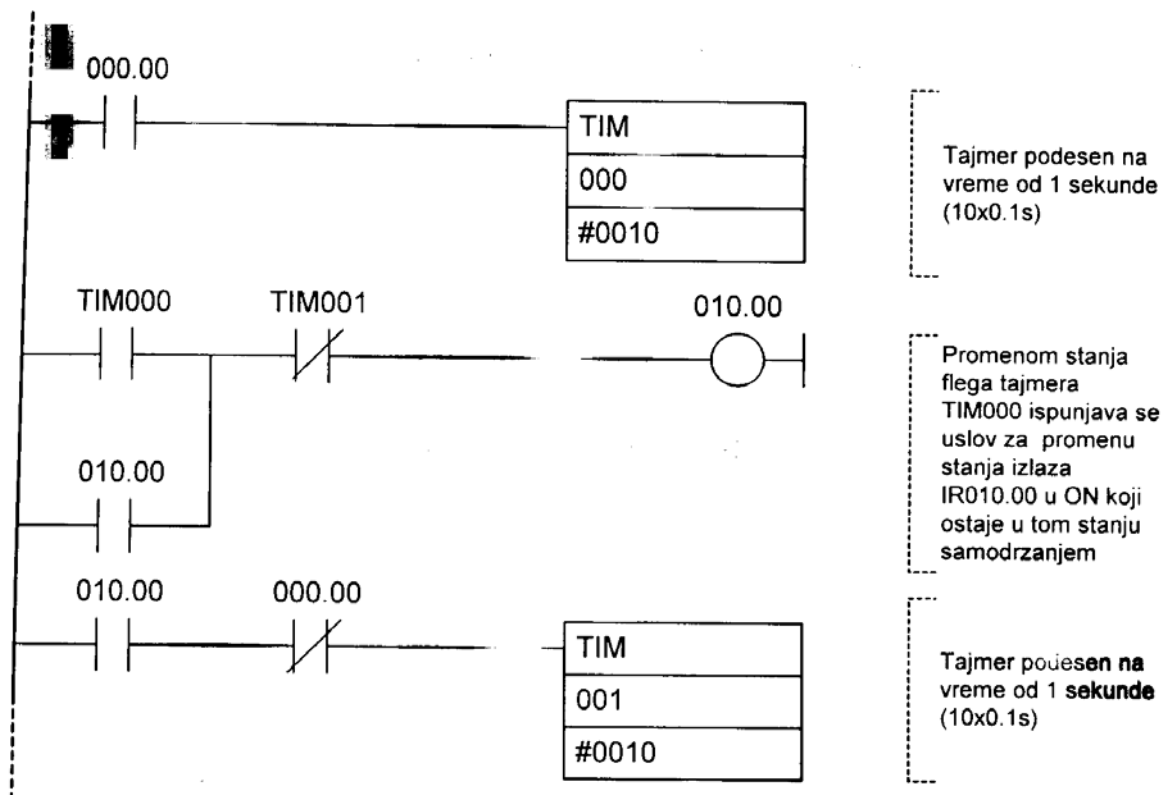
Lestvičasti dijagram primera sa velikim vremenskim intervalom

2.3. Kašnjenje pri uključenju i isključenju

Primer pokazuje kako napraviti kašnjenje izlaza (IR010.00) u odnosu na ulaz (IR000.00). Ispunjavanjem uslova na ulazu IR000.00 tajmer TIM000 počinje da odbrojava setovanu vrednost 10 u koracima po 0.1 sekund. Nakon isteka jedne sekunde setuje svoj fleg TIM000 koji je uslov za promenu stanja izlaza IR010.00 u ON. Time se između ON stanja ulaza IR000.00 i ON stanja izlaza IR010.00 napravilo kašnjenje od jedne sekunde. Promenom stanja izlaza IR010.00 u ON ispunjena je polovina uslova za aktiviranje drugog tajmera. Druga polovina uslova se ispuni kada ulaz IR000.00 promeni stanje u OFF (normalno zatvoren tip kontakta). Tajmer TIM001 po završetku vremena jedne sekunde na koju je podešen setuje svoj fleg TIM001 čime raskida uslov za držanje izlaza u ON stanju.



Vremenski dijagram primera sa kašnjenjem pri uključenju i isključenju

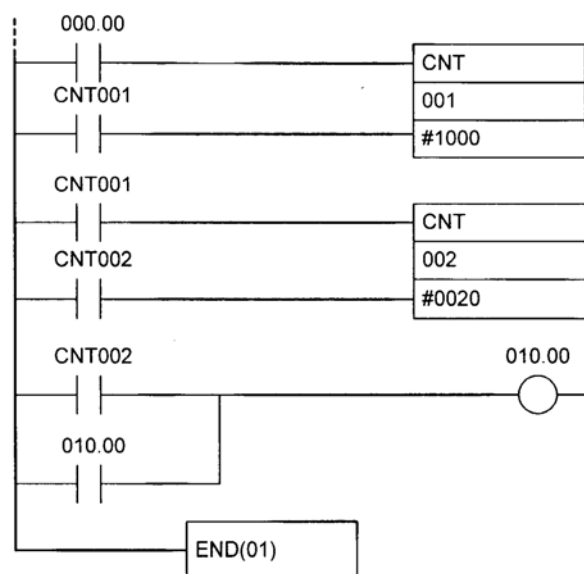


Lestvičasti dijagram primera sa kašnjenjem pri uključenju i isključenju

2.4. Brojač preko 9999

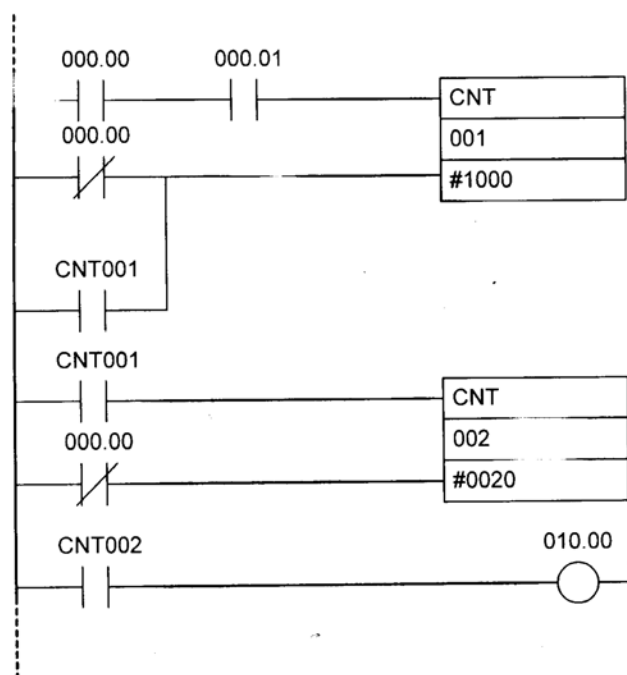
Ukoliko je potrebno izvršiti brojanje preko vrednosti 9999 (na koliko se maksimalno može podesiti brojač) mogu se koristiti dva vezana brojača. Prvi brojač broji do određene vrednosti a drugi broji promene stanja flega prvog brojača. Time se dobija mogućnost brojanja do vrednosti koja je proizvod setovanih vrednosti prvog i drugog brojača. U donjem primeru prvi brojač broji do 1000 a drugi do 20, čime se dobija mogućnost brojanja do 20000. Ispunjavanjem uslova na ulazu IR000.00 (linija čije se promene prate se dovodi na njega) prvi brojač umanjuje svoju vrednost za jedan, što se ponavlja sve dok ne dođe do nule kada setuje svoj fleg CNT001 čime se ujedno i resetuje (bude spreman za novi ciklus brojanja od 1000 do 0). Svako setovanje CNT001 utiče na drugi brojač koji setuje svoj fleg nakon dvadesetog setovanja flega prvog brojača. Setovanjem flega

CNT002 drugog brojača ispunjava uslov da se izlaz IR010.00 aktivira i ostane u tom stanju samoodržanjem.



Lestvičasti dijagram primera sa brojačem preko 9999

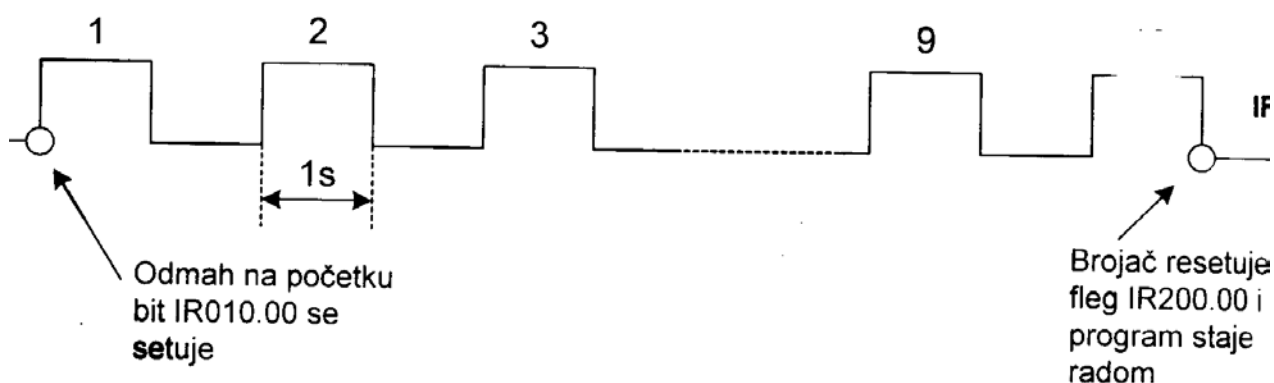
Isti efekat se može postići i donjim modifikovanim programom. Prva promena je da postoji "prekidač" za ceo program i to je ulaz IR000.00 (samo dok je on aktivan program može da vrši svoju funkciju). Druga promena je da se linija čije se stanje prati dovodi na ulaz IR000.01. Ostalo je isto ko i u prethodnoj verziji programa. Brojač CNT002 broji promene stanja flega brojača CNT001. Kada ih odbroji menja stanje svog flega CNT002 čime se ispunjava uslov za promenu stanja izlaza IR010.00. Time se stanje izlaza IR010.00 promeni nakon 20000 promena ulaza IR000.01.



Lestvičasti dijagram modifikovanog primera sa brojačem preko 9999

2.5. Takter

Primer proizvodi određeni broj impulsa željene dužine trajanja na izlazu IR010.00 PLC kontrolera. Broj impulsa se zadaje u instrukciji brojača (ovde je to konstanta #0010, odnosno deset impulsa) a vreme trajanja impulsa u dve tajmerske instrukcije. Prvi tajmer definiše trajanje ON stanja a drugi trajanje OFF stanja izlaznog bita IR010.00. U primeru su ta dva vremena ista ali se drugačijim zadavanjem parametara mogu razlikovati tako da vreme stanja ON bude različito od vremena stanja OFF.



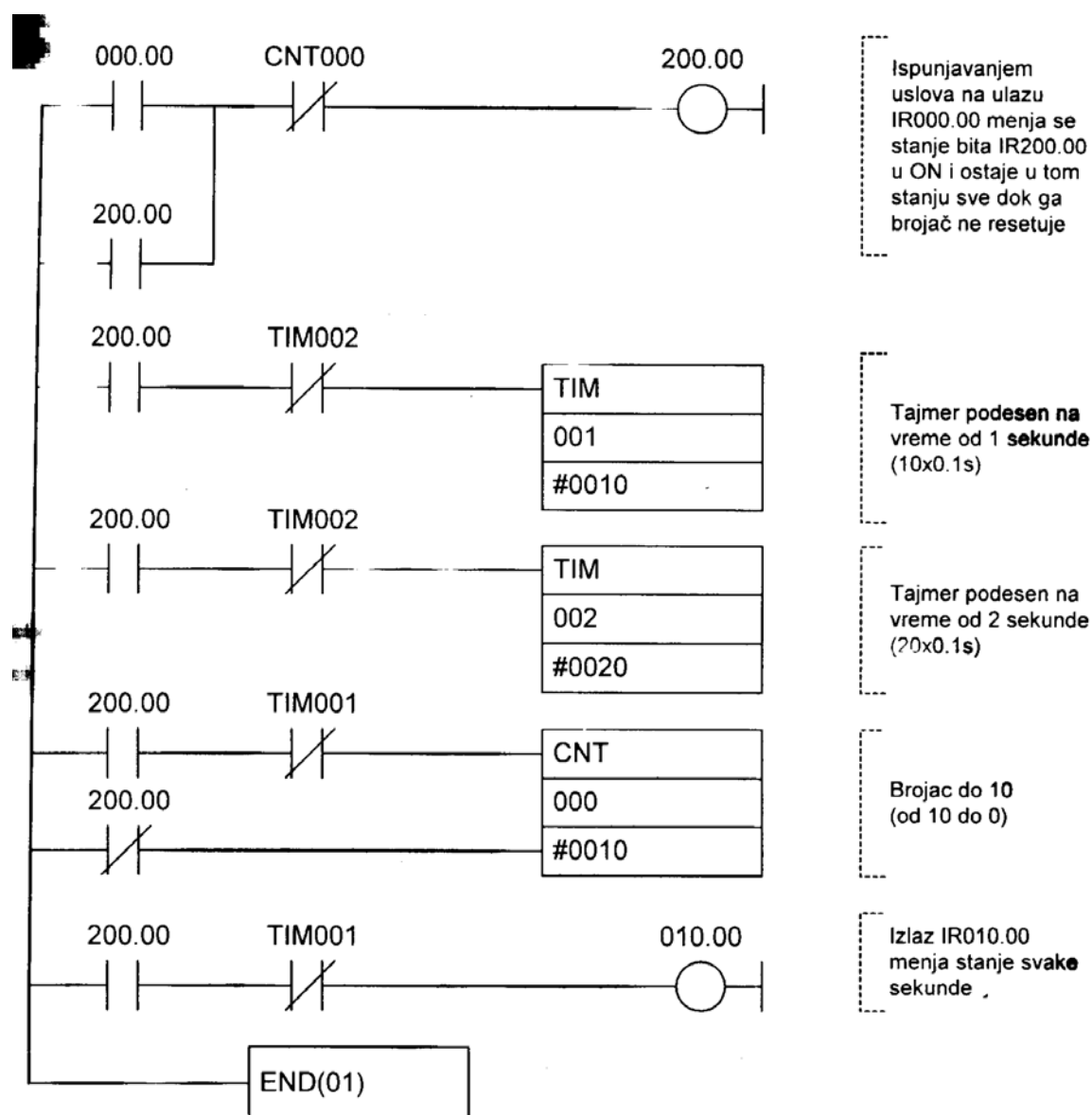
Vremenski dijagram na izlazu taktera

Program počinje sa izvršavanjem ispunjenjem uslova na bitu IR000.00. Kako je sa njim vezan u "I" kolo normalno.zatvoreni kontakt koji se odnosi na fleg brojača koji nije setovan to će se stanje bita IR200.00 promeniti u ON. Bit IR200.00 zadržava svoje stanje samodržanjem sve dok se fleg brojača ne setuje i tako raskine uslov.

Po setovanju bita IR200.00 tajmeri TIM001 i TIM002 počinju da odbrojavaju setovani broj intervala po 0.1 s (u primeru je taj broj 10 za prvi tajmer odnosno 20 za drugi čime se setuje vreme od jedne odnosno dve sekunde). Kod oba tajmera sa bitom IR200.00 je povezan i normalno zatvoren kontakt koji se odnosi na fleg tajmera TIM002. Kada se taj fleg setuje što se dešava svake dve sekunde resetuju se oba tajmera. Tajmer TIM002 resetuje tajmer TIM001 i sam sebe čime počinje novi ciklus.

Odmah na početku programa izlazni bit IR010.00 menja stanje u ON i ostaje u tom stanju sve dok fleg TIM001 ne promeni stanje u ON (nakon jedne sekunde). Promenom stanja flega TIM001 u ON raskida se uslov (jer je predstavljen kao normalno zatvoren kontakt) i bit IR010.00 menja stanje u OFF.

Stanje izlaza IR010.00 se menja u ON ponovo posle isteka vremena na tajmeru TIM002 čime se resetuje tajmer TIM001 i njegov fleg čime se ispunjava uslov za promenu stanja izlaza IR010.00. Ciklus se tako ponavlja sve dok brojač ne odbroji 10 promena stanja flega TIM001. Promenom stanja flega brojača CNT000 raskida se uslov za pomoćni bit IR200.00 i program staje sa radom.

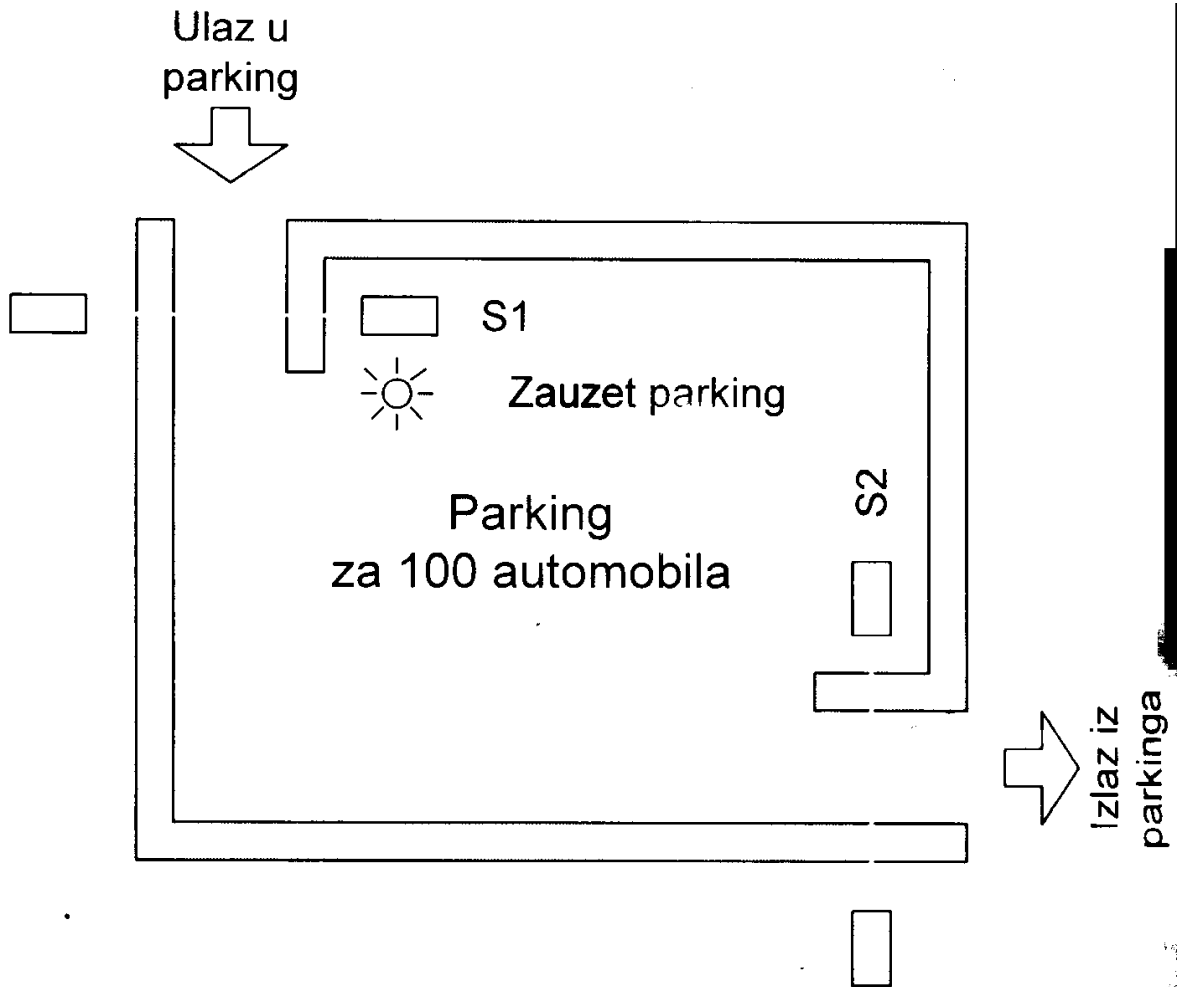


Lestvičasti dijagram modifikovanog primera sa takterom

2.6. Automatizacija parkinga

Radi se o jednostavnom sistemu koji može da kontroliše maksimalno 100 automobila. Svaki put kada automobil uđe, PLC ga automatski dodaje na zbir automobila koji se već nalaze na parkingu. Svaki automobil koji izađe biće automatski oduzet. Kada se parkira 100 automobila, upaliće se znak da je parking pun radi obaveštavanja vozila koja nailaze da ne ulaze jer nema više mesta na parkingu.

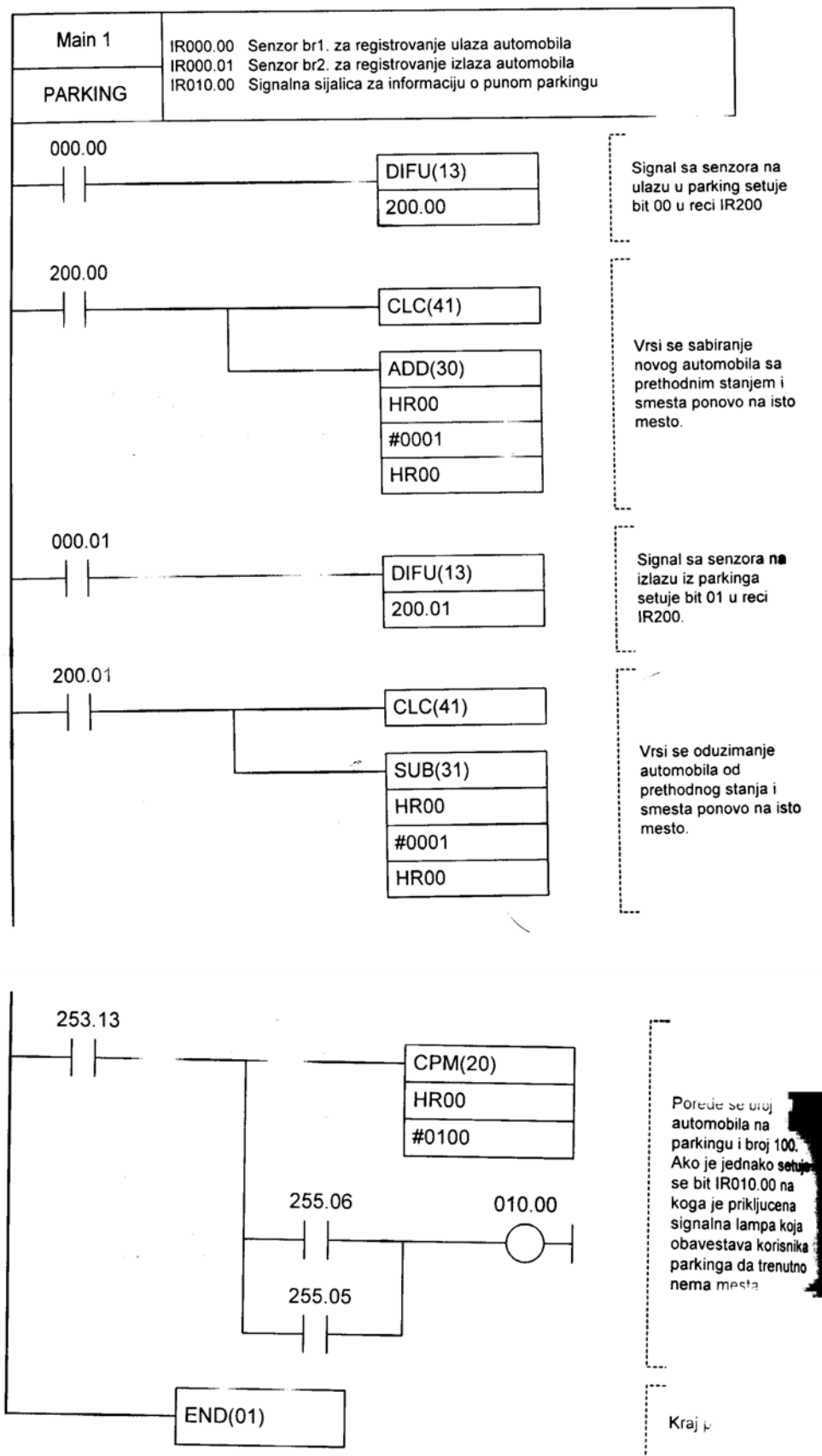
Signal sa senzora na ulazu u parking setuje bit IR200.00. Taj bit je uslov izvršenja naredne dve instrukcije u programu. Prvom instrukcijom se resetuje keri bit CY (uvek se radi pre neke računске operacije koja utiče na njega) a drugom se na broj automobila u reči HR00 dodaje jedan i tako dobijeni zbir ponovo smešta u HR00. HR oblast memorije je odabrana za smeštanje ukupnog broja automobila zbog toga što zadržava svoje stanje i nakon nestanka napajanja.



Parking

Znak "#" u instrukcijama sabiranja i oduzimanja definiše decimalnu konstantu koja se oduzima ili dodaje na broj automobila koji se već nalaze na parkingu. Uslov izvršavanja instrukcije poređenja CPM je uvek ispunjen jer je bit 5R253.13 je uvek setovan, to praktično znači da će se poređenje raditi u svakom ciklusu bez obzira da li je neki automobil ušao ili izašao.

Signalna lampica za "pun parking" je povezana na izlaz IR010.00. Njeno uključenje i isključenje kontrolišu flegovi EQ (eng. equal, u prevodu jednako) na adresi SR255.06 i GR (eng. Greather Than, u prevodu veće ili jednako) na adresi SR255.05. Oba bita se nalaze u ILI vezi sa izlazom IR010.00 na kome je signalna lampica. Na ovaj način lampica će svetleti kada je broj automobi veći ili jednak broju 100. Broj automobila u realnim uslovima zaista može da bude veći od 100 jer neki nepoverljivi vozač može poželeti da proveri da zaista nema mesta i tako broj automobila koji se trenutno nalaze na parkingu povećati sa 100 na 101. Njegovim izlaskom sa parkinga broj automobila smanjuje na 100 koliko i ima parking mesta.

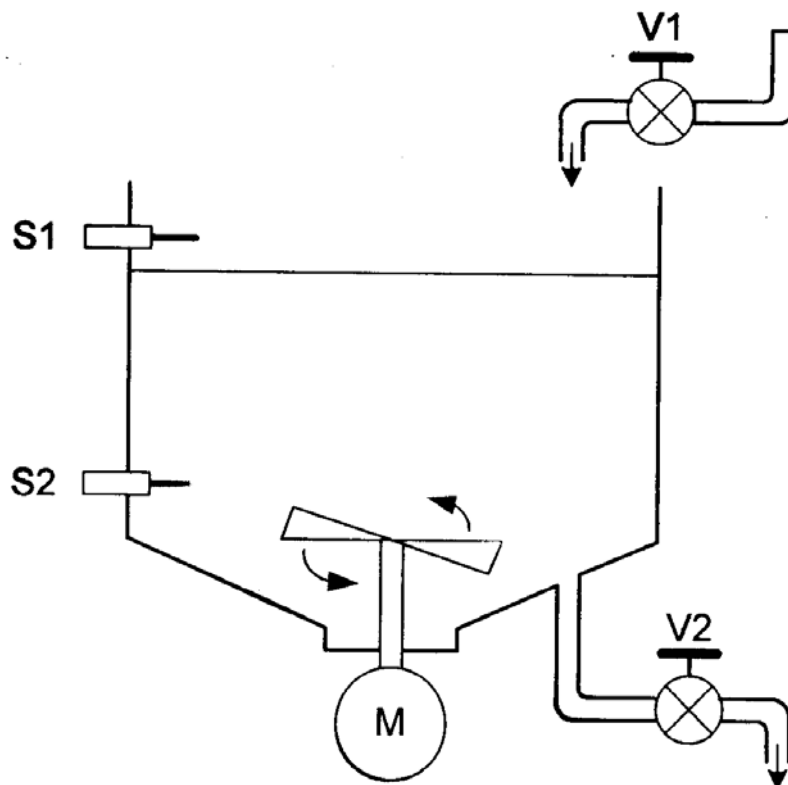


Lestvičasti dijagram primera automatizacije parkinga

2.7. Upravljanje procesom punjenja / pražnjenja

Punjenje i pražnjenje rezervoara kao i potreba za mešanjem dve ili više supstanci je dosta čest slučaj u industriji. Upotrebom automatskih ventila taj proces se može u potpunosti automatizovati. Neka je tečnost u primeru voda i neka treba četiri puta napuniti i isprazniti rezervoar.

Kada se na upravljačkom panelu pritisne T1, otvara se ventil V1 i voda počinje da puni rezervoar. U isto vreme, motor M mešalice počinje sa radom. Punjenjem rezervoara raste nivo vode i dostiže nivo određen senzorom S1, zatvara se ventil V1 a motor mešalice zaustavlja. Nakon toga otvara se ventil V2 i rezervoar počinje da se prazni. Kada nivo vode padne ispod nivoa određenog senzorom S2 zatvara se ventil V2. Ponavljanjem istog ciklusa četiri puta aktivira se lampica za indicaciju kraja ciklusa. Pritiskom na taster T1 pokreće se novi ciklus.



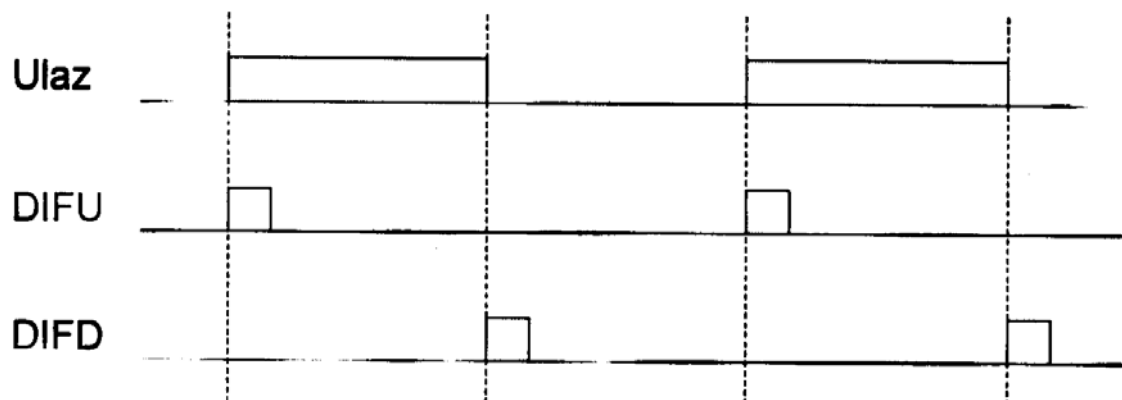
Rezervoar sa elementima

U primeru se koriste dve vrste diferencijatora. Njihova uloga se može videti na slici ispod. Senzori nivoa S1 i S2 daju informaciju o tome da li nivo tečnosti prelazi određenu vrednost. Informacije tog tipa nisu od značaja kada se želi znati da li se u određenoj sekvenci nivo tečnosti podiže ili spušta. Tu stupaju na scenu diferencijatori. Naime, u segmentu broj 3 lestvičastog dijagrama se detektuje događaj dostizanja gornjeg nivoa, odnosno trenutak kada tečnost puneći rezervoar pređe gornji nivo i aktivira senzor S1. Kratkotrajno aktiviranje izlaza IR200.02 ima za posledicu isključenje izlaza V1 (ventila za vodu, sprečava dalji dotok vode ali i rad motora mešalice). Trenutak pre toga (segment 5) uključuje se ventil V2 što označava početak isticanja tečnosti. Druga dva diferencijatora (u segmentima 6 i 7) imaju zadatak da registruju događaje zatvaranja ventila V2 i pada nivoa tečnosti ispod dozvoljen minimuma.

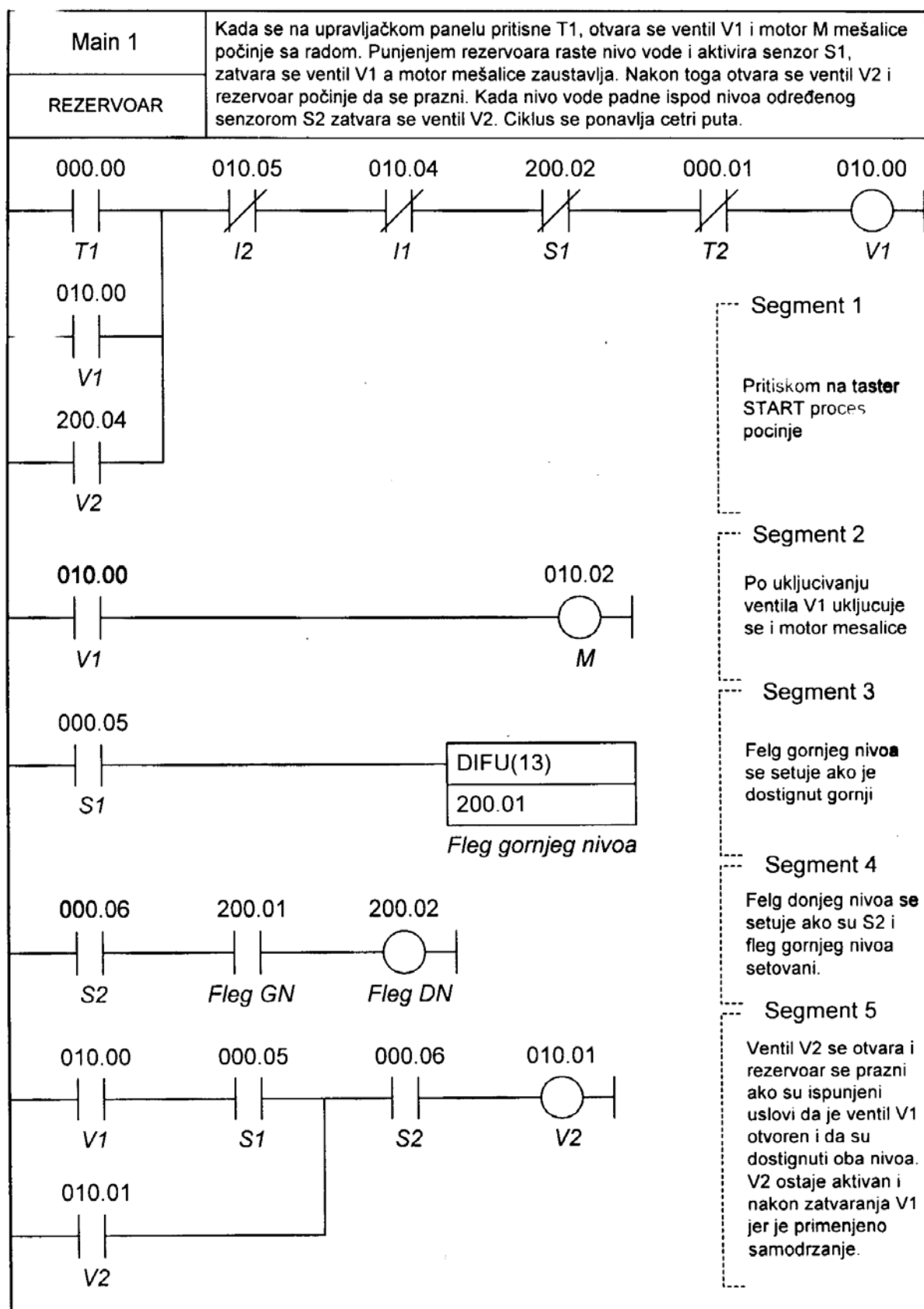
ULAZ	FUNKCIJA
IR000.00	Start taster T1
IR000.01	Stop taster T2
IR000.02	Reset taster T3
IR000.05	Senzor gornjeg nivoa S1
IR000.06	Senzor donjeg nivoa S2

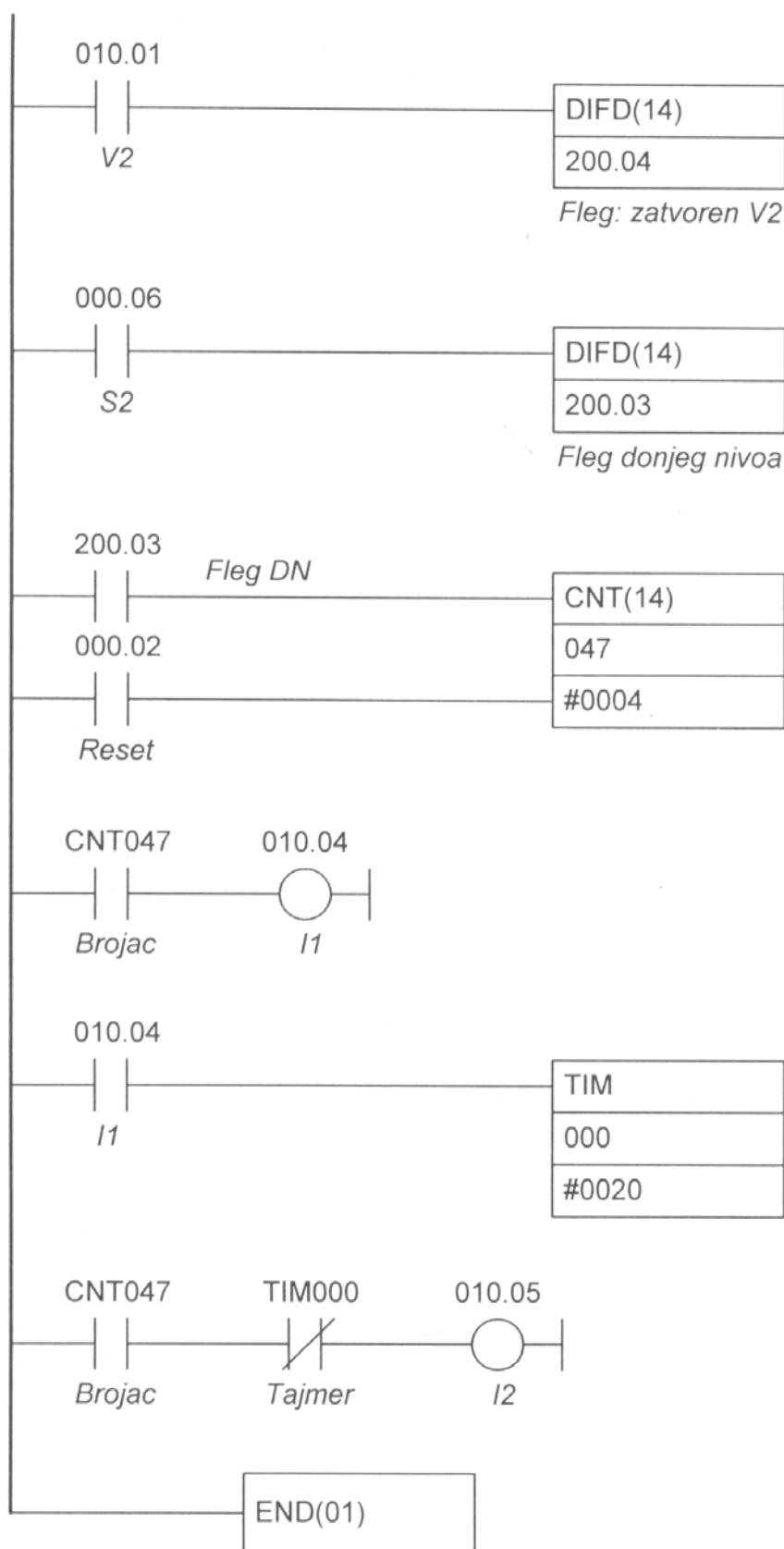
IZLAZ	FUNKCIJA
IR010.00	Ventil za punjenje V1
IR010.01	Ventil za pražnjenje V2
IR010.02	Motor za mešanje M
IR010.04	Svetlosni indikator I1
IR010.05	Zvučni indikator I2

Ulazi i izlazi primera upravljanja procesom punjenja i pražnjenja



Razlika između DIFU i DIFD instrukcija





Segment 6

Po Ventil V2 setuje fleg koji oznacava da je V2 otvoren.

Segment 7

Senzor S2 koji registruje dostizanje donjeg nivoa setuje fleg donjeg nivoa.

Segment 8

Fleg dostizanja donjeg nivoa se broji i poredi sa brojem cetri. Brojac se moze resetovati reset tasterom T2

Segment 9

Po izvršenju cetri ciklusa aktivira se lampica kao indikator kraja ciklusa.

Segment 9

Podesavanje tajmera na duzinu od dve sekunde

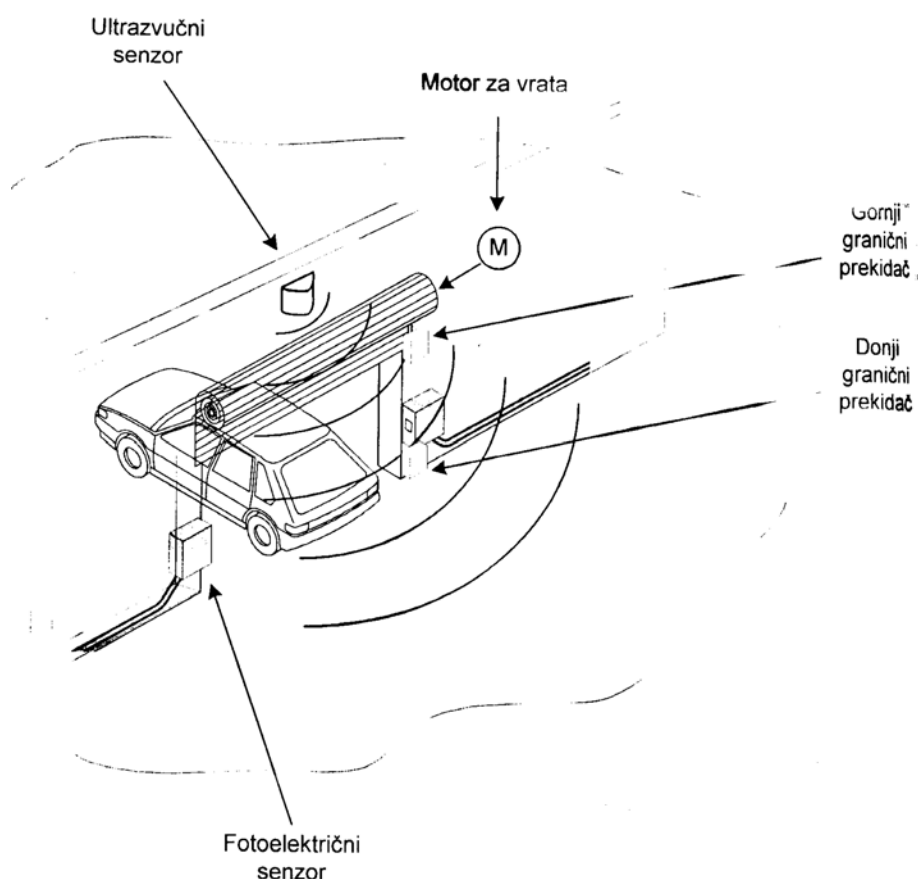
Segment 10

Zvucni signal nakon cetri ciklusa u trajanju od dve sekunde

Kraj programa

2.8. Automatizacija vrata sladišta

Vrata skladišta ili uopšteno bilo koja vrata se mogu automatizovati tako da o njihovom otvaranju i zatvaranju ne brine čovek. Primenom jednog trofaznog motora kome će se menjati smer okretanja može se obezbediti i podizanje i spuštanje vrata. Za registrovanje prisustva vozila ispred vrata koristi se ultrazvučni senzor a za prolaz vozila fotoelektrični senzor. Nailaskom vozila vrata se podižu a prolaskom vozila kroz vrata (prekine se zrak svetlosti fotoelektričnom senzoru) vrata se spuštaju.



Vrata skladišta sa potrebnom opremom

ULAZ	FUNKCIJA
IR000.00	Ultrazvučni senzor
IR000.01	Fotoelektročni senzor
IR000.02	Gornji granični prekidač
IR000.03	Donji granični prekidač

IZLAZ	FUNKCIJA
IR010.00	Motor za podizanje vrata
IR010.01	Motor za spuštanje vrata

Dodeljeni ulazi i izlazi

Main 1	Ultrazvucni senzor detektuje prisustvo vozila cime se aktivira otvaranje vrata. Vozilo prolazi kroz vrata gde ga detektuje fotoelektricni prekidač nakon čega se vrata zatvaraju.
SKLADISTE	


```

graph LR
    subgraph Segment_1 [Segment 1]
        direction LR
        A["000.00  
vozilo prisutno"] --- B["010.00  
vrata, gore"]
        A --- C["000.02  
g.gr.prek."]
        B --- C
        C --- D["010.01  
vrata, dole"]
        D --- E["010.00  
vrata, gore"]
    end

    subgraph Segment_2 [Segment 2]
        direction LR
        F["000.01  
vozilo prolazi"] --- G["200.00  
DIFD(14)  
pomocni fleg"]
    end

    subgraph Segment_3 [Segment 3]
        direction LR
        H["200.00  
pomocni fleg"] --- I["010.01  
vrata, dole"]
        H --- J["010.01  
vrata, dole"]
        I --- K["000.03  
d.gr.prek."]
        J --- K
        K --- L["010.00  
vrata, gore"]
        L --- M["010.01  
vrata, dole"]
    end

    END["END(01)"]
  
```

Segment 1
 Pojavom vozila ispunjava se uslov za podizanje vrata. Pored tog uslova potrebno je da gornji granicni prekidač bude OFF kao i pogon motora za spustanje vrata.

Segment 2
 Setuje se pomocni fleg kada vozilo prekine zrak svetlos na senzoru.

Segment 3
 Pomocni fleg aktivira spustanje vrata. Pored tog uslova potrebno je da je iskljucen pogon za podizanje vrata i da se vrata ne nalaze donjem poloaju.

Kraj programa

72

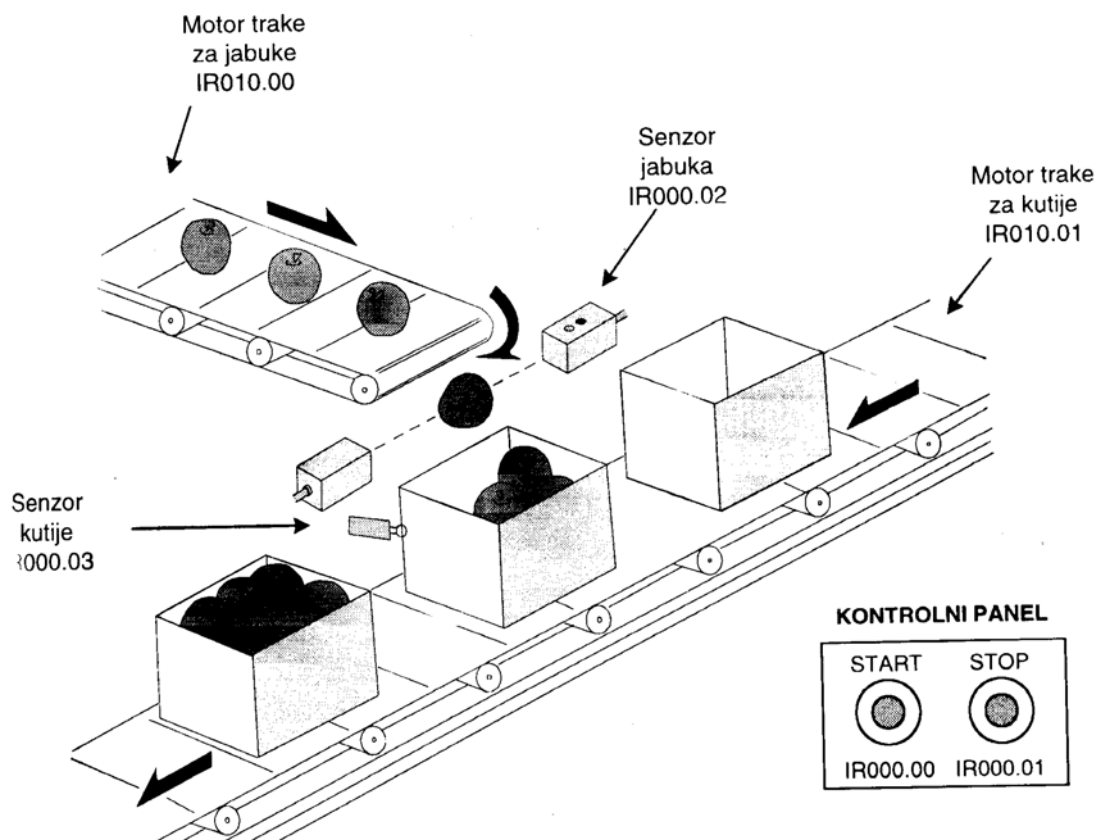
Setovanjem bita IR000.00 na ulazu PLC kontrolera na koga je povezan ultrazvučni senzor aktivira se izlaz IR010.00 (na koga je povezana sklopka), tako da motor podiže vrata. Pored ovog uslova potrebno je da pogon za spuštanje vrata ne bude aktivan (IR010.01) kao i da vrata nisu već u gornjem položaju (IR000.02). Uslov za gornji granični prekidač je dat kao normalno zatvoren tako da će promenom njegovog stanja iz OFF u ON (kad se vrata podignu) prestati uslov za bit IR010.00 na kome je pogon za podizanje vrata (Segment 1.).

Fotoelektrični prekidač registruje prolazak vozila i setuje fleg IR200.00. Upotrebljena je instrukcija DIFD koja se aktivira kada uslov koji joj prethodi promeni stanje sa ON u OFF. Kada vozilo prolazi kroz vrata prekida zrak i stanje bita IR000.01 prelazi sa ON u OFF (Segment 2.).

Menjanjem stanja pomoćnog flega sa OFF u ON ispunjava se uslov za spuštanje vrata (Segment 3.). Pored tog uslova potrebno je da pogon za podizanje vrata bude isključen kao i da se vrata ne nalaze već u donjem položaju. Bit koji upravlja pogonom za spuštanje IR010.01 je samodržeci tako da se vrata spuštaju dok se ne dođe do donjeg graničnog prekidača koji je u uslovu predstavljen kao normalno zatvoren. Njegova promena stanja iz OFF u ON prekida uslov pogona za spuštanje vrata. Dolaskom novog vozila ciklus se ponavlja.

2.9. Automatizacija pakovanja proizvoda

Problem pakovanja proizvoda je jedan od najčešćih slučajeva automatizacije u industriji. Može se sresti na malim mašinama (npr. pakovanje zrnastih prehrambenih proizvoda) pa do velikih sistema kao što su mašine za pakovanje lekova. Primer o kome je ovde reč rešava klasičan problem pakovanja sa malim brojem elemenata automatizacije. Mali broj potrebnih ulaza i izlaza omogućuje korišćenje CPMIA PLC kontrolera koji predstavlja jednostavno i ekonomično rešenje.



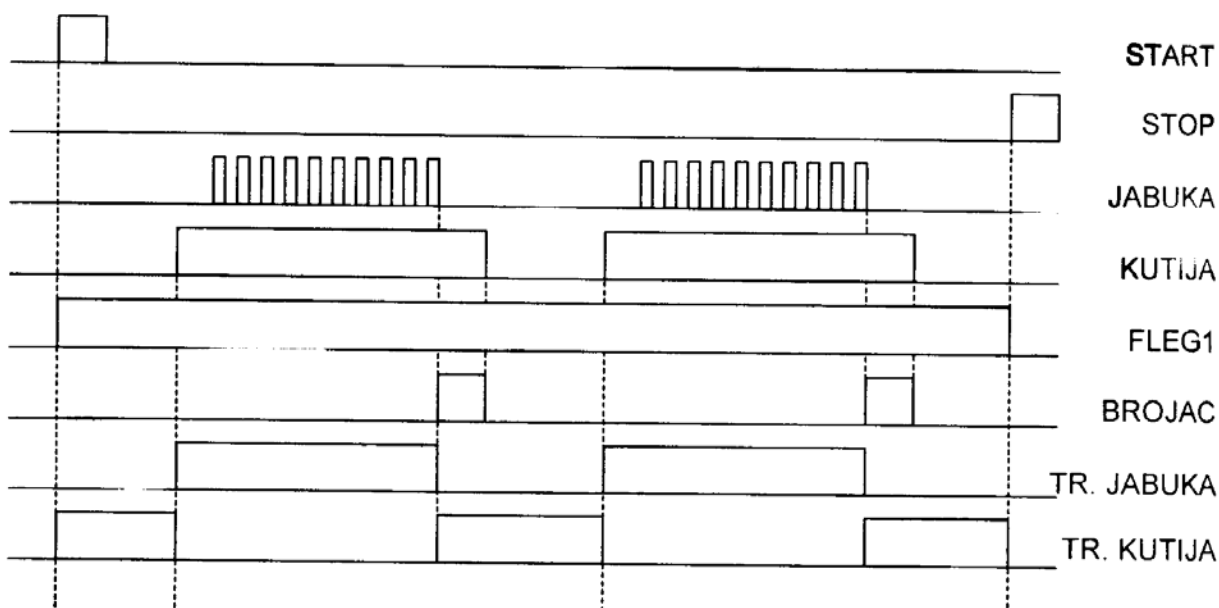
Traka za pakovanje proizvoda sa elementima

ULAZ	FUNKCIJA
IR000.00	Start taster T1
IR000.01	Stop taster T2
IR000.02	Senzor jabuka
IR000.03	Senzor kutija

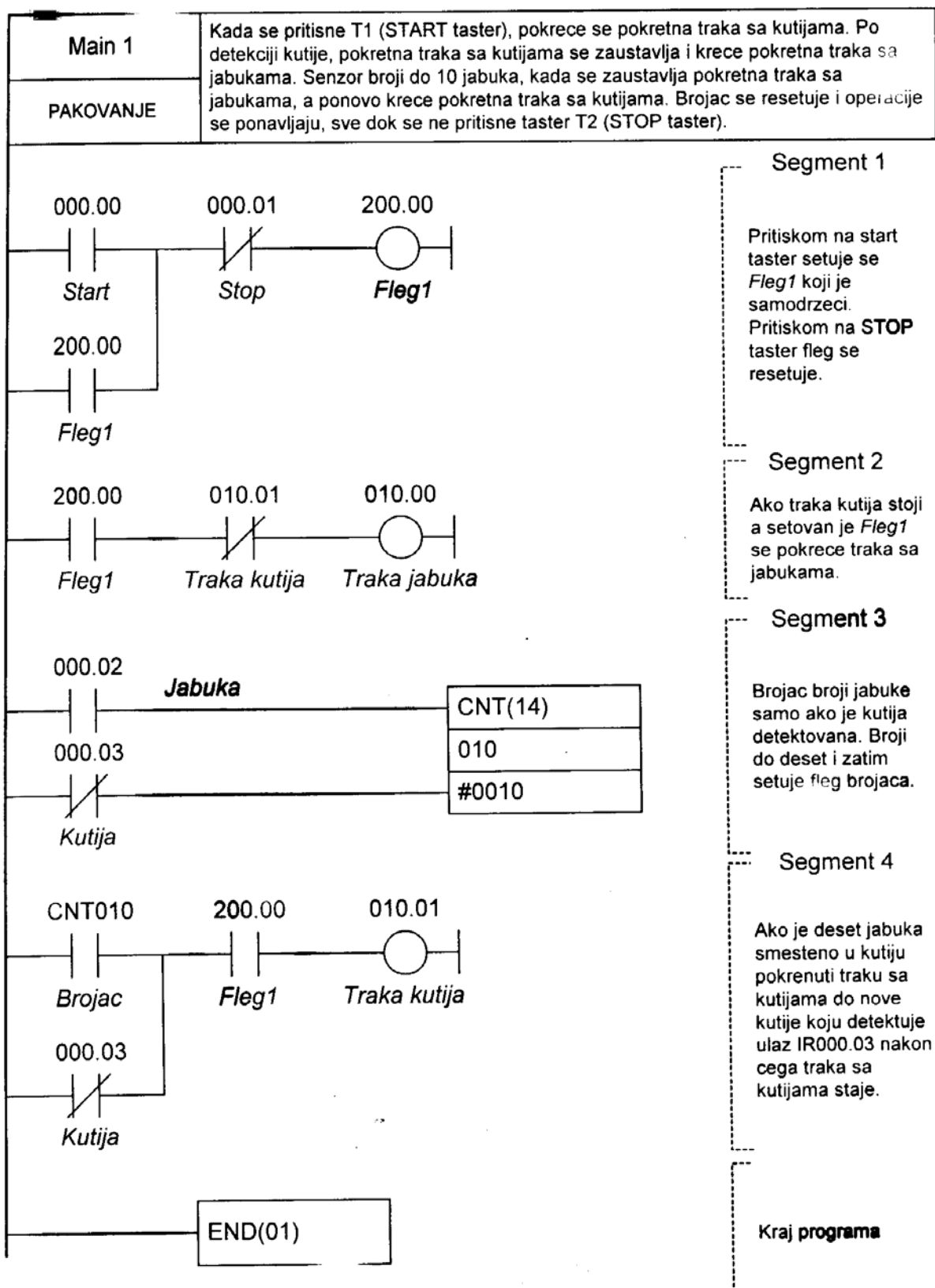
IZLAZ	FUNKCIJA
IR010.00	Traka jabuka
IR010.01	Traka kutija

Dodeljeni ulazi i izlazi

Pritiskom na taster START, aktivira se Flegl koji predstavlja pomoćni fl (Segment 1) koji se pojavljuje kao uslov u daljem programu (njegovo resetovanje zavisi samo od tastera STOP). Po startovanju aktivira se motor trake za kutije koji nosi kutiju sve do graničnog prekidača kada se motor zaustavlja (Segment 4). Uslov za pokretanje trake sa jabukama je upravo granični prekidač za kutije. Po detekciji kutije kreće pokretna traka sa jabukama (Segment 2). Prisutnost kutije dozvoljava brojaču da preko senzora za jabuke odbroji 10 jabuka i generiše fleg brojača CNT010 koji je uslov za ponovo pokretanje trake sa kutijama (Segment 3). Po pokretanju trake za kutije granični prekidač resetuje brojač koji je opet spreman da odbroji 10 jabuka. Operacije se ponavljaju, sve dok se ne pritisne taster STOP, kada se uslov setovanja flega Flegl gubi. Na slici ispod je dat vremenski dijagram signala linije za pakovanje.



Vremenski dijagram signala linije za pakovanje.



Lestvičasti dijagram u primeru automatizacije pakovanja proizvoda

PRIMENA I PROGRAMIRANJE PROGRAMIBILNIH KONTROLERA U ELEKTROMOTORNIM POGONIMA

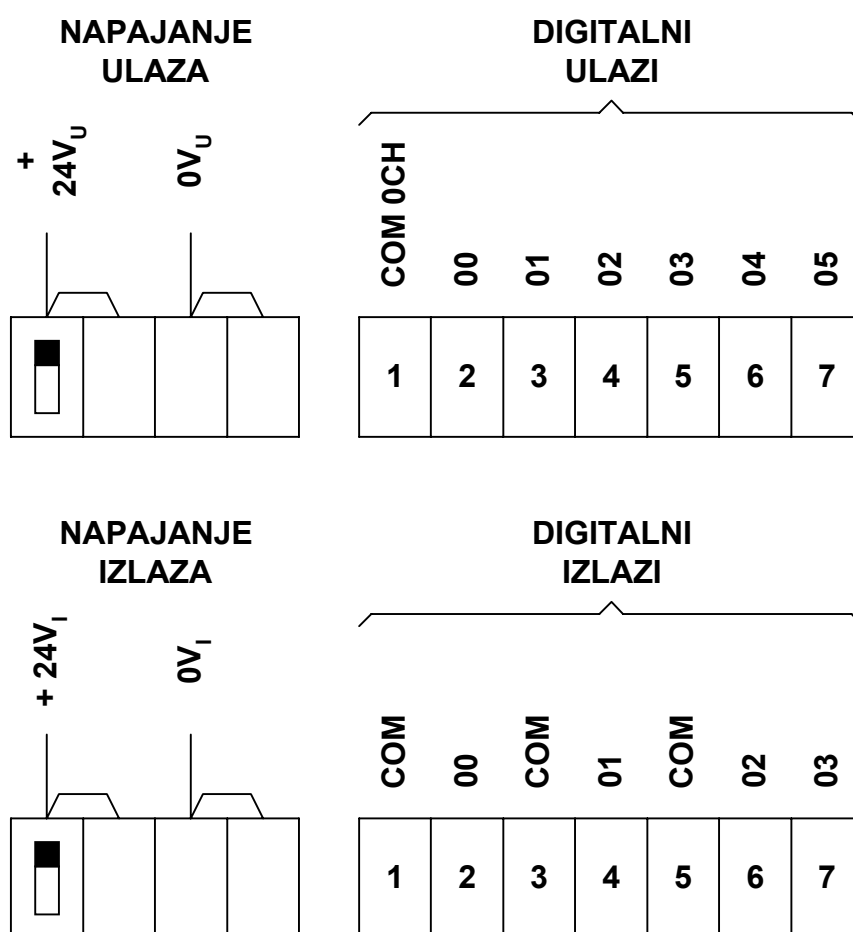
1. Cilj laboratorijske vežbe:

- Odvežbati priložene primere upravljačkih zadataka u elektromotornim pogonima rešenih sa PLC-em CQM1H firme "Omron".
- Odvežbati samostalno dodatno zadate primere upravljačkih zadataka u elektromotornim pogonima.

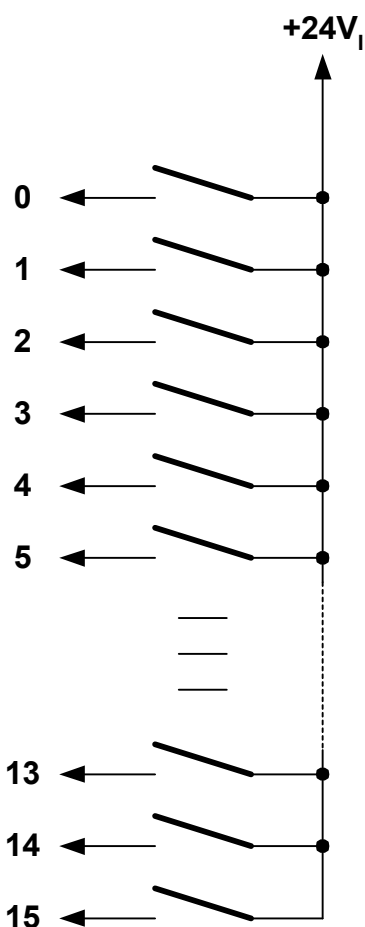
2. Laboratorijska vežba se sastoji od:

- PC računara sa instaliranim softverom SYSWIN za upravljanje PLC kontrolerima firme OMRON;
- Tabla sa PLC kontrolerom CPM1;
- Simulator ulaza i izlaza PLC kontrolera.

2.1. Raspored priključnih stezaljki na tabli sa PLC kontrolerom



2.2. Raspored priključnih stezaljki na simulatoru ulaza i izlaza



Napomena: Izlaze pri simulaciji rešenja zadataka, nije potrebno povezati, njihovo stanje može se pratiti jednostavno preko LED indikatora stanja digitalnih izlaza na samom PLC-u.

3. Primeri upravljačkih zadataka

Primeri počinju od jednostavnijih i idu postupno ka složenijim. Potreban preduslov za programiranje je poznavanje upravljačkog programa SISWIN firme "Omron".

3.1 Jednomotorni jednokvadrantni pogon sa zaštitama

Zaštita od pregrevanja trofaznog asinhronog motora izvođena je ugradnjom davača temperature *PT100* ugrađenim u njegov namotaj. Merenje otpora temperaturnog davača izvodi se kontrolnikom sa relejnim kontaktom. Relejni kontakt je mirni, a raskida se pri prekoračenju temperature namotaja iznad maksimalne vrednosti, određene klasom izolacije namotaja motora. Motor se sem toga štiti i od preopterećenja motornom zaštitnom sklopkom. Upravljanje motorom izvodi se tasterom "START" i "STOP".

Na osnovu zadatka utvrđujemo da upravljanje ima četiri ulaza i jedan izlaz.

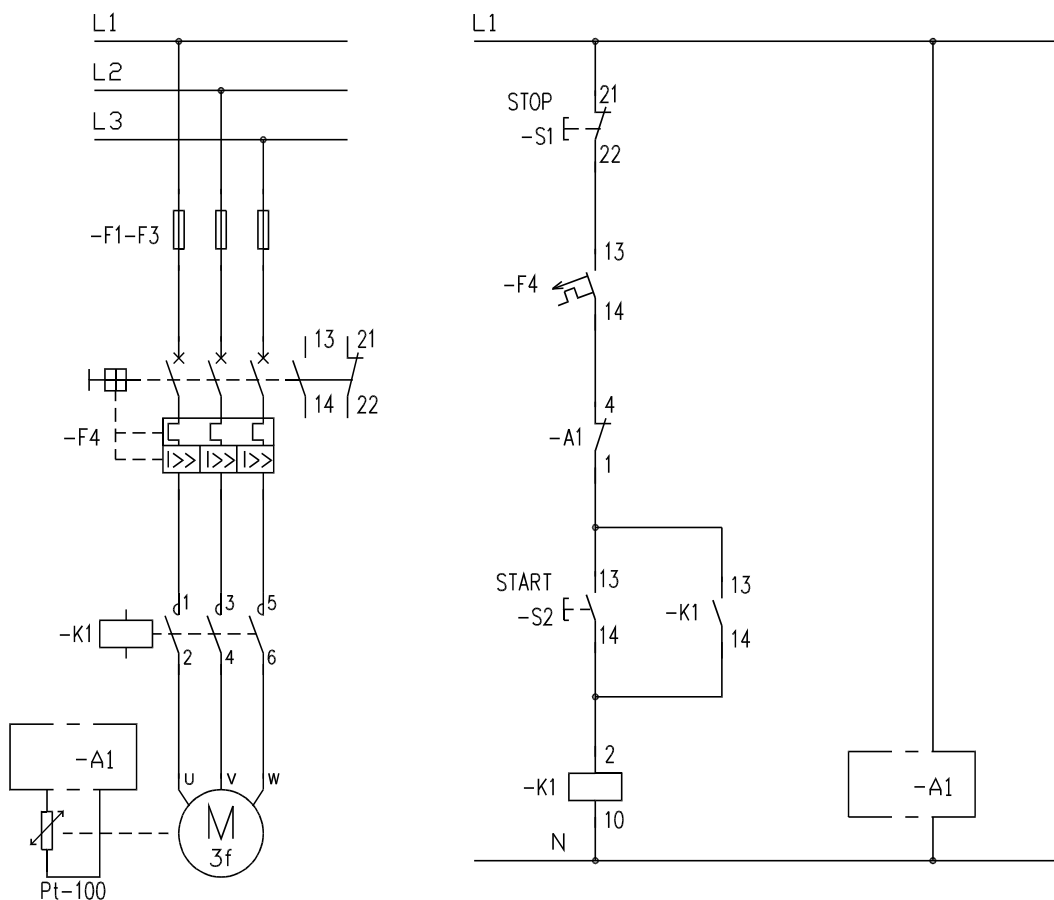
Tablica ulaza i izlaza je:

adrese ulaza		adrese izlaza	
000.00	taster $STOP S_1$	010.00	kontaktor motora K_1
000.01	taster $START S_2$		
000.02	zaštita motora F_4		
000.03	kontrolnik temperature A_1		

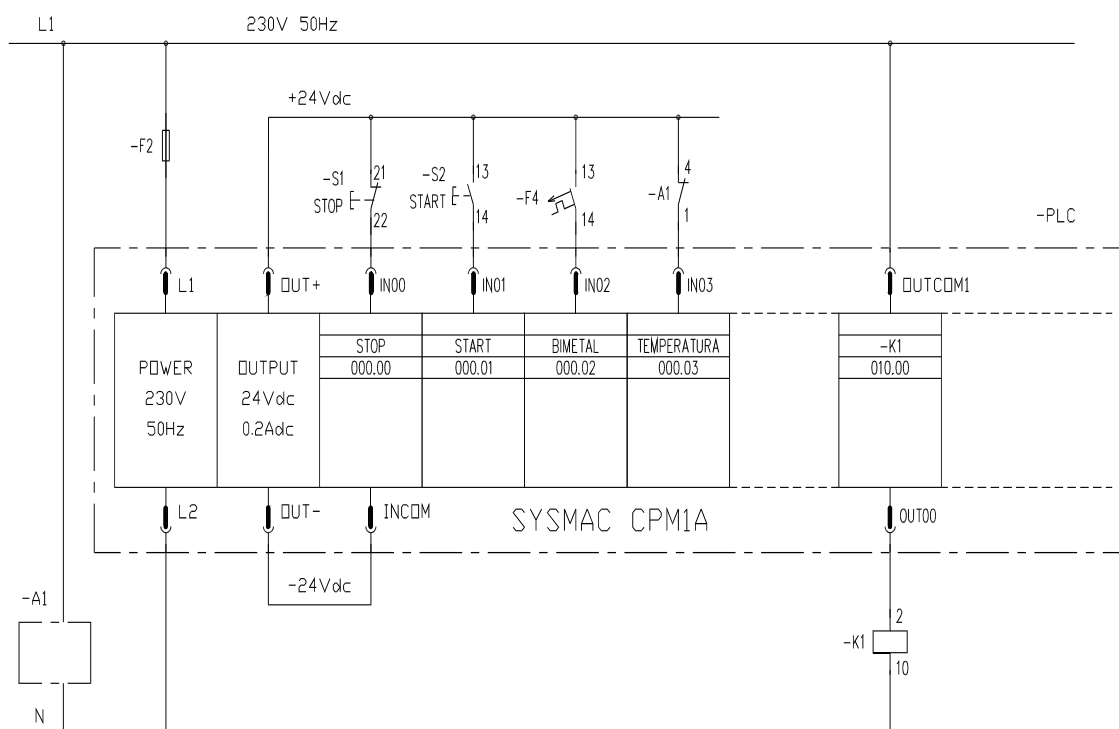
Relejna šema upravljanja bazira se na upravljanju sklopkom za uključanjem motora sa samodržanjem.

Primer samodržanja je čest u konkretnim primerima. Zasniva se na sledećem, mirni kontakti uslova koji isključuju kontaktor (resetuju ga) vezuju se na red sa namotajem kontaktora. Na red se vezuje i radni kontakt uslova koji uključuje kontaktor (setuje ga). Paralelno sa njim vezuje se slobodni radni kontakt kontaktora (kontakt za samodržanje), kao i drugi radni kontakti koji takođe uključuju kontaktor. Ako nije pritisnut nijedan od mirnih kontakata uslova za isključenje, ako se pritisne radni kontakt uslova za uključanje, namotaj kontaktora se aktivira i premošćava uslov (uslove) za uključanje kontaktom za samodržanje. Pošto je uslov za uključanje premošćen on se može i ukinuti, te kontaktor i dalje ostaje aktiviran. Isključenje se vrši aktiviranjem bilo kog mirnog kontakta uslova za isključenje, koji prekida napajanje kontaktora te se raskida i kontakt za samodržanje, te kontaktor ostaje na dalje deaktiviran.

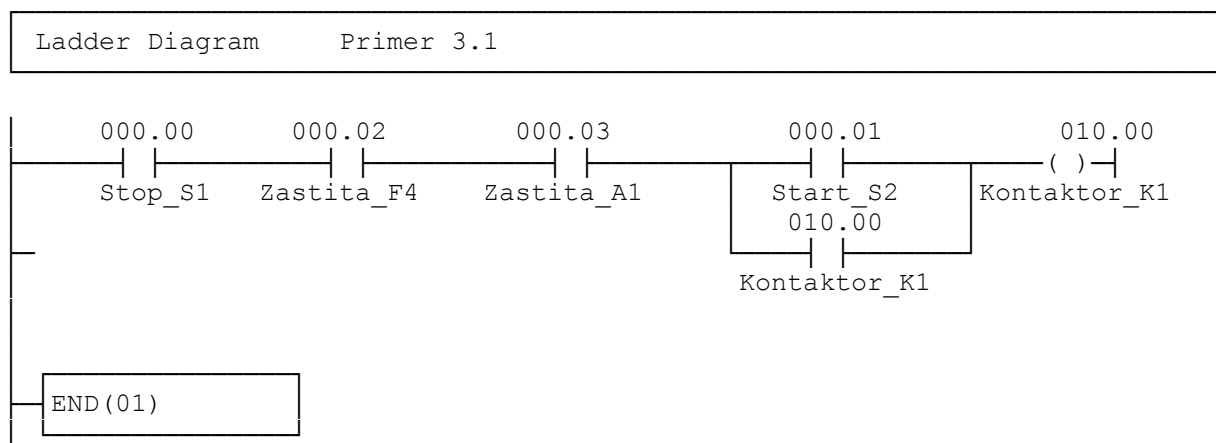
Relejna šema je:



Na osnovu tabele ulaza i izlaza crta se šema povezivanja sa *PLC* uređajem:



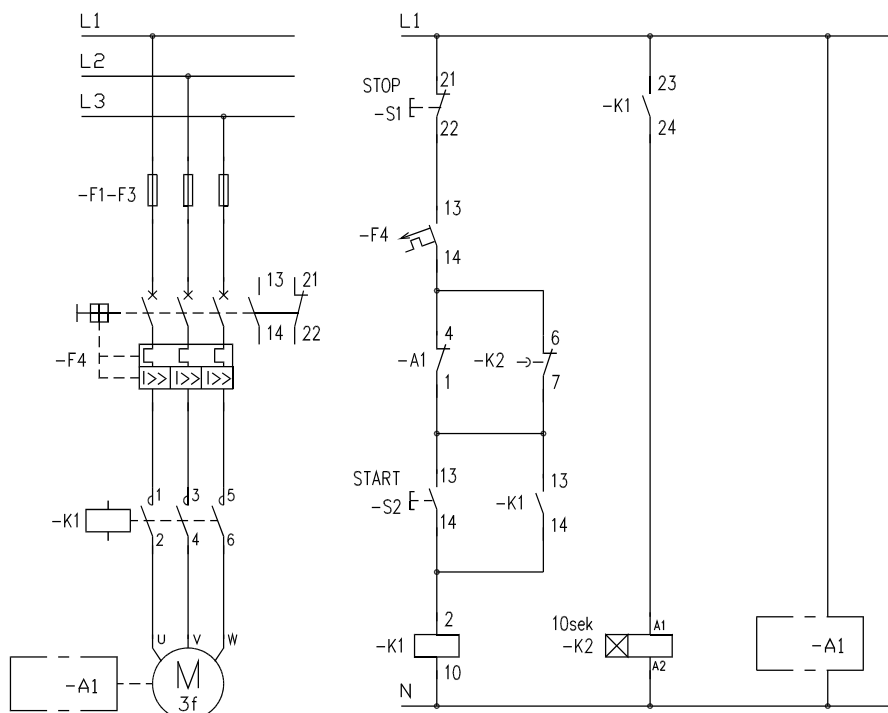
Na osnovu relejne šeme i tablice ulaza i izlaza dobija se lestvičasti dijagram. Primetimo da i mirni kontakti tastera S_I i kontrolnika temperature A_I moraju u lestvičastom dijagramu da se nacrtaju kao radni jer u normalnom stanju provode.



3.2. Jednomotorni pogon sa zaštitom od zaglavljivanja

Zaštita od zaglavljivanja pogona sa trofaznim asinhronim motorom izvodi se kontrolnikom obrtanja postavljenim na njegovu osovinu. Kontrolnik ima na svom izlazu mirni kontakt koji se raskida pri padu broja obrtaja ispod minimalne vrednosti. Motor se sem toga štiti i od preopterećenja motornom zaštitnom sklopkom. Upravljanje motorom izvodi se tasterom "START" i "STOP".

Relejna šema, slična je šemi iz prethodnog primera, data je na sledećoj slici. Vremenski rele K_2 pri startovanju premošćuje kontakt kontrolnika obrtanja u trajanju od 10 [s], sve do dostizanja nominalne brzine obrtanja.

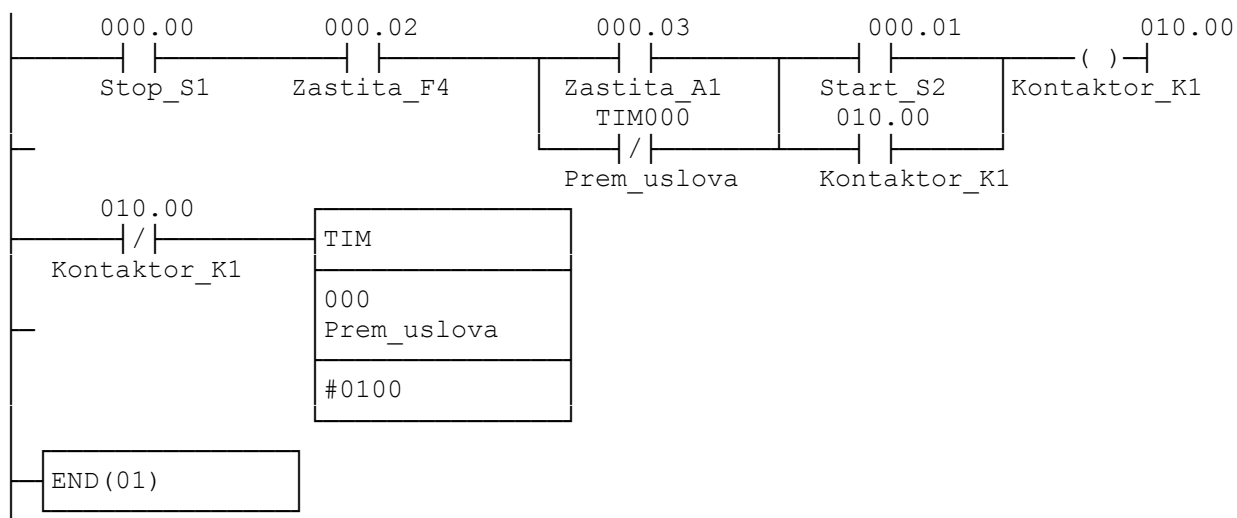


Na osnovu relejne šeme dobijamo tabelu ulaza i izlaza:

adrese ulaza		adrese izlaza	
000.00	taster <i>STOP S₁</i>	010.00	kontaktor motora <i>K₁</i>
000.01	taster <i>START S₂</i>		
000.02	zaštita motora <i>F₄</i>		
000.03	kontrolnik obrtanja <i>A₁</i>		

Na osnovu tabele ulaza i izlaza dobija se šema povezivanja sa *PLC* uređajem ista kao i u prethodnom primeru. Na osnovu uslova zadatka dobija se konačno lestvičasti dijagram:

Ladder Diagram Primer 3.2



3.3. Jednomotorni pogon sa promenom smera obrtanja

Promena smera obrtanja trofaznog asinhronog motora izvodi se promenom redosleda faza napona napajanja motora. Upravljanje motorom izvodi se tasterima "START-DESNO", "START-LEVO" i "STOP". Motor se sem toga štiti i od preopterećenja motornom zaštitnom sklopkom.

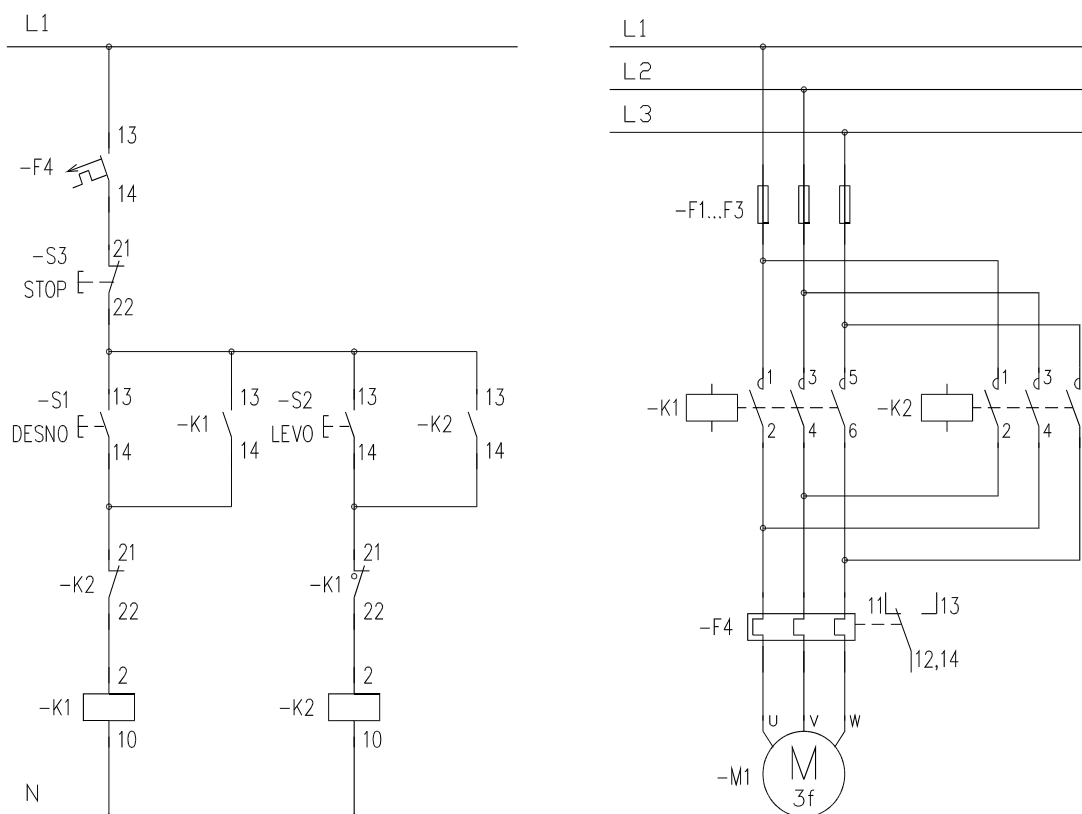
Tablica ulaza i izlaza je:

adrese ulaza		adrese izlaza	
000.00	taster <i>START-DESNO</i> S_1	010.00	kontaktor za desno K_1
000.01	taster <i>START-LEVO</i> S_2	010.01	kontaktor za levo K_2
000.02	taster <i>STOP</i> S_3		
000.03	zaštita motora F_4		

Relejna šema, slična je šemama iz prethodnih primera, data je na sledećoj slici.

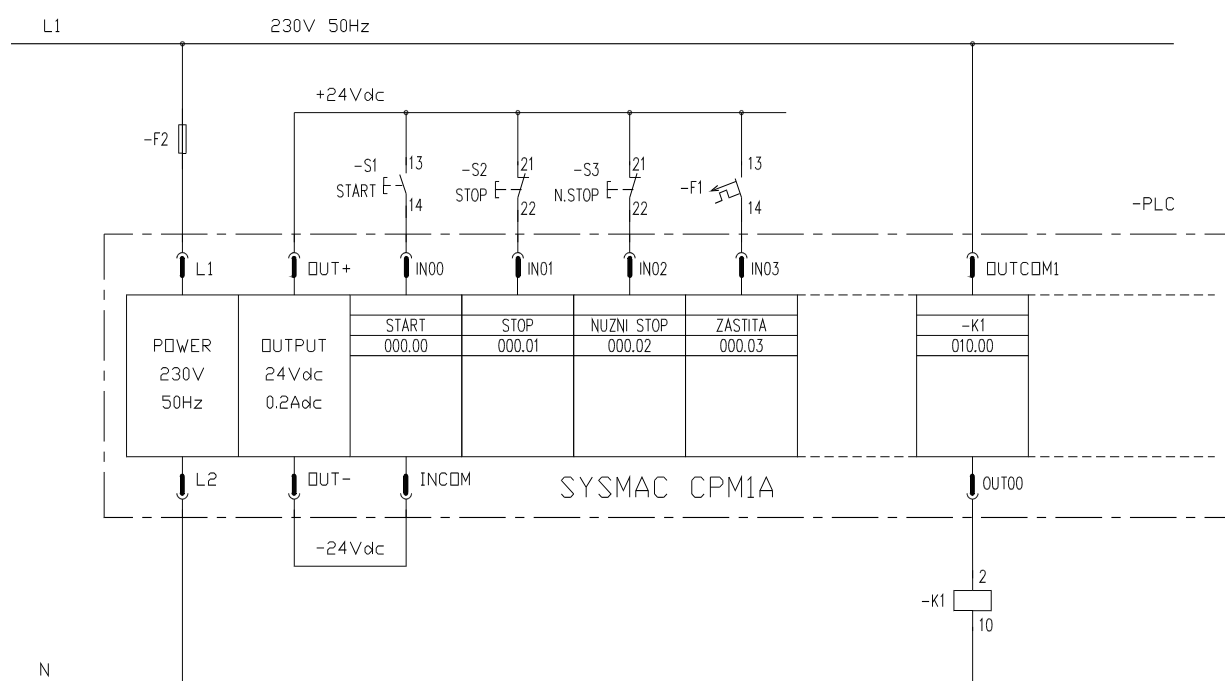
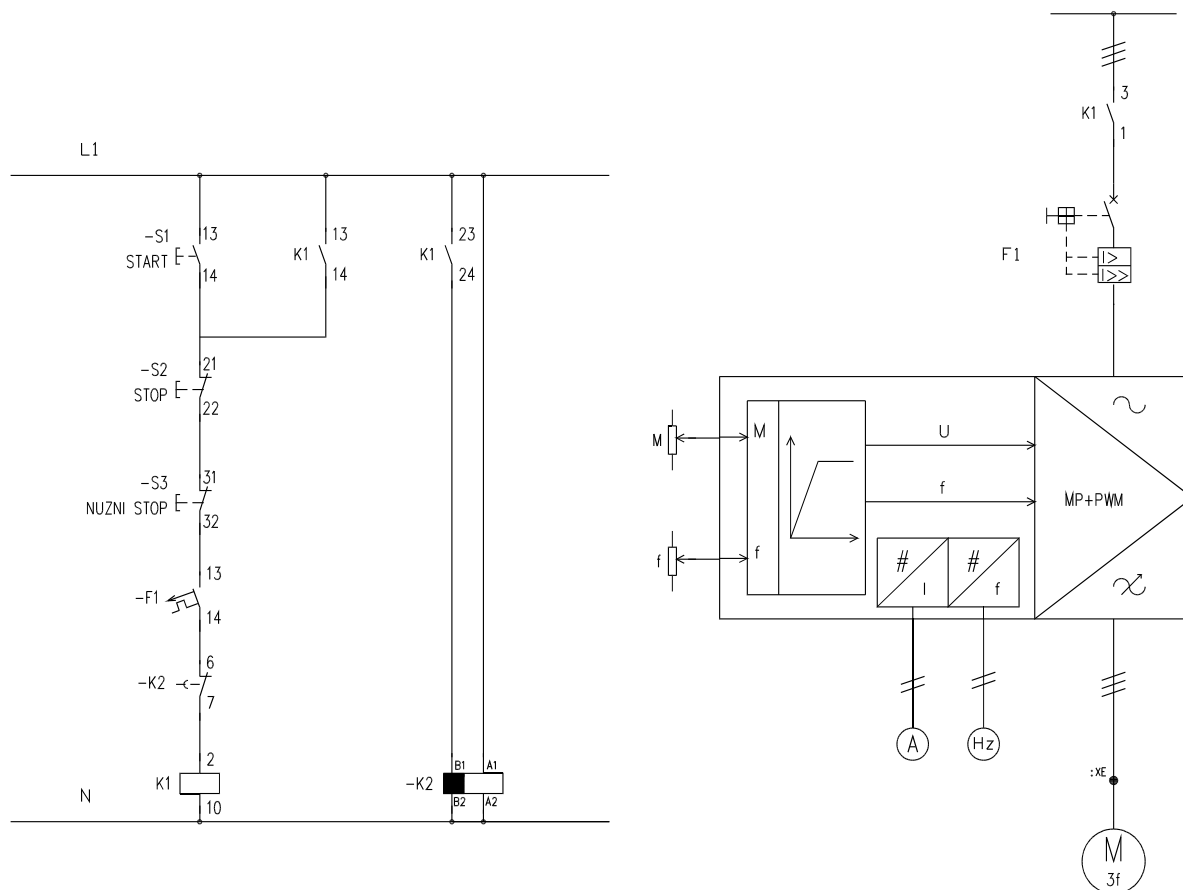
Promena smera obrtanja motora izvodi se spajanjem dva kontaktora K_1 i K_2 paralelno između mreže i izvoda motora. Ako se uključi kontaktor K_1 motor se okreće u desnom smeru a ako se uključi kontaktor K_2 motor se okreće u levom smeru. Jasno je da pri tome nikad ne smeju biti uključena oba kontaktora jer u tom slučaju nastaje kratak spoj.

Šema je izvedena sa dve grane u kom se obezbeđuje samodržanje pojedinačnih sklopki posle pritiska na taster koji zahteva određeni smer. pri tome mirni kontakti sklopki su stavljeni u grane samodržanja sklopke za suprotan smer, da bi se onemogućilo uključjenje obe sklopke istovremeno. Samodržanje se raskida ili pritiskom na taster "STOP" ili po proradi motorne zaštitne sklopke.



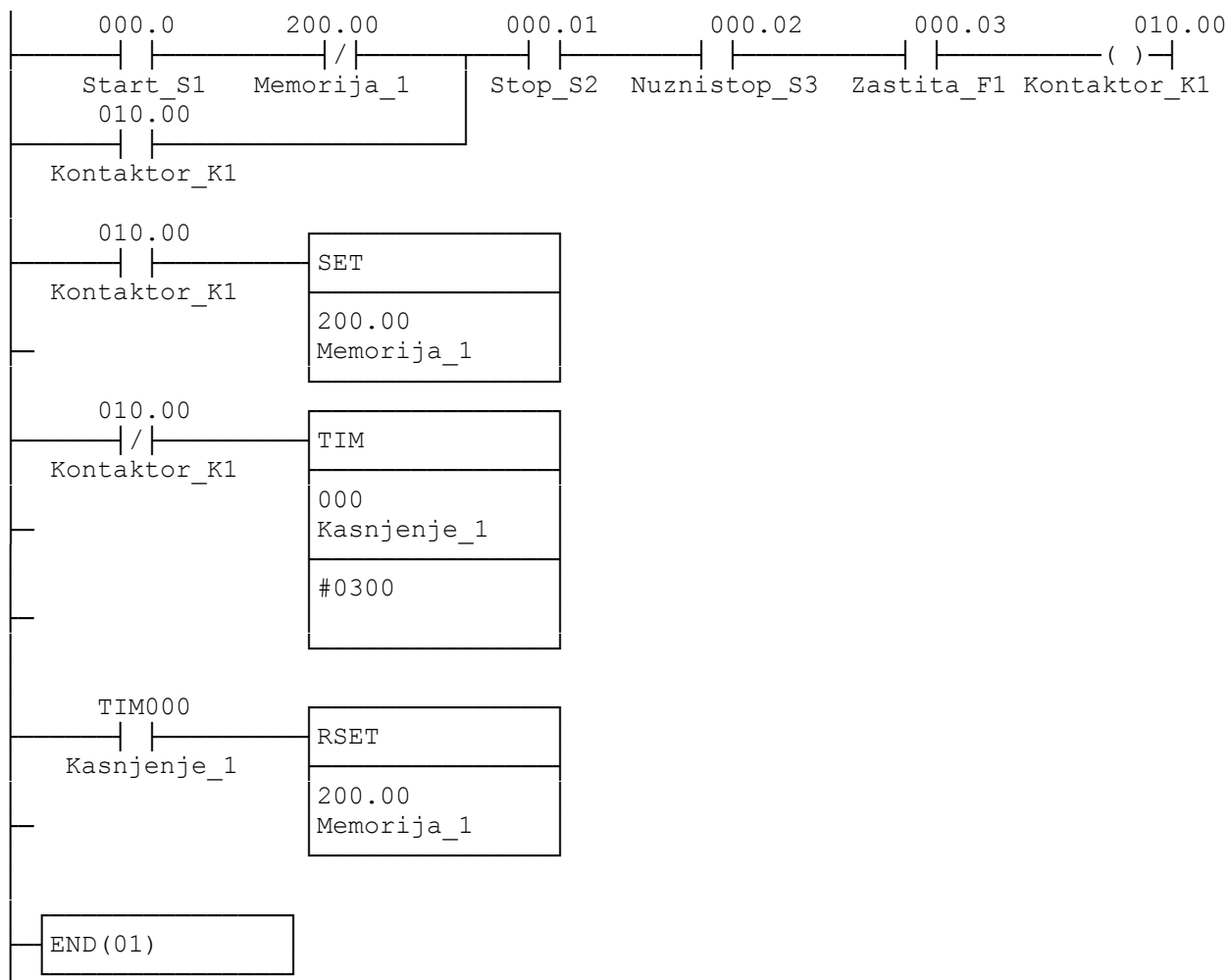
Na osnovu tabele ulaza i izlaza crta se šema povezivanja sa *PLC* uređajem:

adrese ulaza		adrese izlaza	
000.00	taster $START S_1$	010.00	kontaktor pretvarača K_1
000.01	taster $STOP S_2$		
000.02	taster $NUŽNI STOP S_3$		
000.03	zaštita motora F_1		



Lestvičasti dijagram se dobija na osnovu relejne šeme, vodeći računa o činjenici da je vremenski rele K_2 sa kašnjenjem pri isključenju.

Ladder Diagram Primer 3.4



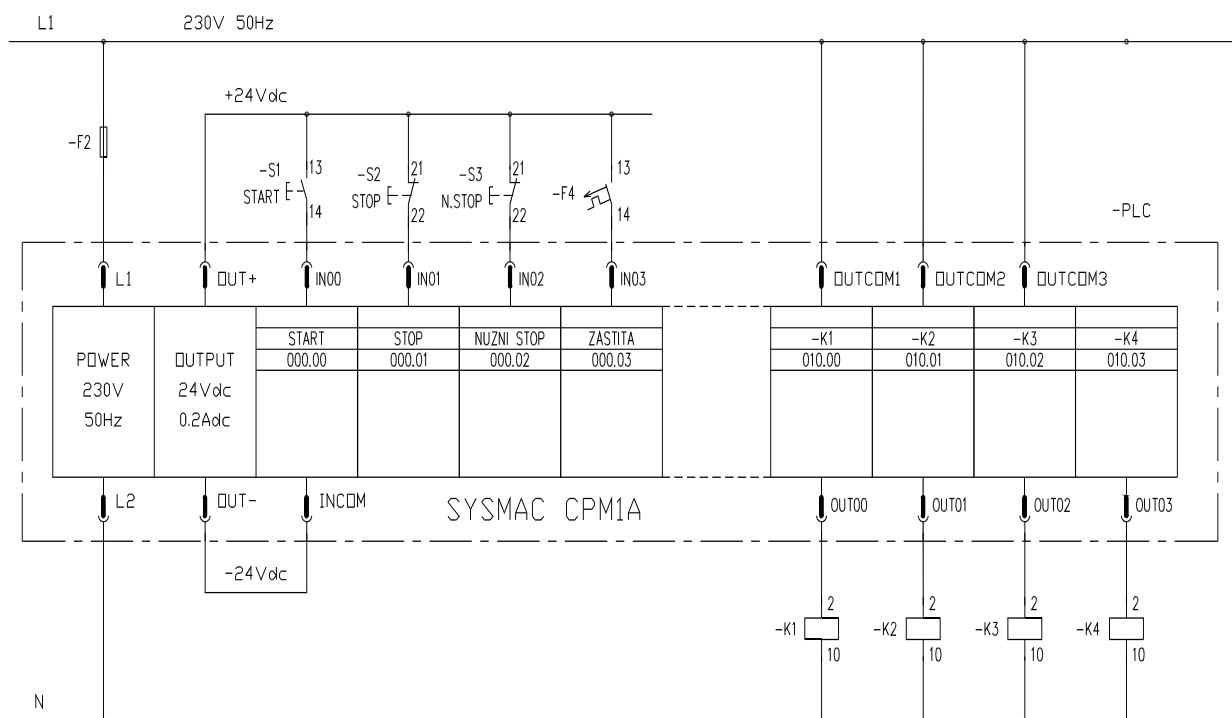
3.5. Upuštač asinhronog motora sa namotanim rotorom.

Elektromotorni pogon je rešen sa asinhronim motorom sa namotanim rotorom. Upuštanje asinhronog motora izvodi se trostepenim rotorskim otpornim upuštanjem sa vremenskim upravljanjem. Trajanje pojedinih vremena upuštanja iznosi pet sekundi. Komande za upravljanje upuštanjem ostvaruju se komandnim tasterima "START", "STOP" i "NUŽNI STOP". Zaštita motora izvedena je sa bimetalom sa dojavnim kontaktom.

Tablica ulaza i izlaza je:

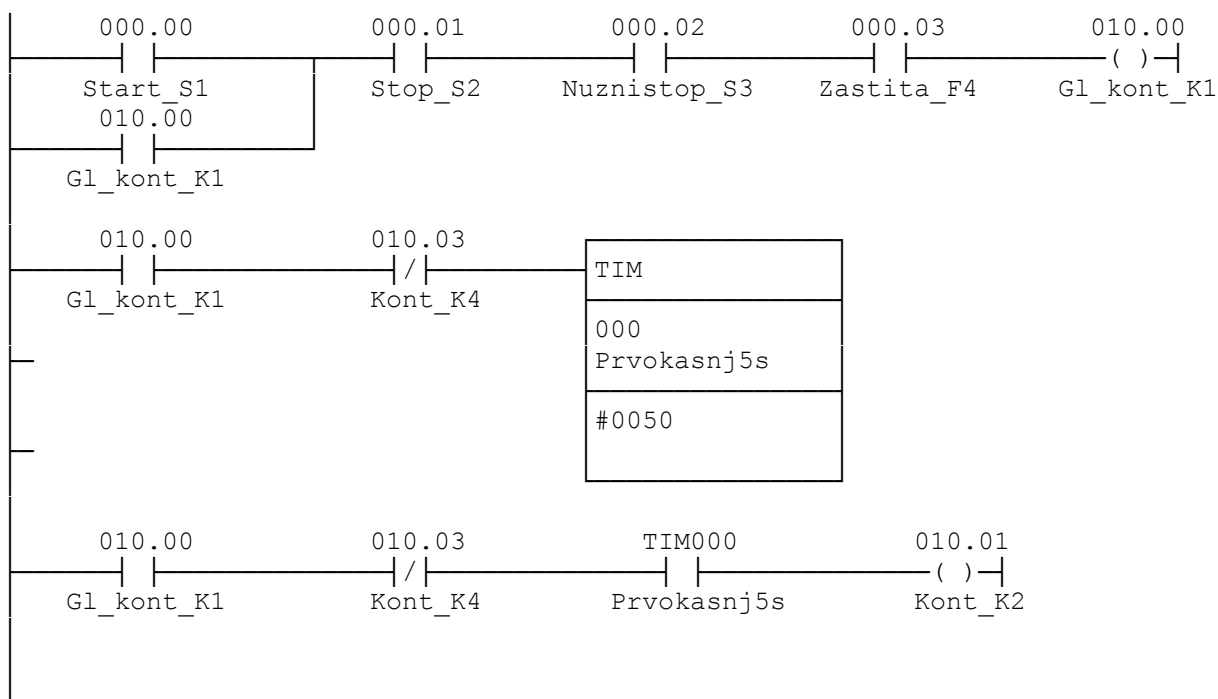
adrese ulaza		adrese izlaza	
000.00	taster $START S_1$	010.00	glavni kontaktor K_1
000.01	taster $STOP S_2$	010.01	kontaktor upuštača K_2
000.02	taster $NUŽNI STOP S_3$	010.02	kontaktor upuštača K_3
000.03	zaštita motora F_4	010.03	kontaktor upuštača K_4

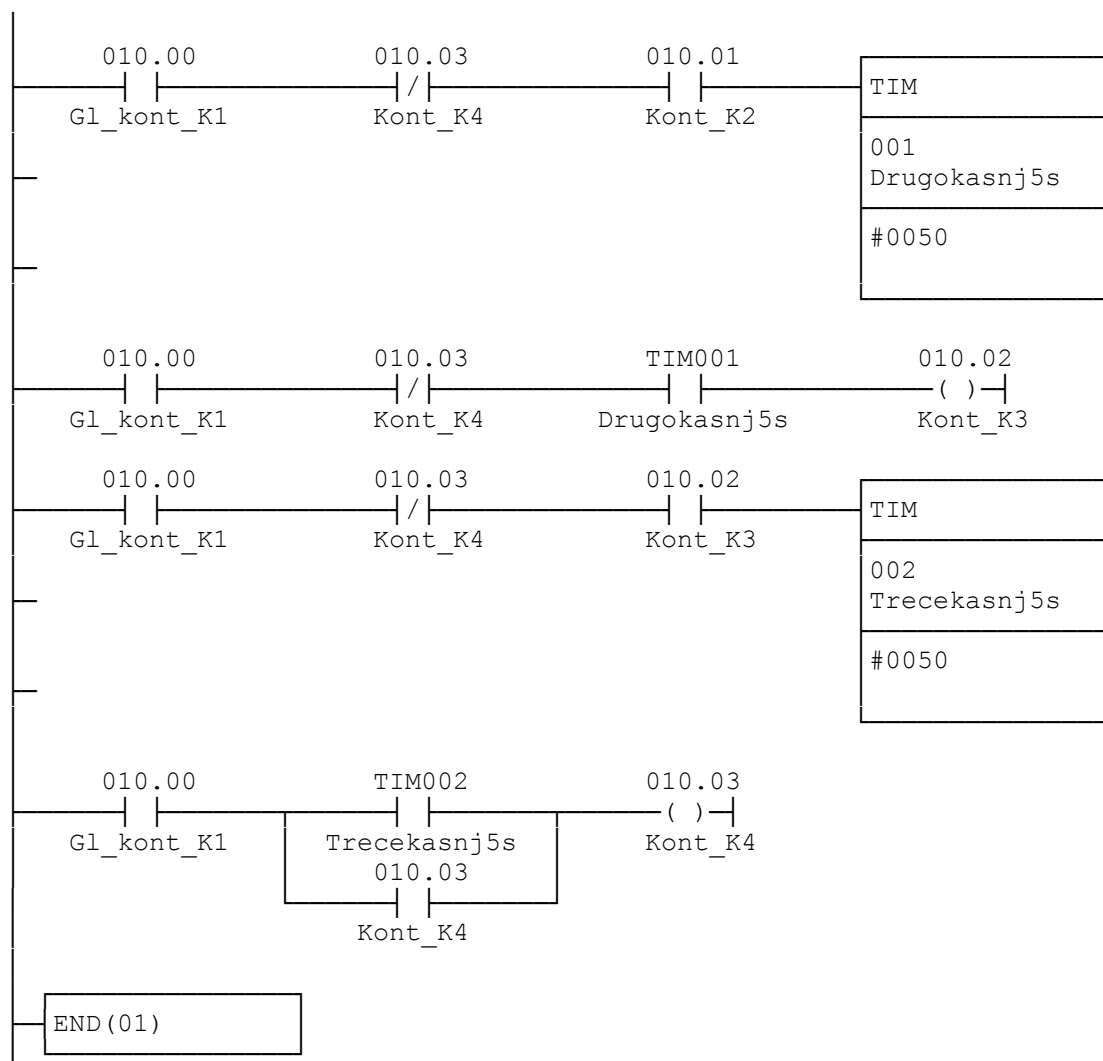
Šema povezivanja spoljašnjih upravljačkih elemenata sa *PLC* uređajem je:



Lestvičasti dijagram *PLC* upravljanja upuštačem dobijen na osnovu relejne šeme i šeme povezivanja je:

Ladder Diagram Primer 3.5





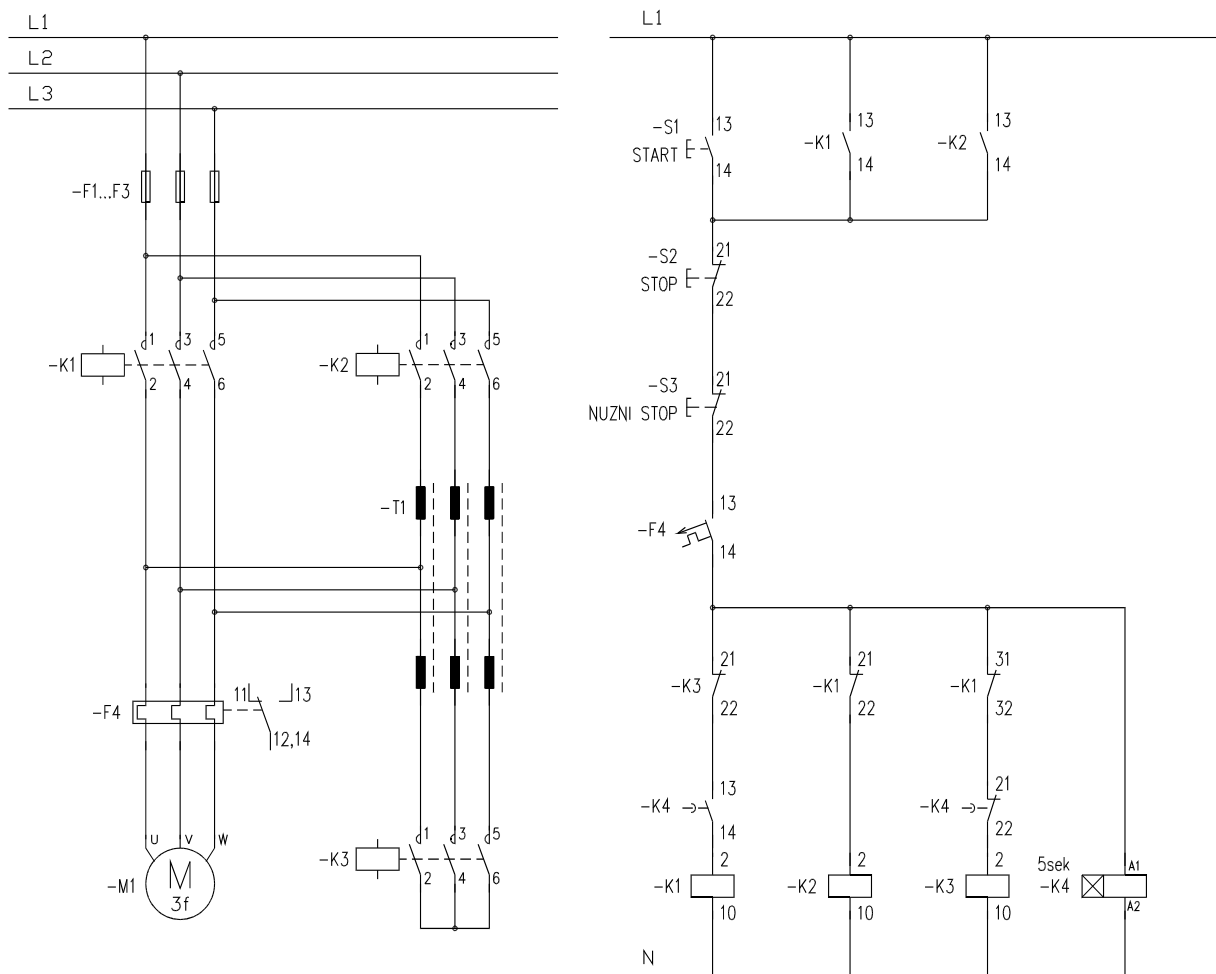
3.6. Autotransformatorski upuštač asinhronog motora sa kratkospojenim rotorom.

Elektromotorni pogon je rešen sa asinhronim motorom sa kratkospojenim rotorom. Upuštanje asinhronog motora izvodi se jednostepenim autotransformatorskim upuštačem sa vremenskim upravljanjem. Trajanje upuštanja iznosi pet sekundi. Komande za upravljanje upuštačem ostvaruju se komandnim tasterima "START", "STOP" i "NUŽNI STOP". Zaštita motora izvedena je sa bimetalom sa dojavnim kontaktom.

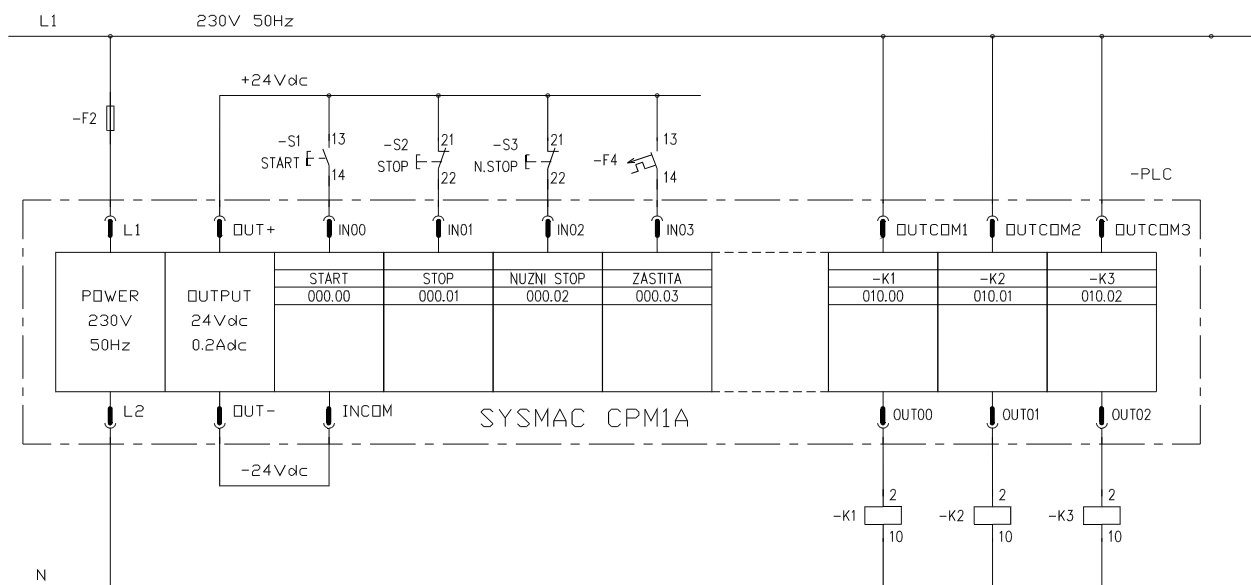
Tablica ulaza i izlaza je:

adrese ulaza		adrese izlaza	
000.00	taster $START S_1$	010.00	glavni kontaktor K_1
000.01	taster $STOP S_2$	010.01	kontaktor upuštača K_2
000.02	taster $NUŽNI STOP S_3$	010.02	kontaktor upuštača K_3
000.03	zaštita motora F_4		

Energetska šema upuštača i relejna šema upravljanja upuštačem prikazana je na sledećoj slici:

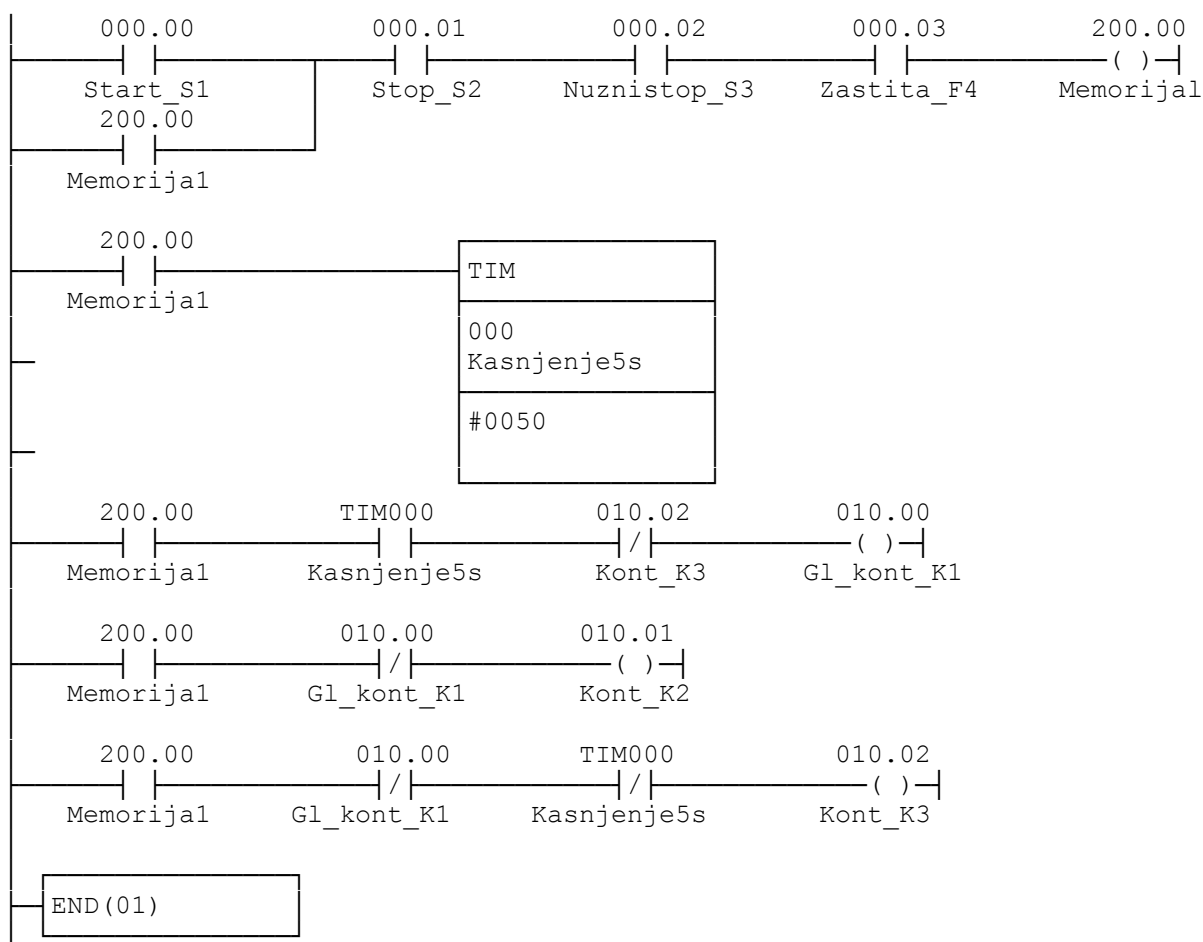


Šema povezivanja spoljašnjih upravljačkih elemenata sa *PLC* uređajem je:



Lestvičasti dijagram *PLC* upravljanja upuštačem dobijen na osnovu relejne šeme i šeme povezivanja je:

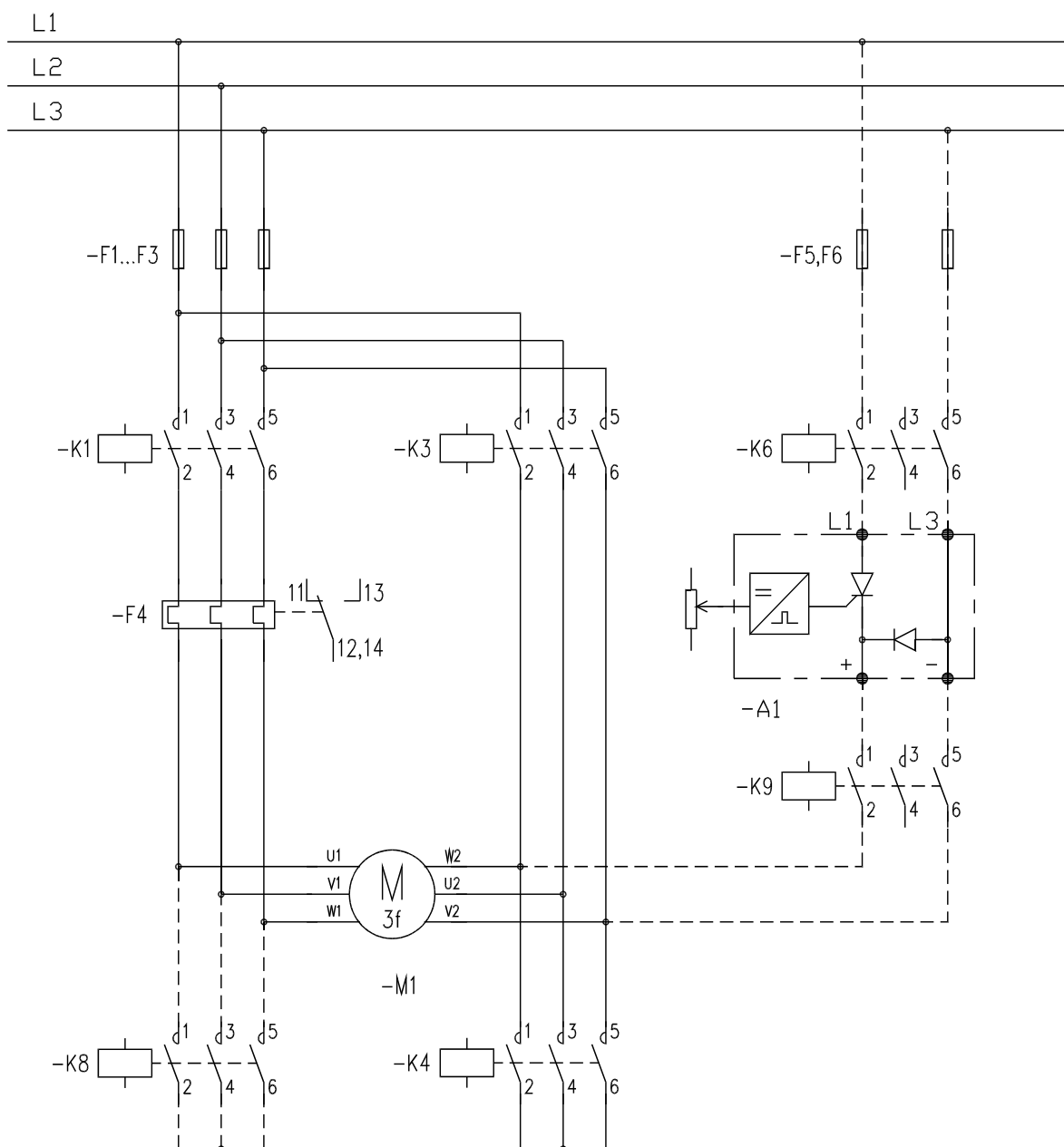
Ladder Diagram Primer 3.6.



3.7. Elektromotorni pogon asinhronog motora sa kočenjem pri isključenju

Elektromotorni pogon mlina rešen je sa asinhronim motorom sa upuštanjem zvezda / trougao. Vremensko trajanje upuštanja iznosi pet sekundi. Pri isključenju pogon se mora što pre zakočiti. Kočenje se izvodi jednosmernom strujom, priključenjem regulisanog ispravljača na namotaj motora u trajanju od pet sekundi, nakon isključenja. Pre isključenju motora sa mreže na njegovim namotajima nastaje prenapon, izazvan nagomilanom elektromagnetnom energijom. Prenapon može izazvati proboj energetskih poluprovodnika u ispravljaču, ako se on odmah priključi po isključenju sa mreže. Zbog toga se po isključenju prvo namotaj motora kratko spaja u trajanju od jedne sekunde, radi pražnjenja elektromagnetne energije, a tek onda uključuje ispravljač. Radi izbegavanja nastanka prenapona na namotajima kod isključenja ispravljača, mora se prvo isključiti njegovo mrežno napajanje a tek onda prekinuti jednosmerni krug napajanja namotaja. Komande za upravljanje pogonom ostvaruju se komandnim tasterima "START" i "STOP". Zaštita motora izvedena je sa bimetalom sa dojavnim kontaktom.

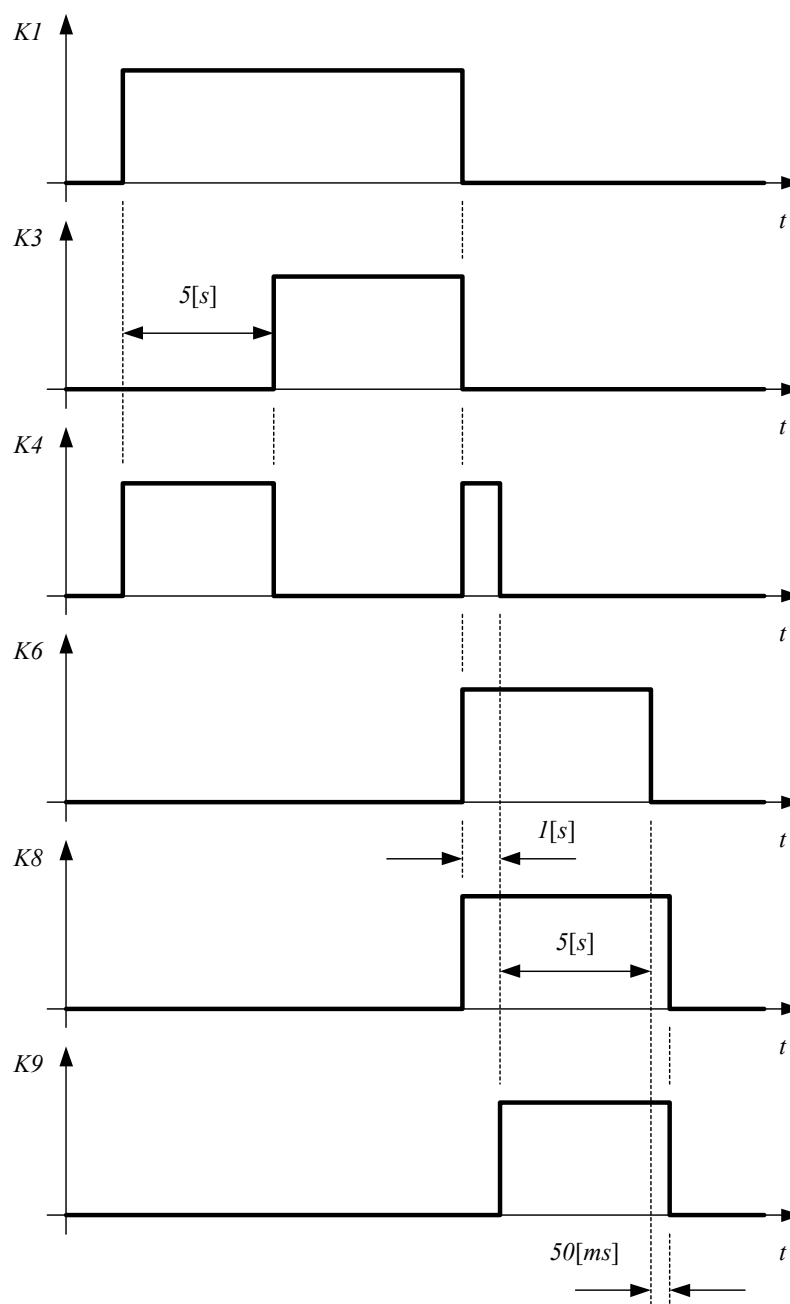
Energetska šema upravljanja pogonom prikazana je na sledećoj slici, gde je deo šeme koji se odnosi na upuštač zvezda / trougao povezan punim linijama, a deo koji se odnosi na kočenje jednosmernom strujom povezan isprekidanim linijama.



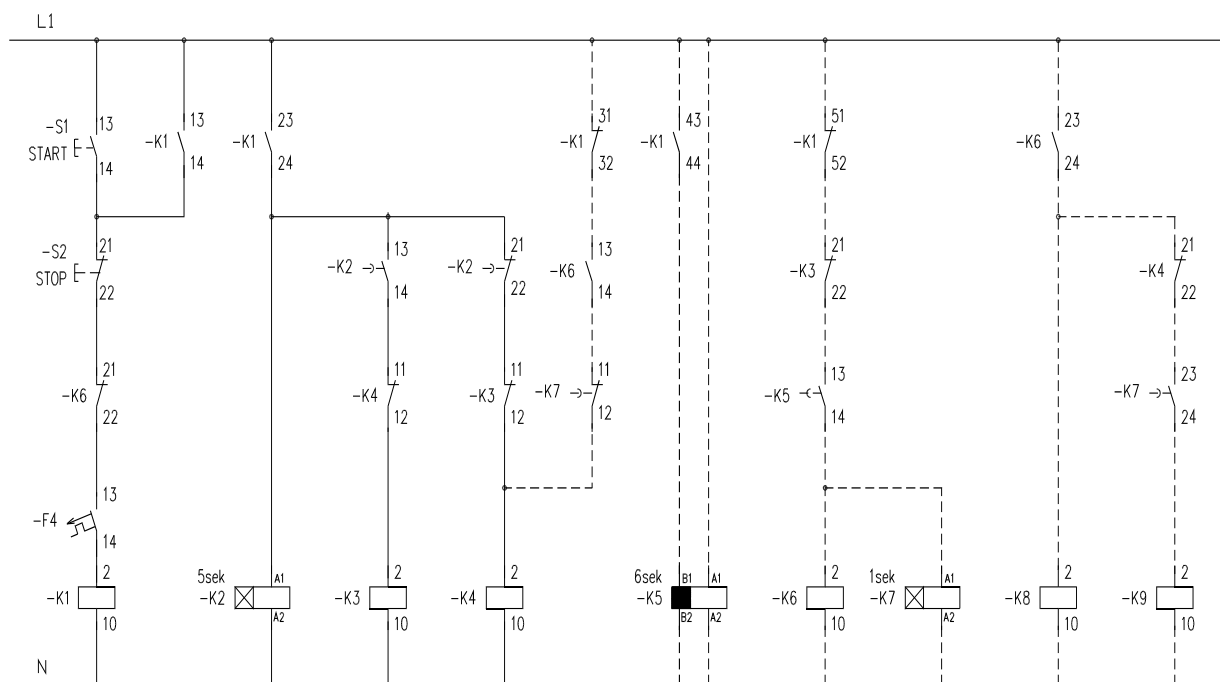
Potreban broj izlaza šest je veći od četiri, koji ima osnovna jedinica *PLC*-a, pa se mora koristiti odgovarajuće proširenje, pa je prema tome tablica ulaza i izlaza:

adrese ulaza		adrese izlaza	
000.00	taster <i>START S₁</i>	010.00	glavni kontaktor <i>K₁</i>
000.01	taster <i>STOP S₂</i>	010.01	kontaktor <i>K₃</i>
000.02	zaštita motora <i>F₄</i>	010.02	kontaktor <i>K₄</i>
		010.03	kontaktor <i>K₆</i>
		011.00	kontaktor <i>K₈</i>
		011.01	kontaktor <i>K₉</i>

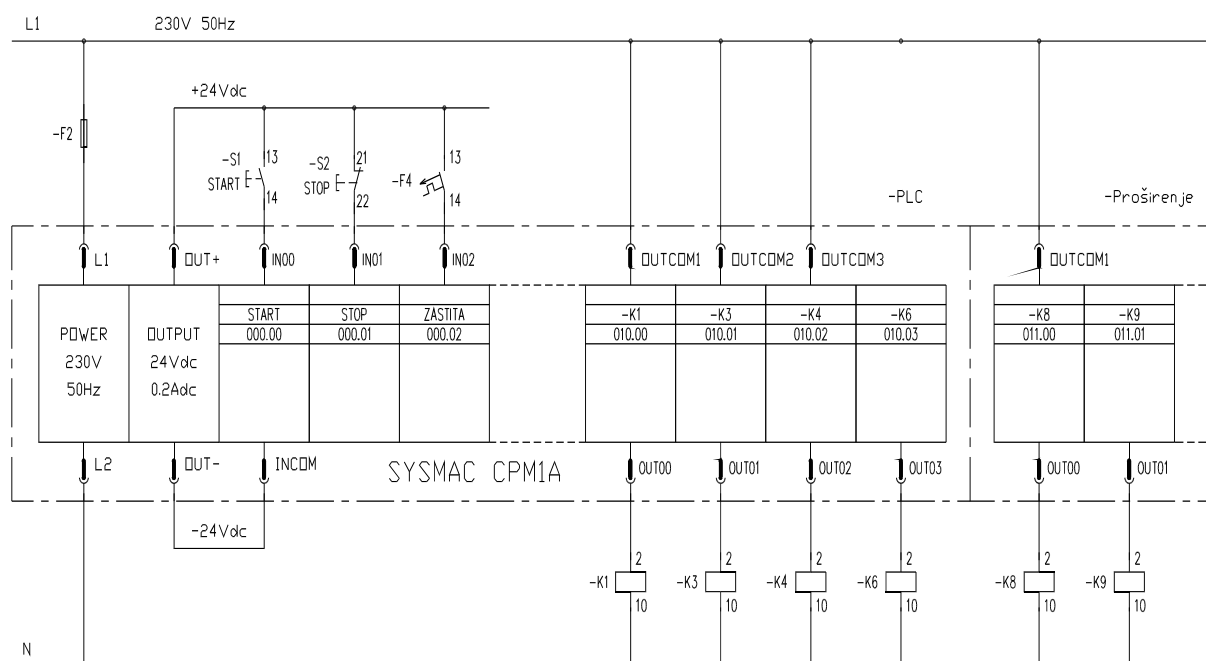
Prema uslovima zadatka crta se dijagram vremenskog procesa uključjenja i isključenja pojedinih energetskih kontaktora:



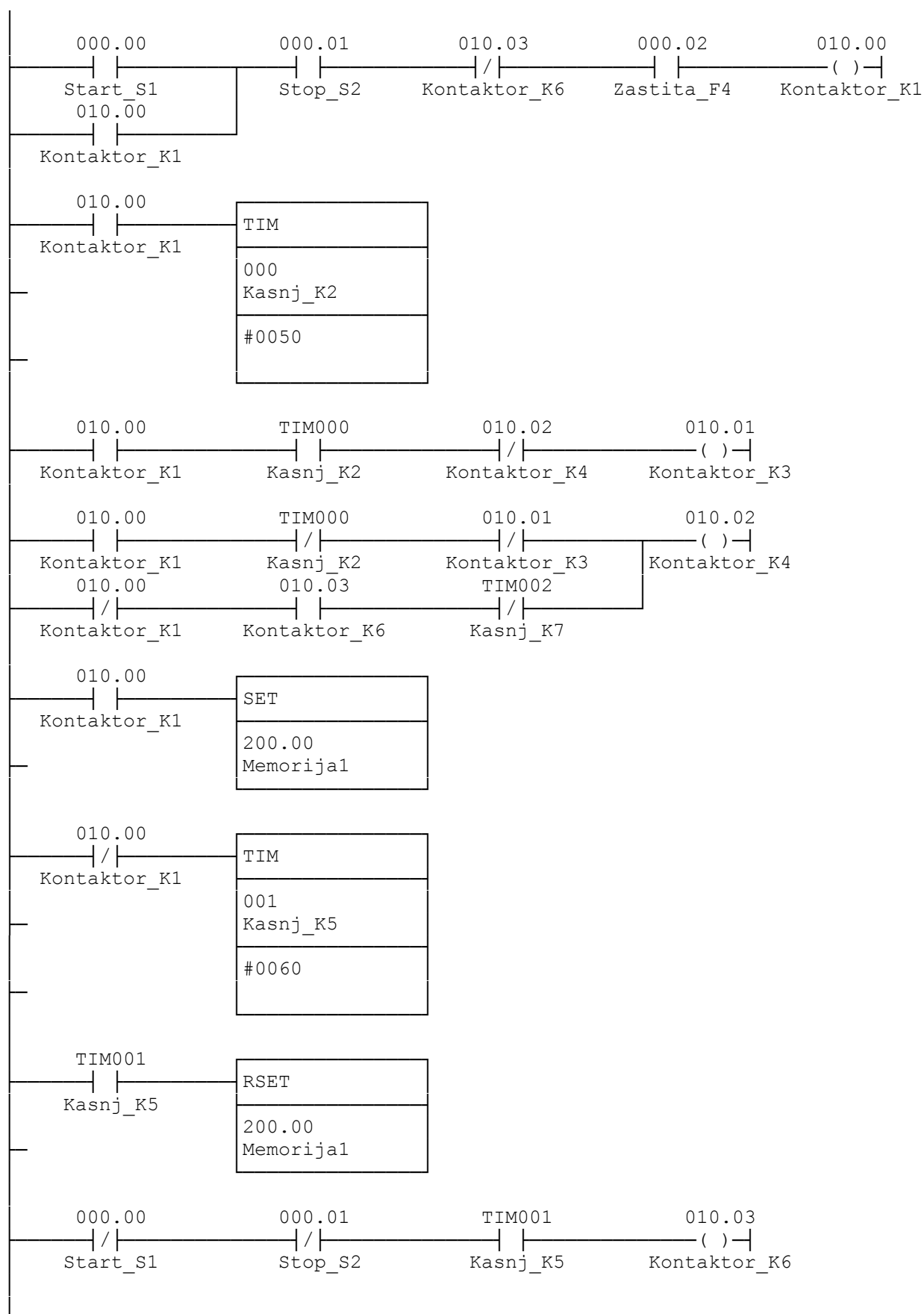
Na osnovu vremenskog dijagrama sintetizuje se relejna šema upravljanja. Posle komande za uključenje odmah se motor povezuje u zvezdu ($-K_1 = -K_4 = \text{uključ.}$), a nakon kašnjenja određenog vremenskim releom $-K_2$ sa kašnjenjem pri uključanju motor se povezuje u trougao ($-K_1 = -K_3 = \text{uključ.}$ $-K_4 = \text{isključ.}$). Pri isključenju procesom kočenja upravlja vremenski rele $-K_5$ sa kašnjenjem pri isključenju, tokom koga se uključuje ispravljač ($-K_6 = \text{uključ.}$), pri čemu se tokom vremena zatezanja vremenskog relea $-K_7$, prvo kratko spaja namotaj ($-K_4 = -K_8 = \text{uključ.}$) a zatim u preostalom vremenu predviđenom za kočenje priključuje ispravljač na namotaj ($-K_8 = -K_9 = \text{uključ.}$ $-K_4 = \text{isključ.}$). Po isteku vremena kašnjenja pri isključenju određenog releom $-K_5$, prvo se isključuje napajanje ispravljača ($-K_6$) a zatim jednosmerni krug ($-K_8$; $-K_9$) nakon kašnjenja koje ostvaruje vreme otpuštanja samog kontaktora (cca. 50 [ms]). To kašnjenje je dovoljno za pražnjenje akumulisane elektromagnetne energije, pa nije potrebno koristiti poseban vremenski rele.

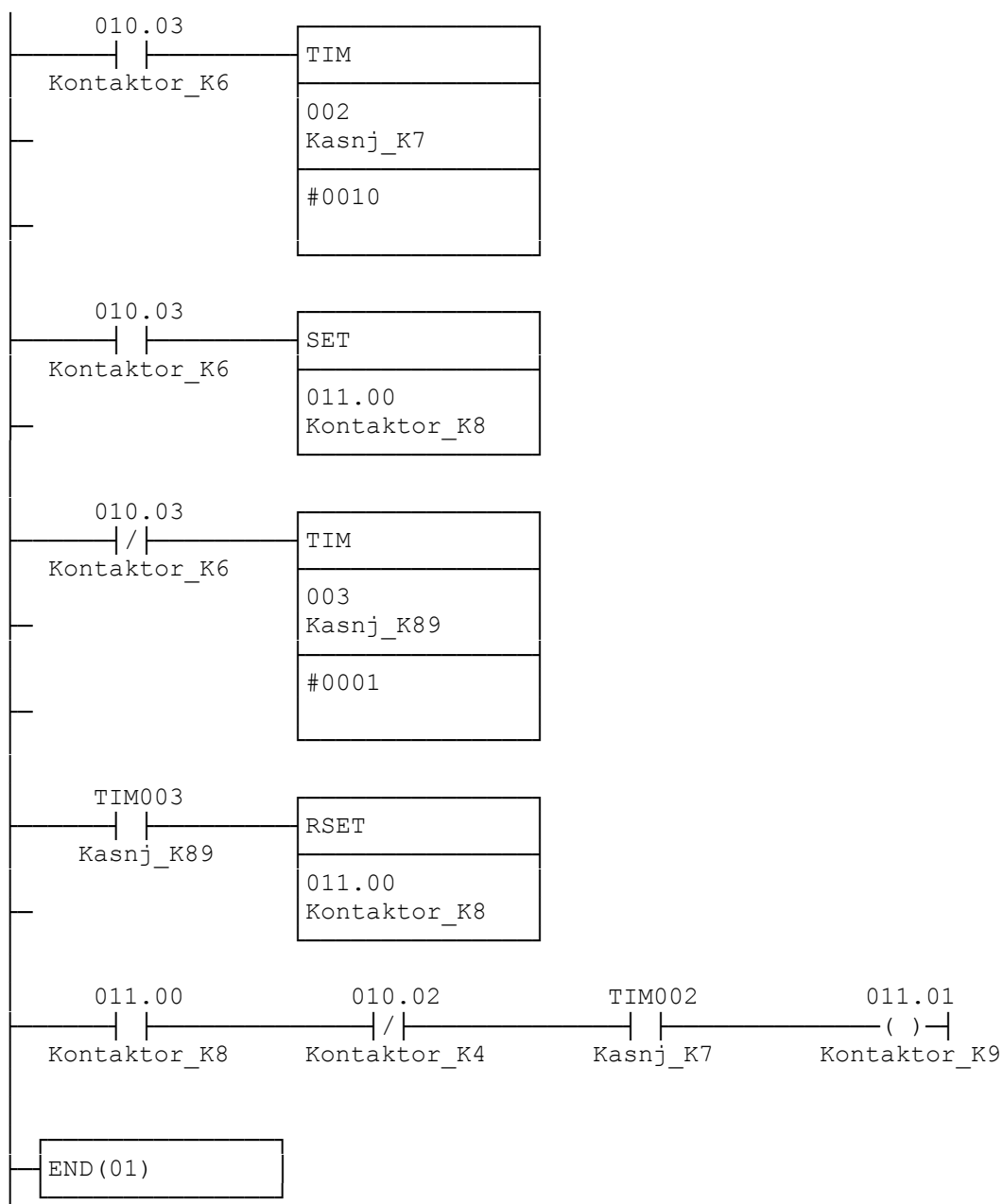


Šema povezivanja spoljašnjih upravljačkih elemenata sa osnovnim *PLC* uređajem i njegovim proširenjem je:



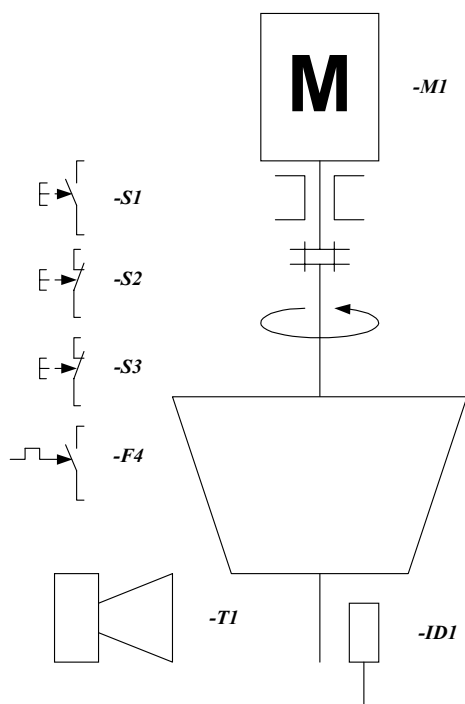
Lestvičasti dijagram *PLC* upravljanja pogonom dobija se na osnovu vremenskog dijagrama, relejne šeme i šeme povezivanja. Pri tome za ostvarenje potrebnog kašnjenja pri isključenju između napajanja i jednosmernog kola ispravljača mora se dodati poseban vremenski član, pošto fiktivni relei u programa *PLC* upravljanja ne ostvaruju kašnjenje otvaranja kontakata.





3.8 Pogon centrifuge sa zaštitom od debalansa

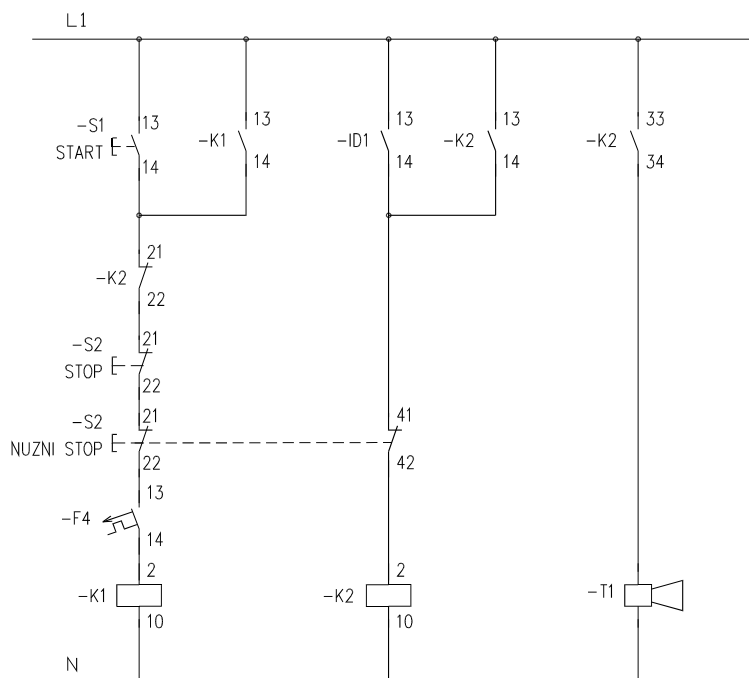
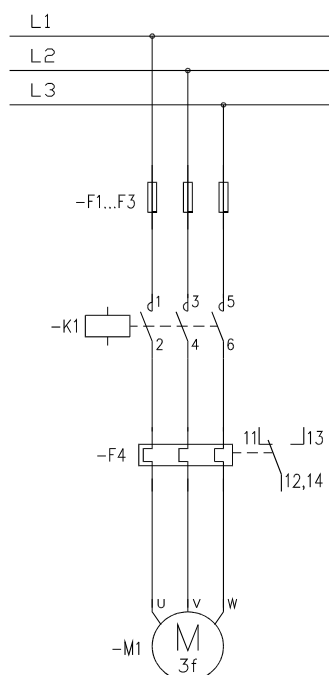
Vertikalna centrifuga štiti se od debalansa induktivnim davačem postavljenim u blizini donjeg kraja osovine centrifuge. Pojava debalansa isključuje motor centrifuge i pali trubu za signalizaciju opasnosti. Komande za upravljanje pogonom su ostvarene tasterima: “Start”, “Stop” i “Nužni stop”. Taster “Nužni stop” ujedno služi i za kvitiranje zvučnog signala trube, tako da treba obezbediti da motor centrifuge ne može da krene dok se ne zaustavi posle pojave debalansa, odnosno kvitira zvučni signal. Zaštita motora izvedena je sa zaštitnom sklopkom sa dojavnim kontaktom.



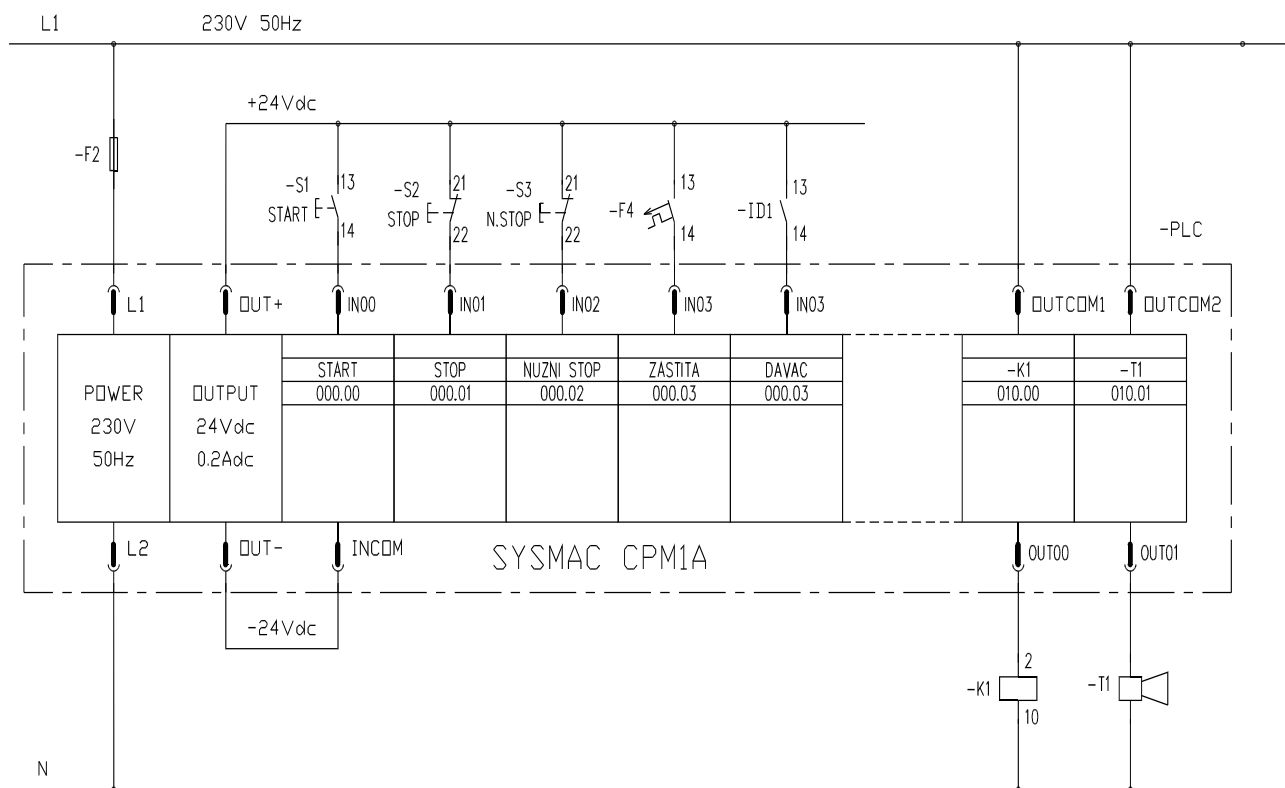
Tablica ulaza i izlaza je:

adrese ulaza		adrese izlaza	
000.00	taster $START S_1$	010.00	kontaktor motora K_1
000.01	taster $STOP S_2$	010.01	truba T_1
000.02	taster $NUŽNI STOP S_3$		
000.03	zaštita motora F_4		
000.04	induktivni davač ID_1		

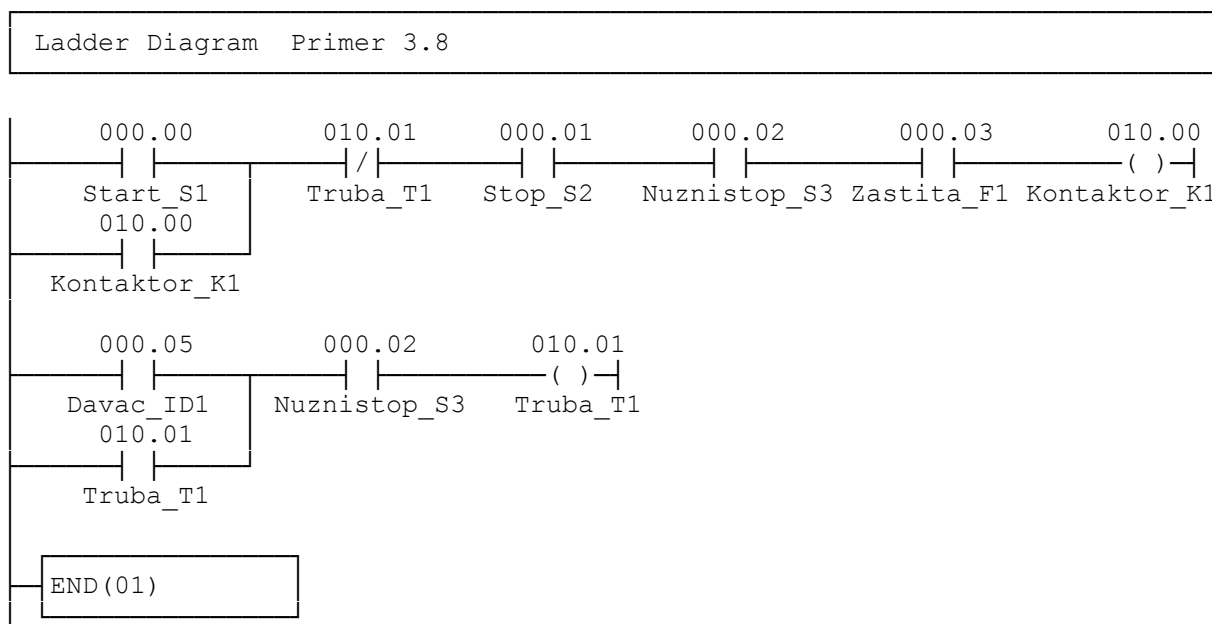
Relejna šema upravljanja crta se na osnovu logičkih uslova zadatka:



Na osnovu tabele ulaza i izlaza crta se šema povezivanja sa *PLC* uređajem:



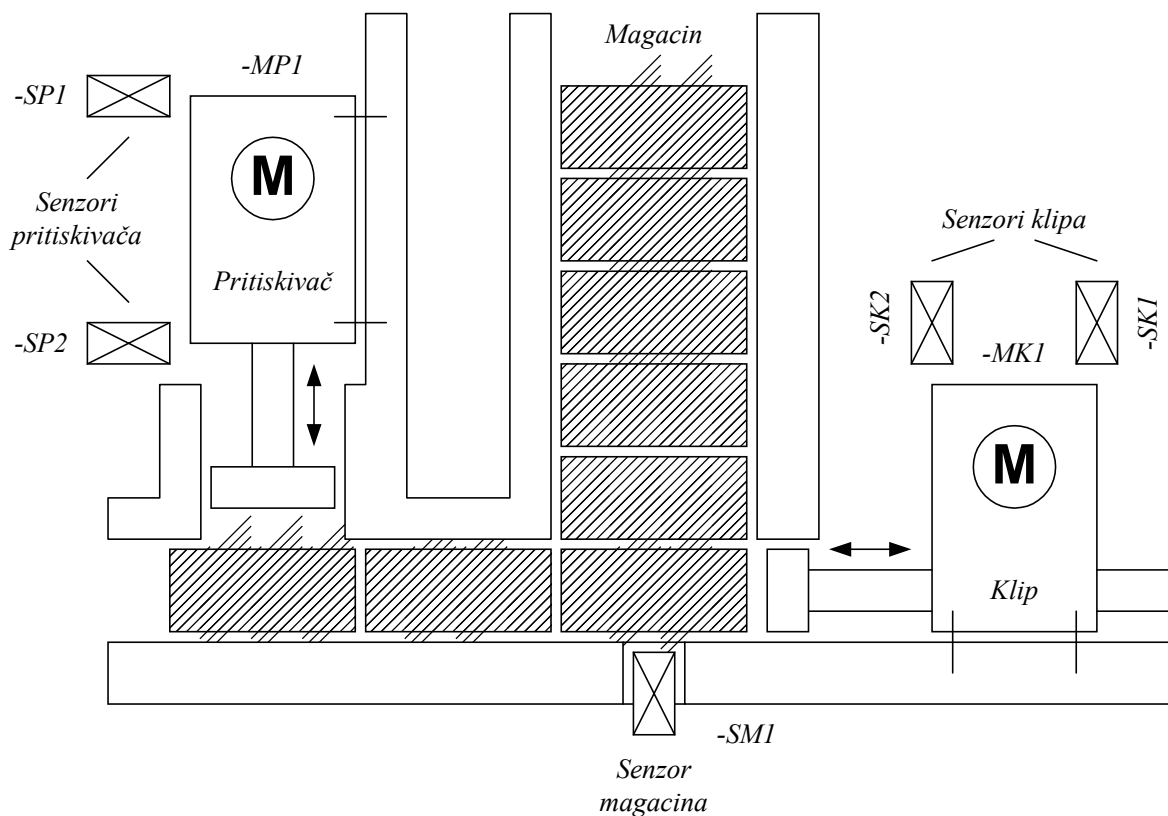
Lestvičasti dijagram se dobija na osnovu relejne šeme:



3.9. Višemotorni pogon štamparske mašine

Razraditi rešenje upravljanja štamparskom mašinom za štampanje na mehaničkim delovima, sa automatskim prinošenjem delova.

Tehnološka šema mašine data je na slici.

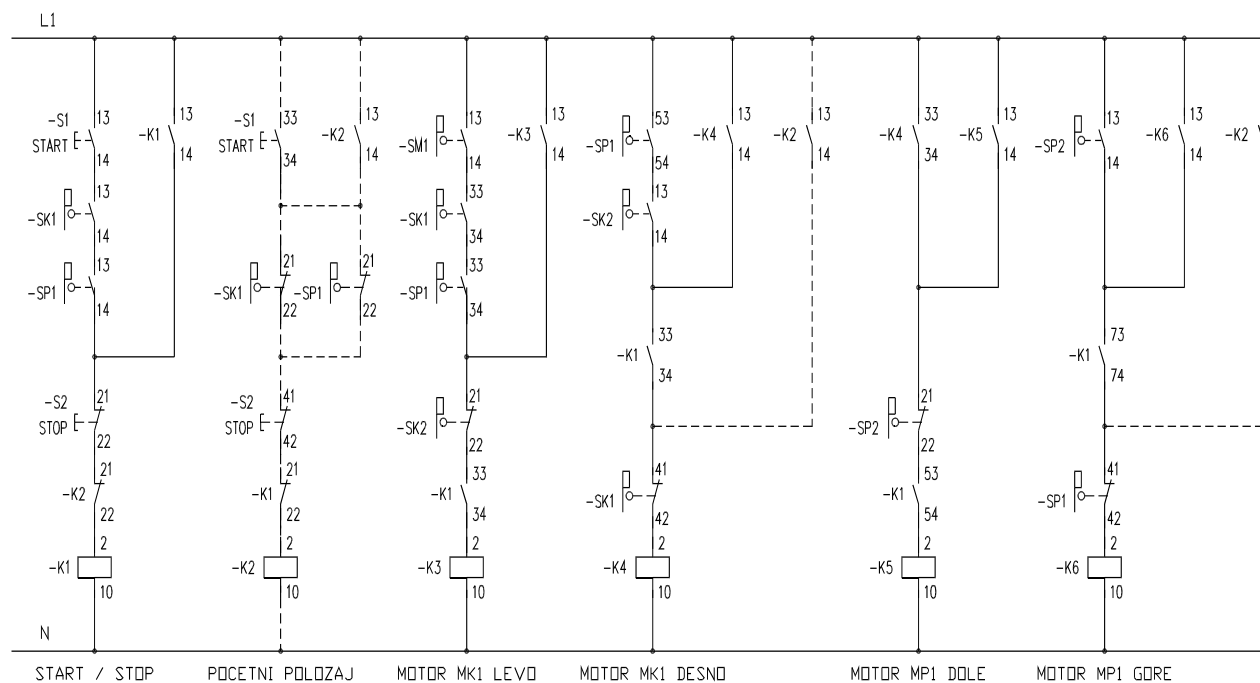
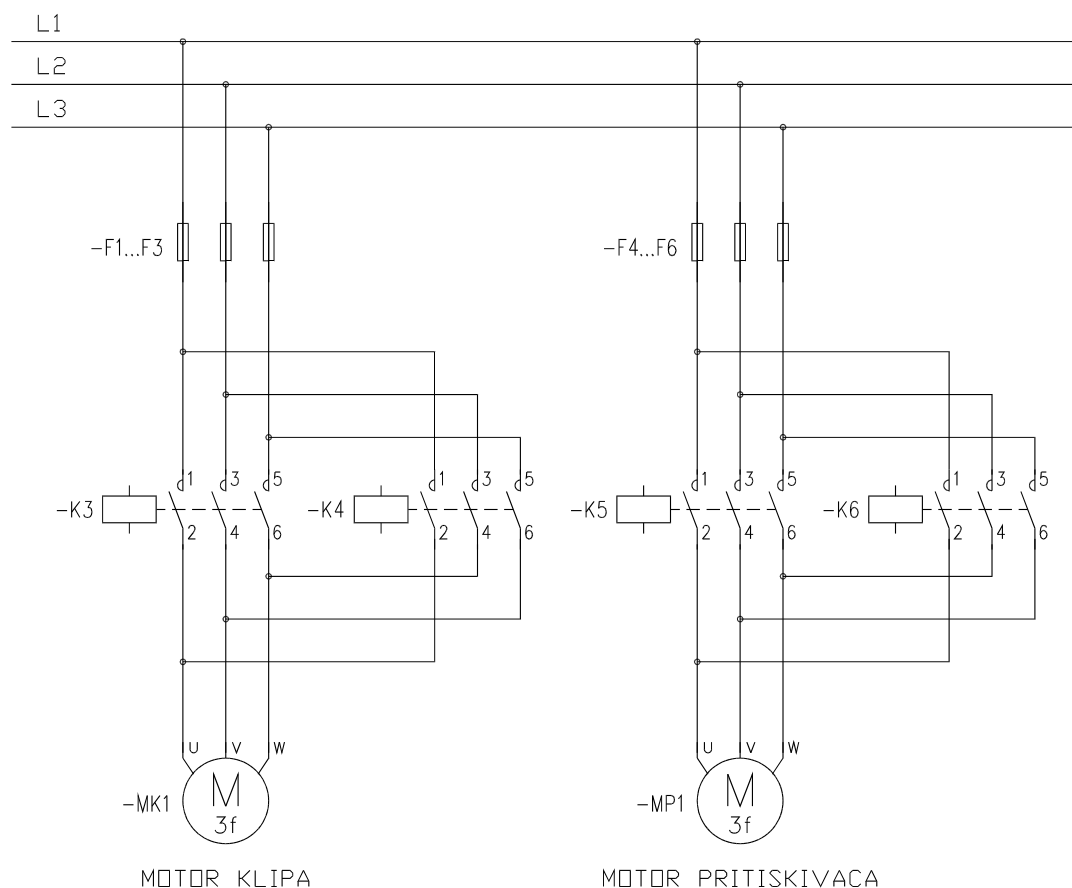


Pokretanje klipa za dodavanje delova i pritiskivača za štampanje izvedeno je sa trofaznim asinhronim motorima. Detekcija krajnjih položaja pritiskivača, klipa i detekcija popunjenosti magacina izvedena je krajnjim induktivnim prekidačima. Komande za upravljanje pogonom su ostvarene tasterima: "Start" i "Stop".

Tablica ulaza i izlaza je:

adrese ulaza		adrese izlaza	
000.00	taster $START S_1$	010.00	motor klipa $MK_1 LEVO$
000.01	taster $STOP S_2$	010.01	motor klipa $MK_1 DESNO$
000.02	senzor $MAGACIN SM_1$	010.02	motor pritiskivača $MP_1 DOLE$
000.03	senzor pritiskivača $GORE SP_1$	010.03	motor pritiskivača $MP_1 GORE$
000.04	senzor pritiskivača $DOLE SP_2$		
000.05	senzor klipa $DESNO SK_1$		
003.00	senzor klipa $LEVO SK_2$		

Energetska šema upravljanja motora, predstavljena je na sledećoj slici, a obezbeđuje sa po dva kontaktora smeštena u vodove za napajanje motora njihovo pokretanje u oba smera, radi pokretanja klipa levo - desno i pritiskivača dole – gore.



Relejna šema upravljanja crta se na osnovu logičkih uslova zadatka, te obezbeđuje ciklični red operacija štamparske mašine. Jedan ciklus rada sastoji se od sledećih operacija, koje se izvode redom:

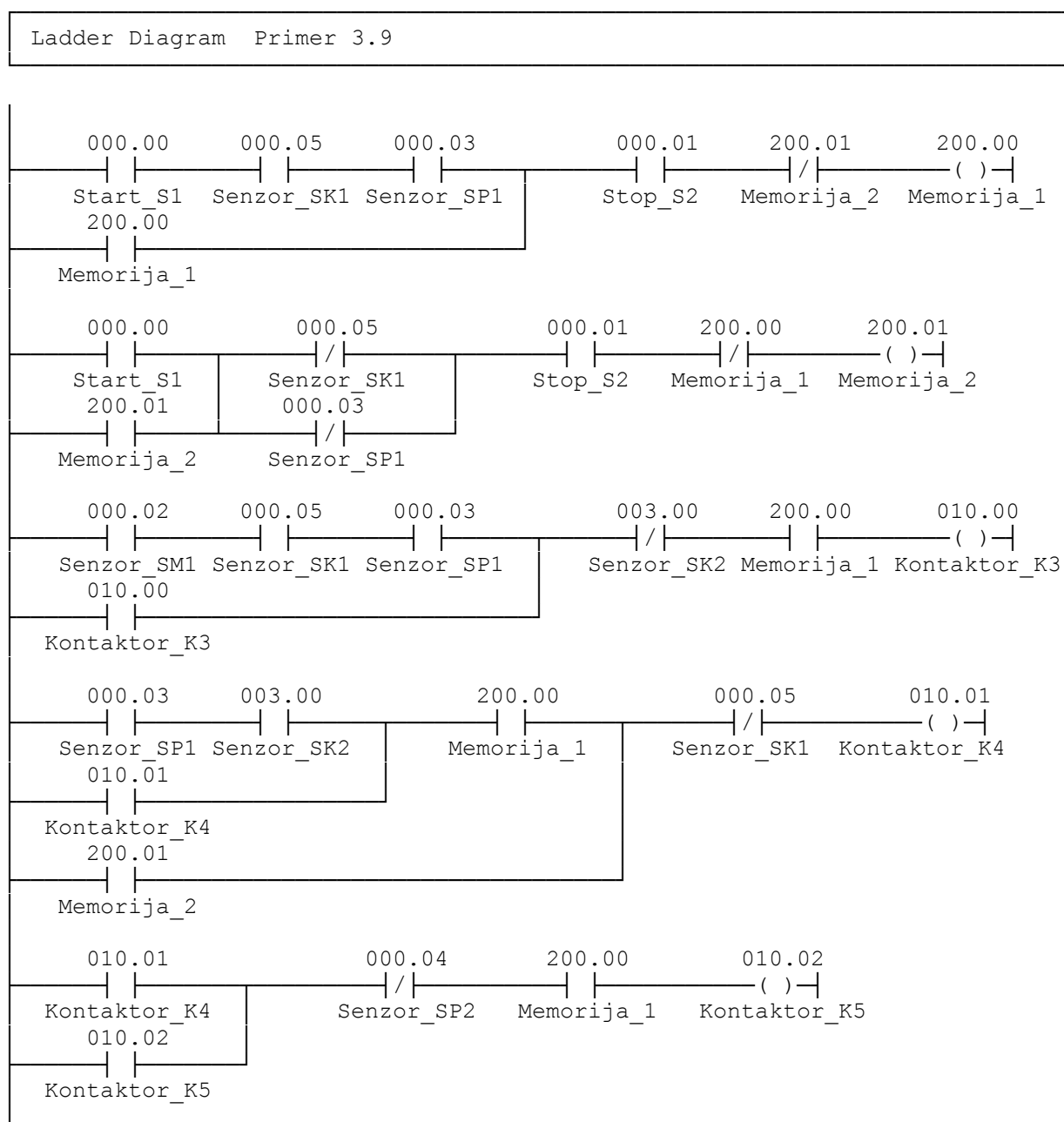
- Pomeranje klipa u levo iz levog krajnjeg položaja do krajnjeg desnog položaja, radi pomeranja dela do pritiskivača.

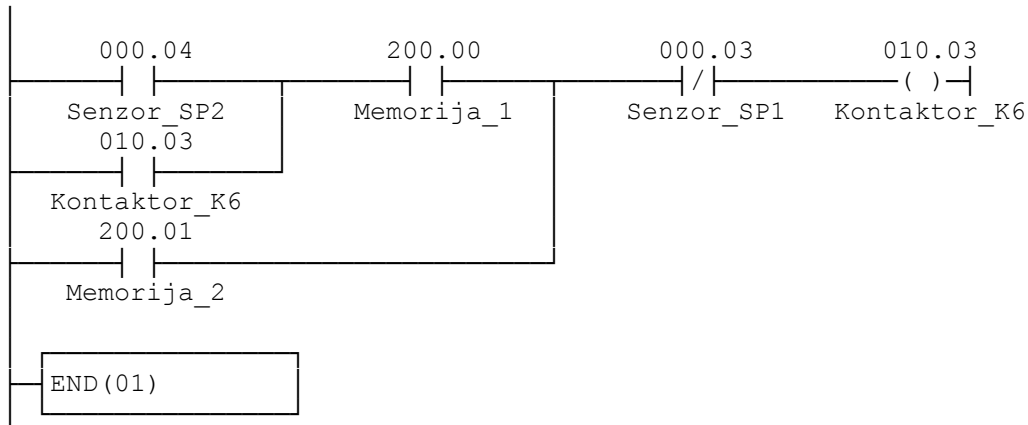
- Po dostizanju krajnjeg desnog položaja, klip se vraća nazad u početni položaj, a istovremeno pritiskivač se spušta naniže iz gornjeg krajnjeg položaja na niže.
- Pritiskivač pri dostizanju krajnjeg donjeg položaja, štampa i odmah se vraća na gore u početni položaj.
- Po dostizanju krajnjeg gornjeg položaja pritiskivača, ako je i klip u krajnjem desnom položaju i ako je magacin pun, ponavlja se postupak, odnosno ciklus.

Relejna šema koja obezbeđuje opisano upravljanje prikazana je na slici iznad.

Ciklus se automatski pokreće pritiskom na taster “Start” ako su pritiskivač i klip u početnim položajima, a prekida se pritiskom na taster “Stop”. Ako pritiskivač i klip nisu u početnim položajima, potrebno ih je prvo dovesti u njih. Deo upravljačke šeme koji to obezbeđuje na slici je prikazan isprekidanim linijama.

Lestvičasti dijagram se dobija na osnovu upravljačke relejne šeme:

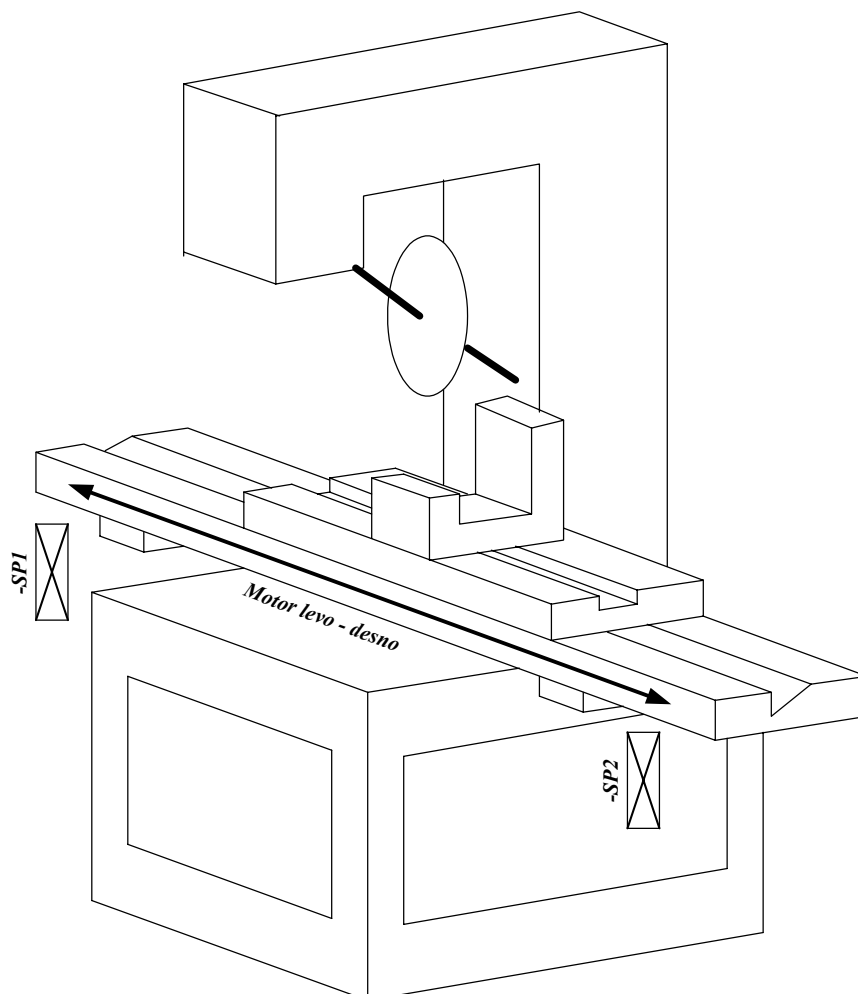




3.10. Upravljanje glodalicom

Razraditi rešenje upravljanja glodalicom na slici, po sledećem algoritmu:

Komande za upravljanje su ostvarene tasterima: “Start” i “Stop”. Kada se uključi start motor za posmak se obrće tako da pomera suport na desno, sve do desnog krajnjeg položaja koji se detektuje aktiviranjem graničnog prekidača –SP2. Tada se motor isključuje na dve sekunde, nakon čega se obrće na drugu stranu tako da pomera suport na levo, sve do levog krajnjeg položaja koji se detektuje aktiviranjem graničnog prekidača –SP1, nakon čega se ceo postupak ponavlja.



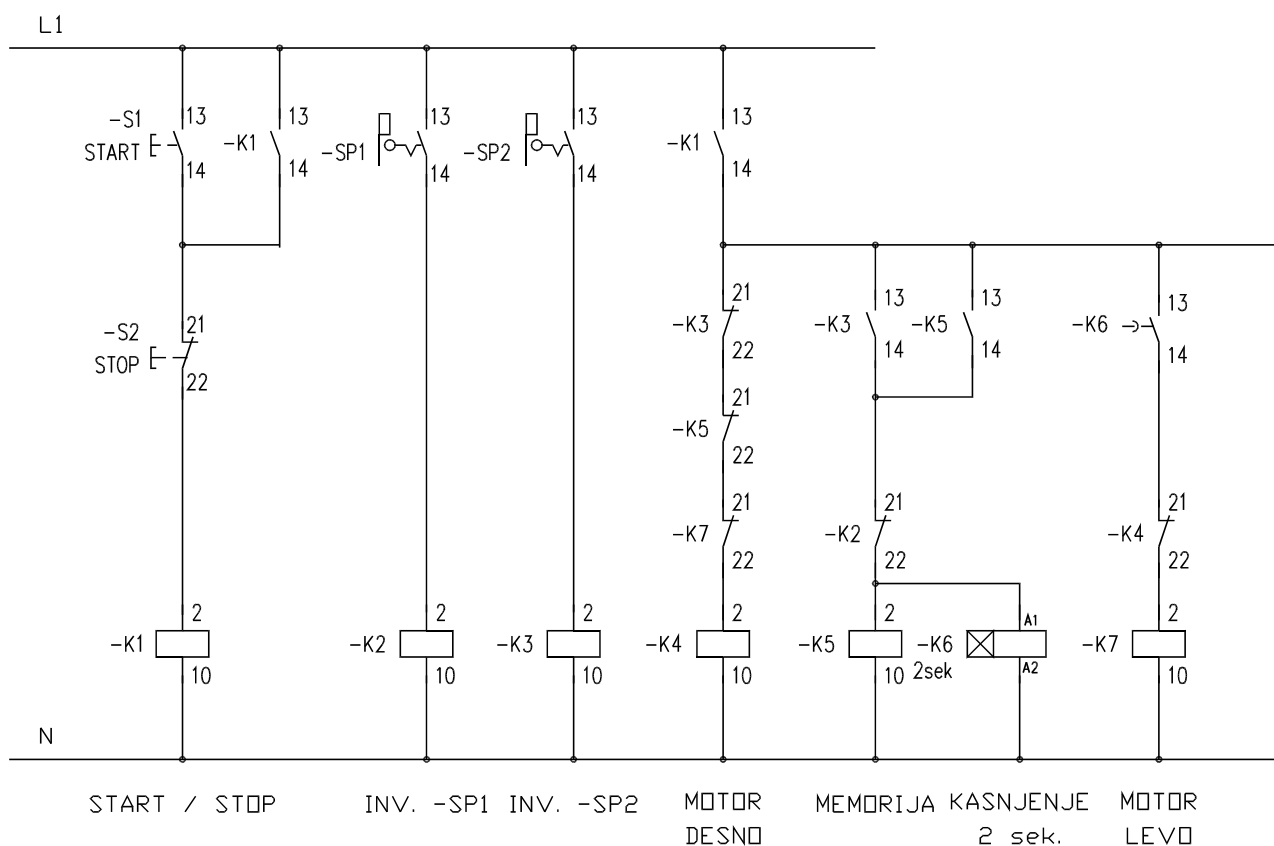
Tablica ulaza i izlaza je:

adrese ulaza		adrese izlaza	
000.00	taster <i>START S₁</i>	010.00	motor posmaka <i>MP₁ LEVO</i>
000.01	taster <i>STOP S₂</i>	010.01	motor posmaka <i>MP₁ DESNO</i>
000.02	granični prekidač <i>LEVO-SP₁</i>		
000.03	granični prekidač <i>DESNO-SP₂</i>		

Relejna šema upravljanja crta se na osnovu logičkih uslova zadatka, te obezbeđuje :

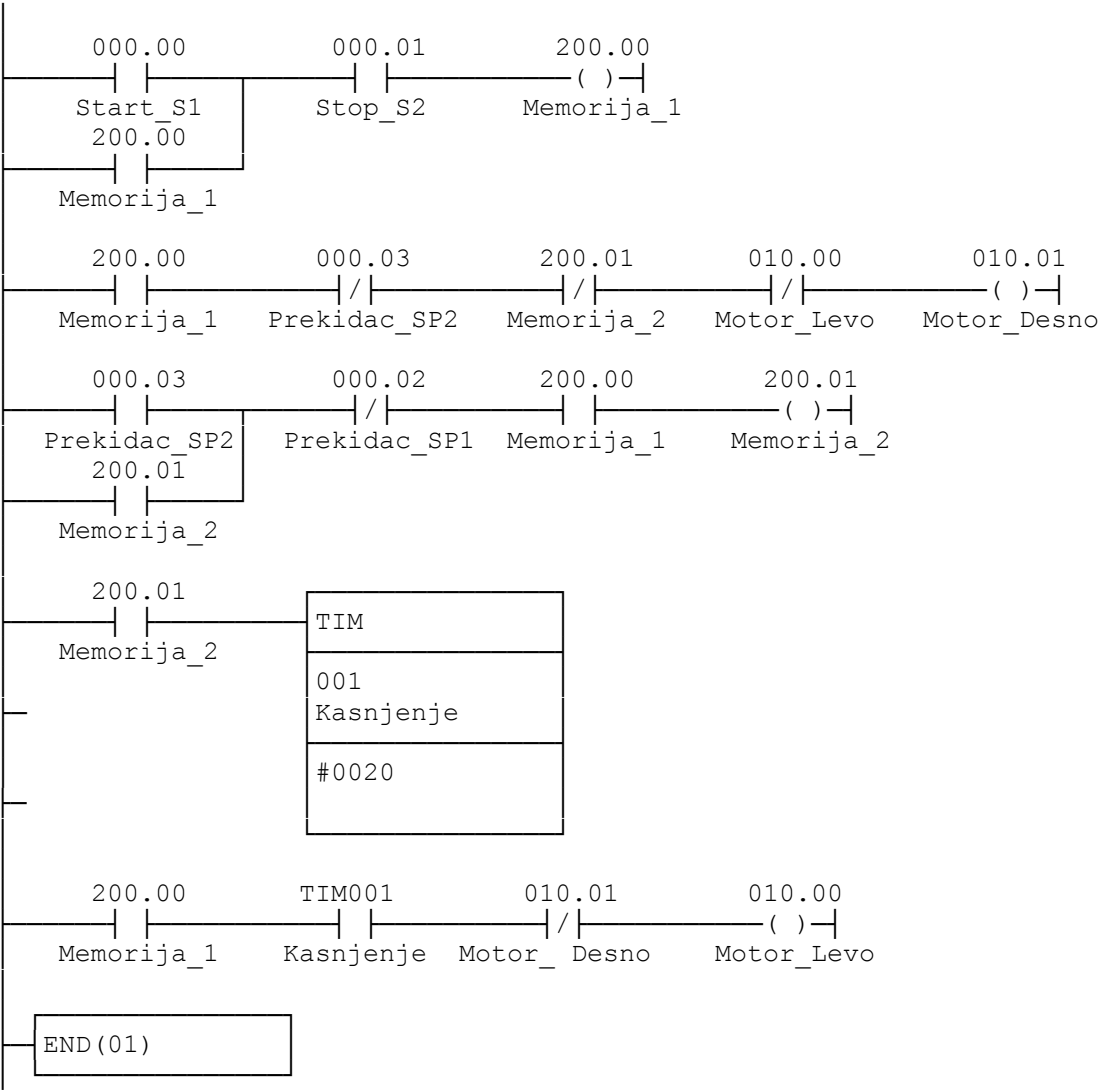
- Upravljanje se pokreće pritiskom na taster “Start“, a prekida se pritiskom na taster “Stop“ (Rele – *K1*).
- Potrebna inverzija logičkih stanja graničnih prekidača izvedena je releima –*K2* i –*K3*.
- Motor ide na desno (kontaktor –*K4*) ako je startovan pogon i ako nije izdata komanda za kretanje levo, sve do pritiska na desni granični prekidač –*SP2*.
- Dostizanje desnog graničnog prekidača se pamti (rele –*K5* sa pamćenjem). Pamćenje se resetuje sa sledećim pritiskom na levi granični prekidač –*SP1*.
- Potrebno vreme mirovanja motora posle dostizanja desnog graničnog položaja obezbeđeno je vremenskim releom –*K6*, čijim se radnim kontaktom upravlja radom motora na desno (kontaktor –*K7*).
- Isključuje se istovremena mogućnost uključanja rada motora na levo i desno.

Relejna šema koja obezbeđuje opisano upravljanje prikazana je na slici ispod.



Lestvičasti dijagram se dobija na osnovu upravljačke relejne šeme:

Ladder Diagram Primer 3.10



PRIMENA ANALOGNIH ULAZNO - IZLAZNIH PLC MODULA

1. Cilj laboratorijske vežbe:

- Upoznati elemente, način upotrebe analognih ulaznih i izlaznih modula na primeru analognog modula MAB42 u sprezi sa PLC-om CQM1H firme "Omron".
- Odvežbati priložene primere primene analognih ulaznih i izlaznih modula u akviziciji i obradi podataka.
- Odvežbati samostalno dodatno zadate primere primene analognih ulaznih i izlaznih modula.

2. Laboratorijska vežba se sastoji od:

- PC računara Pentium III sa instaliranim softverom CX-Programmer.
- Tabla sa PLC kontrolerom CQM1H-CPU51 i analognim modulom MAB 42.
- Generator / izvori signala.
- Simulator ulaza i izlaza PLC kontrolera.
- Merni instrumenti.

3. Uvod

PLC kontroler je upravljački elemenat automatizovanog sistema koji na osnovu prihvaćenih ulaznih signala sa ulaznih uređaja i određenog programa, formira izlazne signale kojima se upravlja izlaznim uređajima.

Ulazni uređaji čije signale prihvata PLC mogu biti vrlo različiti. Po tipu signala koji ti uređaji daju na svojim izlazima, mogu se podeliti na digitalne i analogne. Karakteristični analogni ulazni uređaji su: termoelementi, otpornički termometri i drugi pretvarači električnih i neelektričnih veličina u standardne strujne i naponske signale.

Ulazni signali se pri tome moraju prilagoditi sa odgovarajućim PLC ulaznim modulima. Ulazni moduli se konstruišu za prihvatanje jednosmernih i naizmeničnih električnih signala, naponskih nivoa od 5 do 250 V, strujnih nivoa od 0 do 20 mA.

Izlazni analogni moduli daju na svojim izlazima naponske i strujne signale koji zavise od njihovih karakteristika i koji se moraju prilagoditi potrebnim naponskim i strujnim nivoima.

Ovi moduli takođe obezbeđuju i raznorazne dijagnostičke funkcije: detekciju prekida kola, alarme za signale van dozvoljenog opsega itd.

Pomoću programa u PLC-u moguće je praktično realizovati bilo kakav željeni oblik signala bez korišćenja klasičnih elektronskih komponenti. Moguće je programski izvesti aproksimirani rad nekog kola bez potrebe za njegovom fizičkom realizacijom što znatno umanjuje troškove i olakšava izmene sistema.

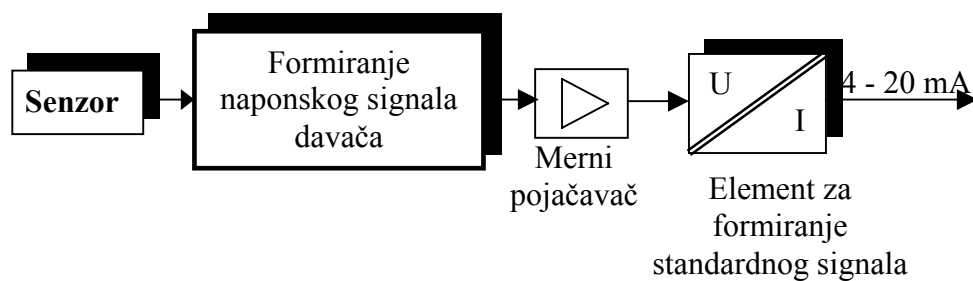
U datim primerima prikazane su osnovne operacije pri radu sa analognim modulima (akvizicija podataka, upravljanje na osnovu ulaznih analognih signala, neke jednostavne matematičke obrade ulaznih vrednosti, generisanje izlaznih signala po nekim pravilima) kao i neki kompleksniji primeri upravljanja i obrade signala.

4. Osnove rada sa analognim signalima

Analogni signali mogu biti naponski ili strujni. Koriste se po potrebi. Neki uređaji na svojim izlazima imaju naponske, a neki strujne signale. Postoje industrijski standardi opsega koje maksimalne vrednosti tih signala mogu poštovati. Međutim uvek se može desiti da neki uređaj generiše nivoe signala koji nisu po tim standardima.

Ako takav signal želimo upotrebiti kao ulaz u svoj sistem moramo ga pomoću odgovarajućeg pretvarača prilagoditi ulazu u PLC. Ulazi PLC-a poštuju industrijske standarde. Standardni opsezi naponskih signala su 0 do 5 [V], 0 do 10 [V], -10 do +10 [V], a strujnih 0 do 20 [mA], i 4 do 20 [mA].

Analogue ulazne veličine se primaju direktno sa mernog davača (transducer - senzor), ili iz sklopa koji zajednički čine merni davač i element za prilagođavanje (kondicioniranje) signala (transmitter) ili iz sklopa koji signal sa mernog davača priprema za direktan prihvrat od strane PLC-a. Pod kondicioniranjem signala u ovom slučaju podrazumeva se filtriranje, pojačanje, linearizacija, baferovanje, sample / hold, prigušenje itd. signala sa mernog davača. Na slici je prikazana šema pripreme analognih signala za prihvrat od strane PLC-a.



Ove vrednosti se obrađuju u programu koji je upisan u PLC. Pošto program radi sa numeričkim veličinama u digitalnom obliku, mora se izvršiti konverzija tih analognih u digitalne vrednosti i obrnuto (za izlaze). To je osnovni zadatak modula.

Parametri potrebni za ovu konverziju su naponski ili strujni opseg signala i rezolucija A/D odnosno D/A konverzije. Kod PLC modula postoji mogućnost izbora između više unapred definisanih opsega, ali je rezolucija konverzije (broj bitova digitalne konvertovane vrednosti) fiksiran.

Broj bitova konvertovanog podatka je bitan zbog preciznosti koju možemo postići. Ako je na primer opseg signala od 0 do 10 [V], a broj bitova je 8, to znači da postoje 256 kvantizaciona nivoa i da je preciznost svedena na $10 [V] / 256 [nivoa] = 39.0625 [mV/nivou]$, a ako je broj bitova 10 tada je preciznost $9.76 [mV/nivou]$. Za strujni opseg 0 do 20 [mA] i rezoluciju 12 bitova preciznost iznosi $4.88 [\mu A]$.

Za konvertovanje signala koriste se razni tipovi A/D konvertora. Razlikuju im se metod konverzije (paralelni komparatori, sukcesivna aproksimacija itd.), preciznost, vreme konverzije (vreme neophodno za konverziju jednog uzorka) i da li rade kontinualno ili na zahtev.

A/D konvertor naredbu za početak konverzije dobija od PLC-a, počinje konverziju, a kada je završi obaveštava PLC da je konverzija obavljena i da on može da očita konvertovanu vrednost.

A/D konvertori su opremljeni analognim vremenskim multiplekserima pomoću kojih mogu da obrađuju više analognih kanala naizgled istovremeno. Oni ustvari sprovode jedan po jedan analogni kanal na A/D konvertor i pojedine konvertovane vrednosti smeštaju na odgovarajuće adrese.

Zbog vremena potrebnog za konverziju neophodno je koristiti "Sample and Hold" kola koja zadržavaju konstantnu vrednost na ulazu A/D konvertora tokom cele periode konverzije.

Konvertovane vrednosti su celi brojevi koji mogu biti označeni ili neoznačeni. Ako želimo izvršiti neku precizniju obradu tih vrednosti moramo ih konvertovati u neki prigodan oblik (na primer "*floating point*") ili skalirati u skladu sa željenim pravilima.

Analogni izlazi zahtevaju D/A konvertore da bi digitalni podatak pretvorili u jednu od dve fizičke veličine: napon ili struju. Ovi naponi i struje moraju biti u određenim opsezima koji su dati industrijskim standardima.

Moduli međutim imaju i mnogo drugih funkcija koje služe za dijagnostiku, pouzdanost rada u raznim uslovima sredine, alarme itd.

Pri radu sa analognim signalima mnogo su izraženije smetnje nego pri radu sa digitalnim signalima. Zbog toga se preporučuje sprovođenje signala oklopljenim kablovima i uvrnutim paricama.

Osnovni redosled operacija pri radu sa analognim signalima je:

- Određivanje opsega analognih ulaza i broja ulaza.
- Određivanje opsega analognih izlaza i broja izlaza.
- Pisanje programa u skladu sa tako određenim opsezima.
- Povezivanje ulaznih i izlaznih tačaka u sistem.
- Ako se podešavanja opsega ne mogu izvršiti iz programa (na primer kod modula MAB42) to se obavlja postavljanjem odgovarajućih vrednosti u kontrolnim registrima PLC-a uz pomoć programa na PC-računaru ili pomoću upravljačke konzole.
- Startovanje programa u PLC-u.

U periodu razvoja programa korisno je koristiti generatore signala za simuliranje svih mogućih vrednosti ulaza i njihov uticaj na sistem. Isto tako korisno je mernim uređajima (multimetrima, osciloskopima) proveriti izlazne signale pre njihovog konačnog priključenja u sistem.

Moguće je koristiti i povratnu vezu na samom PLC-u. Programski kontrolišemo analogni izlaz koji je spojen na analogni ulaz na koji bi inače bio povezan neki davač signala. Na taj način možemo ubrzati razvoj programa time što ćemo simulirati razne promene vrednosti ulaznog signala (na primer temperatura peći sa velikom vremenskom konstantom).

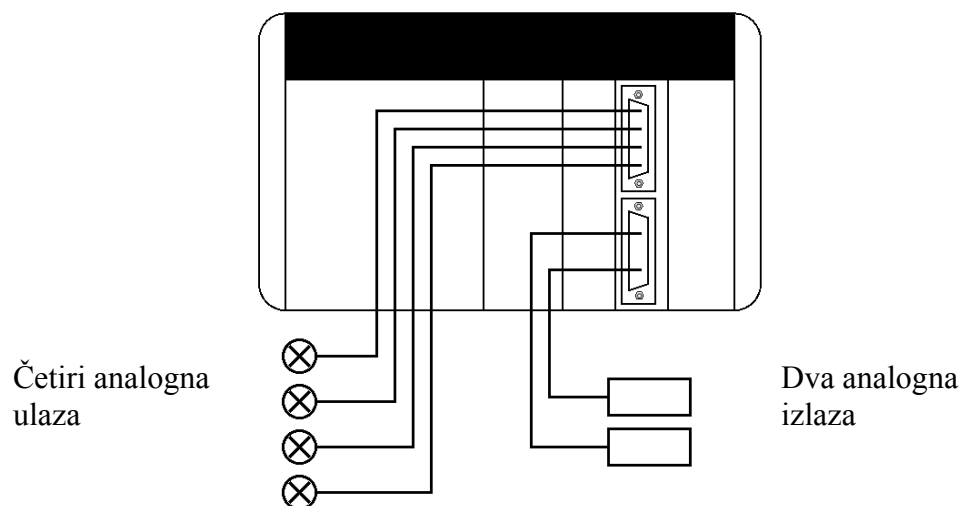
5. Analogni ulazno-izlazni moduli - Model CQM1H-MAB42

CQM1H-MAB42 je takozvana "*Inner Board*" kartica za CQM1H PLC. Ona se može montirati samo u slot broj 2 (desni slot) CQM1H-CPU51/61 CPU jedinice. Osnovne karakteristike ovog modula su date u tabeli:

Ime	Model	Specifikacije
Analogni I/O modul	CQM1H-MAB42	4 analogna ulaza: (-10 do +10 [V]; 0 do 5 [V]; 0 do 20 [mA]; odvojeni opseg signala za svaki ulaz) 2 analogna izlaza: (-10 do +10 [V]; 0 do 20 [mA]; odvojeni opseg signala za svaki izlaz)

5.1. Konfiguracija sistema

Modul poseduje dva standardna DB-15 konektora. Na gornjem su izvodi četiri analogna ulaza, a na donjem su izvodi dva analogna izlaza. Na gornjem kraju modula su i dve led diode koje služe za očitavanje stanja.



5.2. Led indikatori

ERR (Crvena) – Upaljena kada postoji greška u podešavanjima ili je došlo do greške pri analognoj konverziji.

RDY (Zelena) - Upaljena kada se analogni ulaz, izlaz može obaviti.

5.3. Specifikacija adresnog prostora

U adresnom prostoru PLC-a modul zauzima adrese od IR 232 do IR 237, SR 254 - bit 15, i AR 04 - bitove 08 do 15. Funkcija ovih memorijskih lokacija je data u sledećim tabelama.

IR adresno područje:

Word	Bits	Ime	Funkcija
IR 232	00 do 15	Konvertovana vrednost analognog ulaza 1	-10 do +10 [V] ; F800 do 07FF 0 do 20 [mA] ; 0000 do 07FF 0 do 5 [V] / 0 do 20 [mA] 0000 do FFFF
IR 233	00 do 15	Konvertovana vrednost analognog ulaza 2	
IR 234	00 do 15	Konvertovana vrednost analognog ulaza 3	
IR 235	00 do 15	Konvertovana vrednost analognog ulaza 4	
IR 236	00 do 15	Vrednost analognog izlaza 1	-10 do +10 [V] ; F800 do 07FF 0 do 20 [mA] ; 0000 do 07FF
IR 237	00 do 15	Vrednost analognog izlaza 2	

SR adresno područje:

Word	Bit	Funkcija
SR 254	15	Indikator greške modula

AR adresno područje:

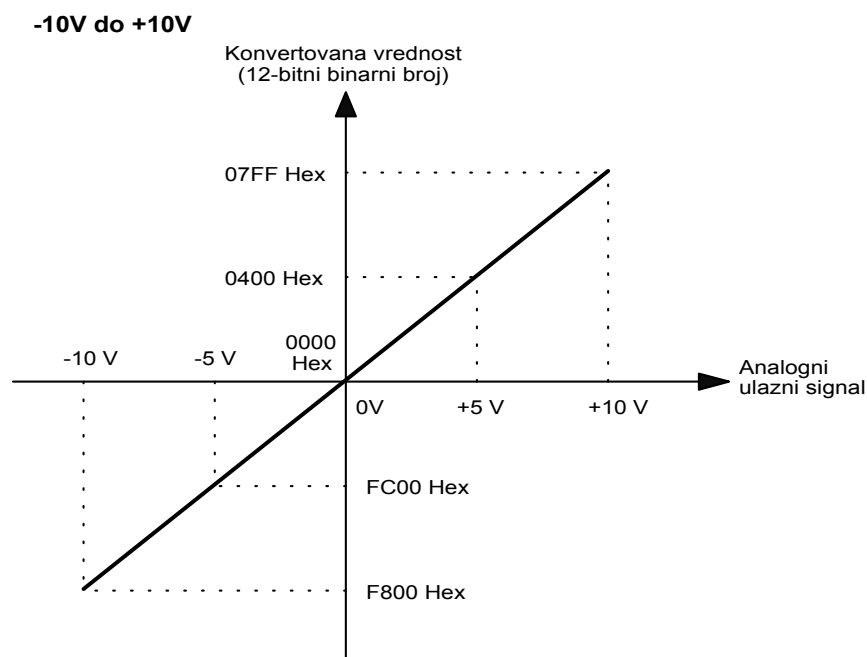
Word	Bitovi	Funkcija	
AR 04	08 do 15	Kodovi greške	00 Hex: Normalno 01 ili 02 Hex: Greška u elektronici 03 Hex: Greška u PC setup-u 04 Hex: Greška pri A/D ili D/A konverziji

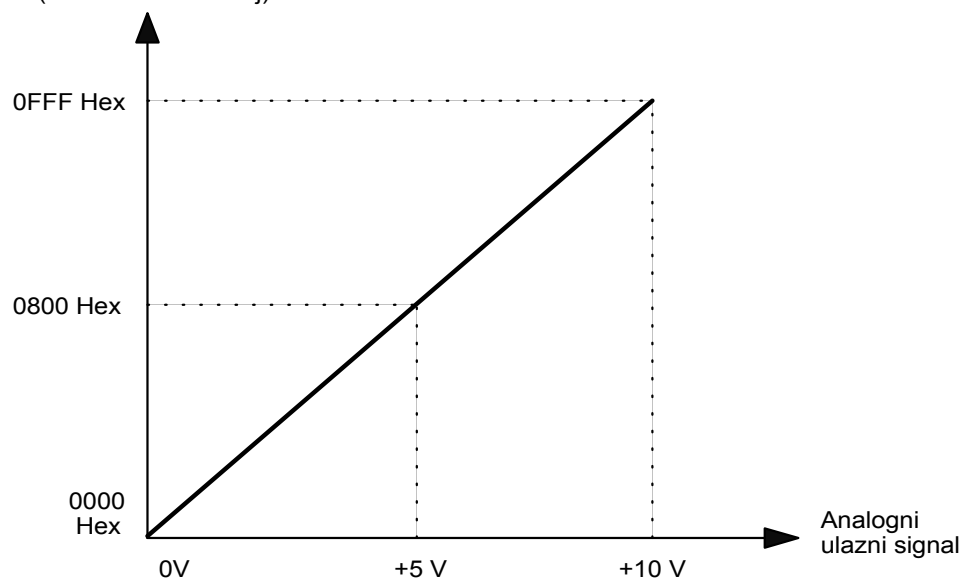
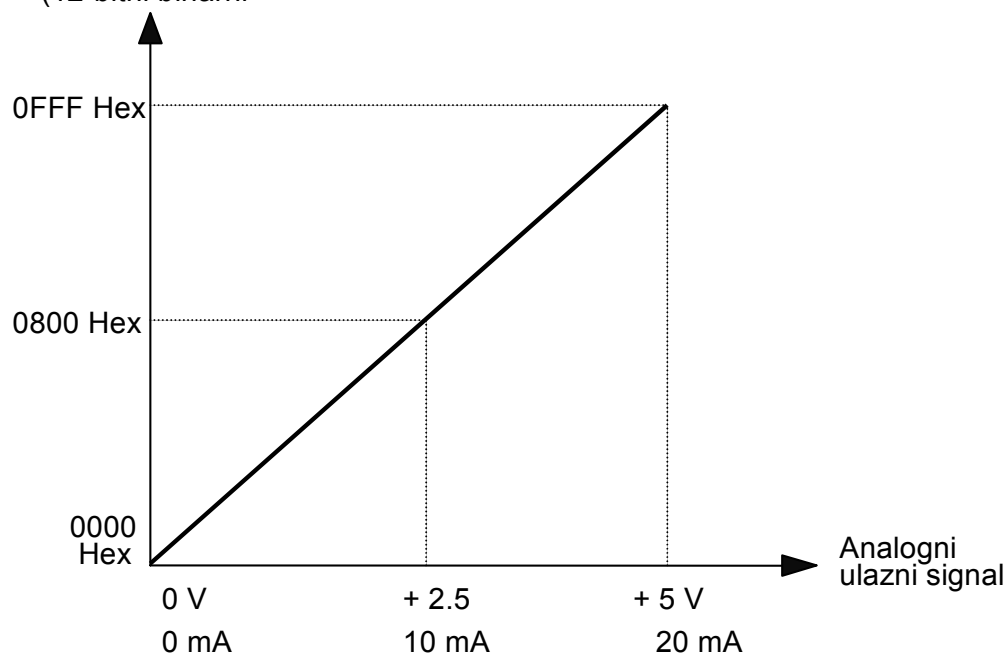
Bitna PC setup podešavanja:

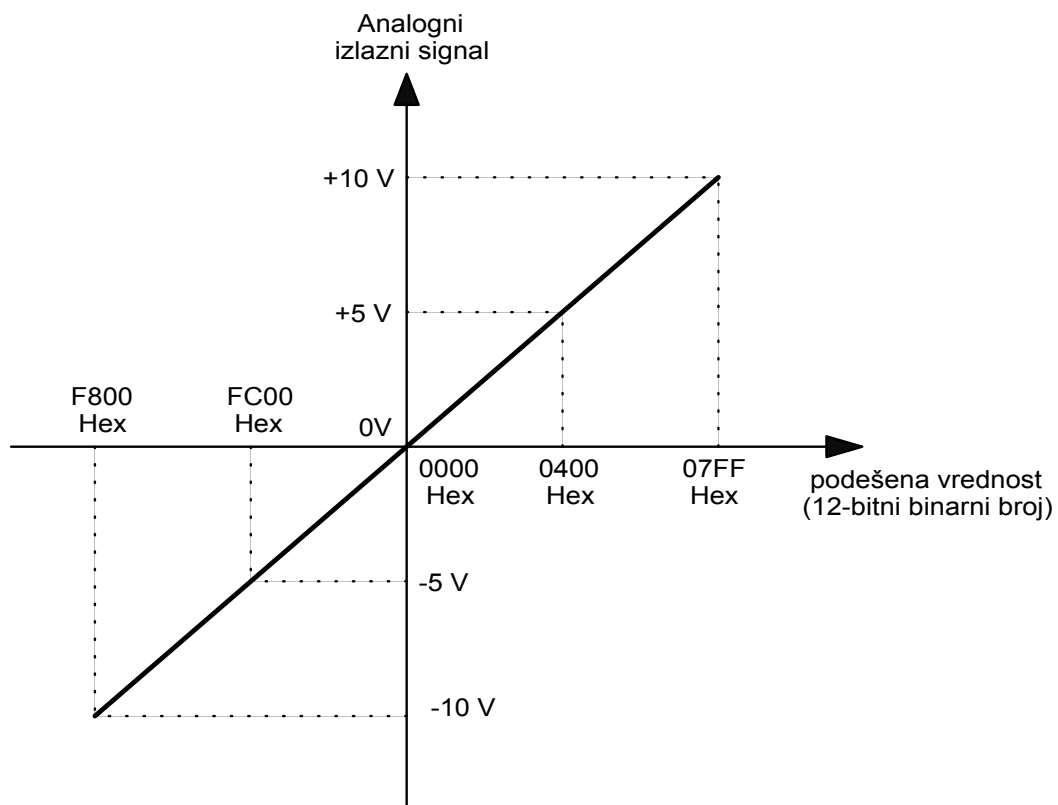
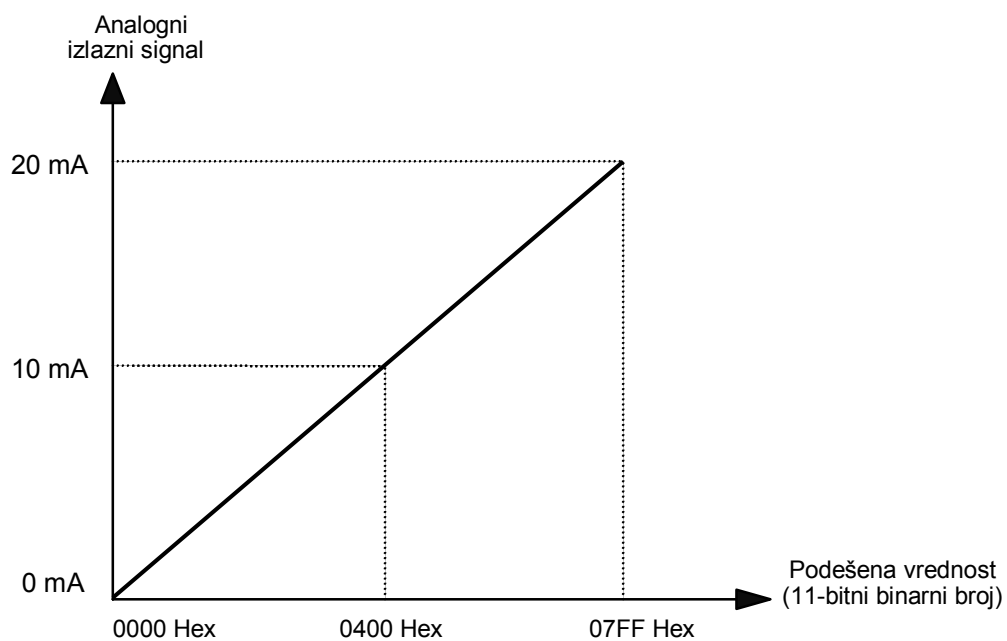
Word	Bitovi		
DM 6611	00 do 07	01 00: Opseg analognog ulaza 1 03 02 Opseg analognog ulaza 2 05 04: Opseg analognog ulaza 3 07 06: Opseg analognog ulaza 4	00: -10 [V] do +10 [V] 01: 0 [V] do +10 [V] 10: 0 [V] do +5 [V] i 0 [mA] do 20 [mA] 11: Ne koristi se
	08	Selekcija ulaza 1	Određuje da li se koristi A/D konverzija za određeni ulaz. 0: Ulaz se koristi 1: Ulaz se ne koristi
	09	Selekcija ulaza 2	
	10	Selekcija ulaza 3	
	11	Selekcija ulaza 4	
	12 do 15	Ne koriste se (postavljeni su na nulu)	

Napomena: Nivoi izlaznih analognih signala zavise samo od priključenih kola, odnosno ne postoje podešavanja u PC setup-u za njih. Podešavanja u PC setupu imaju uticaja na PLC samo na početku rada, odnosno pri njegovom uključenju. To znači da ako se promeni neko podešavanje na adresi 6611 moramo isključiti napajanje PLC-a i ponovo ga uključiti da bi podešavanja imala efekta. Promene sadržaja memorije PLC-a se obavljaju iz programa koji služe za programiranje i monitoring (na primer "CX-programmer").

5.4. Analogni ulazi: ulazni podaci i konvertovane vrednosti



0V do +10VKonvertovana vrednost
(12-bitni binarni broj)**0V do 5V ili 0 do 20 mA**Konvertovana
(12-bitni binarni)

5.5. Analogni izlazi: podešene vrednosti i vrednosti izlaznog signala**-10V do +10V****0 do 20 mA**

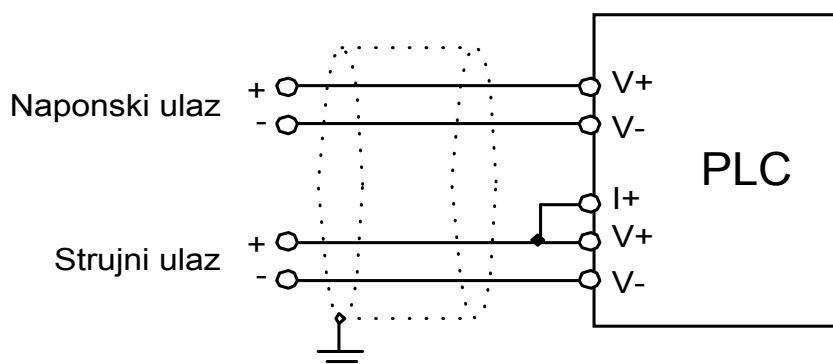
5.6. Primeri korišćenja modula

Modul ne koristi specijalne instrukcije, već se koristi standardna instrukcija MOV. MOV se koristi za iščitavanje analognih ulaza i za postavljanje vrednosti analognih izlaza.

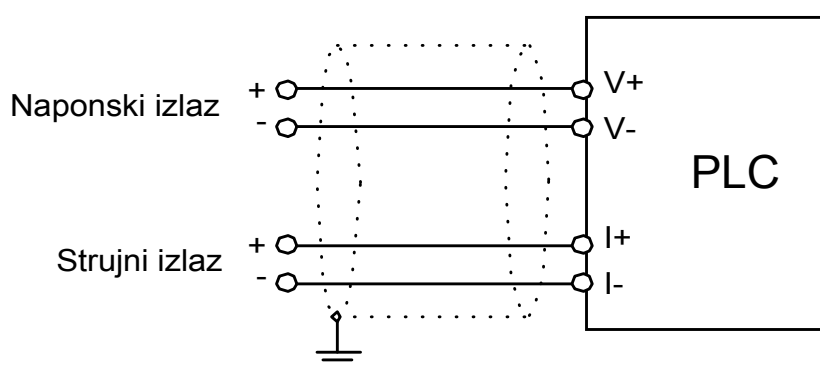
Osnovni redosled operacija pri radu sa ovim modulom je sledeći:

- Određivanje opsega analognih ulaza i broja ulaza.
- Određivanje opsega analognih izlaza i broja izlaza.
- Povezivanje ulaznih i izlaznih tačaka u sistem.
- Podešavanje opsega i korišćenosti ulaza i izlaza postavljanjem odgovarajućih vrednosti na adresu DM 6611 u PLC-u.
- Resetovanje PLC-a njegovim gašenjem i paljenjem.
- Startovanje programa u PLC-u.

5.7. Povezivanje u sistem



Povezivanje ulaza

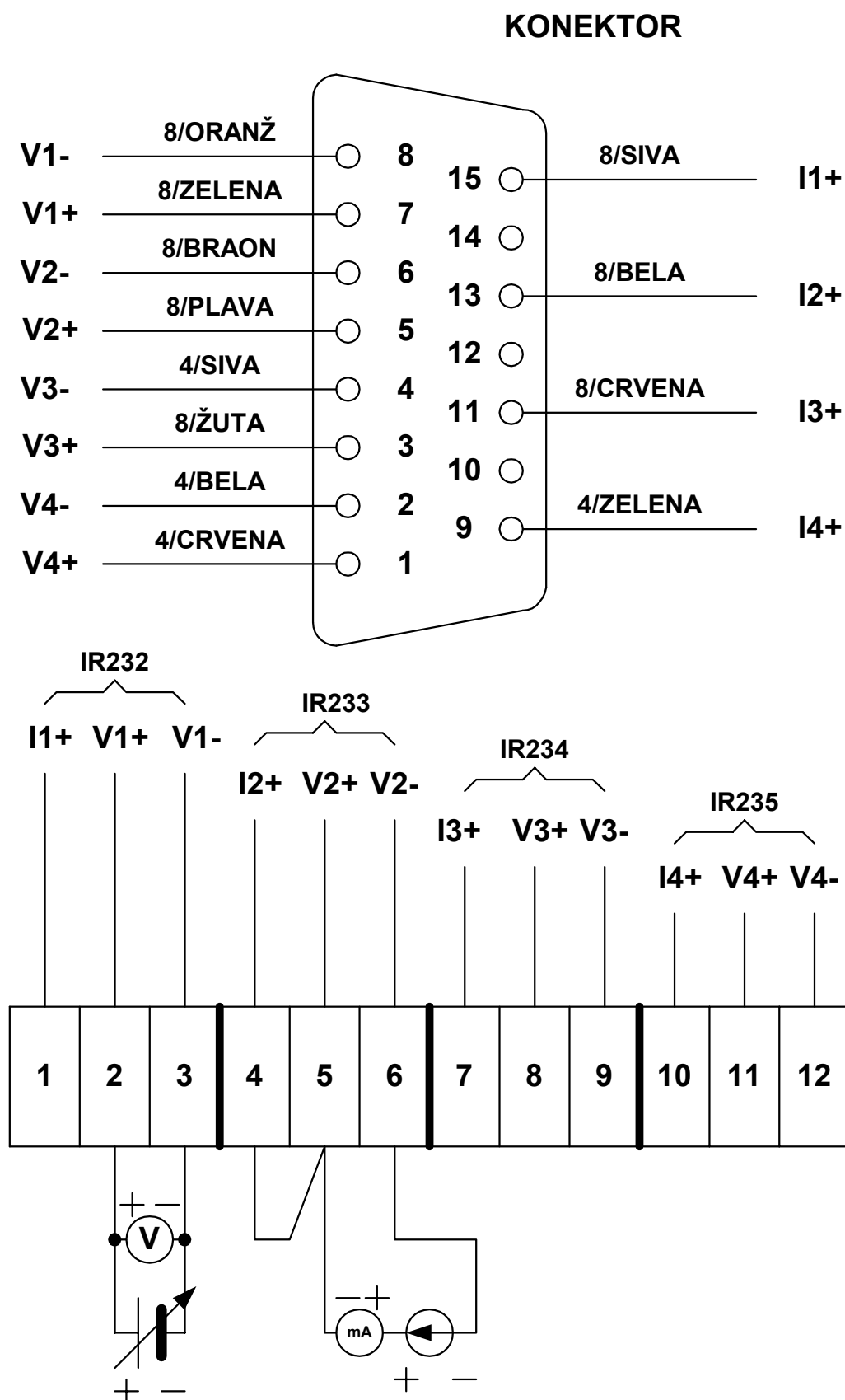


Povezivanje izlaza

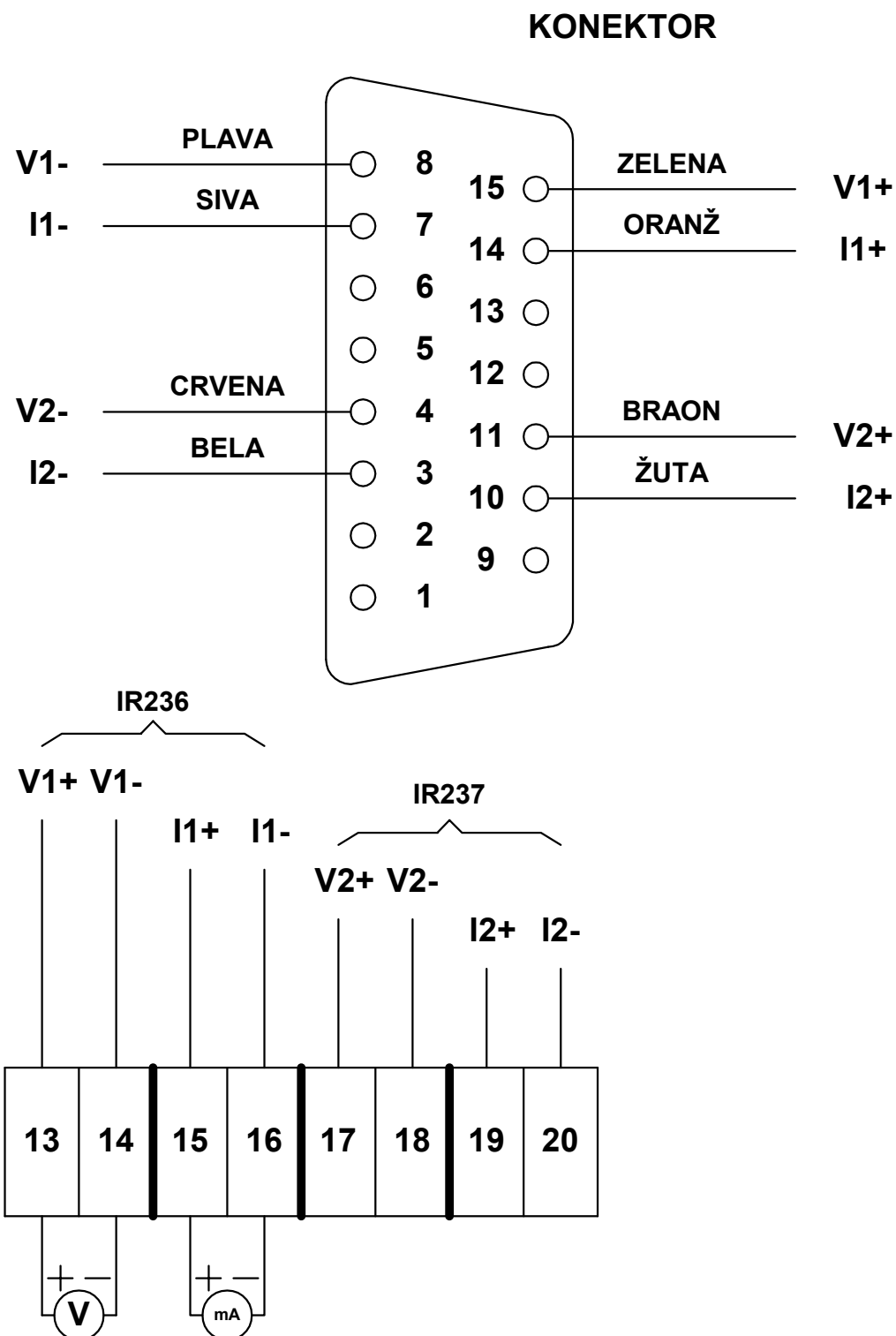
Napomena: Oklop kablova koji povezuju modul sa sistemom, se povezuje sa masom radi neutralisanja eventualnih smetnji.

5.8. Povezivanje stezaljki

CN1: ANALOG INPUTS 1 TO 4



CN2: ANALOG OUTPUTS 1 TO 2



6. Osnovni primeri rada sa analognim modulima

Ovi primeri će pokazati osnovne operacije pri radu sa analognim signalima: očitavanje, generisanje i skaliranje signala. To ćemo ostvariti povezivanjem izvora jednosmernog napona na ulaze MAB42 modula čije ćemo vrednosti očitavati u PLC-u, i povezivanjem multimetara na izlaze MAB42 modula pomoću kojeg ćemo pratiti vrednosti tih izlaza.

6.1. Primer 1.

Zadatak:

Povezati izvor jednosmernog napona na analogni ulaz PLC-a. Napisati leder dijagram koji će vrednost analognog ulaza smestiti na neku lokaciju u memoriji. Ispitati rad PLC-a za sledeće opsege generisanog signala: 0 – 10 [V], 0 – 20 [mA] i 4-20 [mA]. Ove opsege ispitujemo zato što predstavljaju standardne vrednosti signala koji se koriste u industrijskim primenama.

1. korak:

Određivanje opsega analognih ulaza i broja ulaza.

Modul MAB42 može da obradi signale čije vrednosti pripadaju sledećim opsezima:

-10 [V] do +10 [V], 0 [V] do +10 [V], 0 [V] do +5 [V] i 0 [mA] do 20 [mA]

Rad PLC-a za opseg 0 do 10 [V] možemo ispitati na prva dva opsega.

Izabraćemo opseg 0 do 10 [V] zbog veće rezolucije uzorkovanog signala.

Strujni ulaz ćemo za oba tražena opsega ispitati pomoću trećeg opsega modula MAB42 pošto oba opsega pripadaju opsegu 0 do 20 [mA].

2.korak:

Povezivanje ulaznih priključaka u sistem.

Povezivanje izvršiti u skladu sa pravilima vezivanja za MAB42. jednosmerni izvor povezati na analogni ulaz 1. Obratiti pažnju na razliku u načinu vezivanja naponskog i strujnog ulaza.

3. korak:

Podešavanje opsega i korišćenosti ulaza i izlaza postavljanjem odgovarajućih vrednosti na adresu DM 6611 u PLC-u.

To znači da u lokaciju DM 6611 koja služi kao kontrolni registar modula MAB-42 treba staviti sledeću vrednost:

bit		15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DM 6611		0	0	0	0	1	1	1	0	0	0	0	0	0	0	0	1
Ovi bitove se ne koriste i zato ih postavljamo na 0						Koristi se samo prvi ulaz				Ova podešavanja nisu bitna pošto ulaze 2, 3 i 4 ne koristimo					01: 0V do +10V 10: 0 [mA] do 20 [mA]		

Vrednost koju treba da upišemo u DM 6611 za naponski opseg je:

binarno '0000 1110 0000 0001'

odnosno hex ' 0 E 0 1 '

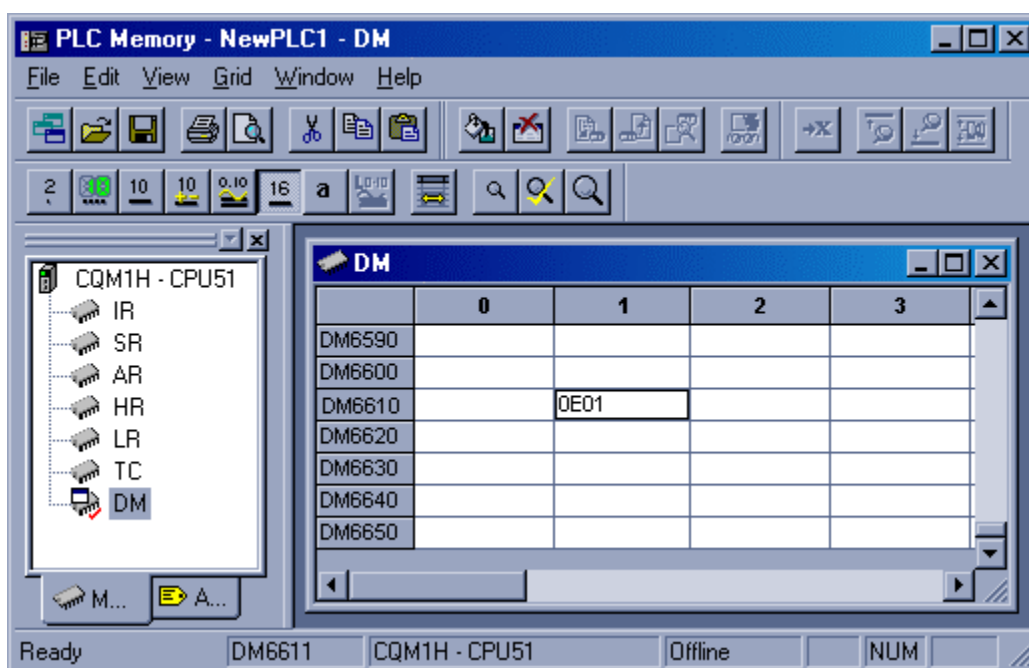
a za strujni opseg je:

binarno '0000 1110 0000 0010'

odnosno hex ' 0 E 0 2 '

To se može uraditi iz "CX-programmera" na sledeći način:

- Pod pretpostavkom da je otvoren projekat u kojem je određen tip PLC-a CQM1H, da je PLC "Online" i u "Program" modu, iz glavnog menija se izabere opcija PLC → "Edit" → "Memory".
- Dvoklikom na labelu DM u levom kraju prozora otvorićete prozor sa vrednostima u DM memorijskom području.
- Pomoću klizača na desnoj strani pronađite adresu DM 6611 gde je s leve strane adresa u koraku od deset, a kolone označavaju desnu cifru adrese.
- Dvoklikom na polje adrese 6611 upišite 0E01 (ili 0E02).
- Desnim dugmetom miša možete birati način prikaza vrednosti upisane u memoriji tako da možete vrednost upisati i u binarnom ili hex-a obliku.
- Na ekranu će te videti sliku:



- Pritiskom na ikonicu "Transfer to PLC"  prebacite vrednost u memoriju PLC-a.
- Zatvorite prozor i odgovorite potvrdno na pitanje da li želite da snimate promene u projekat. Ovo je potrebno da bi unesena vrednost bila zapamćena.
- **Prekinuti vezu PLC-a sa računarom (isključivanjem "Online" moda).**
 - **Ako ovo ne uradimo PLC će dospeti u nedefinisano stanje i biće potrebno ponoviti ceo postupak.**
- Isključite PLC iz struje i posle nekoliko sekundi ga ponovo uključite. Ovim će se izvršiti ponovna inicijalizacija opsega analognog modula i on će od tada sve konverzije vršiti po tim podešavanjima.

4. korak:

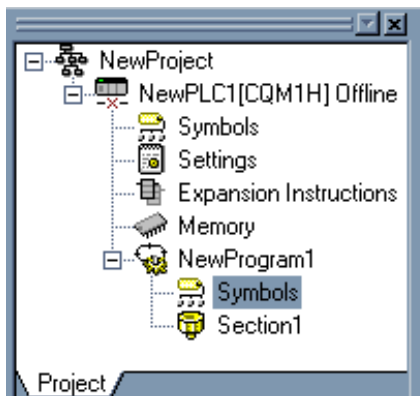
Izrada programa za PLC

Realizovaćemo leder dijagram koji će vrednost analognog ulaza smestiti na neku lokaciju u memoriji.

Definisaćemo dva simbola: ulaz1 - adresa prvog analognog ulaza je 232

vrednost - adresa gde ćemo čuvati vrednost
analognog signala je DM100

Simboli koji pripadaju trenutno aktivnom projektu se mogu videti u stavci "symbols" u "workspace" prozoru. Na listi simbola se mogu videti sve njihove osobine.



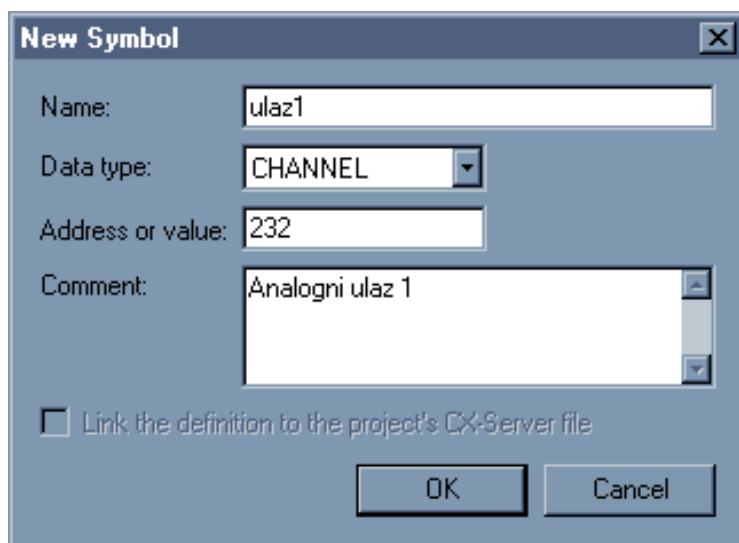
Name	Type	Address / Value	Rack Location	Usage	Comment
ulaz1	CHANNEL	232			Analogni ulaz 1
vrednost	CHANNEL	DM100			Vrednost ulaza

Novi simboli se dodaju opcijom "Insert symbol" iz menija koji se dobija na pritisak desnog dugmeta miša u ovom prozoru.

Potrebno je uneti ime novog simbola, njegov tip, adresu i komentar.

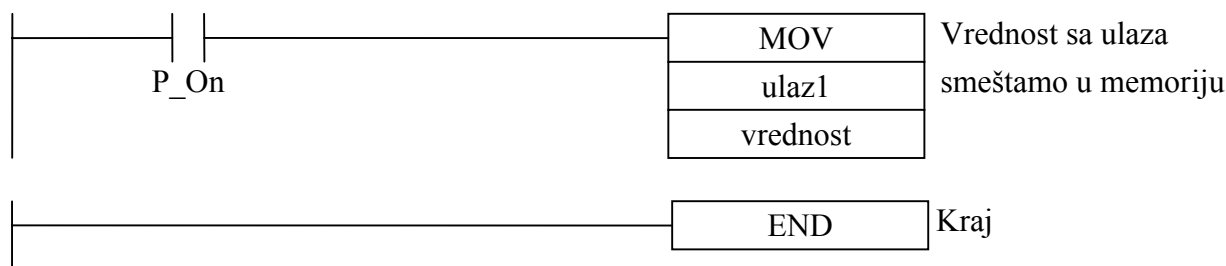
Ovaj postupak treba ponoviti za svaki simbol koji ćemo koristiti u projektu.

Osobine svakog simbola se mogu kasnije promeniti izborom opcije "Edit symbol" iz "Pop-Up" menija koji se dobija desnim klikom na ime simbola u listi simbola.



5. korak:

Sada možemo uneti sledeći lestvičasti dijagram:



6. korak:

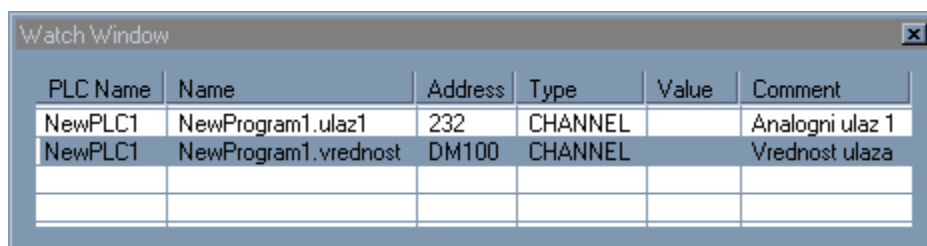
Praćenje rada programa.

Kada pokrenemo program možemo pratiti njegov rad na više načina.

Prvi način je da u "*Monitoring modu*" pratimo vrednosti simbola direktno na lestvičastom dijagramu programa.

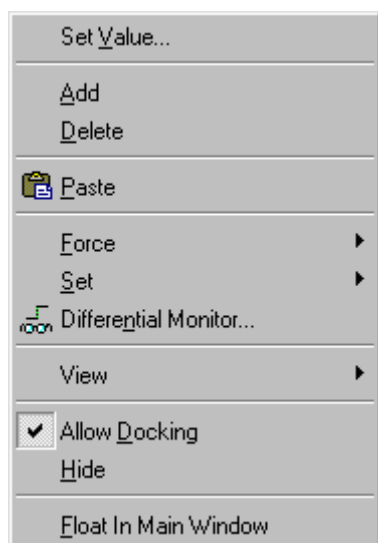
Drugi način je "*Watch window*". Pomoću njega možemo pratiti trenutno stanje više simbola na pregledan način. Watch window se može aktivirati ako se iz glavnog menija izabere "*View*" → "*Windows*" →  ili tastaturom, kombinacijom "*Alt + 3*".

"*Watch window*" izgleda kao na slici:



PLC Name	Name	Address	Type	Value	Comment
NewPLC1	NewProgram1.ulaz1	232	CHANNEL		Analogni ulaz 1
NewPLC1	NewProgram1.vrednost	DM100	CHANNEL		Vrednost ulaza

Simboli koje želimo da posmatramo se dodaju ili brišu iz menija koji dobijamo pritiskom na desni taster miša.



- | | |
|----------------------|---|
| Set Value | - omogućava direktno postavljanje simbola na neku proizvoljnu vrednost. |
| Add | - dodavanje simbola u tabelu. |
| Delete | - brisanje simbola iz tabele. |
| Paste | - umetanje prethodno kopiranog simbola iz liste simbola. |
| Force | - prisilno zadržavanje BOOL simbola na On ili Off (program u PLC-u ne može da promeni ovu vrednost) |
| Set | - postavljanje vrednosti BOOL simbola (program u PLC-u može da promeni ovu vrednost). |
| Differential Monitor | - brojanje opadajućih ili rastućih ivica nekog BOOL simbola. |
| View | - podešavanje vidljivosti pojedinih kolona " <i>watch window</i> " - prozora. |
| Allow Docking | - određuje da li da " <i>watch window</i> " bude u sklopu glavne radne površine CX-programmera. |
| Hide | - sakriva " <i>watch window</i> ". |
| Float in main Window | - Oslobađa " <i>watch window</i> " od svog fiksiranog položaja. |

Treći način je "*Data Trace*" ili "*Time chart monitoring*".

Taj način će biti detaljno opisan u okviru naprednijih primera. Za sada je dovoljno praćenje vrednosti ulaznog signala u okviru watch prozora.

Potrebno je ispitati ponašanje analognog ulaza za vrednosti ulaznog signala iz celog opsega. Iz karakteristika modula MAB42 vidimo da se vrednost ulaznog signala konvertuje u 12-bitni binarni broj, odnosno trocifreni heksadekadni broj.

Pošto je karakteristika ulaznog modula linearna to znači da će se vrednosti preslikati na sledeći način:

0 do 10 V - 000 do FFFh

0 do 20 mA - 000 do FFFh

4 do 20 mA - 333 do FFFh

Vidimo da konvertovane vrednosti nisu pogodne za dalju obradu. Zato je potrebno izvršiti njihovo skaliranje. O tome će biti reči u sledećim poglavljima.

Podešavanjem generatora napona i struje ispitajte više vrednosti signala koje pripadaju izabranom opsegu uključujući i minimalne i maksimalne vrednosti. Zatim probajte i vrednosti koje su malo manje od minimalne ili malo veće od maksimalne.

Napomena: Paziti da vrednost ulaznog signala ne bude prevelika. Inače može doći do oštećenja modula.

6.2. Primer 2.

Zadatak:

Postavljanje vrednosti izlaznih signala raznih vrednosti.

Povezati merni instrument na analogni izlaz PLC-a. Napisati leder dijagram koji će čitajući vrednosti iz memorijske lokacije postaviti vrednost izlaznog signala na neku od vrednosti iz sledećih opsega: 0 – 10 [V], 0 – 20 [mA] i 4-20 [mA].

1. korak:

Određivanje opsega analognih izlaza i broja izlaza.

Koristićemo prvi analogni izlaz za naponski signal, a drugi analogni izlaz za strujni signal. Modul MAB42 može da generiše signale čije vrednosti pripadaju sledećim opsezima: -10 [V] do +10 [V] i 0 [mA] do 20 [mA]

Rad PLC-a za opseg 0 do 10 [V] možemo ispitati na prvom opsegu.

2. korak:

Povezivanje izlaznih priključaka sistema.

Povezivanje izvršiti u skladu sa pravilima vezivanja za MAB42. Merne instrumente povezati na analogne izlaze. Jedan merni instrument koji je podešen da meri napon opsega 0 do 10 [V] povezati na prvi izlaz. Drugi merni instrument koji je podešen da meri struju reda veličine [mA] povezati na drugi izlaz.

Obratiti pažnju na razliku u načinu vezivanja naponskog i strujnog izlaza. Nivoi izlaznih analognih signala zavise samo od priključenih kola, odnosno ne postoje podešavanja u PC setup-u za njih.

3. korak:

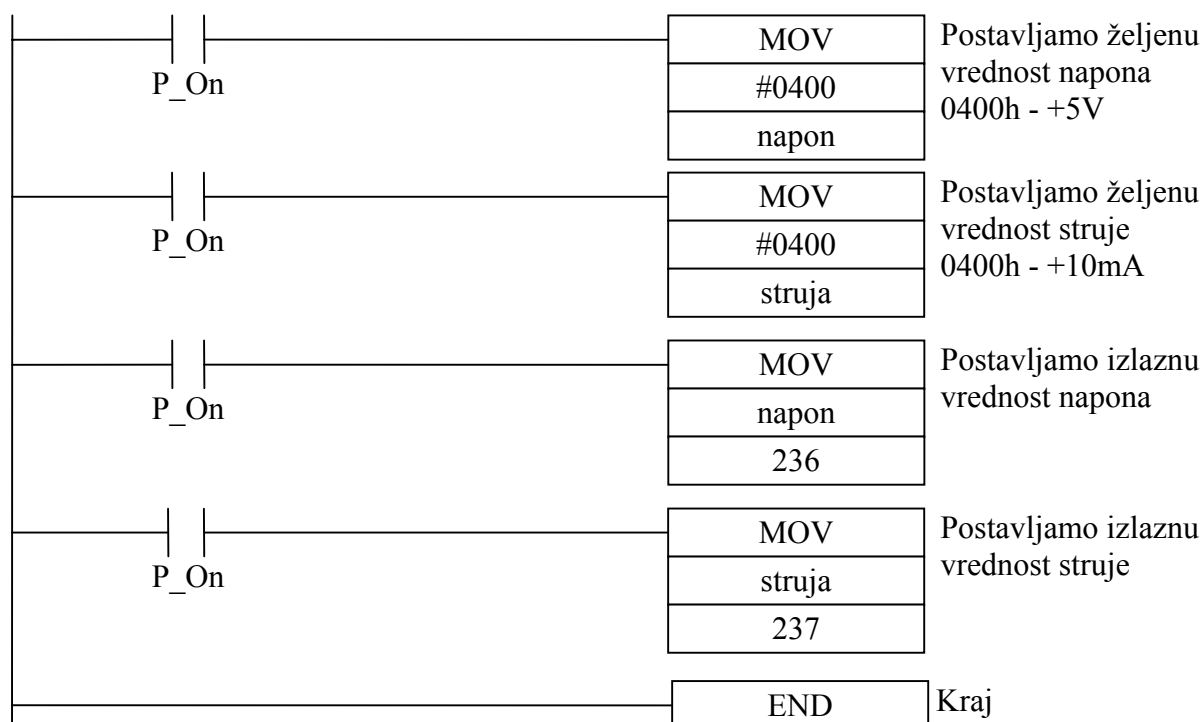
Izrada programa za PLC

Realizovaćemo lader dijagram koji će vrednost iz dve lokacije u memoriji smestiti na analogne izlaze.

Definisaćemo dva simbola:

izlaz1 - adresa prvog analognog izlaza je 236.
 izlaz2 - adresa drugog analognog izlaza je 237.
 napon - adresa gde ćemo čuvati željenu vrednost
 naponskog izlaza je DM100.
 struja - adresa gde ćemo čuvati željenu vrednost
 strujnog izlaza je DM101.

Sada možemo uneti sledeći lestvičasti dijagram:



4. korak:

Praćenje rada programa.

Vrednosti analognih izlaza možete pratiti na mernim instrumentima.

Potrebno je ispitati ponašanje analognih izlaza za vrednosti iz celog opsega. Iz karakteristika modula MAB42 vidimo da vrednost izlaznog signala zavisi od 12-bitnog binarnog broja, odnosno trocifreni heksadekadni broj.

Pošto je karakteristika izlaznog modula linearna to znači da će se vrednosti preslikati na sledeći način:

Za naponski izlaz:

0000 do 07FFh	- 0 do 10 [V]
F800 do FFFFh	- -10 do 0 [V]

Za strujni izlaz:

000 do FFFh	- 0 do 20 [mA]
333 do FFFh	- 4 do 20 [mA]

Promenom željenih vrednosti napona i struje ispitajte više vrednosti signala koje pripadaju izabranom opsegu uključujući i minimalne i maksimalne vrednosti. Zatim probajte i vrednosti koje su malo manje od minimalne ili malo veće od maksimalne.

Primetićete da se za vrednosti signala van definisanih opsega pali indikator greške na analognom modulu.

6.3. Primer 3.

Zadatak:

Realizovati skaliranje ulaznog signala i postaviti odgovarajuću vrednost na izlazu:

Opseg od 4 do 20 [mA] na ulazu skalirati u opseg od 0 do 10 [V] na izlazu.

1. korak:

Određivanje opsega i broja analognih ulaza i izlaza.

Koristićemo prvi analogni ulaz i prvi analogni izlaz modula MAB42. Za ulaz ćemo koristiti strujni opseg 0 do 20 [mA].

2. korak:

Povezivanje ulaznih i izlaznih tačaka sistema.

Povezivanje izvršiti u skladu sa pravilima vezivanja za MAB42. Generator signala povezati na analogni ulaz 1. Obratiti pažnju na način vezivanja strujnog ulaza. Merni instrument povezati na analogni izlaz vodeći računa da je to naponski izlaz.

3. korak:

U lokaciju DM 6611 koja služi kao kontrolni registar modula MAB-42 treba staviti sledeću vrednost:

binarno '0000 1110 0000 0010'

odnosno hex ' 0 E 0 2 '

To se može uraditi iz "CX-programmera" na već navedeni način:

4. korak:

Izrada programa za PLC

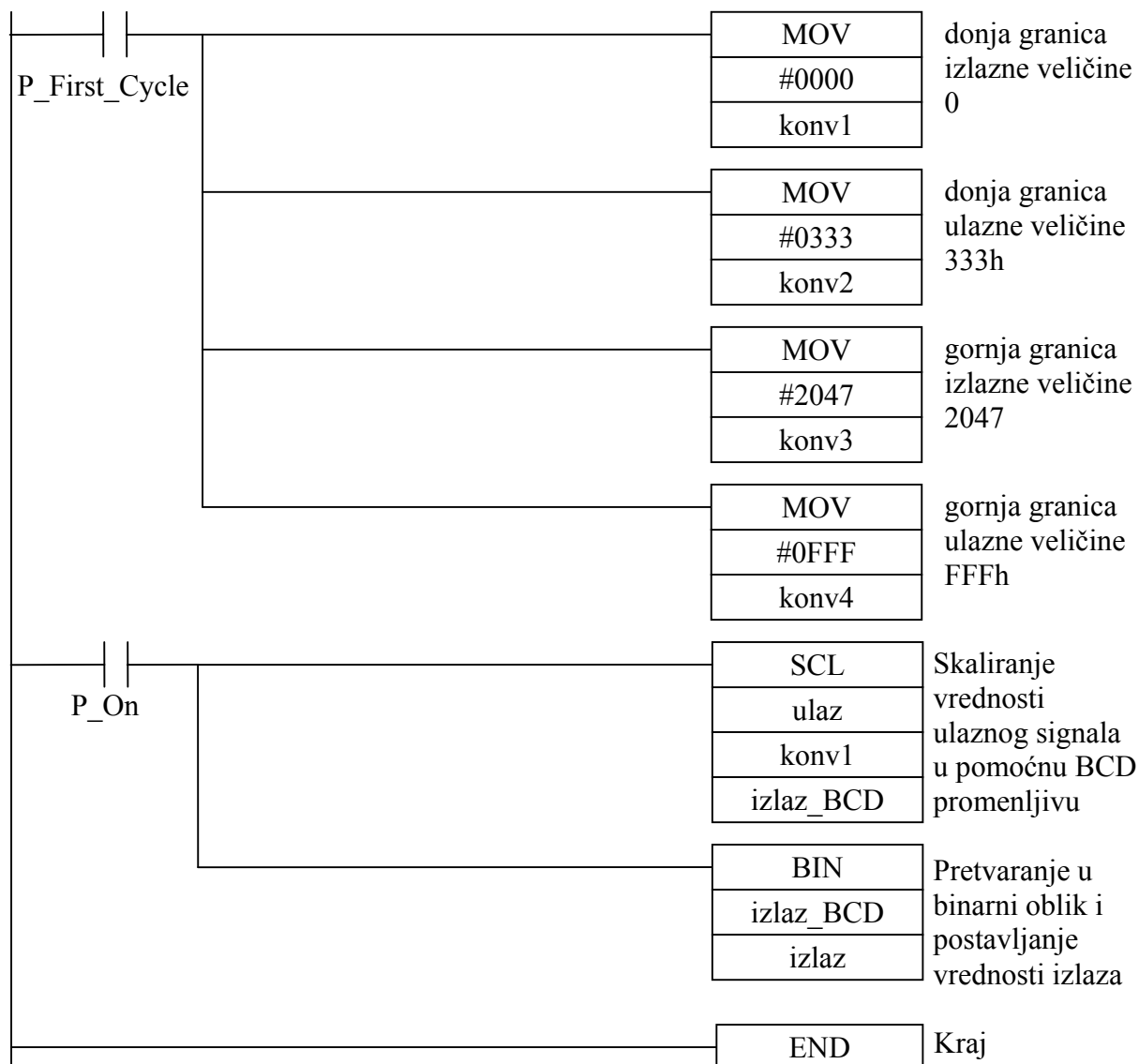
Skaliranje se obavlja pomoću naredbe SCL. Ova naredba skalira vrednost iz jednog opsega u drugi opseg. Za rad su joj potrebna četiri parametra. Parametri se moraju čuvati u četiri uzastopne memorijske lokacije. Vrednost koja se skalira je standardni binarni broj. Rezultat skaliranja je BCD broj. Taj BCD broj ne možemo postaviti direktno na analogni izlaz. Moramo izvršiti njegovu konverziju u binarni broj naredbom BIN. Zbog toga ćemo koristiti i jednu pomoćnu memorijsku lokaciju u kojoj ćemo čuvati rezultat skaliranja. Definišaćemo sledeće simbole:

ulaz	- adresa prvog analognog ulaza je 232
izlaz	- adresa prvog analognog izlaza je 236
izlaz_BCD	- adresa pomoćne promenljive DM100
konv1	- adresa prvog parametra skaliranja DM101
konv2	- drugi parametar DM102
konv3	- treći parametar DM103

konv4 - četvrti parametar DM104

Na početku rada programa potrebno je postaviti vrednosti parametara skaliranja. Funkcija skaliranja je linearna operacija. Zavisnost njenog rezultata od ulazne vrednosti i parametara je data na slici. Ulazni opseg je od 0 [mA] (0333h) do 20 [mA] (0FFFh). Izlazni opseg je od 0 [V] (0000) do 10 [V] (07FFh=2047BCD).

Sada možemo uneti sledeći lestvičasti dijagram:

**5. korak:**

Praćenje rada programa.

Potrebno je ispitati ponašanje analognog izlaza za sve vrednosti ulaznog signala iz celog opsega. Proverite linearnost operacije skaliranja. Podešavanjem generatora struje na više vrednosti signala koje pripadaju opsegu 0 do 20 [mA] uključujući i minimalne i maksimalne vrednosti zapišite odgovarajuće vrednosti izlaznog signala. Nacrtajte grafik na osnovu ovih vrednosti.

Napomena: Paziti da vrednost ulaznog signala ne bude prevelika. Inače može doći do oštećenja modula.

6.4. Primer 4.**Zadatak:**

Realizovati skaliranje vrednosti iz memorije i postaviti odgovarajuću vrednost na izlazu:

Vrednost iz opsega od 0 do 2000 [ob/min] skalirati u opseg od 0 do 20 [mA] na izlazu.

1. korak:

Određivanje opsega i broja analognih ulaza i izlaza.

Koristićemo prvi analogni izlaz modula MAB42.

2. korak:

Povezivanje izlaznih tačaka sistema.

Povezivanje izvršiti u skladu sa pravilima vezivanja za MAB42. Merni instrument povezati na analogni izlaz vodeći računa da je to strujni izlaz.

3. korak:

Nivoi izlaznih analognih signala zavise samo od priključenih kola, odnosno ne postoje podešavanja u PC setup-u za njih.

4. korak:

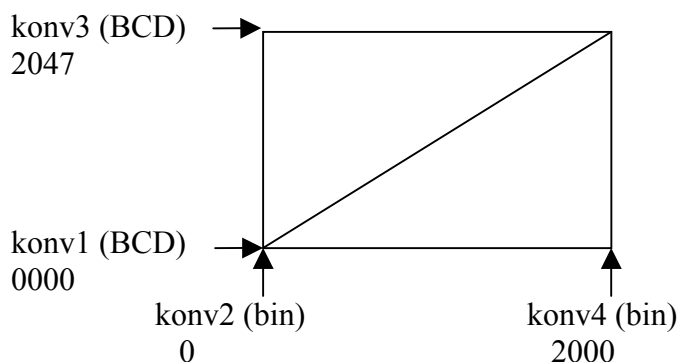
Izrada programa za PLC

Program je sličan prethodnom primeru. Razlika je u tome što se ulazna vrednost nalazi u memoriji. Razlikuju se i parametri konverzije.

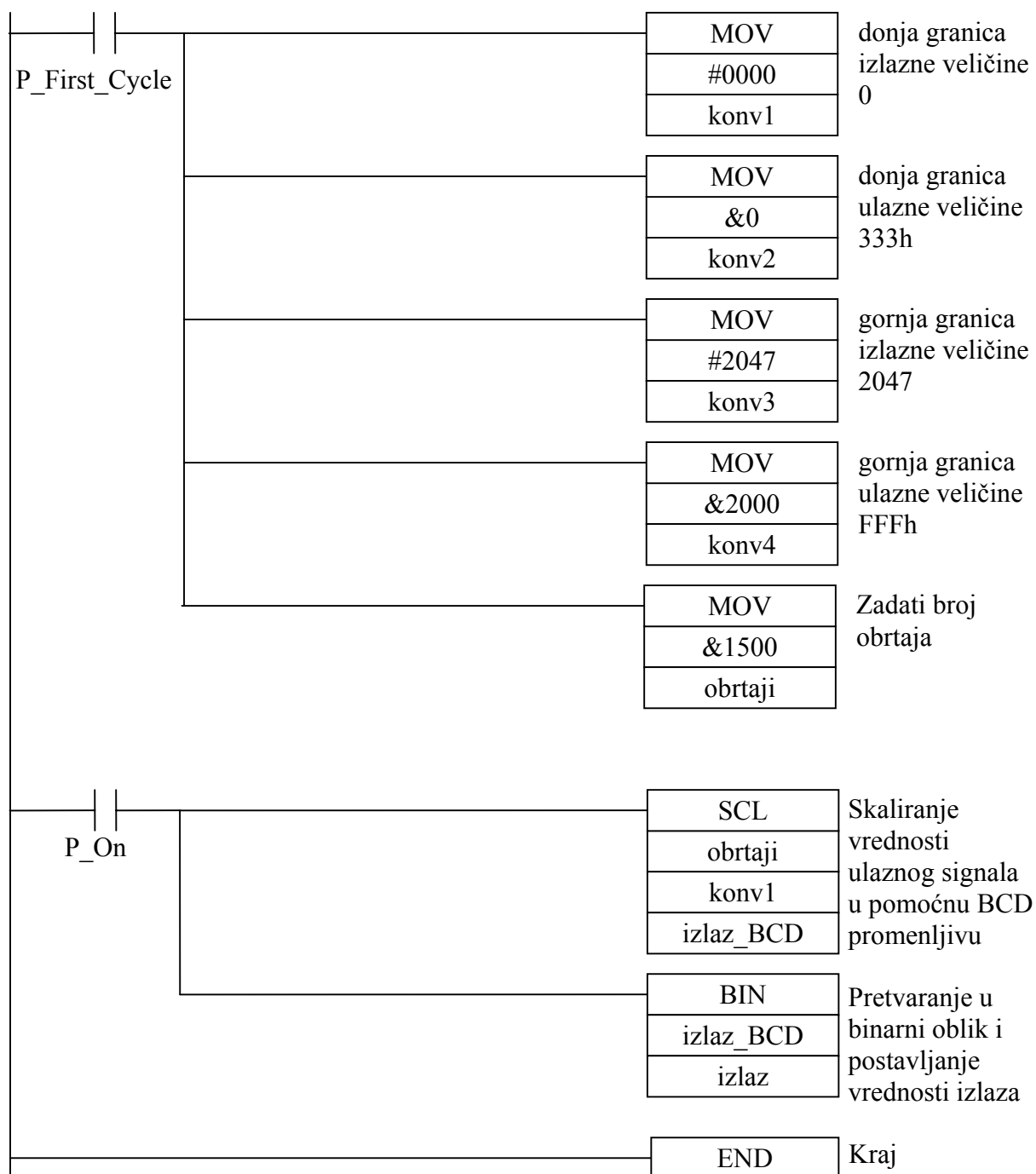
Definisaćemo sledeće simbole:

obrtaji	- adresa lokacije broja obrtaja DM102
izlaz	- adresa prvog analognog izlaza je 236
izlaz_BCD	- adresa pomoćne promenljive DM100
konv1	- adresa prvog parametra skaliranja DM101
konv2	- drugi parametar DM102
konv3	- treći parametar DM103
konv4	- četvrti parametar DM104

Na početku rada programa potrebno je postaviti vrednosti parametara skaliranja. Ulazni opseg je od 0 [ob/min] do 2000 [ob/min]. Izlazni opseg je od 0 [mA] (0000) do 20 [mA] (07FFh=2047BCD).



Sada možemo uneti sledeći lestvičasti dijagram:



5. korak:

Praćenje rada programa.

Potrebno je ispitati ponašanje analognog izlaza za različite vrednosti zadatog broja obrtaja. Proverite linearnost operacije skaliranja. Podešavanjem različitih vrednosti broja obrtaja zapišite odgovarajuće vrednosti izlaznog signala. Nacrtajte grafik na osnovu ovih vrednosti.

7. Napredni primeri rada sa analognim modulima

Ovi primeri će pokazati napredne operacije pri radu sa analognim signalima: upoređivanje vrednosti, odlučivanje, kontrolu, rad sa floating point aritmetikom itd.

7.1. Primer 1.

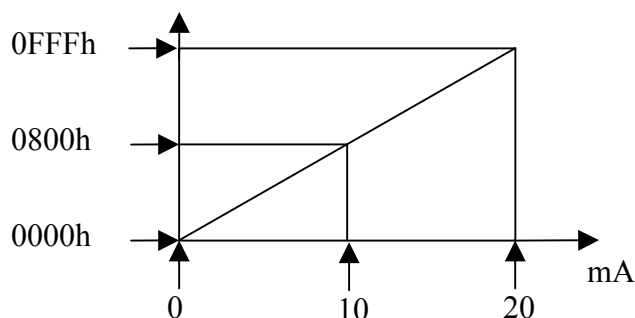
Zadatak:

Koristeći opseg ulaznog signala od 0 do 20 [mA] realizovati program koji će uključivati određene releje u zavisnosti od vrednosti ulaznog signala:

$I_{ul} < 10$ [mA]	-	IR 100.00	-	relej1
$I_{ul} = 10$ [mA]	-	IR 100.01	-	relej2
$I_{ul} > 10$ [mA]	-	IR 100.02	-	relej3

1. korak:

Granična vrednosti ulaznog analognog signala je 10 [mA]. Sa ulazne karakteristike opsega 0 do 20 [mA] vidimo da se ta vrednost konvertuje u broj 0800h. Potrebno je konstantno upoređivati vrednost ulaznog signala sa tom graničnom konstantom i u skladu sa rezultatom komparacije pobuditi odgovarajući relej.



2. korak:

Određivanje opsega i broja analognih ulaza.

Koristićemo prvi analogni ulaz modula MAB42 na opsegu 0 do 20 [mA].

3. korak:

Povezivanje ulaznih tačaka sistema.

Povezivanje izvršiti u skladu sa pravilima vezivanja za MAB42. Generator strujnog signala povezati na analogni ulaz vodeći računa da je to strujni ulaz.

4. korak:

Podešavanje opsega ulaznog signala.

U lokaciju DM 6611 koja služi kao kontrolni registar modula MAB-42 treba staviti sledeću vrednost:

binarno '0000 1110 0000 0010'
odnosno hex ' 0 E 0 2 '

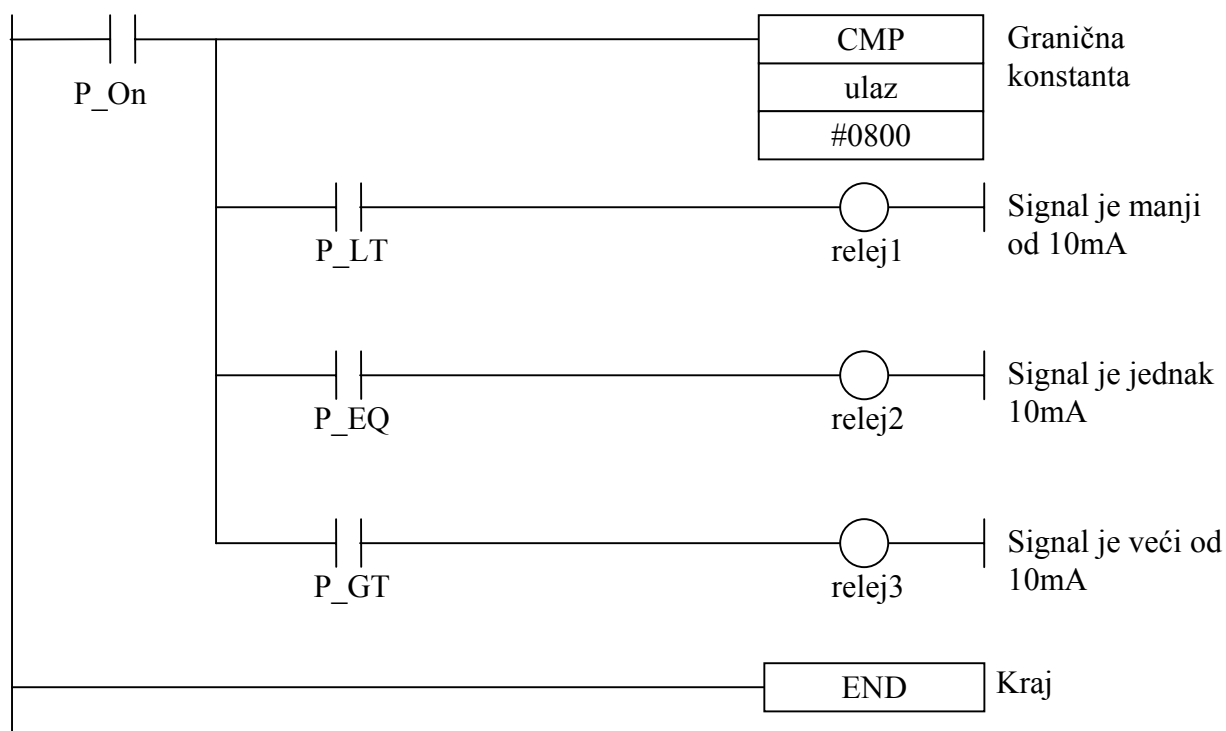
5. korak:

Izrada programa za PLC

Definisaćemo sledeće simbole:

ulaz - adresa prvog analognog ulaza je 232
 relej1 - adresa prvog releja DM100.00
 relej2 - adresa prvog releja DM100.01
 relej3 - adresa prvog releja DM100.02

Sada možemo uneti sledeći lestvičasti dijagram:

**6. korak:**

Praćenje rada programa.

Potrebno je ispitati ponašanje releja za različite vrednosti ulaznog signala. Primetićete da je problem detekcija neke tačne vrednosti zbog oscilacija u ulaznom signalu. To se može rešiti upoređivanjem sa nekim užim opsegom umesto sa konkretnom vrednošću. O tome će biti reči u sledećem primeru.

Napomena: *Paziti da vrednost ulaznog signala ne bude prevelika. Inače može doći do oštećenja modula.*

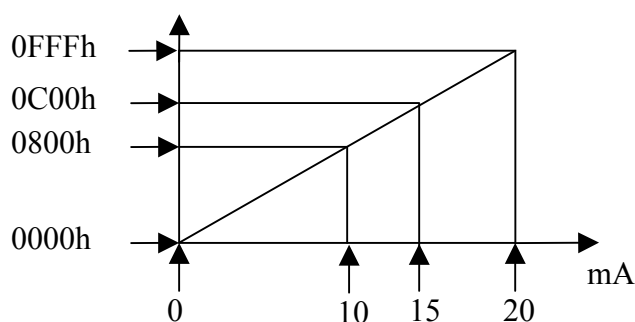
7.2. Primer 2.**Zadatak:**

Na istom strujnom opsegu realizovati dvopozicioni regulator sa histerezom na sledeći način:

$I_{ul} < 10 [mA]$	-	IR 100.00 - relej1
$10 [mA] \leq I_{ul} < 15 [mA]$	-	IR 100.01 - relej2
$I_{ul} \geq 15 [mA]$	-	IR 100.02 - relej3

1. korak:

Granične vrednosti ulaznog analognog signala su 10 i 15 [mA]. Sa ulazne karakteristike opsega 0 do 20 [mA] vidimo da se te vrednosti konvertuju u brojeve 0800h i 0C00h. Potrebno je konstantno upoređivati vrednost ulaznog signala sa tim graničnim konstantama i u skladu sa rezultatom komparacije paliti odgovarajući relej.

**2. korak:**

Određivanje opsega i broja analognih ulaza.

Koristićemo prvi analogni ulaz modula MAB42 na opsegu 0 do 20 [mA].

3. korak:

Povezivanje ulaznih tačaka sistema.

Povezivanje izvršiti u skladu sa pravilima vezivanja za MAB42. Generator strujnog signala povezati na analogni ulaz vodeći računa da je to strujni ulaz.

4. korak:

Podešavanje opsega ulaznog signala.

U lokaciju DM 6611 koja služi kao kontrolni registar modula MAB-42 treba staviti sledeću vrednost:

binarno	'0000 1110 0000 0010'
odnosno hex	' 0 E 0 2 '

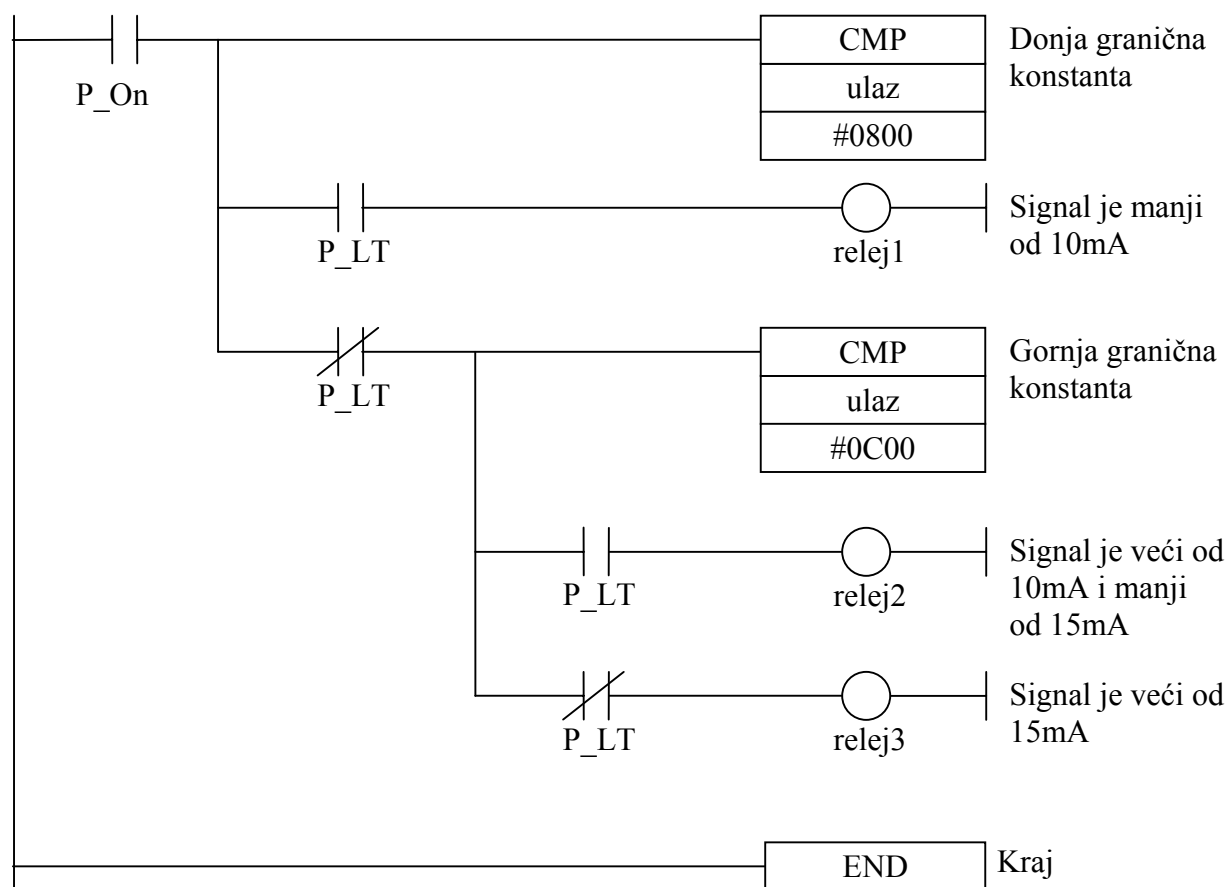
5. korak:

Izrada programa za PLC

Definisaćemo sledeće simbole:

ulaz	-	adresa prvog analognog ulaza je 232
relej1	-	adresa prvog releja DM100.00
relej2	-	adresa prvog releja DM100.01
relej3	-	adresa prvog releja DM100.02

Sada možemo uneti sledeći lestvičasti dijagram:



6. korak:

Praćenje rada programa.

Potrebno je ispitati ponašanje releja za različite vrednosti ulaznog signala.

Napomena: *Paziti da vrednost ulaznog signala ne bude prevelika. Inače može doći do oštećenja modula.*

7.3. Primer 3.

Zadatak:

Realizovati lestvičasti dijagram koji će vršiti upravljanje vrednosti analognog izlaza u skladu sa sledećim pravilima:

- Pomoću dva tastera (plus i minus) povećavati odnosno smanjivati vrednost analognog izlaza za neku proizvoljno određenu vrednost.
- Opseg izlaznog signala je 0 do 20 [mA].
- Vrednost izlaznog signala na početku rada programa treba da je 10 [mA].
- Program napraviti tako da signal nikad ne prekorači granice opsega.

1. korak:

Program ćemo realizovati tako da se struja na izlazu menja za 1 [mA] za svaki pritisak tastera. Željenu vrednost izlaza čuvamo u memoriji, a u svakom ciklusu je postavljamo na izlaz.

2. korak:

Određivanje opsega analognih izlaza i broja izlaza.

Koristićemo prvi izlaz i to kao strujni.

3. korak:

Povezivanje ulaznih i izlaznih tačaka u sistem.

Povezivanje izvršiti u skladu sa pravilima vezivanja za MAB42. Multimetar povezati na analogni izlaz 1. Obratiti pažnju na način vezivanja strujnog izlaza.

4. korak:

Nivoi izlaznih analognih signala zavise samo od priključenih kola, odnosno ne postoje podešavanja u PC setup-u za njih.

5. korak:

Izrada programa za PLC

Dati problem ćemo realizovati na sledeći način:

- Program treba u trenutku kad je neki taster pritisnut da promeni vrednost u memoriji (da je uveća ili smanji) za neku konstantnu vrednost. Pošto je ceo opseg od hex000 do hex7FF, odnosno od 0 do 2047 decimalno, uzećemo konstantu 100 što nam daje korak od oko 1 [mA] na izlazu (programski kontrolisana vrednost u memoriji: $2048/100 \approx 20$ koraka, vrednost na izlazu: $20 \text{ [mA]} / 20 \text{ koraka} = 1 \text{ [mA / koraku]}$).
- Pošto želimo da se na jedan pritisak tastera vrednost promeni za jedan korak, ne smemo to raditi na signal sa tastera, već samo na opadajuću ivicu signala sa tastera.
- To znači da će do promene doći samo prilikom otpuštanja tastera.
- Zato uvodimo simbole gore i dole koji će biti aktivni samo u momentu otpuštanja tastera što ćemo programski rešiti pomoću instrukcije za detekciju opadajuće ivice DIFD.
- Ostanak vrednosti signala u sigurnim granicama ćemo obezbediti tako što uvećanje i smanjenje vrednosti vršimo samo kada znamo da vrednost posle uvećanja ili smanjenja neće izaći iz sigurnog intervala. Upoređivaćemo vrednost signala sa granicama sigurnog intervala. Te granice će biti 100 i 1900 desimalno.

- U programu koristimo sledeće simbole:

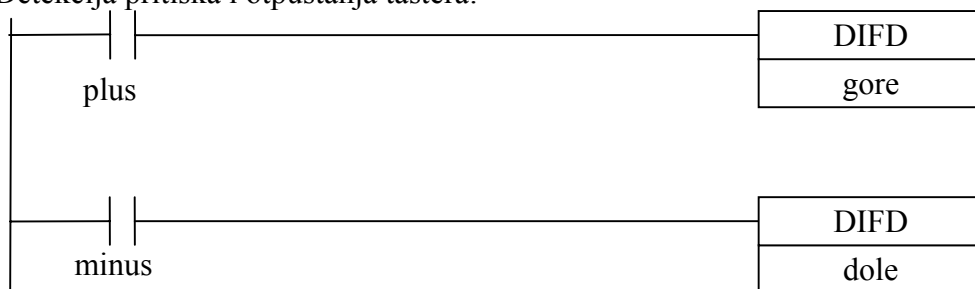
Ime	Tip	Adresa	Opis
plus	BOOL	1.15	Taster za povećanje vrednosti signala
minus	BOOL	1.14	Taster za smanjenje vrednosti signala
signal	UINT	DM101	Željena vrednost signala na izlazu
izlaz	CHANNEL	236	Adresa pomoću koje se podešava vrednost analognog izlaza
gore	BOOL	150.00	Detekcija opadajuće ivice tastera plus
dole	BOOL	150.01	Detekcija opadajuće ivice tastera plus

Program možemo podeliti na nekoliko celina:

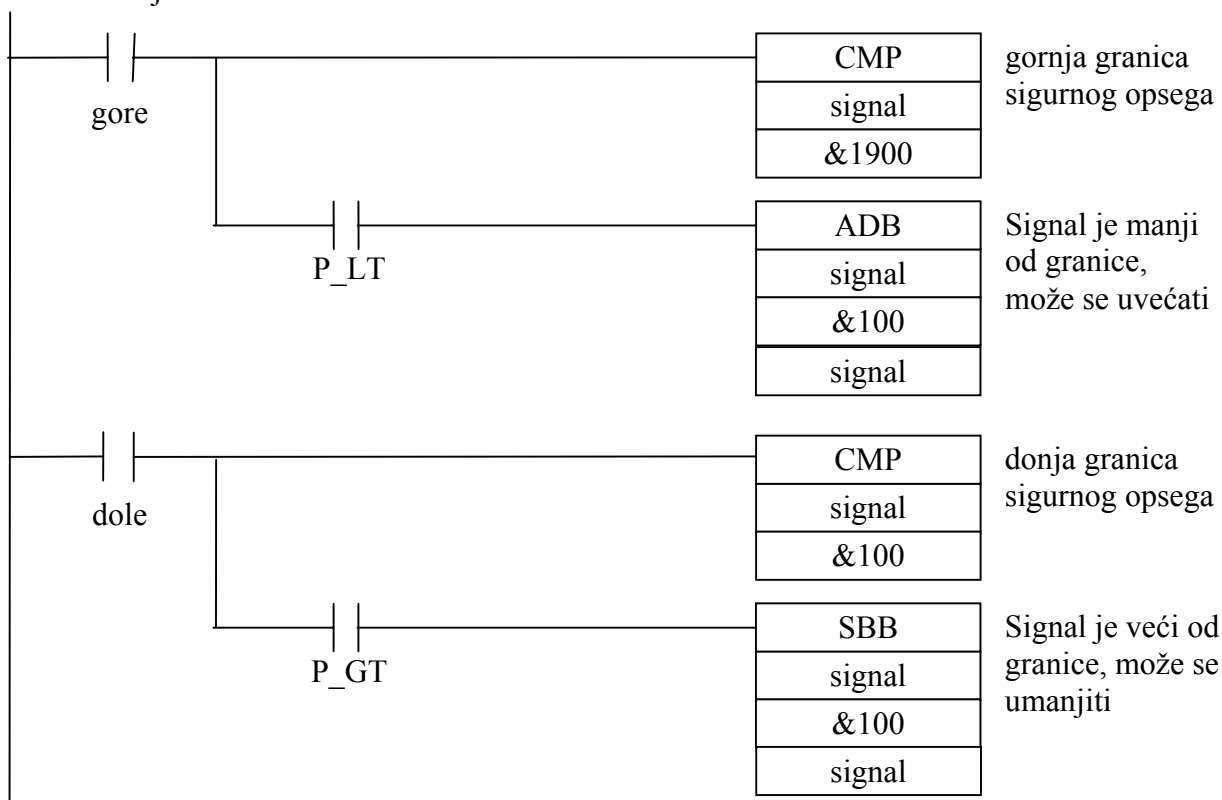
Inicijalizacija:



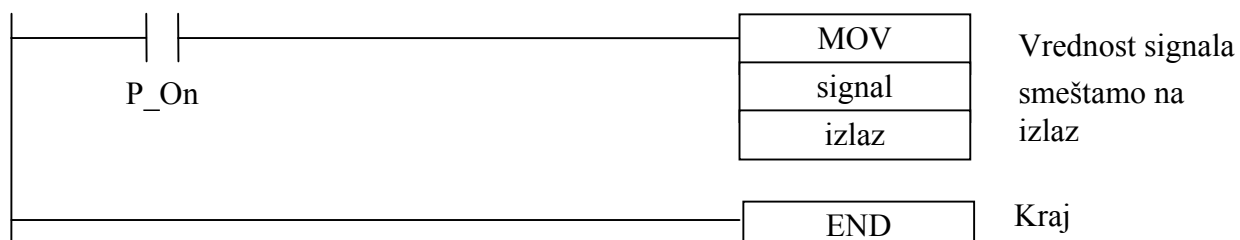
Detekcija pritiska i otpuštanja tastera:



Promena željene vrednosti:



Još je potrebno željenu vrednost postaviti na izlaz:



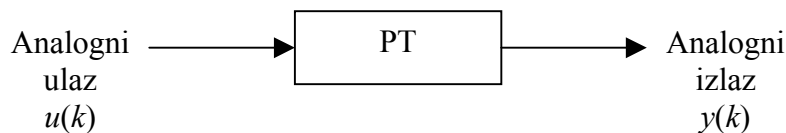
6. korak:

Praćenje rada programa.

Proverite vrednosti na analognom izlazu za sve moguće slučajeve. Isprobajte i druge vrednosti koraka ali obratite pažnju na granice. Donja granica ne sme biti manja od koraka, a gornja ne sme biti veća od (2047-korak).

7.4. Primer 4.**Zadatak:**

Realizovati lestvičasti dijagram koji će simulirati PT član na sledeći način:



Proporcionalan član sa kašnjenjem prvog reda je veoma čest element u sistemima automatskog upravljanja. Zato je zgodno realizovati njegovu simulaciju pomoću PLC lestvičastog dijagrama. Ponašanje člana sa kašnjenjem prvog reda (PT1-član) opisuju sledeće jednačine :

1) Prenosna funkcija :

$$W(p) = \frac{k_{ob}}{Tp + 1}$$

2) Diferencijalna jednačina :

$$T \frac{dy}{dt} + y = k_{ob} u$$

3) Diferentna jednačina :

$$a_0 \cdot y(k) + a_1 \cdot y(k-1) = b_0 \cdot u(k)$$

Gde je T_0 vreme uzorkovanja a koeficijenti su određeni izrazima :

$$a_0 = \frac{T}{T_0} + 1 \quad ; \quad a_1 = -\frac{T}{T_0} \quad ; \quad b_0 = k_{ob}$$

Iz diferentne jednačine sledi da je $y(k)$ odnosno odziv PT1 člana :

$$y(k) = \frac{b_0 \cdot u(k) - a_1 \cdot y(k-1)}{a_0}$$

Na osnovu diferentne jednačine (3) PT1 člana napisaćemo program koji realizuje PT1 član u floating-point aritmetici.

1. korak:

Određivanje opsega i broja analognih ulaza i izlaza.

Koristićemo prvi analogni ulaz i prvi analogni izlaz modula MAB42. Za ulaz ćemo koristiti naponski opseg od -10 do 10 [V].

2. korak:

Povezivanje ulaznih i izlaznih tačaka sistema.

Povezivanje izvršiti u skladu sa pravilima vezivanja za MAB42. Generator signala povezati na analogni ulaz 1. Obratiti pažnju na način vezivanja naponskog ulaza. Merni instrument povezati na analogni izlaz vodeći računa da je to naponski izlaz.

3. korak:

U lokaciju DM 6611 koja služi kao kontrolni registar modula MAB-42 treba staviti sledeću vrednost:

binarno '0000 1110 0000 0000'

Odnosno hex ' 0 E 0 0 '

To se može uraditi iz "CX-programmera" na već navedeni način.

4. korak:

Podešavanja opsega izlaznog signala ne postoje. Opseg je fiksiran na -10 [V] do +10 [V].

5. korak:

Izrada programa za PLC

Program možemo realizovati na sledeći način. Trenutnu vrednost ulaza (koja je za pozitivnu vrednost napona na ulazu 16-bitni binarni broj od 0 do 7FF) ćemo konvertovati u floating point i zatim podeliti sa konstantom 2047 (7FFh) da bi dobili floating point vrednost u opsegu od 0 do 1 što je ustvari vrednost ulaznog napona svedena na navedeni opseg. Zatim ćemo izračunati vrednost izlaza PT člana u skladu sa diferentnom jednačinom i konvertovati je u binarni broj obrnutim postupkom (množenjem sa 2047 i konvertovanjem u 16-bitni binarni broj) koji ćemo postaviti na izlaz.

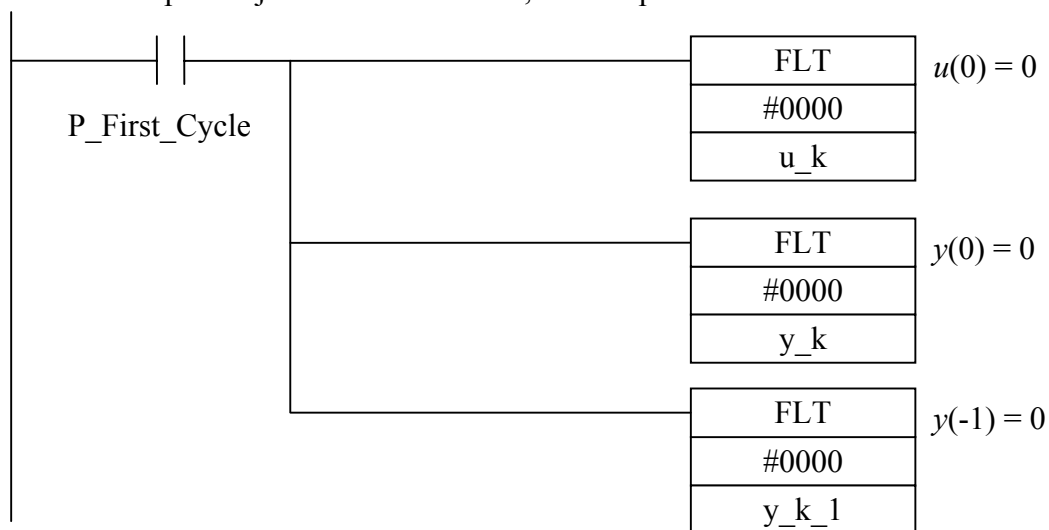
Definisaćemo sledeće simbole:

Ime	Tip	Adresa	Opis
izlaz	CHANNEL	236	vrednost analognog izlaza
ulaz	CHANNEL	232	vrednost analognog ulaza
u_k	UINT	DM100 i DM101	$u(k)$ normalizovana vrednost ulaza (0 - 1)
y_k	UINT	DM102 i DM103	$y(k)$ normalizovana vrednost izlaza (0 - 1)
y_k_1	UINT	DM104 i DM105	$y(k-1)$ prethodna vrednost izlaza (0 - 1)
a_0	UINT	DM106 i DM107	koeficijent a_0
a_1	UINT	DM108 i DM109	koeficijent a_1
b_0	UINT	DM110 i DM111	koeficijent b_0
temp	UINT	DM112 i DM113	privremena promenljiva
maksimum	UINT	DM114 i DM115	konvertovana maksimalna vrednost napona

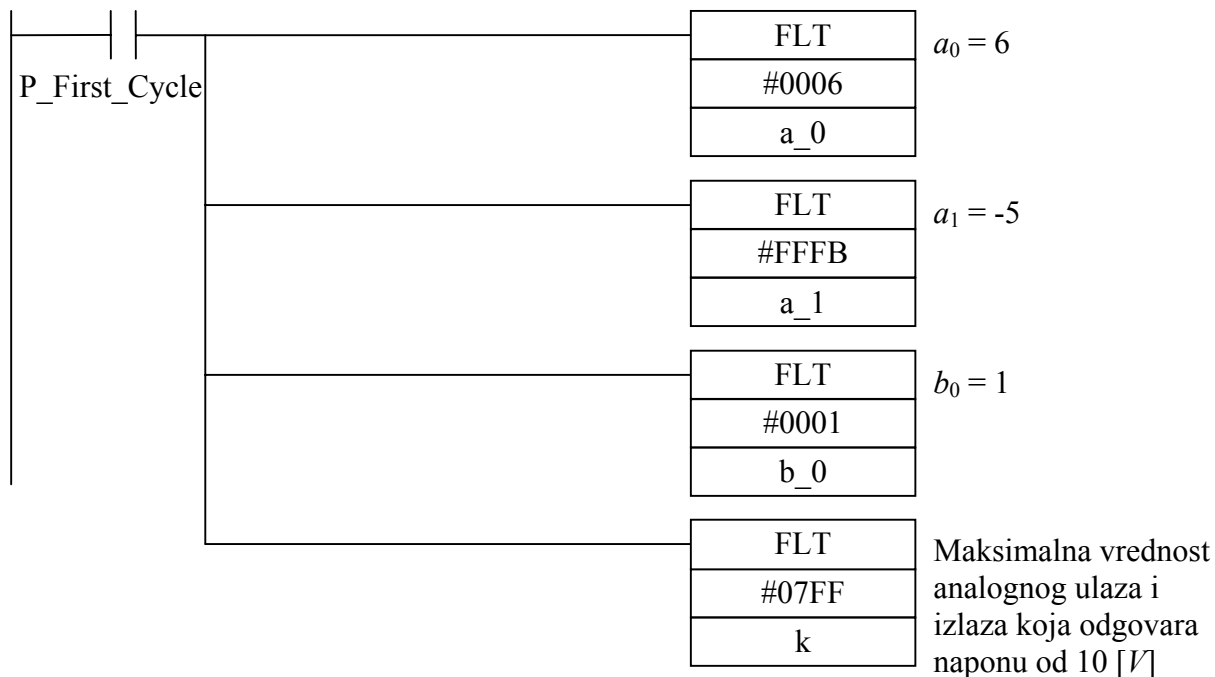
Primećujete da je za svaki floating point broj potrebno odvojiti dve memorijske lokacije. Kao simbol rezervisaćemo samo parne adrese. Neparne će automatski biti iskorišćene.

Na početku je potrebno postaviti početne vrednosti.

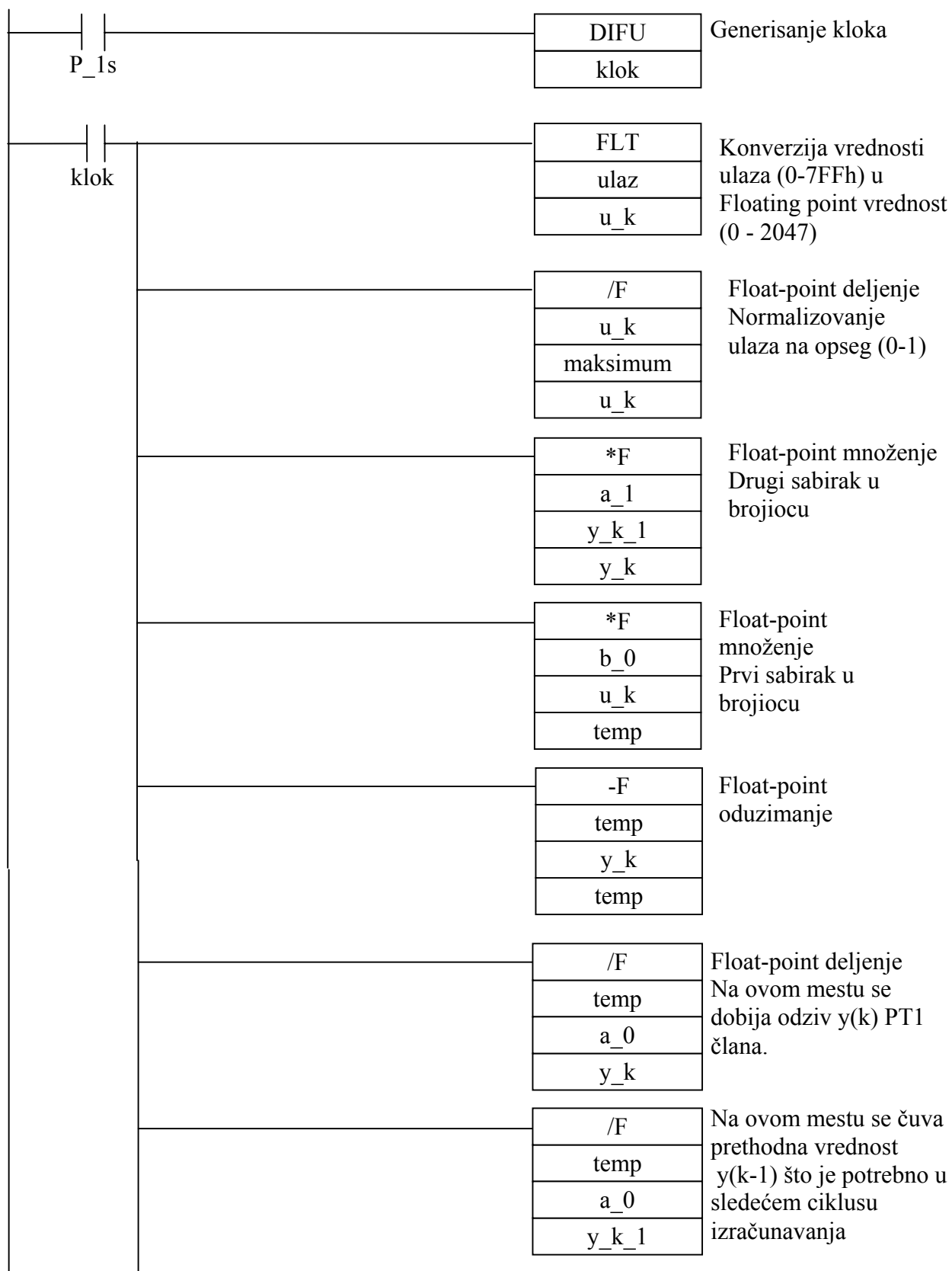
Prvo postavljamo vrednosti ulaza, izlaza i prethodne vrednosti izlaza na nulu:

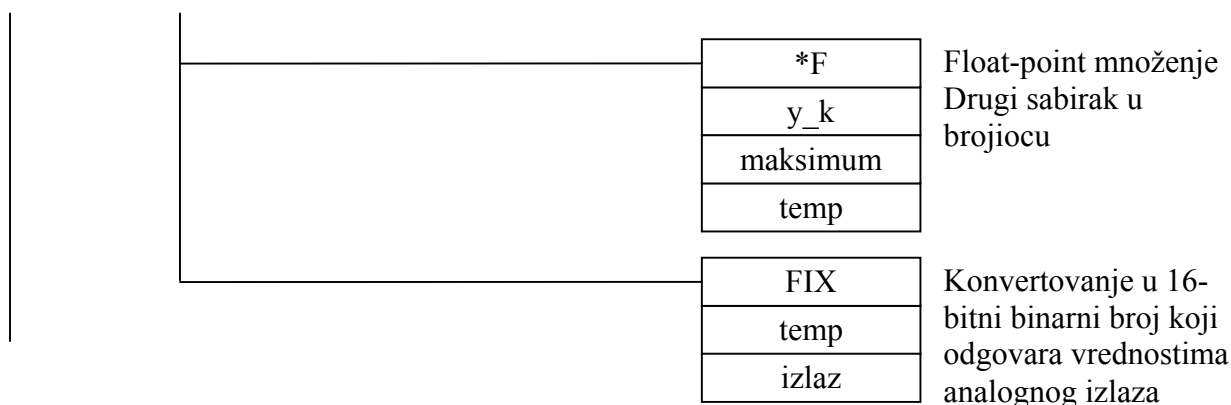


Zatim postavljamo vrednosti koeficijenata diferentne jednačine.



I glavni deo programa koji vrši izračunavanje diferentne jednačine.



**6. korak:**

Praćenje rada programa.

Generišite odskočnu funkciju na naponskom generatoru. Pazite da naponski nivo ne pređe 10 [V]. Izlazni napon pratite preko mernog instrumenta. Takođe je korisno pratiti odziv simuliranog PT člana pomoću time-chart dijagrama.

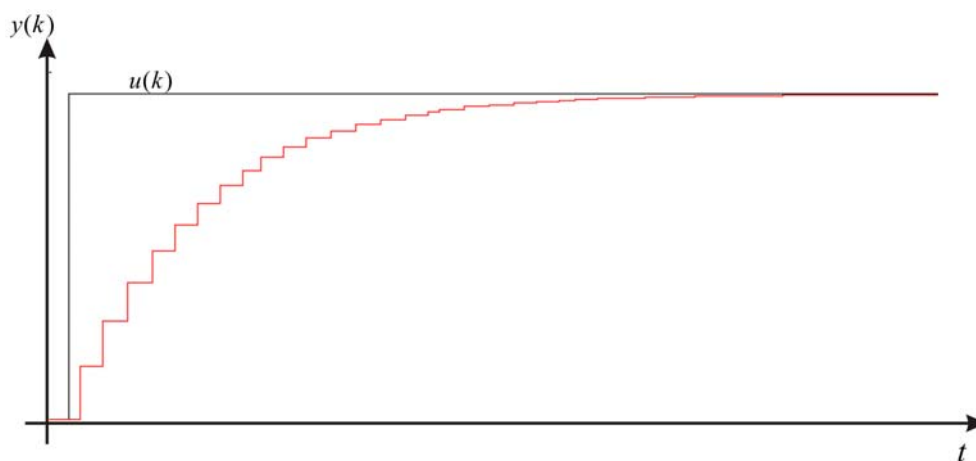
U CX-programmer-u u okviru menija PLC izaberite opciju "*Time chart monitoring*". U okviru njega konfigurišite pointe koje želite pratiti. U ovom primeru zanimaju nas vrednosti ulaza i izlaza. Na žalost ne postoji mogućnost praćenja floating-point vrednosti.

Jednačina PT1-člana, i njegovi koeficijenti čiji je odziv dat na sledećoj slici glasi :

$$W(p) = \frac{1}{5p + 1}$$

$$a_0 = \frac{5}{1} + 1 = 6 \qquad b_0 = 1$$

$$a_1 = -\frac{5}{1} = -5 \qquad T_0 = 1[s]$$



Odziv PT1-člana

7.5. Primer 5.

Zadatak:

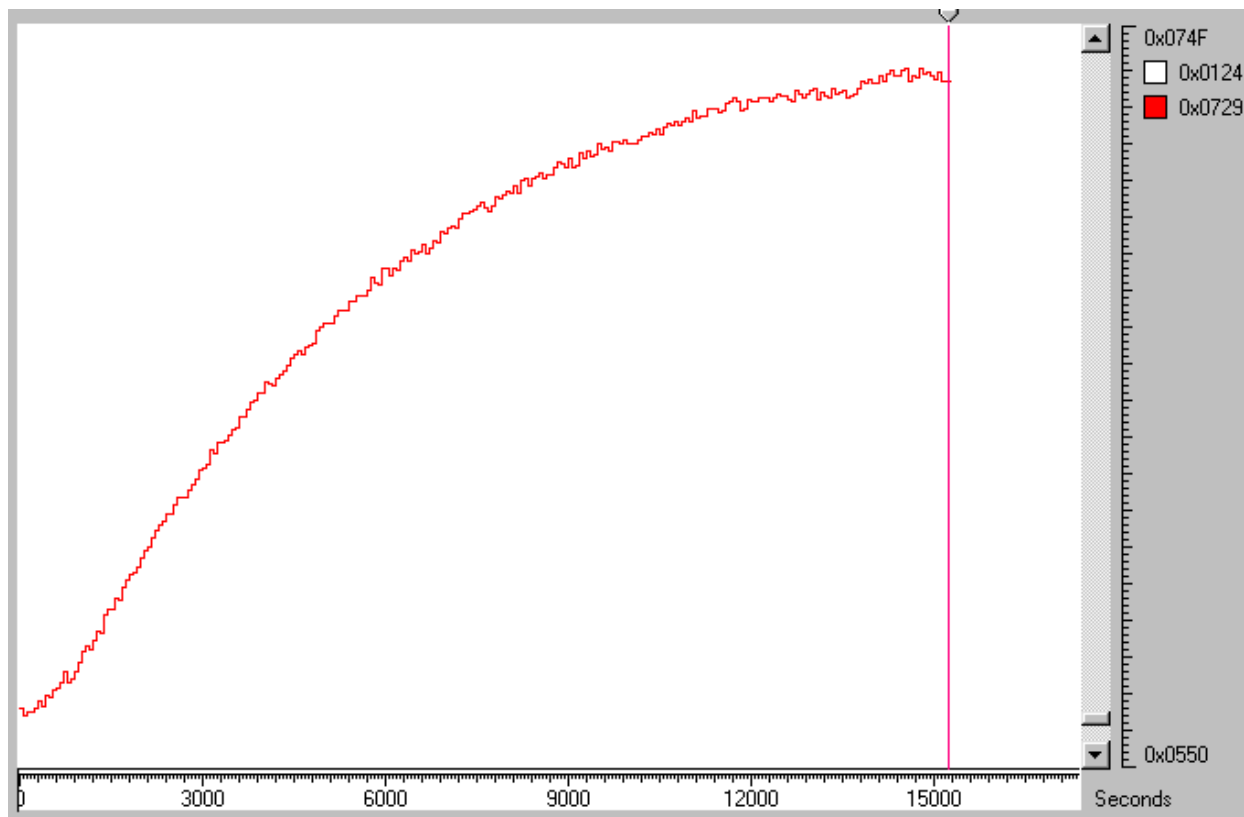
Identifikacija parametara matematičkog modela realnog objekta upravljanja.

Identifikacija predstavlja merenje i obradu rezultata merenja u cilju utvrđivanja karaktera i parametara matematičkog modela koji može reprezentovati objekat upravljanja nad kojim su merenja završena.

Identifikacija je izvršena na peći za kaljenje. Temperaturna karakteristika peći je nelinearna. Identifikacija parametara peći mora se izvršiti na radnoj temperaturi zato što onda možemo aproksimirati realnu karakteristiku peći kao linearnu. Uzeli smo radnu temperaturu od 100 [°C]. Za nju možemo smatrati opseg od 70 [°C] do 130 [°C] kao linearan. Da bi identifikaciju pravilno izvršili skokovita promena mora biti u tom opsegu. To smo postigli skokovitom promenom napona sa 100 [V] na 150 [V].

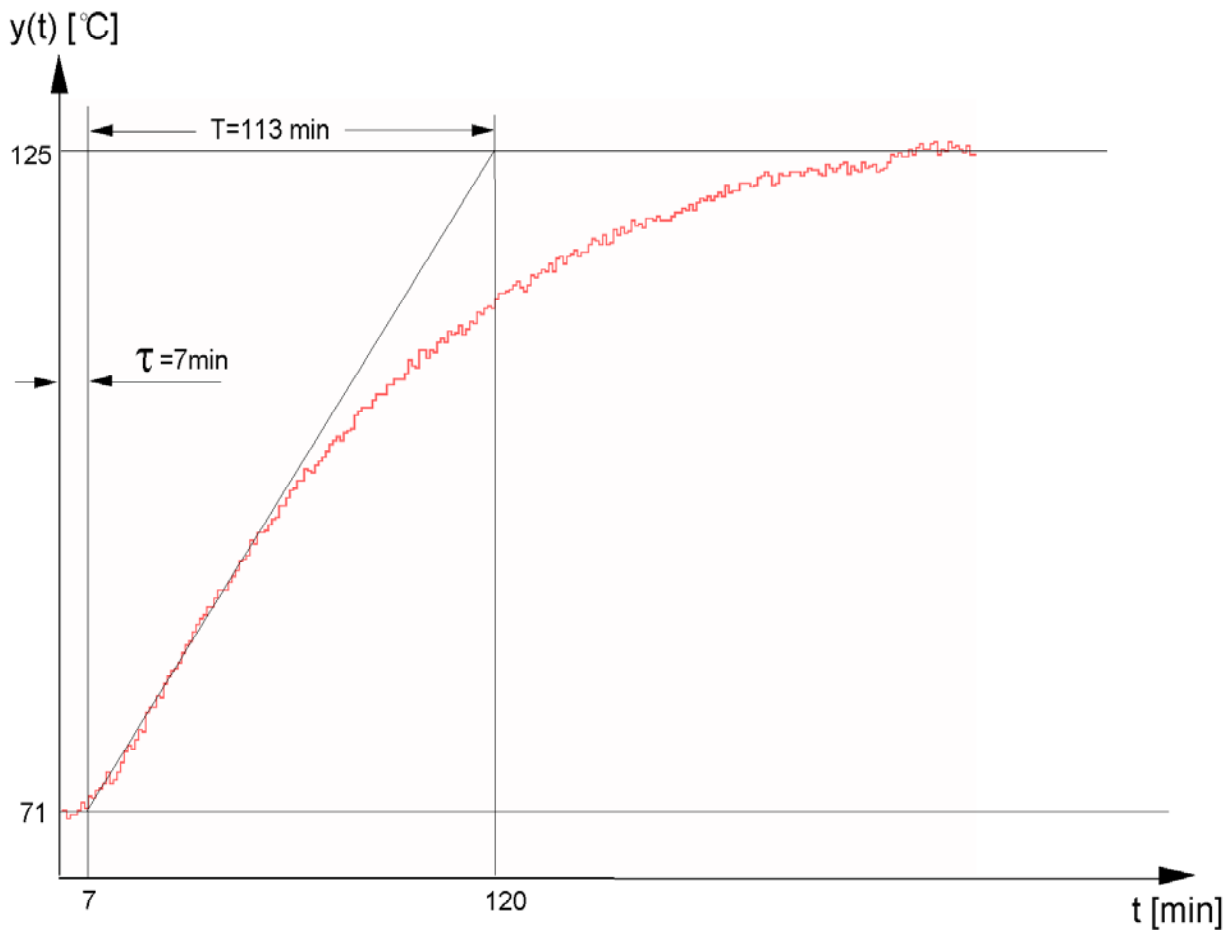
Na grejač je prvo doveden napon od 100 [V] i pri tome se peć zagrejala na 71 [°C]. Kada se temperatura ustalila na grejač je dovedena jedinična skokovita promena (promena napona sa 100 [V] na 150 [V]) i tada se temperatura u peći menjala na način kao što je prikazano na slici 4.1. i dostigla maksimalnu vrednost od 125 [°C].

Sledeća slika predstavlja rezultat merenja obavljenog pomoću "Time chart" dijagrama. X osa predstavlja vreme od početka skokovite promene. Y osa predstavlja podatak sa analognog ulaza modula MAB-42 u hexadekadm brojnom sistemu.



Time chart rezultati merenja.

Na osnovu ove slike mogu se proceniti parametri peći metodom tangente i dobijeno je da je mrtvo vreme $\tau = 7$ [min] a vremenska konstanta $T = 113$ [min].



Prelazna karakteristika peći

Prenosna funkcija peći se izračunava sledećim postupkom:

$$y(\infty) = 125[^\circ\text{C}] - 71[^\circ\text{C}] = 54[^\circ\text{C}]$$

$$u(\infty) = 150[\text{V}] - 100[\text{V}] = 50[\text{V}]$$

$$k_{\text{ob}} = \frac{y(\infty)}{u(\infty)} = \frac{54[^\circ\text{C}]}{50[\text{V}]} = 1.08 \left[\frac{^\circ\text{C}}{\text{V}} \right]$$

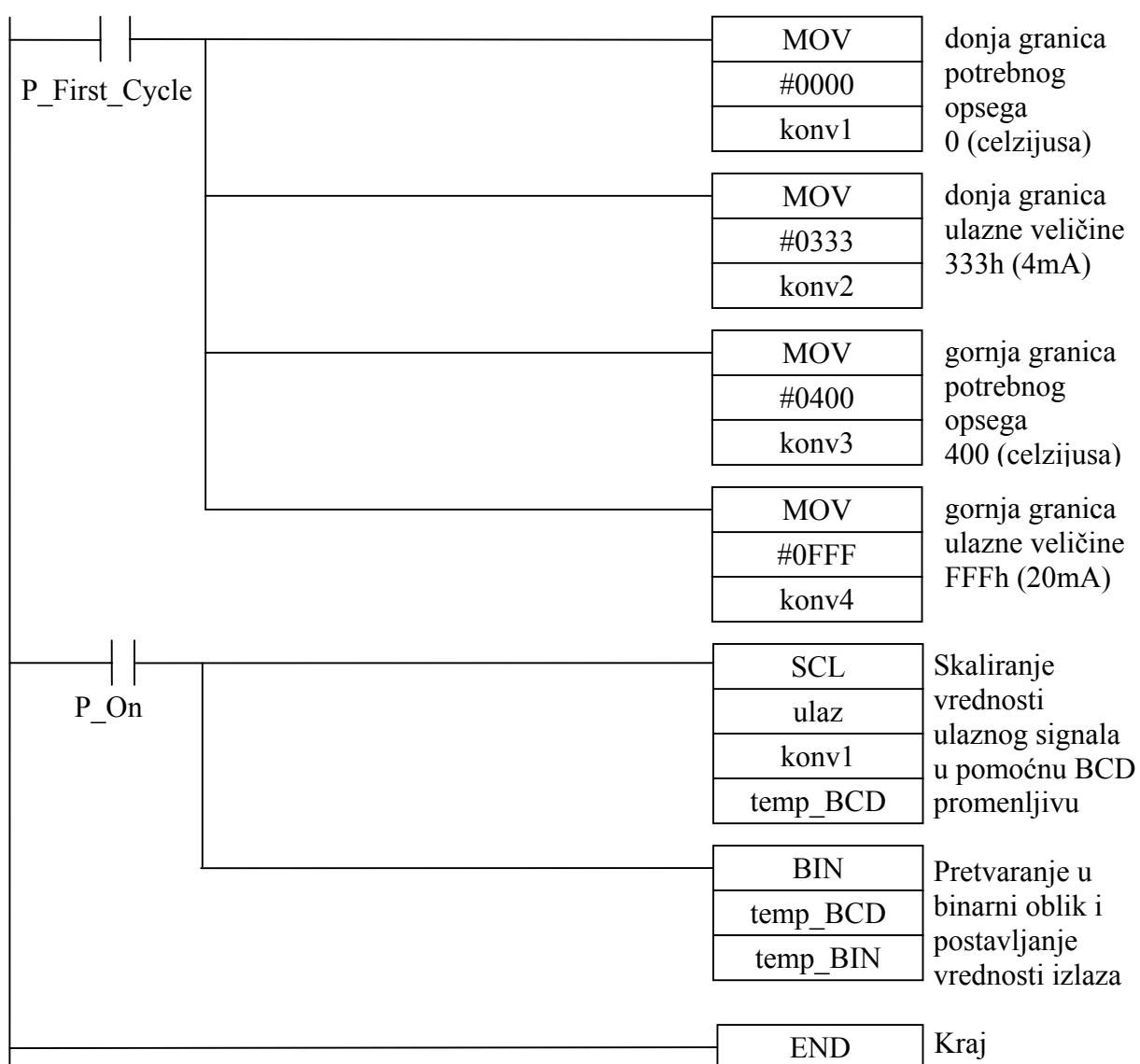
$$W(p) = \frac{k_{\text{ob}}}{T_{\text{ob}}p + 1} e^{-\tau_{\text{ob}}p} \approx \frac{k_{\text{ob}}}{(T_{\text{ob}} + \tau_{\text{ob}})p + 1} = \frac{1.08}{(113 + 7)p + 1} = \frac{1.08}{120p + 1}$$

$$W(p) = \frac{1.08}{113 \cdot 60p + 1} e^{-7 \cdot 60p} = \frac{1.08 \left[\frac{^\circ\text{C}}{\text{V}} \right]}{6780[s]p + 1} e^{-420p}$$

Prethodne jednačine prikazuju matematički model peći koja je bila predmet identifikacije. Iz tih jednačina je moguće proračunati parametre regulatora kao i izvršiti simulacije u MatLab programskom paketu.

Za merenje temperature korišćen je Pt100 otpornik, kao senzor, povezan na merni pretvarač. Izlaz iz mernog pretvarača je vođen na strujni analogni ulaz PLC-a (CQM1H) gde je merena temperatura. Pošto je izlaz iz mernog pretvarača od 4 [mA] do 20 [mA] a ulaz PLC-a od 0 [mA] do 20 [mA] bilo je potrebno izvršiti skaliranje signala. To je postignuto (SCL) instrukcijom što se može videti na listingu programa koji je korišten pri merenju.

Definisani su simboli temp_BCD i temp_BIN kao vrednosti temperature. BCD vrednost je rezultat skaliranja ulazne vrednosti sa AD konvertora, a BIN vrednost je potrebna zato što "Time chart" dijagram nema mogućnost praćenja BCD vrednosti.



Na sličan način se može izvršiti identifikacija mnogih drugih objekata upravljanja.

OSNOVI PRIMENE PROGRAMABILNIH TERMINALA

1. Cilj laboratorijske vežbe:

- Upoznati se sa načinom funkcionisanja i primene programabilnog terminala “NT11S” firme “Omron”.
- Upoznati se sa softverskim paketom za programiranje programabilnih terminala “NT-series Support Tool”.
- Odvežbati priloženi primer upravljanja primenom programabilnog terminala “NT11S”.

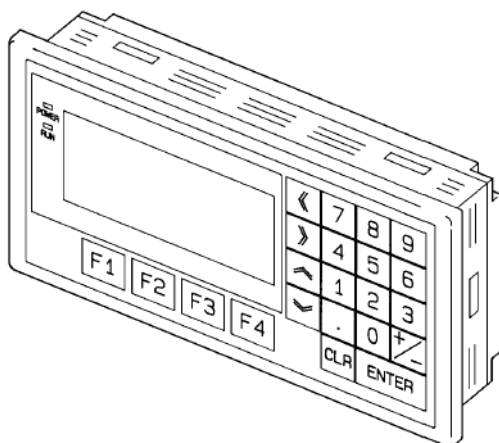
2. Laboratorijska vežba se sastoji od:

- PC računar Pentium I sa instaliranim softverom NT-series Support Tool.
- Tabla sa programabilnim terminalom NT11S.

3. Uvod

Nakon programiranja PLC kontrolera i prelazak u RUN režim rada u mnogim slučajevima više nije potreban monitoring sa PC računarom u daljem radu. U ovim slučajevima se monitoring obavlja sa programabilnim terminalom. Uz pomoć terminala prati se dalji rad PLC kontrolera, proverava se stanje ulaza i izlaza, unose podaci u kontroler, isčitavaju se podaci iz njega a sem toga podaci se vizualno prikazuju.

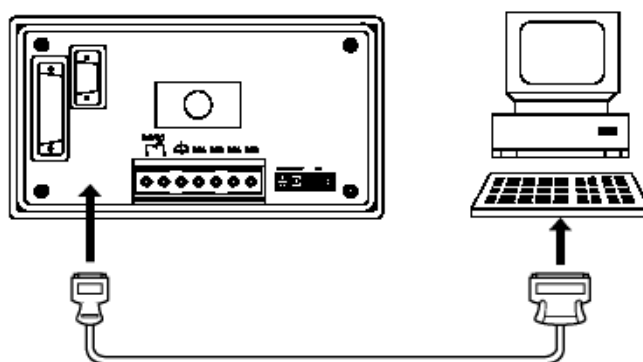
Programabilni terminal NT11S firme Omron je samostalna mikrokontrolerska programabilna jedinica u sistemu upravljanja sa LCD ekranom, tasterima za upravljanje i komunikacionim portovima (RS-232C, RS-422A printer port). Terminal sadrži ugrađeni Flash RAM memoriju, koja ne zahteva napajanje (dodatne baterije). Za komunikaciju sa PLC kontrolerom koristi Host Link ili NT Link standard. Za programiranje programabilnog terminala koristi se PC računar sa softverskim paketom NT-series Support Tool.



Slika 1. Programabilni terminal NT11S

NT11S je, što se dimenzije tiče, najmanji je u Omronovoj NT seriji (218mm x 113mm x 31 mm). Efektivna veličina ekrana je 100 mm (160 tačke) x 40 mm (64 tačke), sa uglom vidljivosti od 35 stepeni. Očekivani radni vek uređaja je minimum 50.000 sati (sa garancijom od 30.000 sati). Životni vek tastera (kojih ukupno ima 22) je 1.000.000 uključenja / isključenja.

4. Instaliranje sistemskog programa programabilnog terminala



Slika 2. Povezivanje PC računara i programabilnog terminala NT11S

PC računar i programabilni terminal NT11S povezuje se preko RS-232C porta uz pomoć odgovarajućeg kabla. Nakon povezivanja terminala sa računarom priključujemo napajanje terminala. Na displeju dobijamo sledeću poruku:

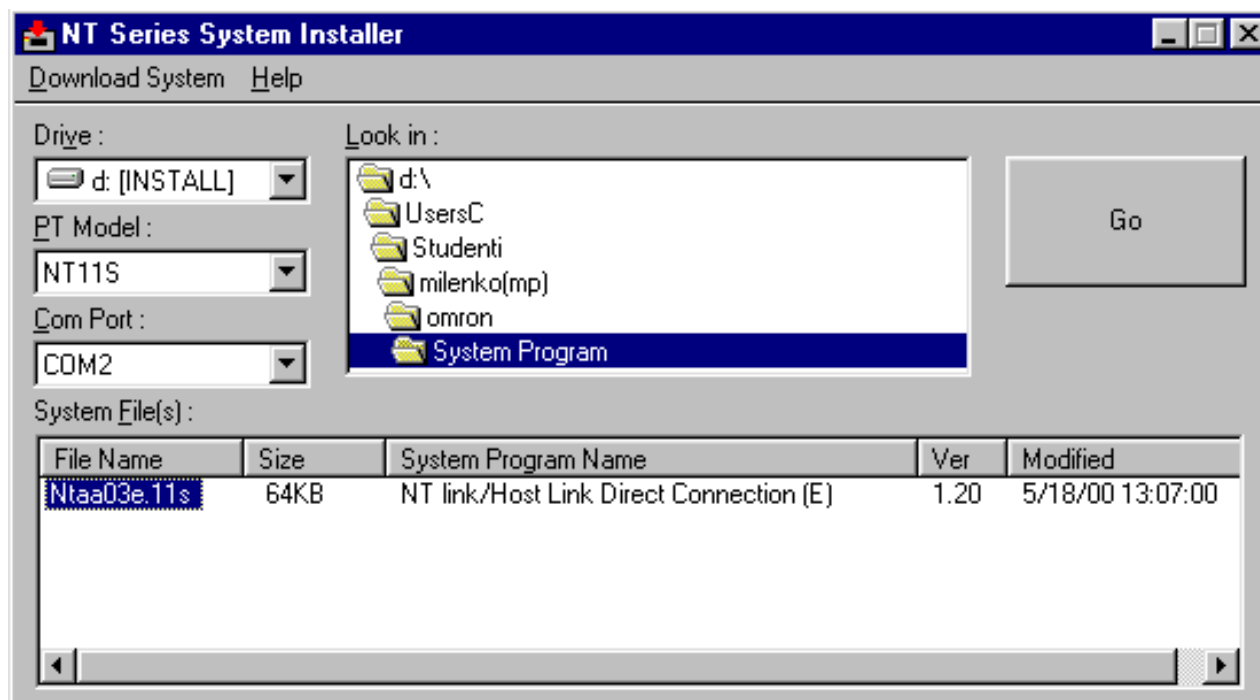
System Program
Isn't installed.
Press ENTER key
To start install.

Ako u memoriji terminala nije postavljen sistemski program treba ga upisati sa alatom: NT-series System Installer, koja se nalazi na ekranu kao prečica.



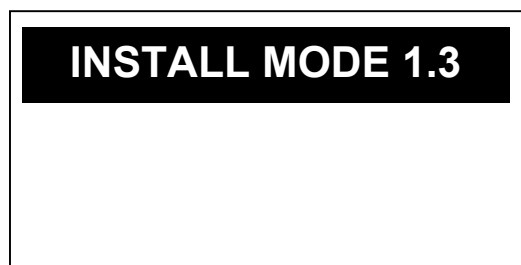
Slika 3. Prečica za NT-series System Installer

Duplim klikom na ovu ikonu otvara se sledeći prozor.

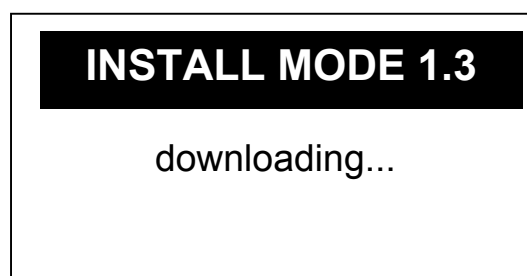


Slika 4. Određivanje parametara programabilnog terminala

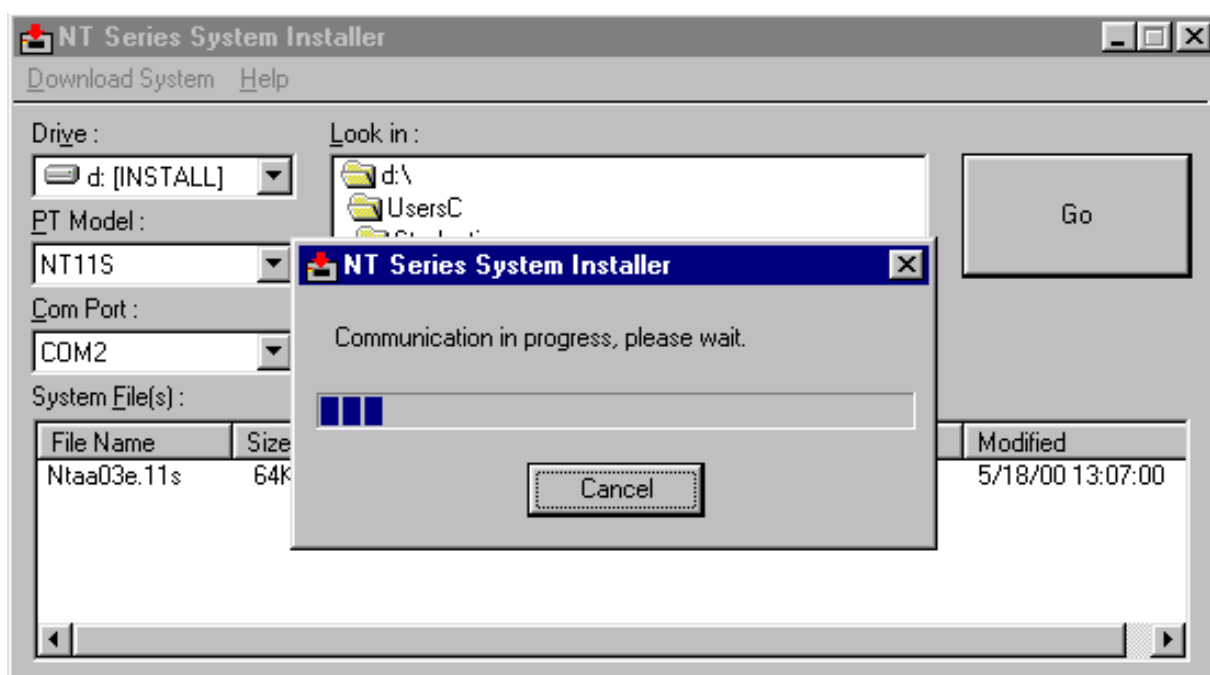
Pre podešavanja parametara potrebno je pritisnuti ENTER na terminalu. Nakon podešavanja klikom na taster GO dobijamo sledeću poruku na ekranu programabilnog terminala:



Pritiskom na taster ENTER program nam javlja da je u toku prenos podataka:



Takođe i na monitoru računara dobijamo poruku da se odvija prebacivanje (downloading) podataka:

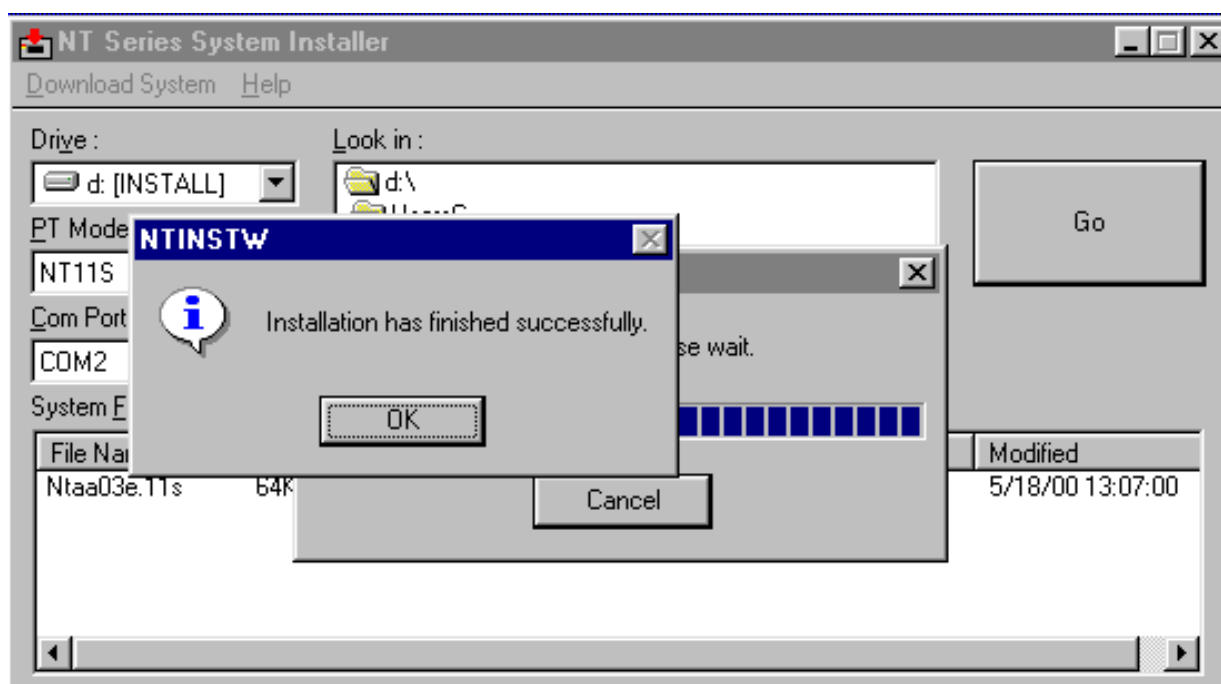


Slika 5. Prenos sistemskog programa u terminal

Na kraju dobijamo informaciju da je postupak uspešno obavljen:

**Downloading is
completed.**

Pres any key to run



Slika 6. Završena instalacija sistemskog programa programabilnog terminala

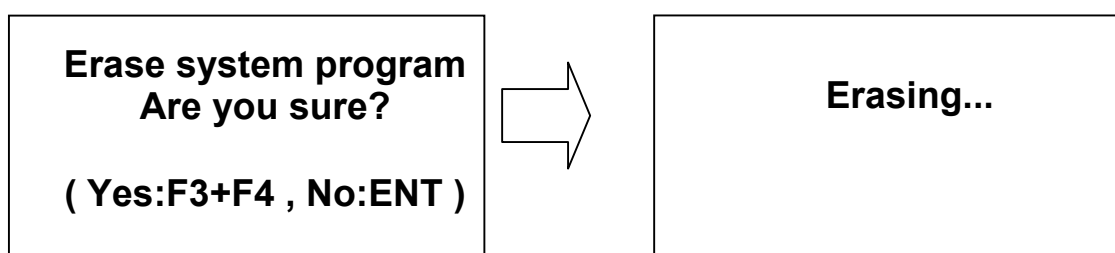
4.1. Brisanje sistemskog programa

Ako u nekom slučaju sami želimo da menjamo sistemski program, to možemo uraditi ako DIP prekidač SW2-6, sa unutrašnje strane terminala, postavimo u položaj ON.



Slika 7. Unutrašnja strana programabilnog terminala NT11S

Nakon što dovedemo napajanje dobijamo sledeću poruku na ekranu terminala.



Slika 8. Brisanje sistemskog programa terminal treba potvrditi sa pritiskom na tastere F3+F4

Nakon brisanja potrebno je vratiti DIP prekidač SW2-6 u OFF položaj i resetovati terminal sa tasterom pored SW2.

**Erasing is complited
Set DIP-SW2-6 to off
and press reset - SW**

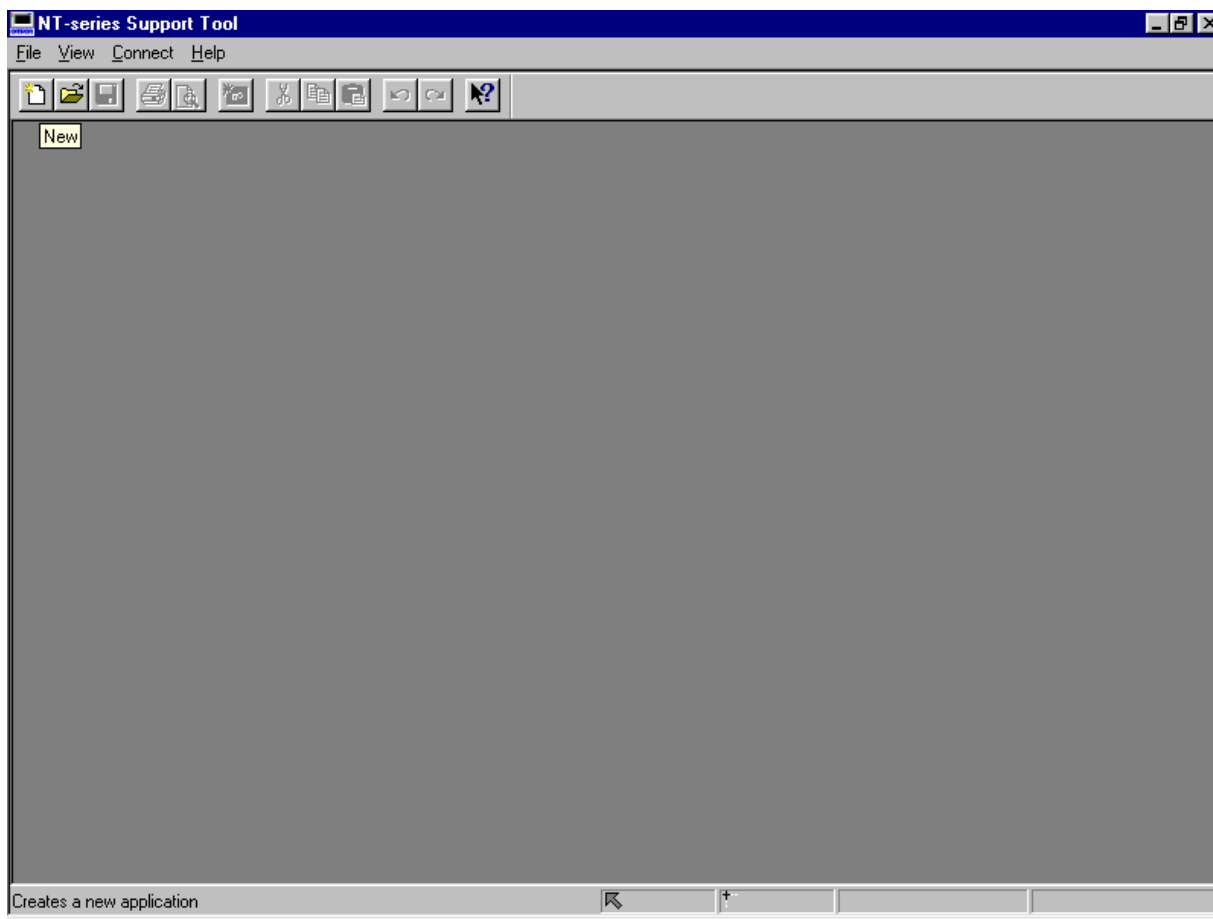
Naravno sada treba ponoviti prethodni postupak instaliranja novog sistemskog programa opisan na početku. Na kraju treba spomenuti da kad se to uradi treba ući u meni i proveriti PT adrese. U ovom slučaju Control Area je DM0400 a Notify Area u DM0800 (pogledati originalni priručnik NT11s Programmable Terminal : OPERATION MANUAL).

5. Softverski paket NT-series Support Tool

Softverski paket NT-series Support Tool poziva se preko prečice na ekranu.

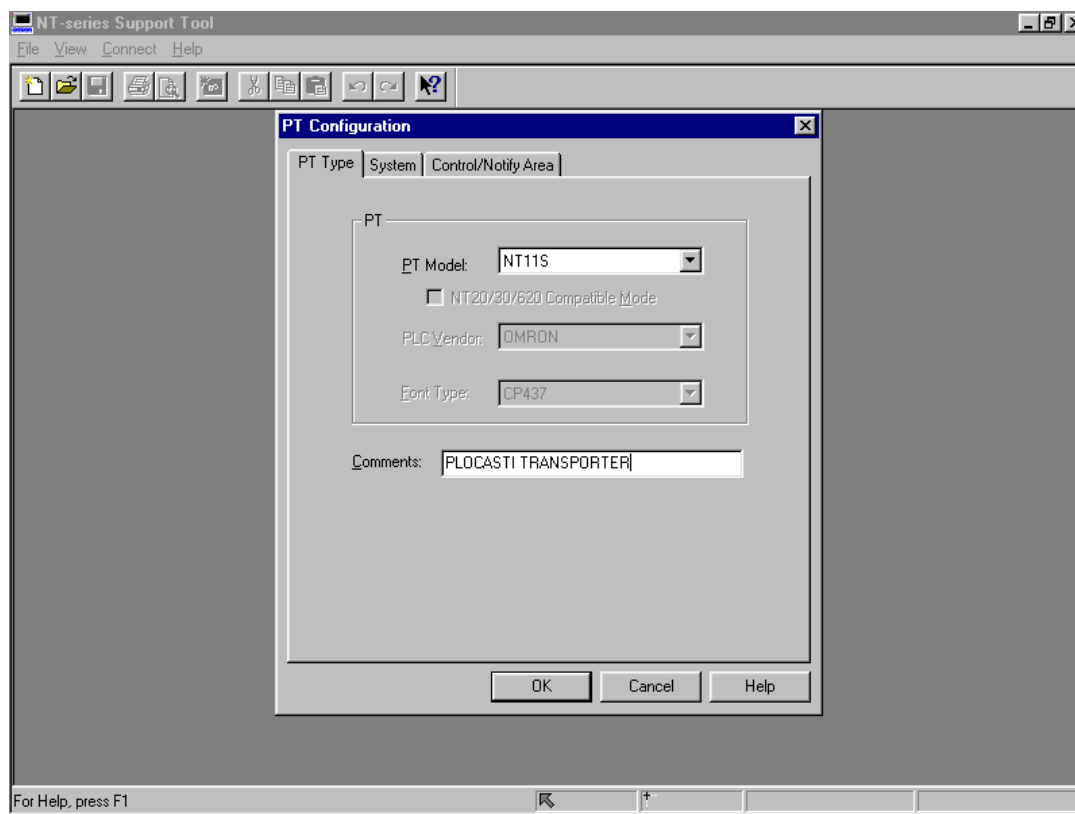


Duplim klikom na ovu ikonicu startujemo program i otvara nam se sledeći prozor:

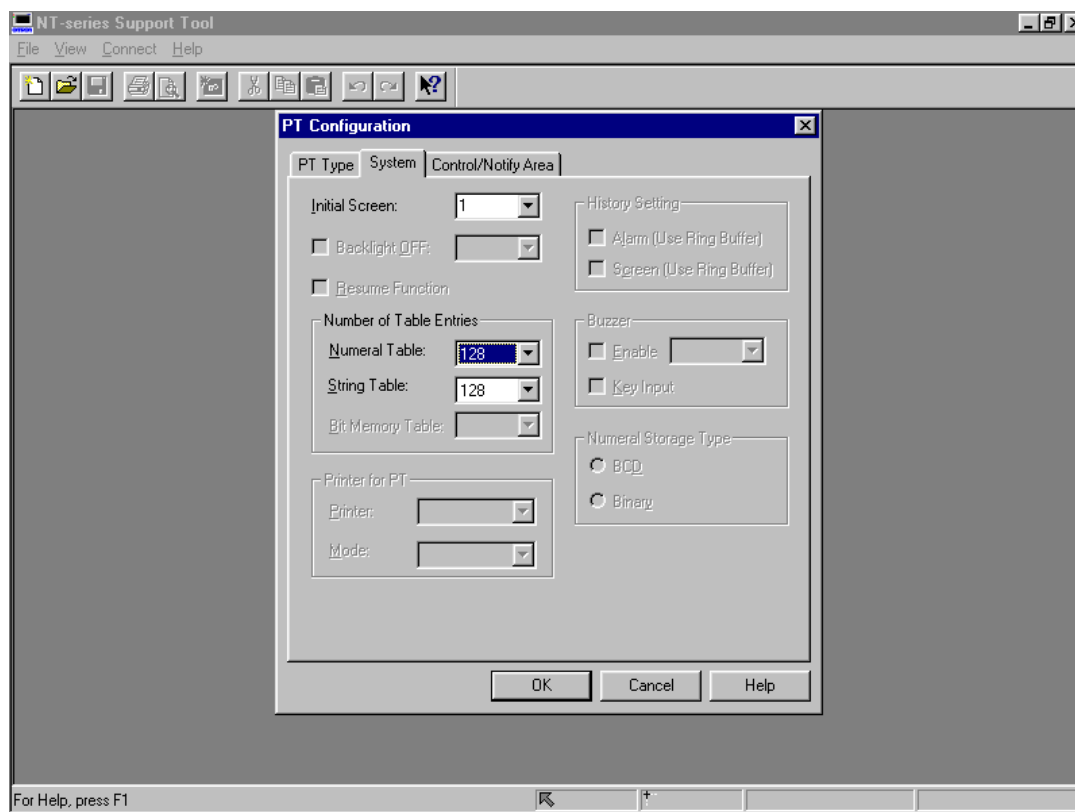


Slika 9. Prvi prozor nakon pokretanja programa

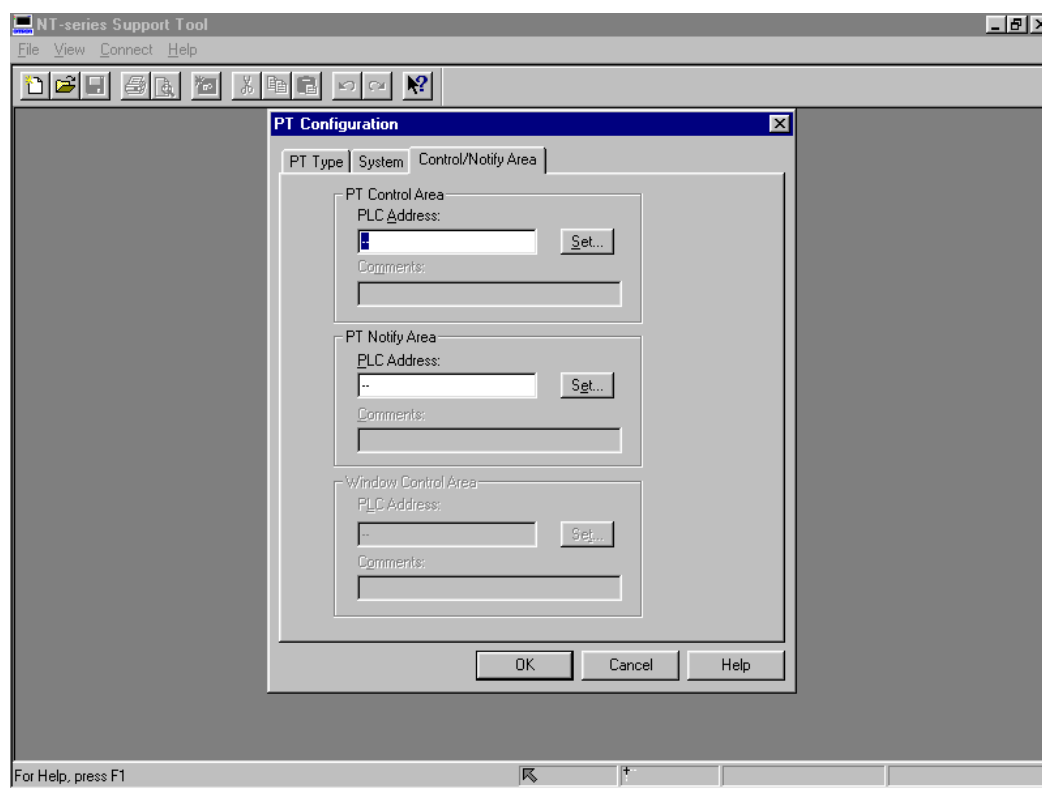
Biramo opciju “novi projekat” (NEW) na paleti sa alatima. Kad smo to uradili program nam otvara PT Configuration prozor gde podešavamo model terminala kojeg koristimo (PT Type). Zatim podešavamo System i Control/Notify Area na način prikazan na sledećim slikama.



Slika 10. Podešavanje modela programabilnog terminala



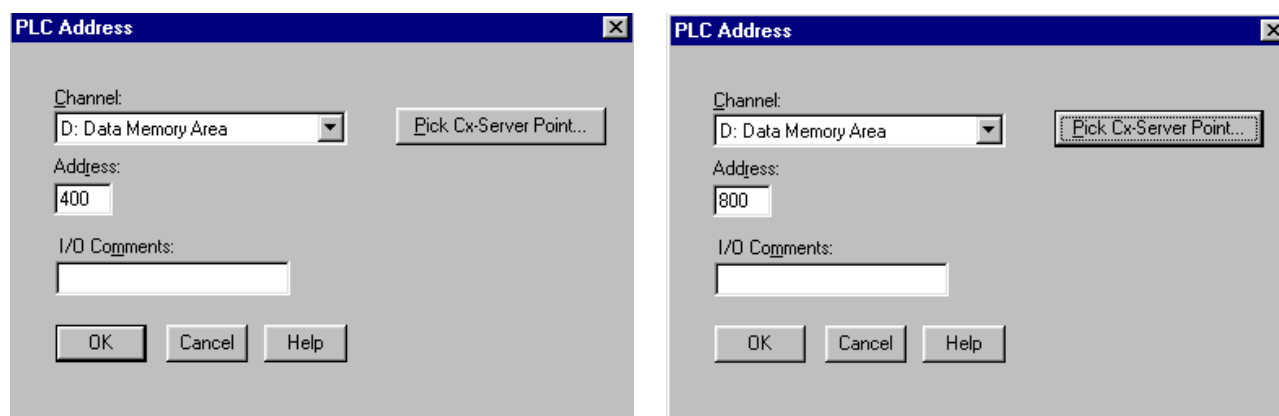
Slika 11. Dijalog box za podešavanje terminala - System



Slika 12. Dijalog box za podešavanje terminala - Control/Notify Area

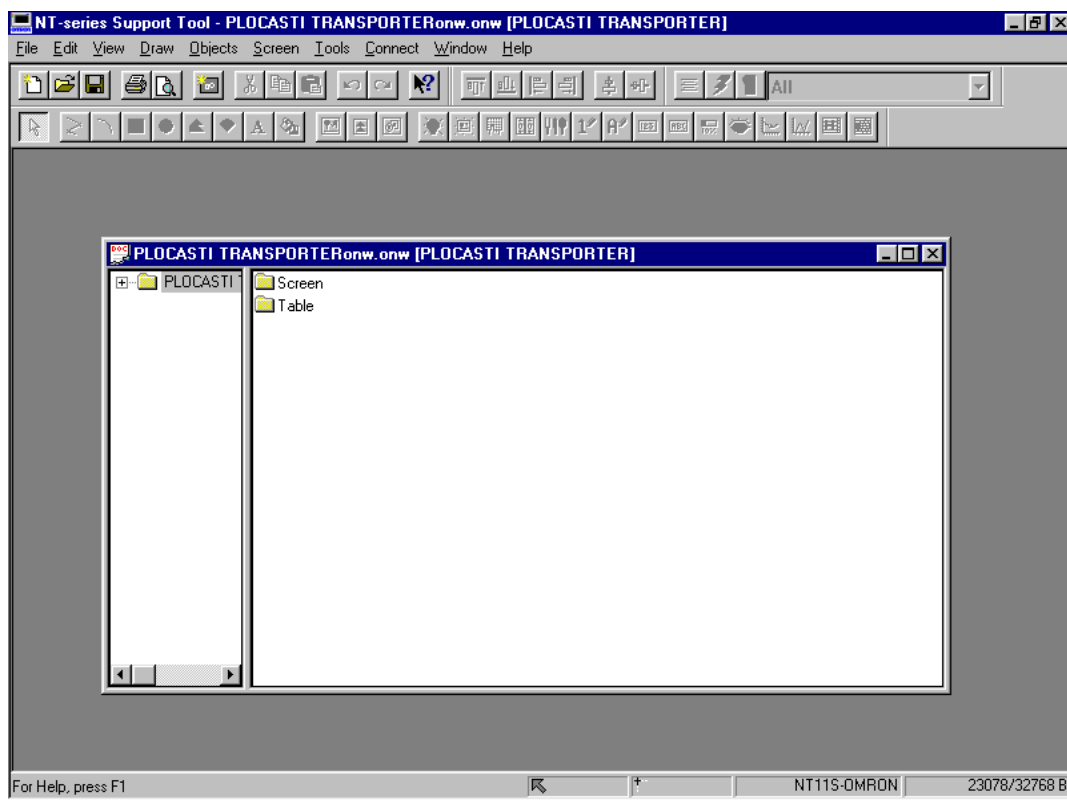
Pritiskajući set i direktno upisujući adresu (Address) ubacujemo vrednosti koje se nalaze u programibilnom terminalu.

Za programabilni terminal NT11S Control Area je DM 0400, a Notify Area je DM 0800.



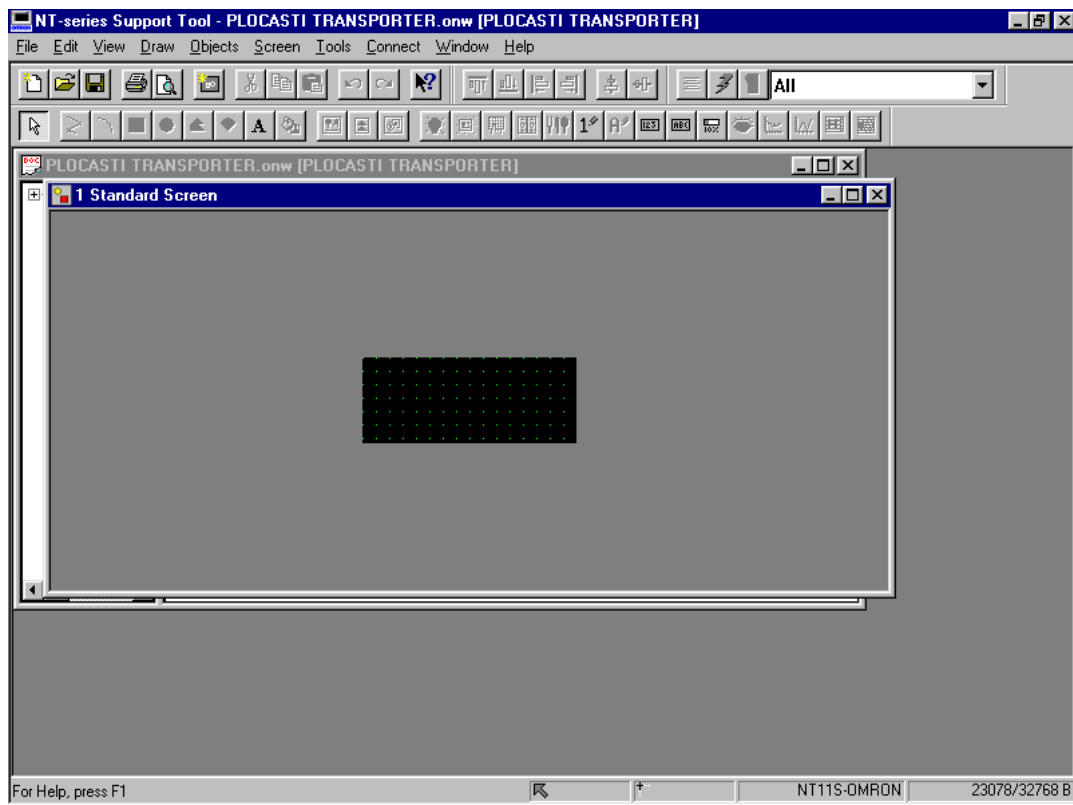
Slika 13. Dijalog box za podešavanje adresa – PLC Address

Posle ovih podešavanja pojavljuje se prozor projekta, prikazan na sledećoj slici. U folderu projekta nalaze se dva foldera: Screen (za standarne ekrane, lozinku i meni ekrane) i Table (za “štampanje” ekrana tj. opcija koja dozvoljava da se štampaju dnevni izveštaji, gde spada vreme displeja, redni broj ekrana, komentari koji su zadati u softveru za NT11S).



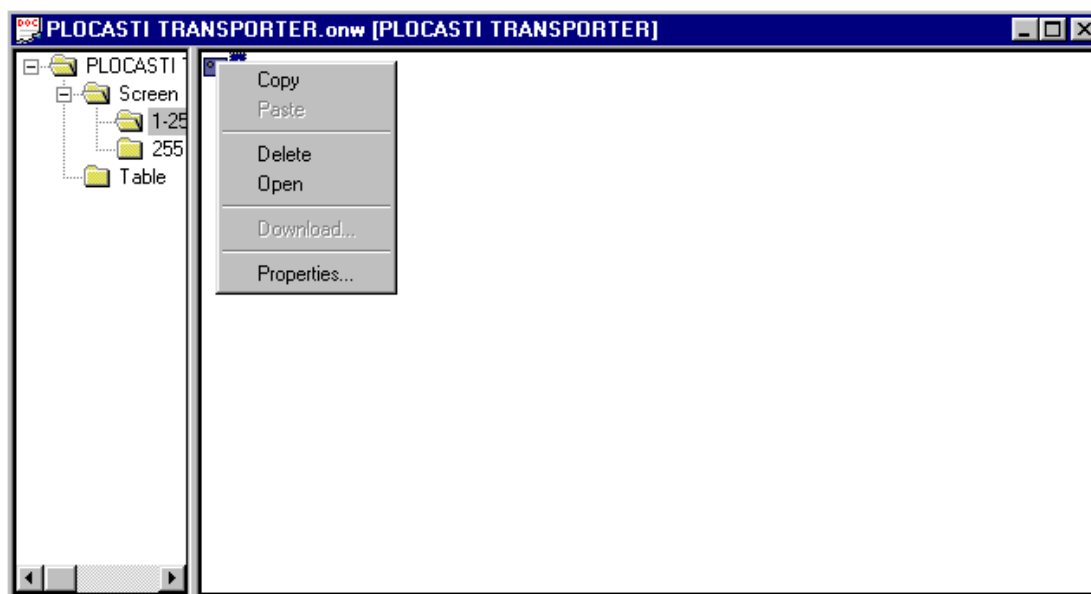
Slika 14. Izgled prozora projekta

Sledeći prozor koji se pojavljuje je prozor standardnog ekrana (Standard Screen) za rad.



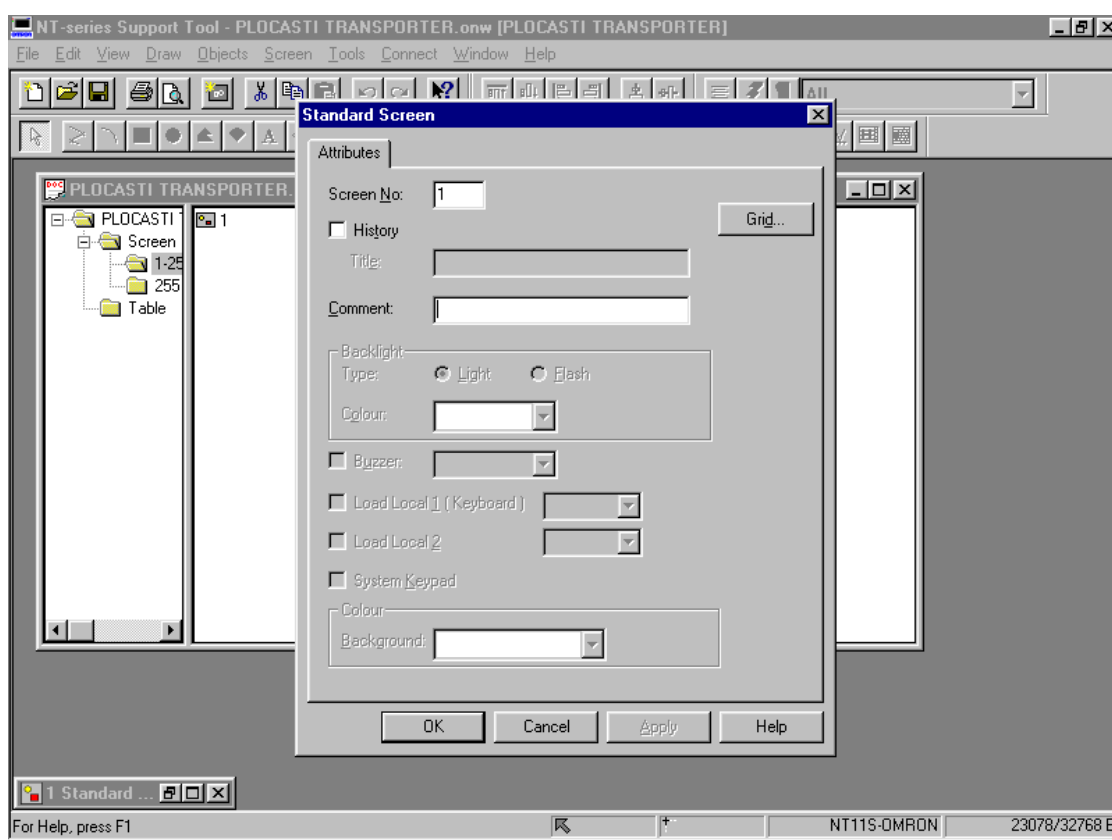
Slika 15. Izgled standardnog ekrana

Sledeći korak je podešavanje ekrana. Desnim klikom na ikonicu prvog ekrana koja se nalazi u folderu Screen/1-250 izaberemo Properties...



Slika 16. Ekran za izbor osobina

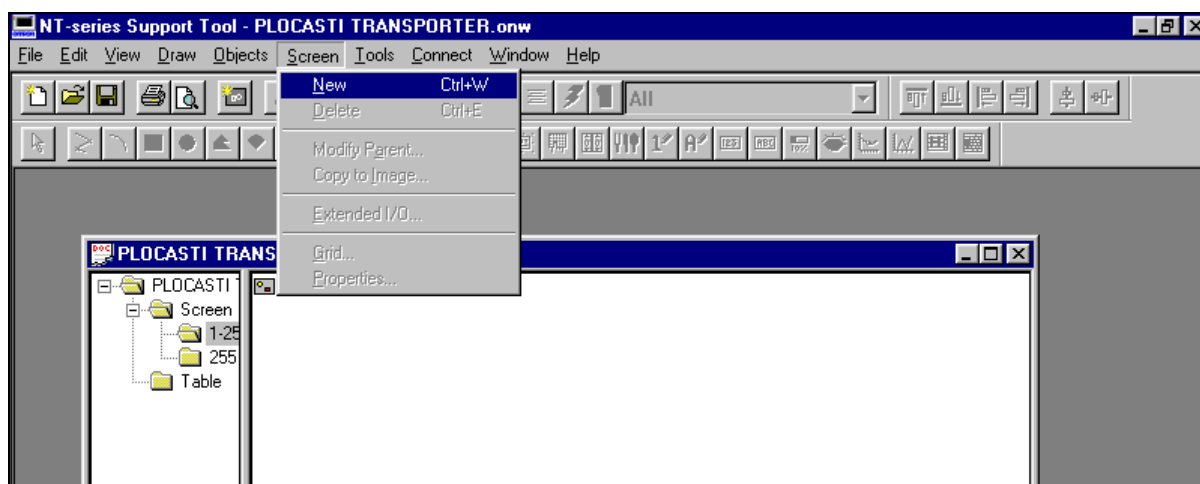
U dijalog box-u koji se pojavljuje možemo promeniti redosled broja ekrana (Screen No:), podesiti pomoćne tačke za crtanje (Grid...) ili upisati komentar (Comment:).



Slika 17. Podešavanje osobina standardnog ekrana

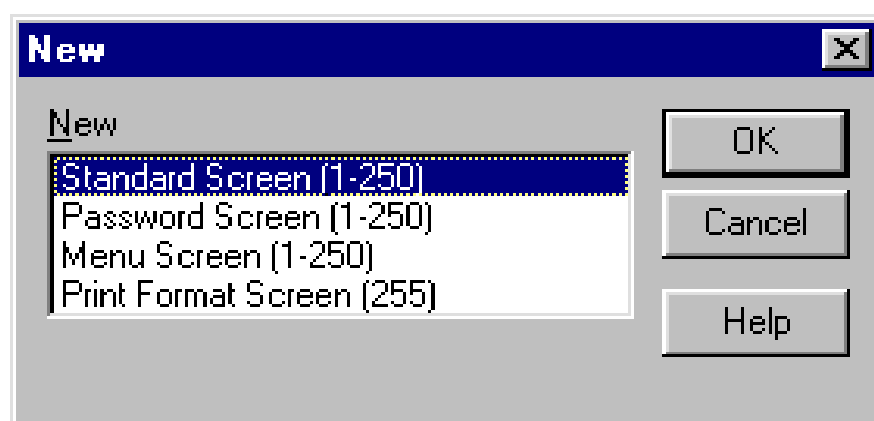
5.1. Kreiranje novog ekrana

Za kreiranje novih ekrana koristi se komanda New u opciji Screen.



Slika 18. Prozor za kreiranje novog ekrana

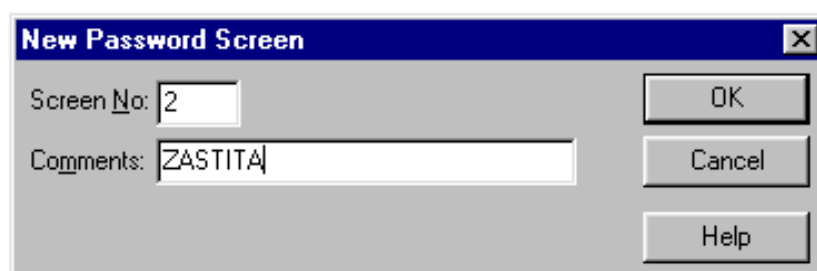
Tada dobijamo mogućnost da izaberemo vrstu ekrana koju ćemo da koristimo.



Slika 19. Izbor vrste ekrana

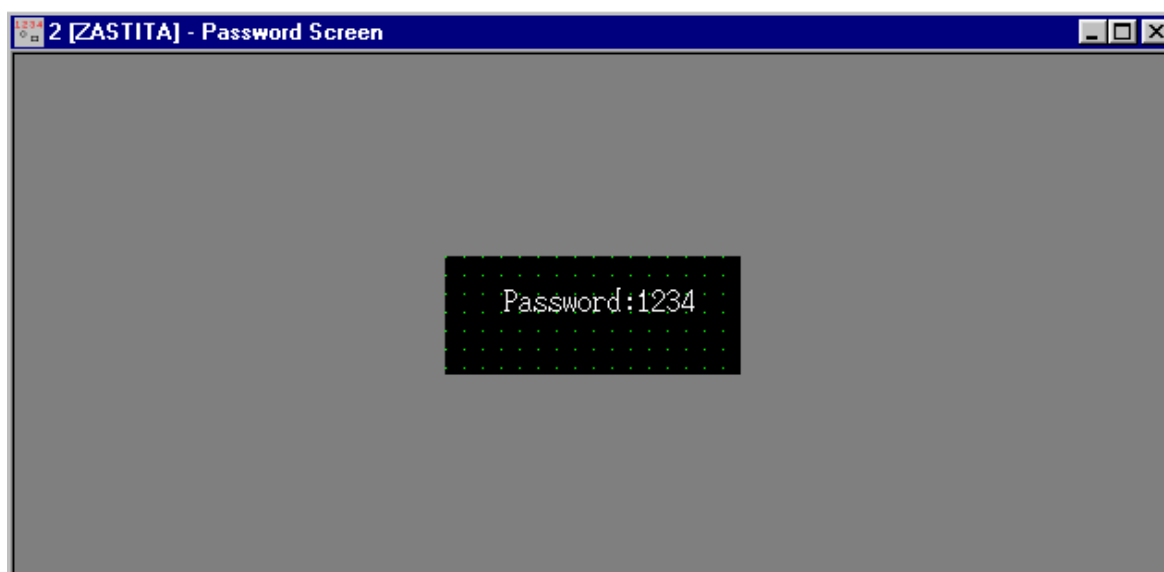
5.1.1. Ekran za lozinku (Password Screen)

Ako smo izabrali opciju Password Screen pojavljuje nam se novi prozor u kome treba zadati osnovne parametre ekrana. (redni broj ekrana i komentar).



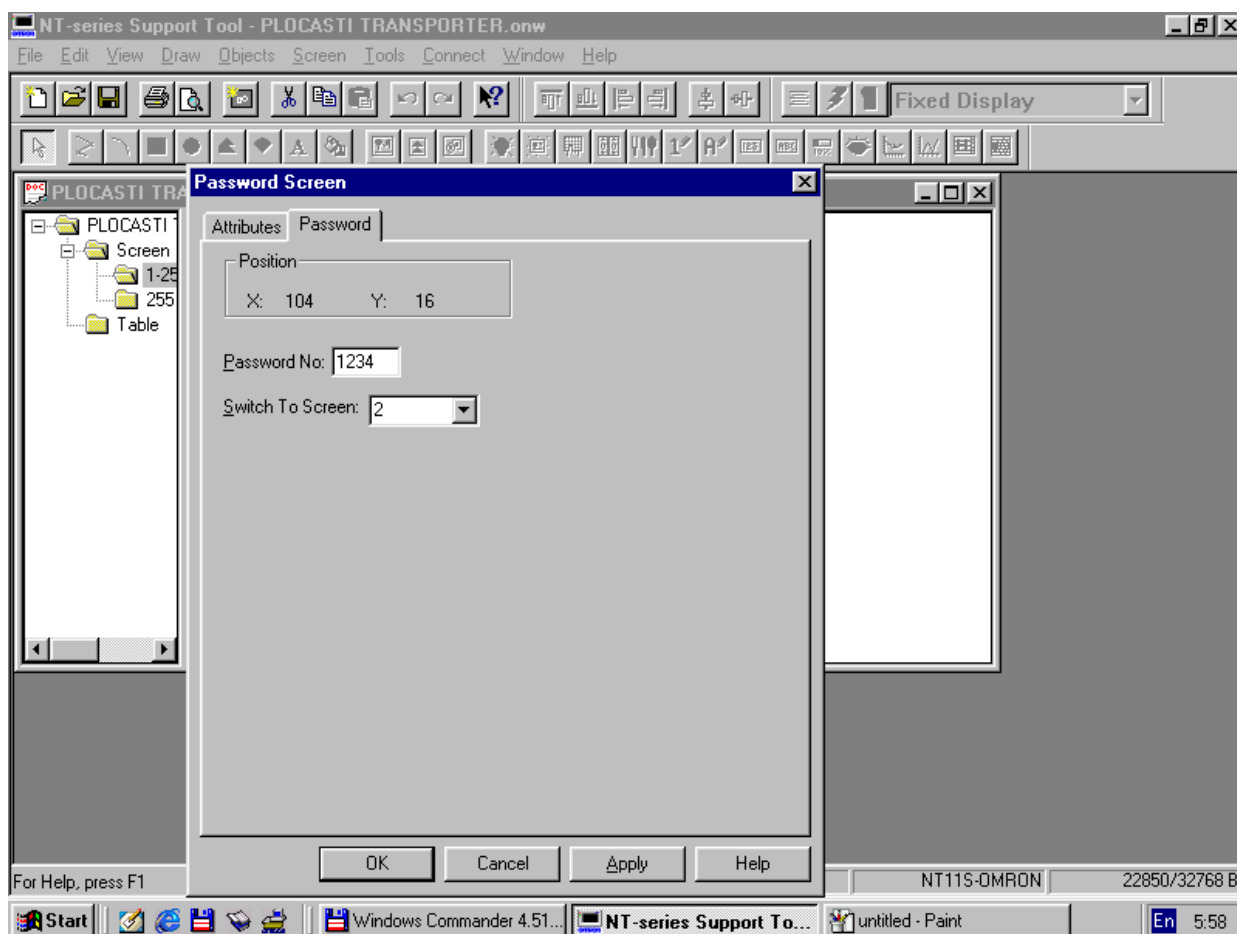
Slika 20. Ekran za lozinku

Tada se u folderu 1-250 dobija još jedna ikonica i otvara nam se sledeći prozor.



Slika 21. Ekran za unos lozinke

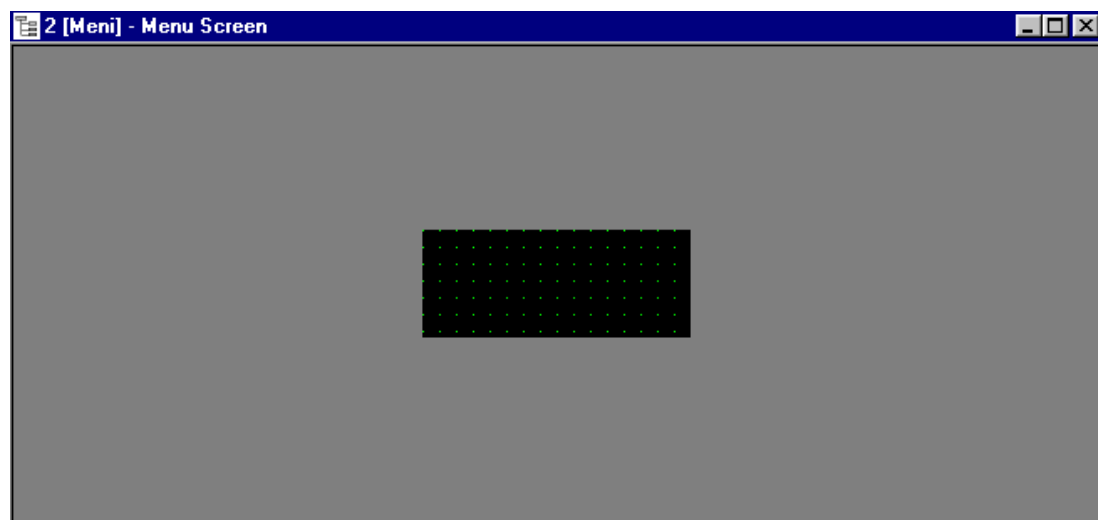
Lozinku možemo menjati u meniju za izmenu osobina (Properties – Password No:). U njemu možemo da podesimo koji će se ekran prikazati nakon ispravno upisane lozinke (Switch To Screen:).



Slika 22. Ekran za unos parametara lozinke

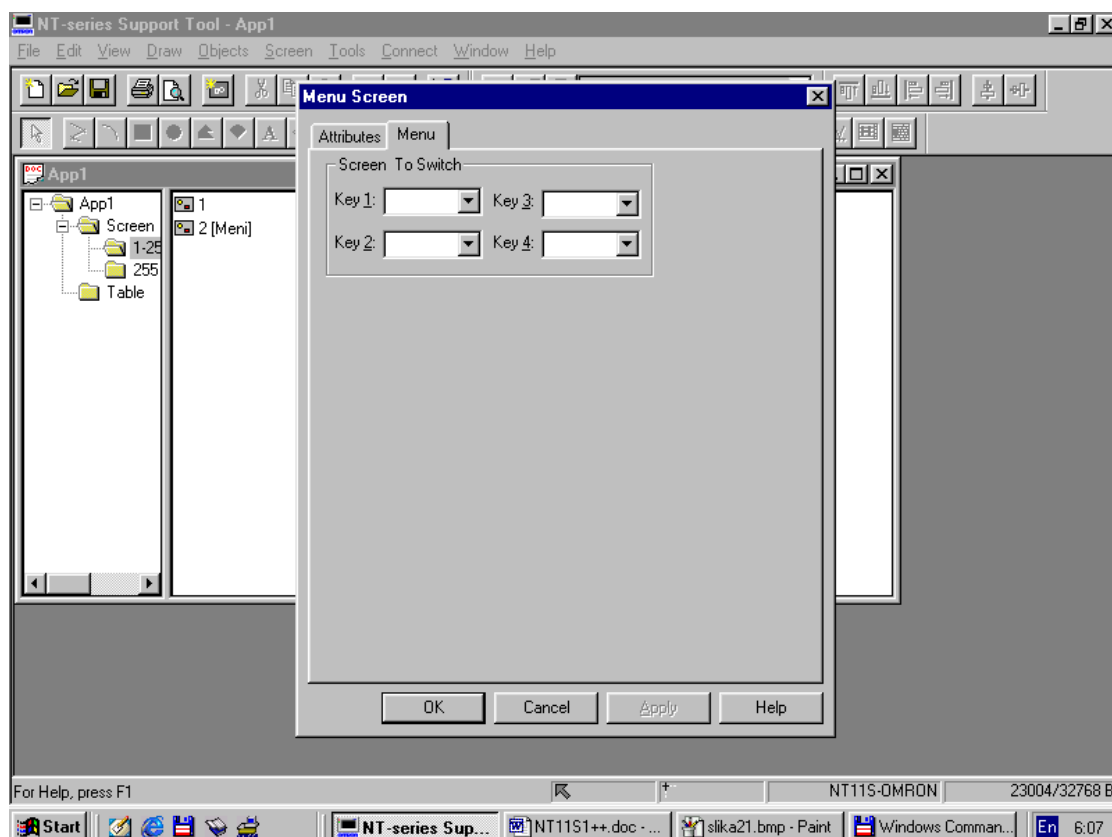
5.1.2. Meni ekran (Menu Screen)

I u ovom slučaju prvi korak je određivanje rednog broja ekrana i komentara. Nakon toga nam se pojavljuje potpuno prazan meni ekran.



Slika 22. Prazan meni ekran

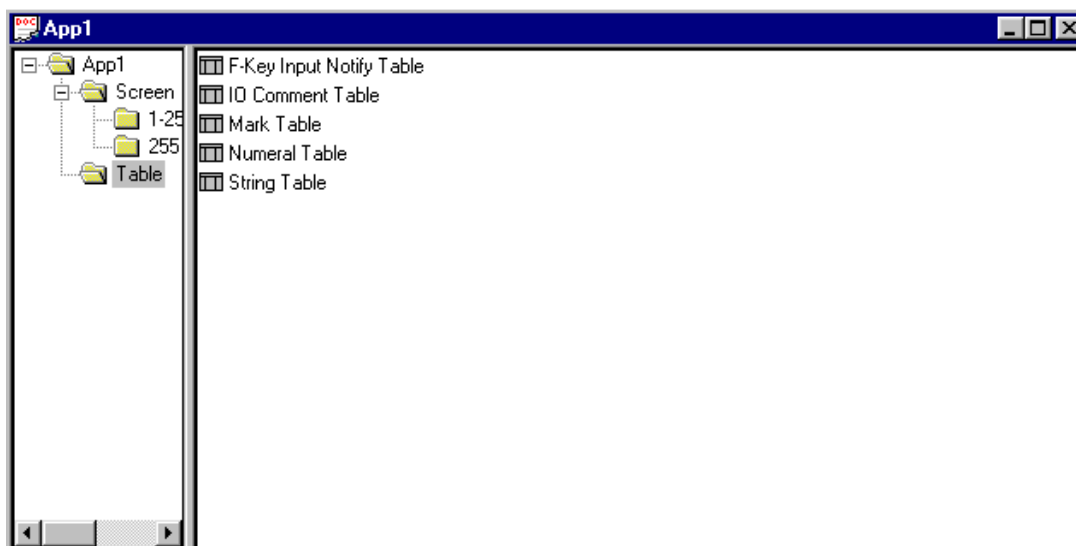
Sledeći korak je podešavanje osobina meni ekrana. U opciji meni možemo izabrati na koji ekran ćemo skočiti kada pritisnemo određeni taster na terminalu.



Slika 23. Izbor skoka na ekran

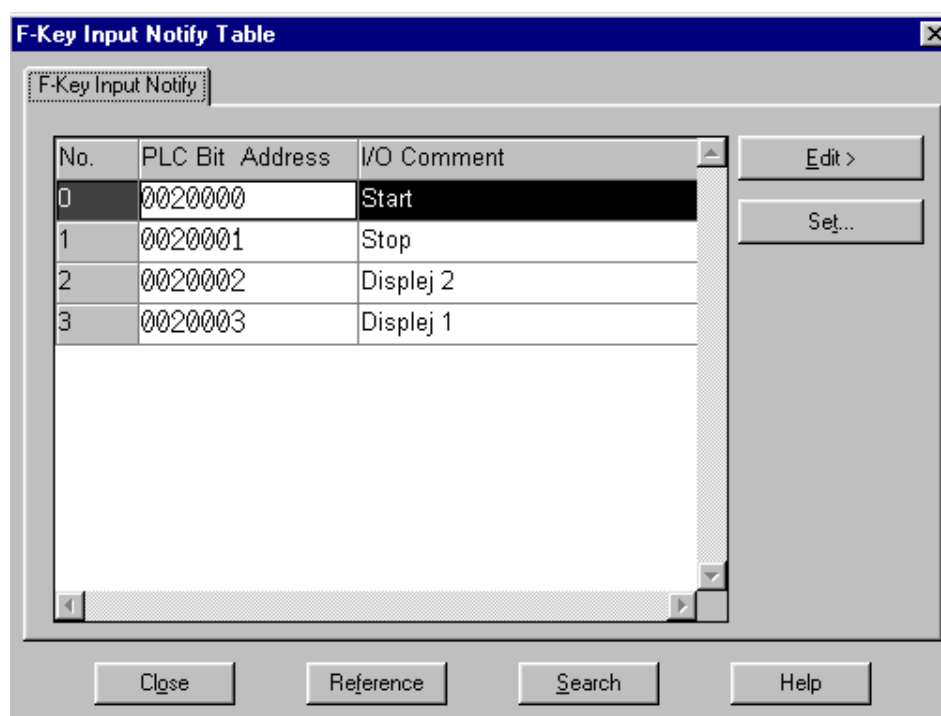
5.2. Memorijske tabele

Memorijske tabele služe za povezivanje promenljivih terminala NT11s sa promenljivima iz PLC-a, odnosno preko odgovarajućih tabela omogućavamo međusobno povezivanje adresa terminala i kontrolera.



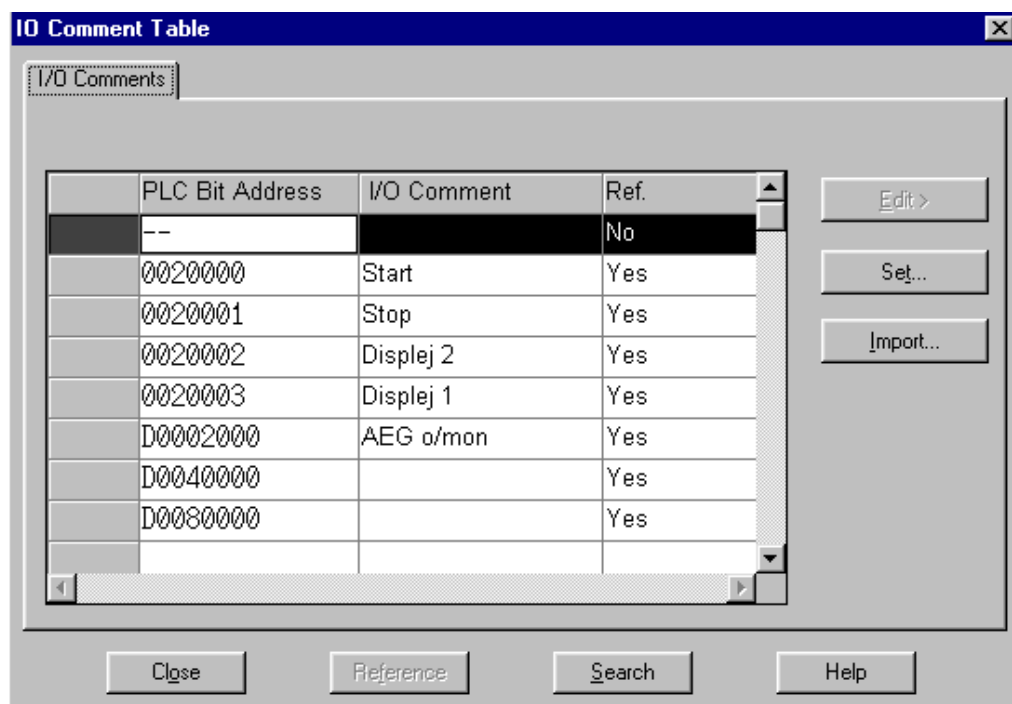
Slika 24. Vrste memorijskih tabela

Preko prve memorijske tabele "F-Key Input Notify Table", mogu povezuju se funkcijski tastera terminala, što obezbeđuje njihovo korišćenje, odnosno može se podesiti da svaki taster ima svoju adresu u PLC-u.



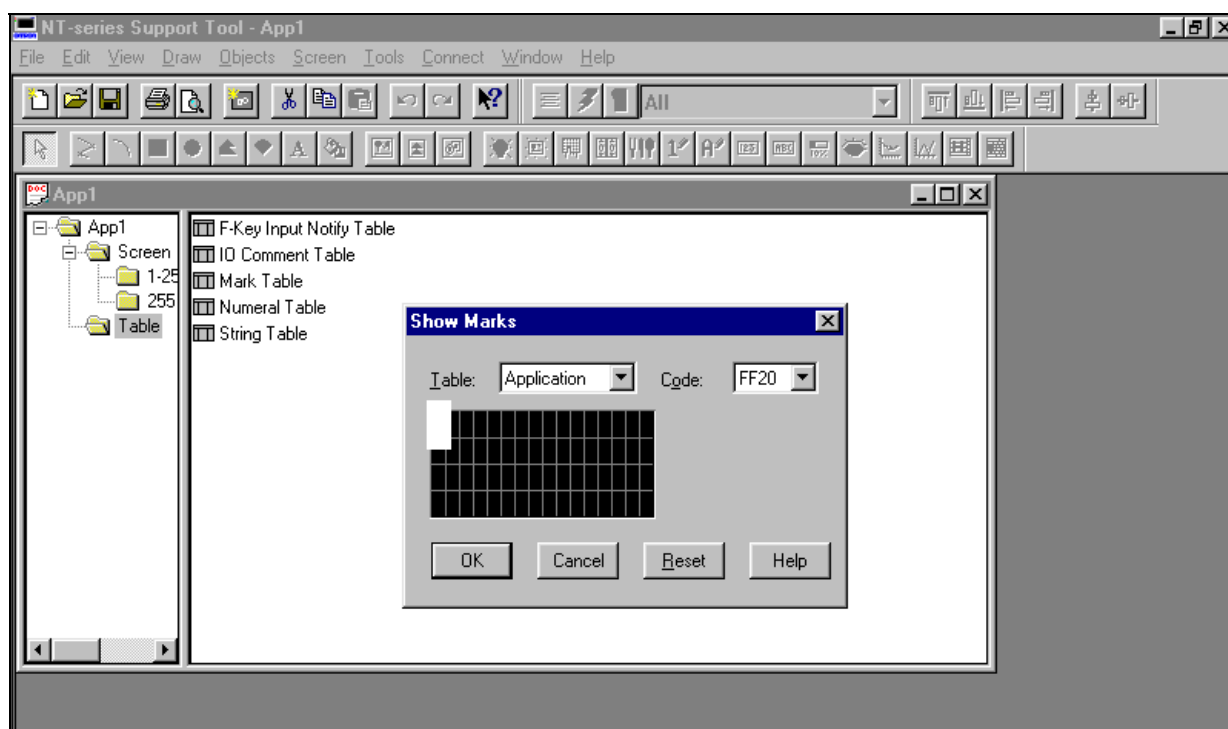
Slika 25. Tabela funkcionalnih tastera

Sledeća tabela "IO comment Table" je za ulazno/izlazne "komentare". Uglavnom u ovoj tabeli se vide i definišu svi bitovi koje se koriste i međusobno razmenjuju između terminala i kontrolera. Naravno u njoj se mogu upisati ili obrisati željene adrese.



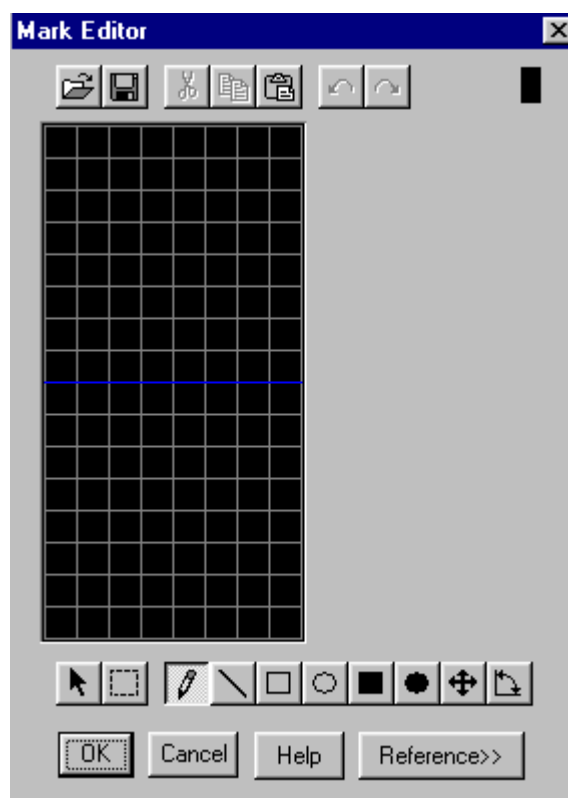
Slika 26. Tabela U/I podataka

Sledeća tabela je "Mark Table" koji sadrži znakove koji se koriste kod crtanja ekrana.



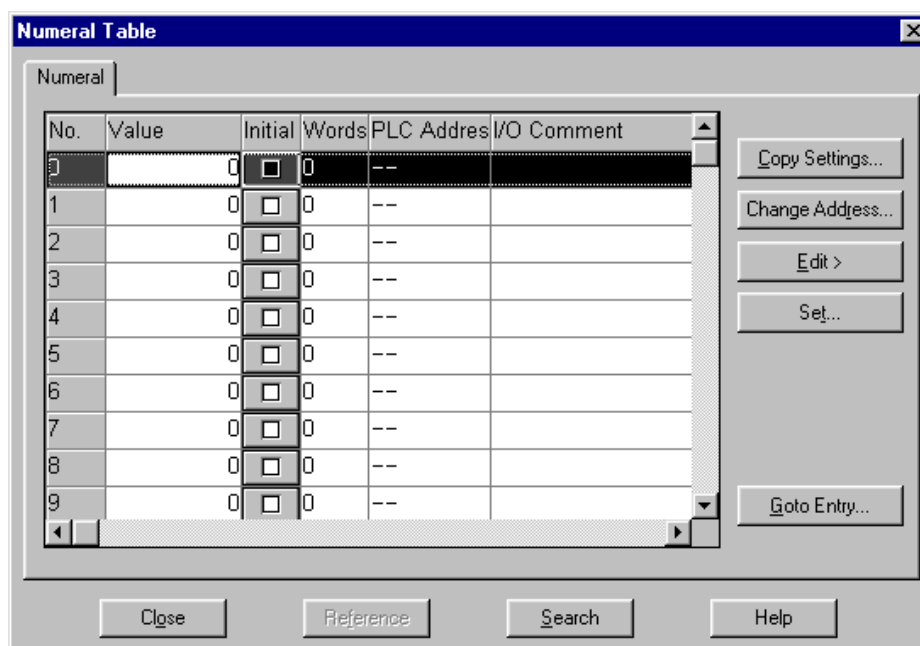
Slika 27. Tabela znakova

Crtanje znakova se vrši pomoću posebnog editora "Tools/Mark Editor".

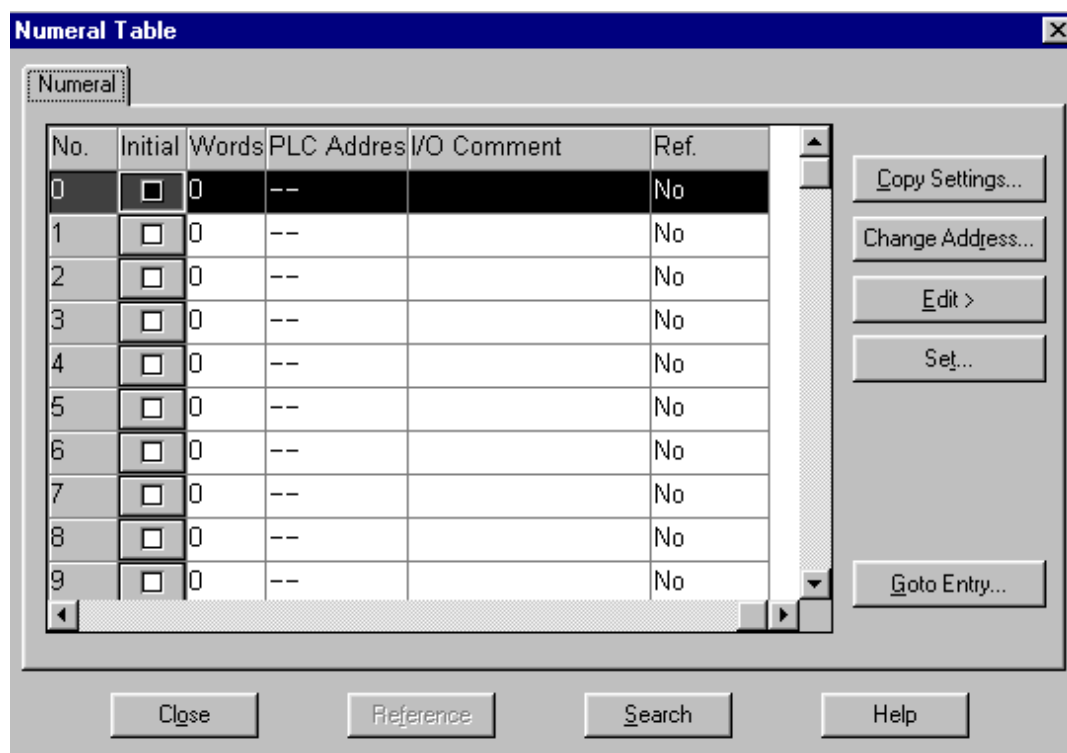


Slika 28. Tabela editora znakova

Numerička tabela (Numeral Table) nam daje mogućnost za unošenje vrednosti na ekran terminala (a sa time i u neku memorijsku lokaciju u kontroleru ako se ta lokacija koristi u programu). Mogu se koristiti 128 polja tabela (0 – 127). Od toga 8 polja može biti registrovano istovremeno na displeju terminala.

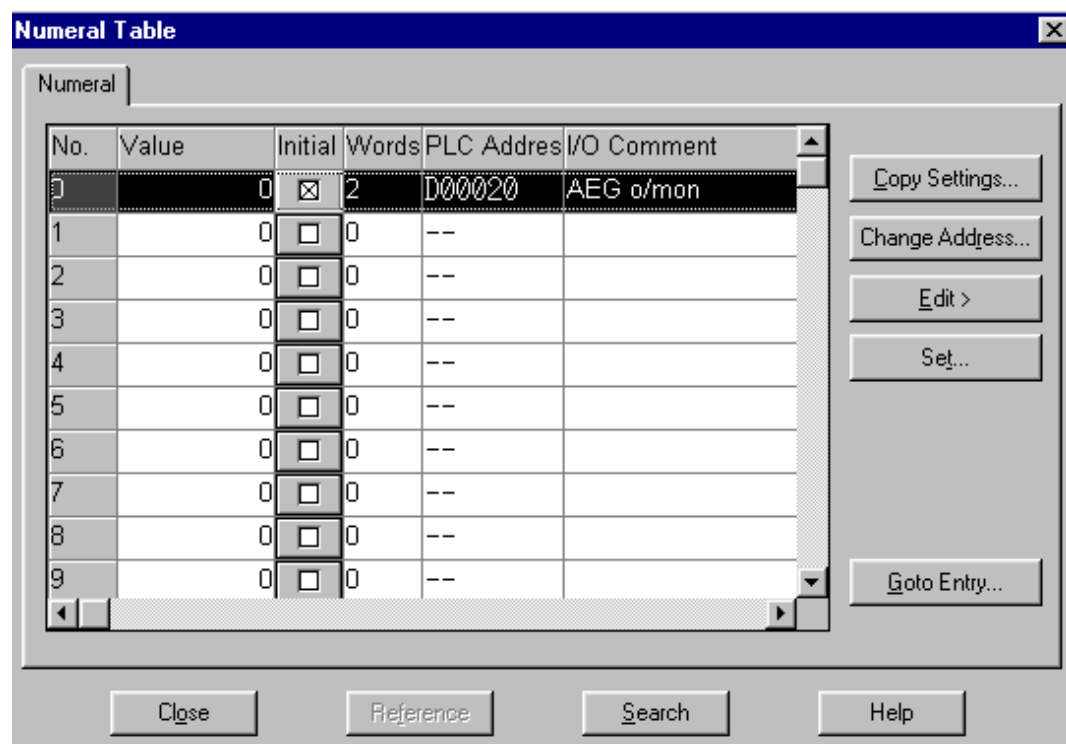


Slika 29. Numeral Table (donji klizač skroz levo)



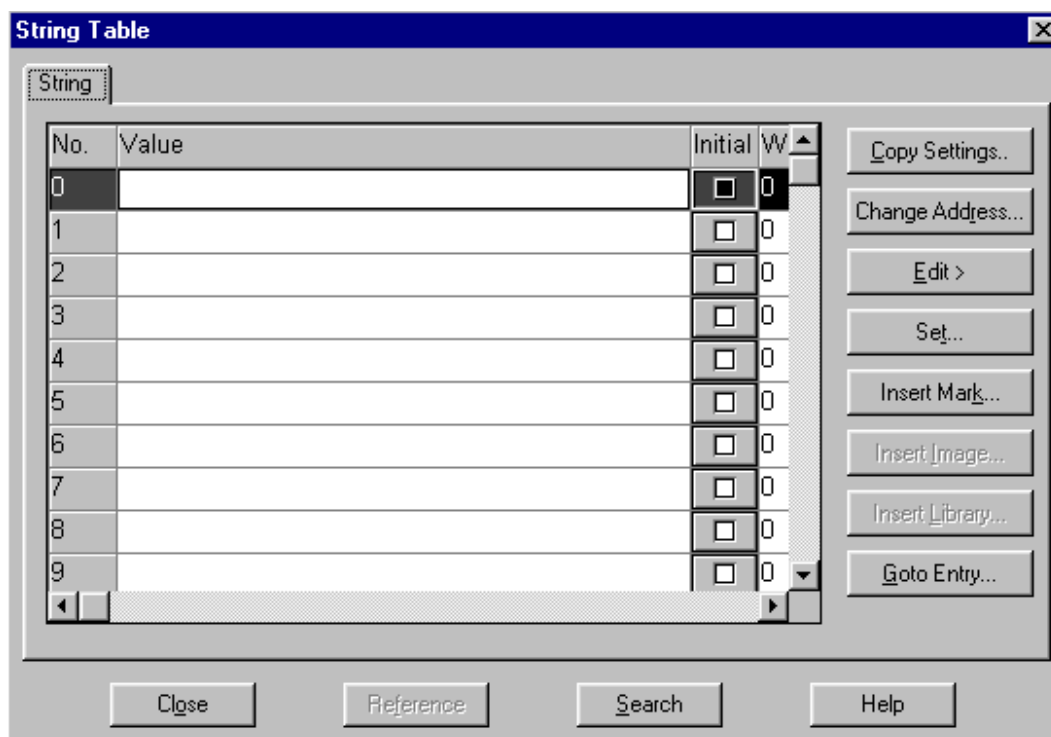
Slika 30. Numeral Table (donji klizač skroz desno)

Primer: Iz memorijske lokacije iščitava vrednost broja obrtaja veličine 2 reči sa adrese DM0020 na PLC-u.

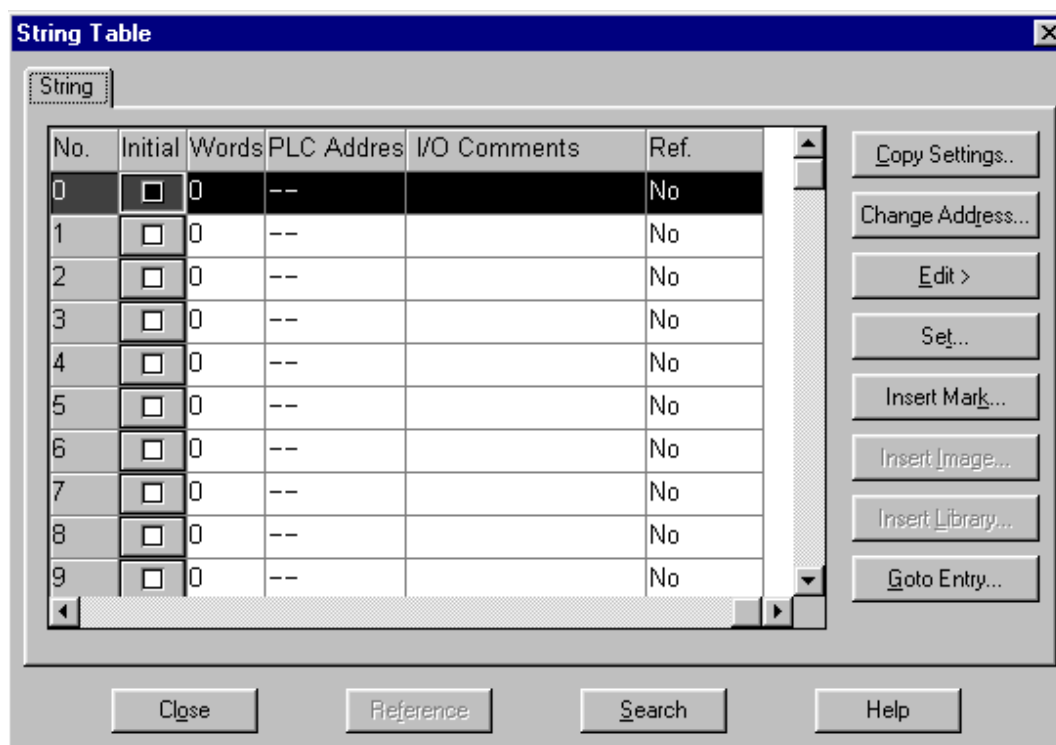


Slika 31. Primer korišćenja tabele Numeral Table za izčitavanje broja obrtaja

Tabela "String Table" namenjena je za skladištenje karakternih, simbolnih, alfa numeričkih (*charracter*) podataka. Moguće je koristiti od 0-127 polja (odnosno 128) "string tabli" u 8 pozicija na ekranu. Kad se kreira tabela i stavi neko karakterni polje na nekoj poziciji na displeju u vezu sa nekim uslovom iz kontrolera, ono će se pojaviti na displeju ako se taj uslov ostvari.



Slika 32. String Table(donji klizač skroz levo)



Slika 33. String Table (donji klizač skroz desno)

5.3. Korišćenje objekata za crtanje kod NT11s

Imamo više mogućnosti za crtanje kod NT11s-a :

- Text-a: (Fixed Display : Text),
- Ulaznih brojčanih podataka: (Data Input: Numeral),
- Numeričkih ekrana: (Numeral Display),
- Simbolnih ekrana: (String Display),
- Grafičkih ekrana: (Graph: Bar Graph),

prečica :



prečica :



prečica:



prečica:



prečica :



Način korišćenja najlakše se objašnjava na primerima. .

6. Primeri programiranja programabilnog terminala

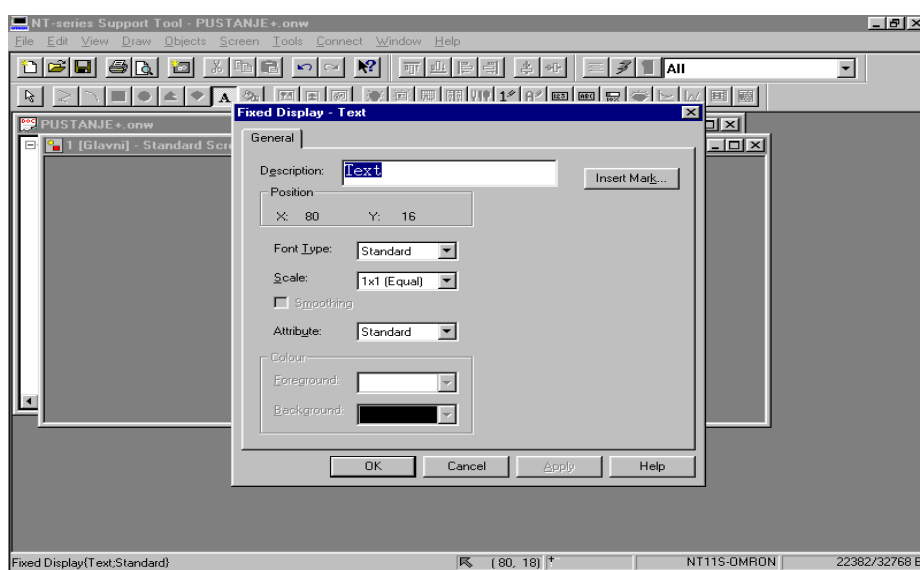
6. 1. Zadavanje vrednosti na analognom izlazu - zadatak 1:

Realizovati lestvičast dijagram koji će vršiti upravljanje vrednošću analognog izlaza u skladu sa sledećim pravilima:

- Pomoću dva tastera (plus F1,minus F2) na programabilnom terminalu NT11S povećava se ili smanjuje analogni izlaz za određenu vrednost.
- Preko tastera F3 pokazuje se analogna izlazna vrednost na terminalu.
- Opseg izlaznog signala je 0 do 20 [mA].
- Vrednost izlaznog signala na početku rada programa treba da je 10[mA].
- Program napraviti tako da signal nikada ne prekorači granice opsega.

6.1.1. MINUS PLUS ekran

Kao što smo objasnili u prethodnom poglavlju, za početak treba pokrenuti softverski paket NT-Series Support Tool i kliknuti na ikonicu za otvaranje novog dokumenta. Kada smo izvršili sva podešavanja dobijamo novi ekran na kojem počinjemo izradu zadatka.

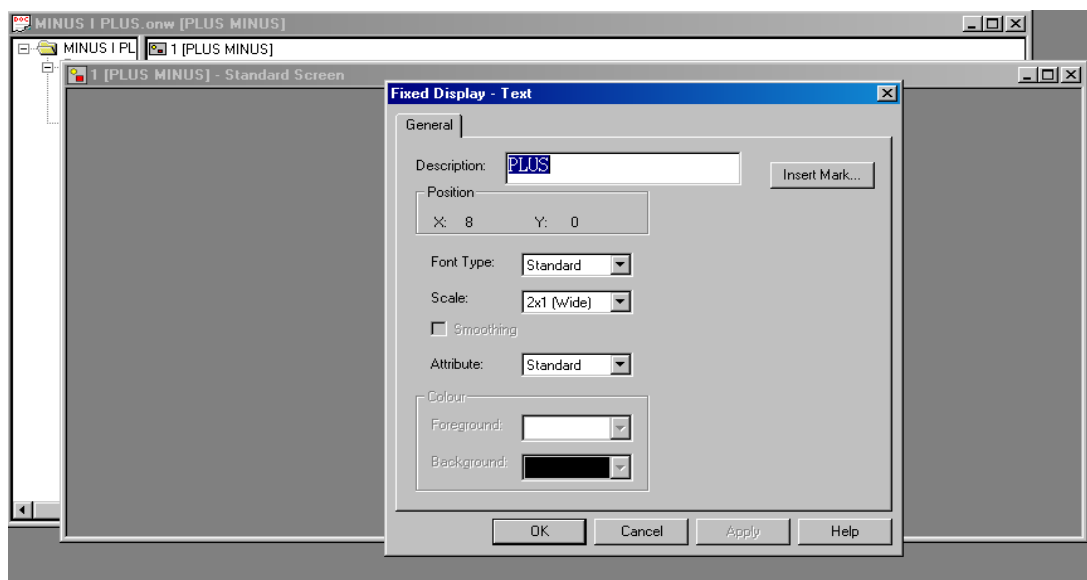


Slika 34. Dijalog box za pisanje teksta na ekran

U paleti sa alatima aktiviramo prečicu za TEXT .

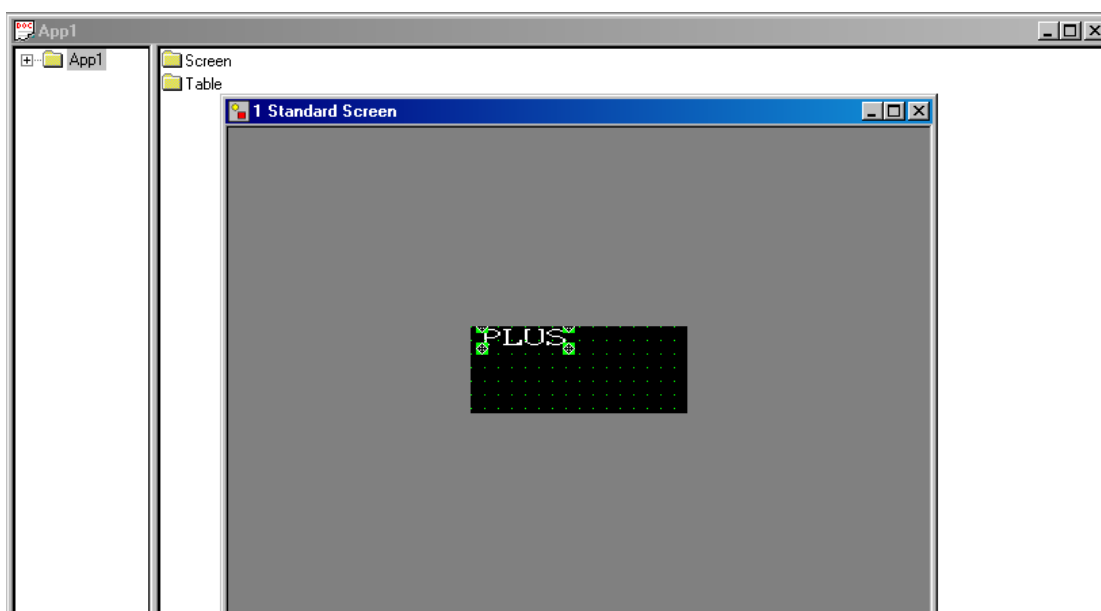
Duplim klikom na ekran pojavljuje se dijalog box za pisanje teksta na ekran, kao na prethodnoj slici.

Ovde možemo upisati željeni tekst (Description), odrediti poziciju teksta na ekranu (Position), podešavati vrstu slova (Font Type), veličinu slova (Scale) i osobenost teksta (Attribute). U našem slučaju podešavanja izgledaju na sledeći način.



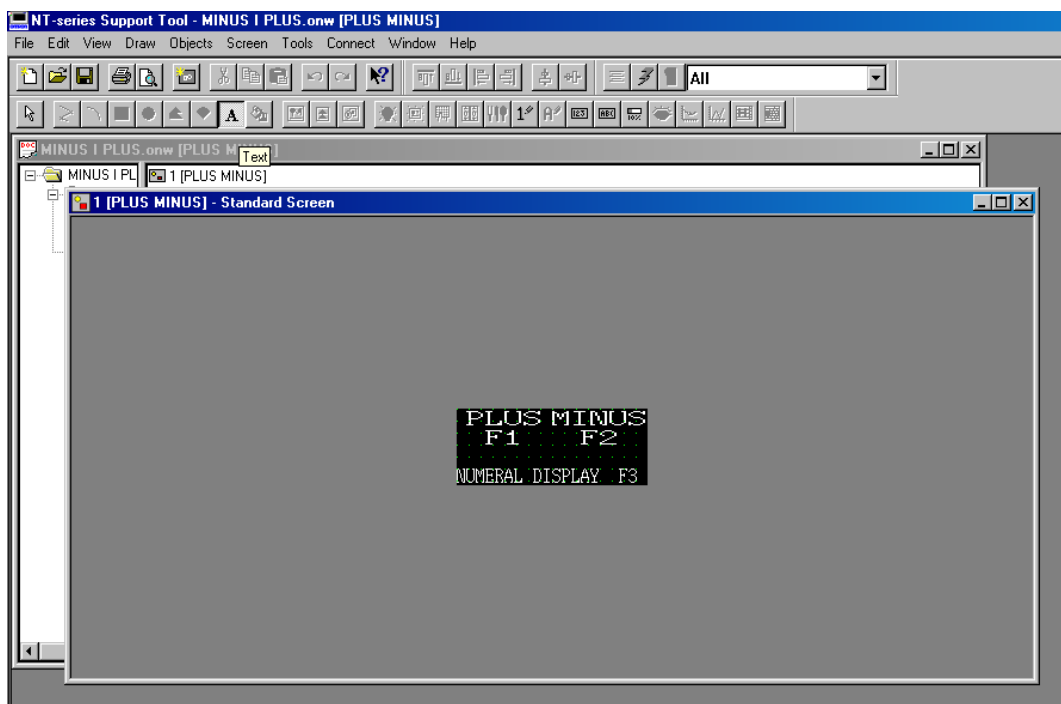
Slika 35. Podešavanja osobina teksta u ekrana

Nakon ovih podešavanja kao rezultat dobijamo sledeći ekran.



Slika 36. Izgled ekrana nakon podešavanja osobina teksta

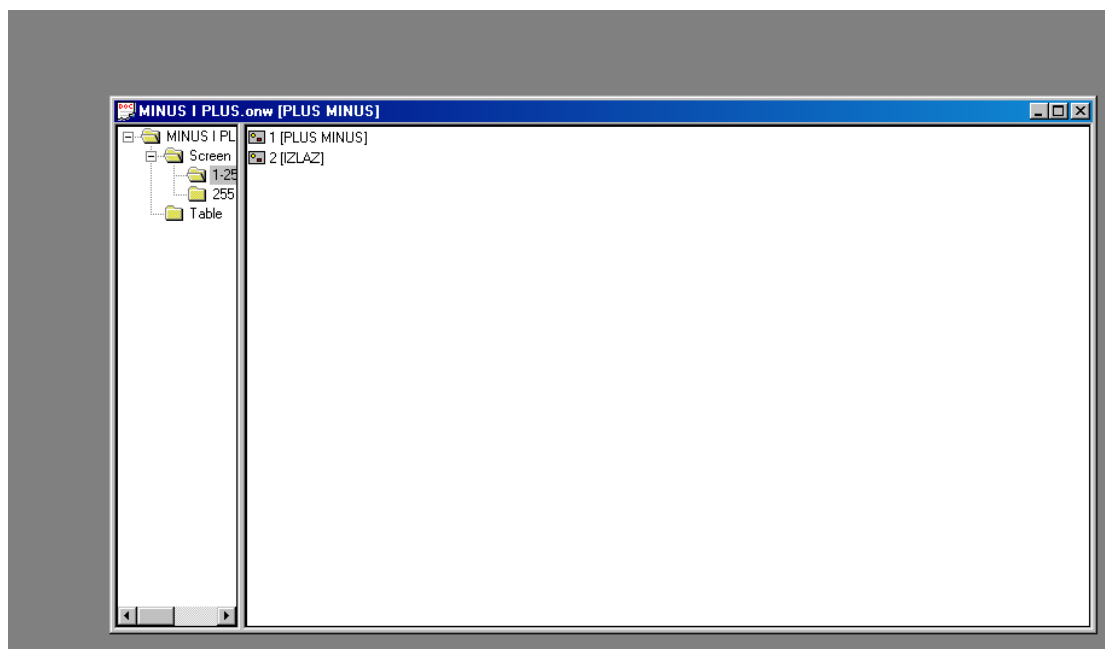
Na isti način upisujemo i preostali deo teksta na ekran. Kada to uradimo prvi ekran bi trebao od prilike da izgleda ovako.



Slika 37. Izgled gotovog ekrana PLUS MINUS

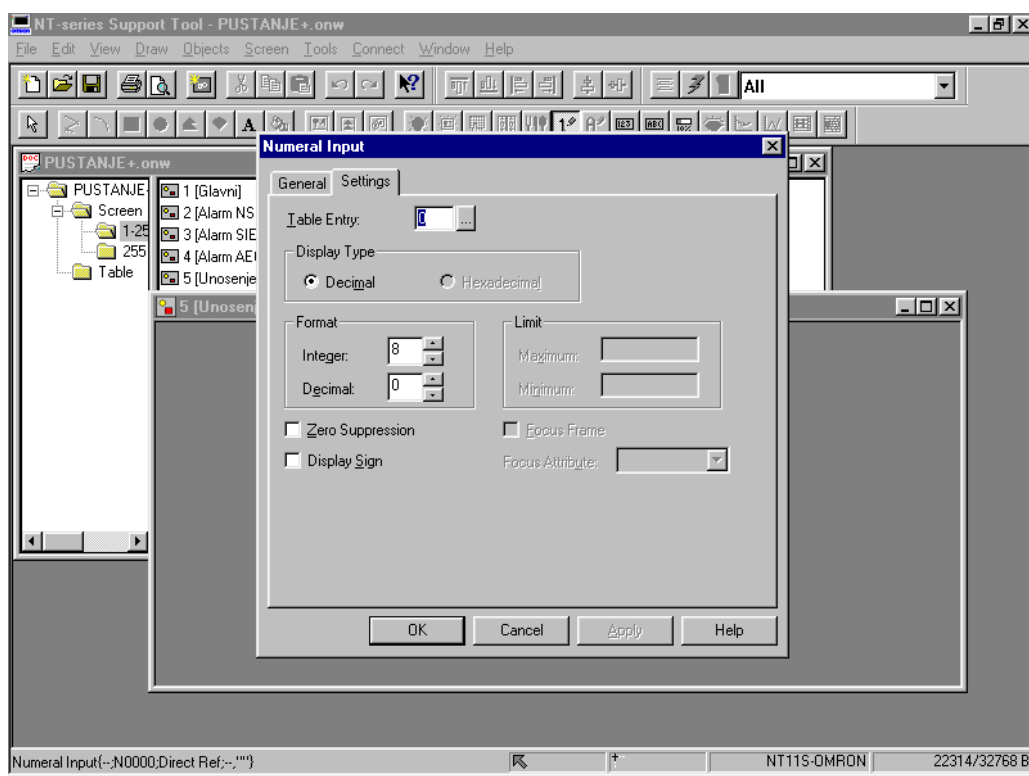
6.1.2. Ekran IZLAZ

Na sličan način pravimo i ostale ekrane potrebne za izradu ovog primera.



Slika 38. Izbor novog ekrana IZLAZ

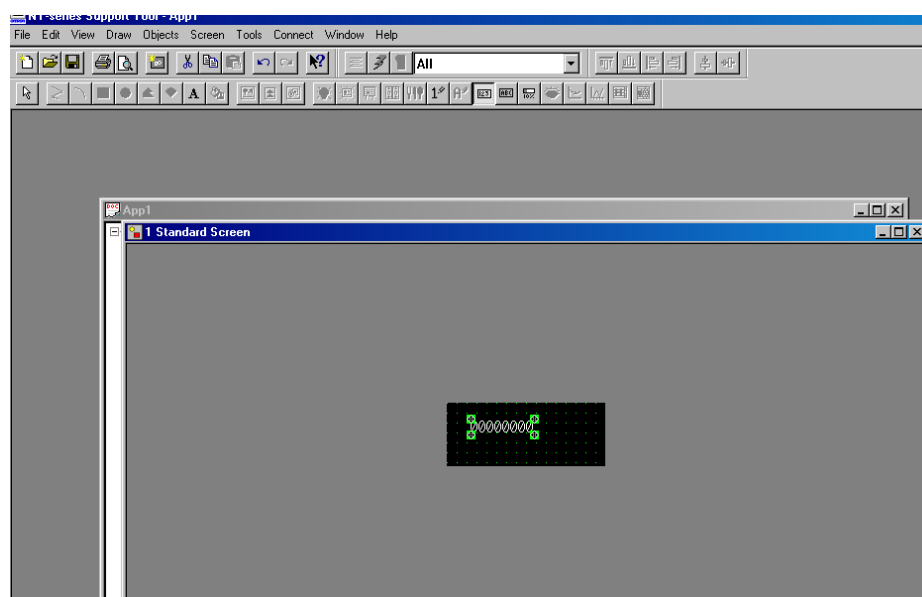
Ako nam treba  ulaz brojčanih podataka, prevučemo ga na ekran i kliknem dva puta levim tasterom. Posle običnog podešavanja u osnovnom, glavno (General) prelazim na podešavanje (Setings):



Slika 38. Prevlačenje analogne vrednosti u ekran IZLAZ

Sada ovde postoje više opcija i to u vezi:

- Biranje memorijske lokacije (Table entry),
- Tip ekrana (Display Type),
- Oblik , raspored (Format),
- Prigušiti, ukinuti nulu (Zero Suppression),
- Znak ekrana (Display Sign),

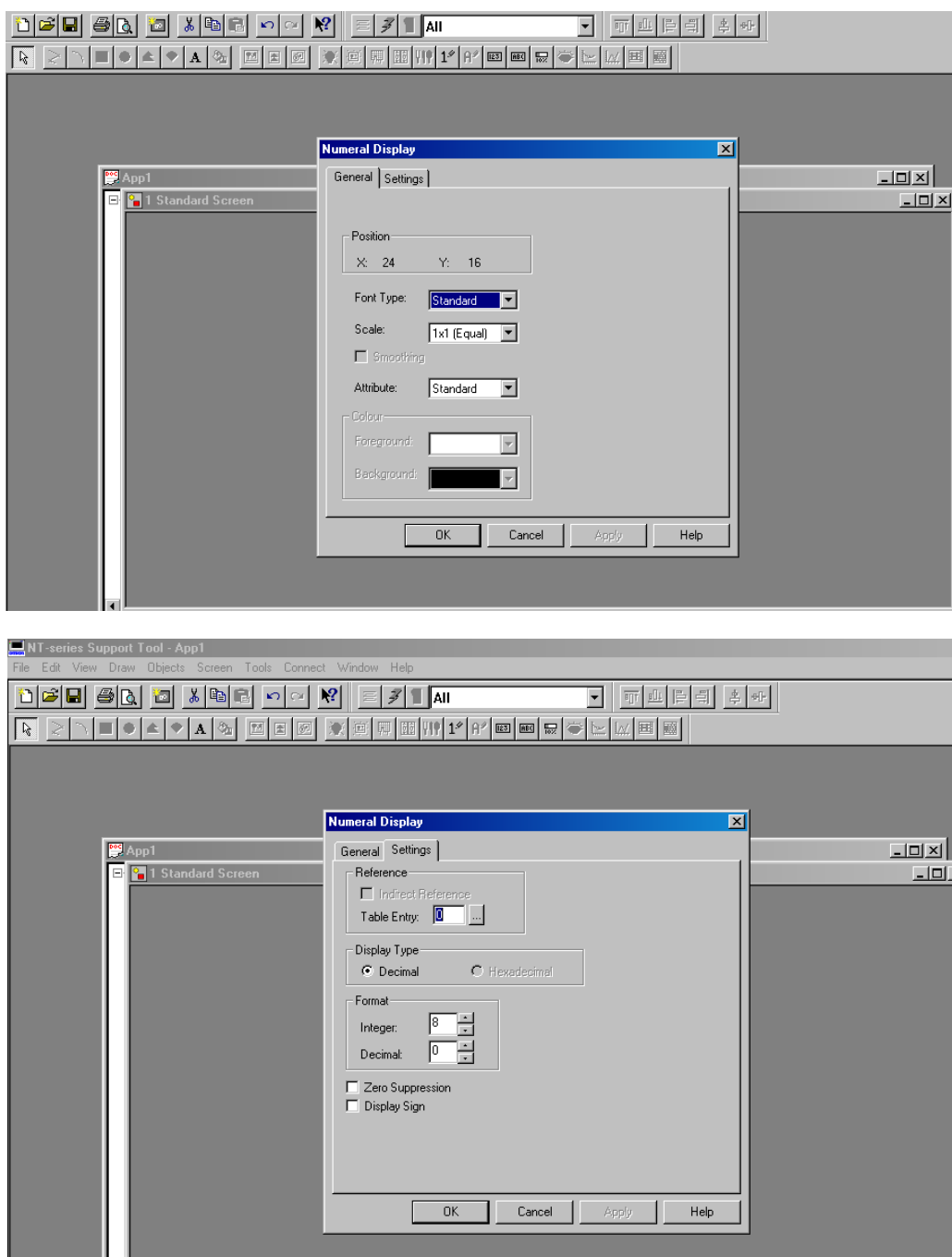


Slika 39. Ekran posle uključenja numeričkog displeja

Treba napomenuti da ove mogućnosti upisivanja, čitanja numeričkih i simbolnih karaktera možemo uraditi u standardnom i ekranu za štampanje. U meniju i lozinki možemo koristiti samo TEXT opciju, a opcija za ispisivanje podataka grafičkim putem omogućena je samo u standardnom ekranu.

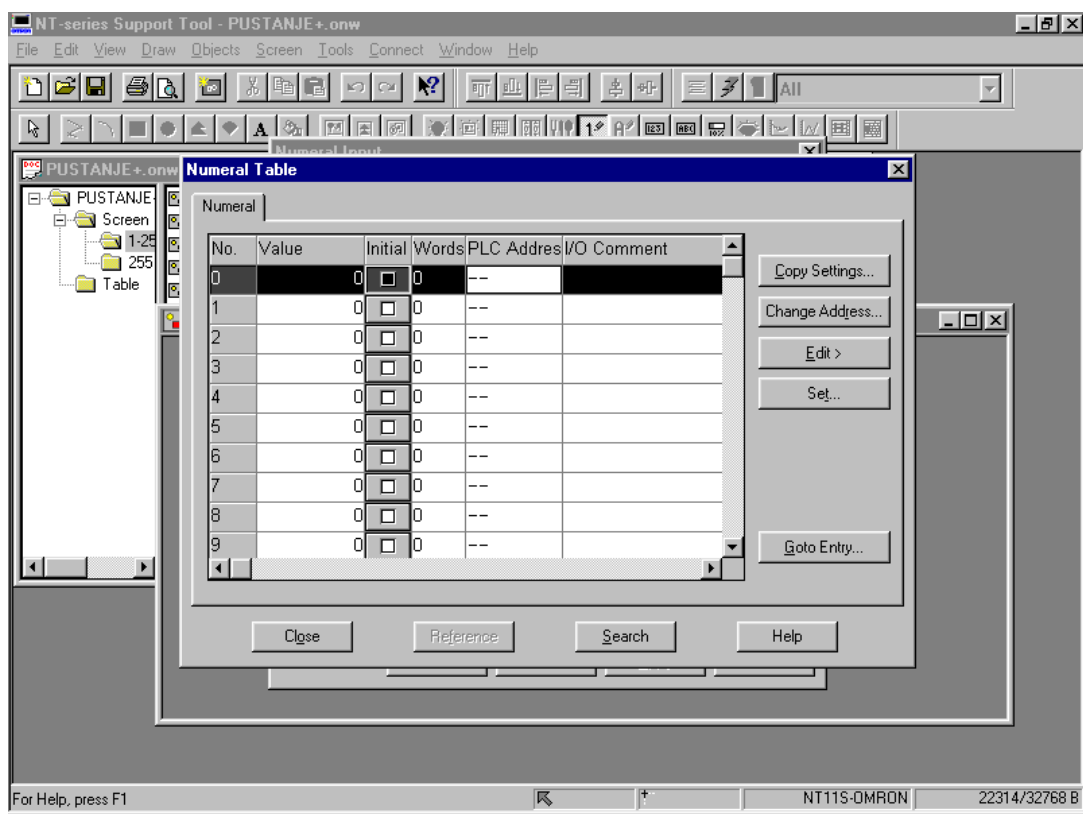
Ovaj postupak je uglavnom isti i za čitanje iz memorijskih lokacija. Što se i koristi u ovom primeru. Kada napravimo novi standardni ekran pomoću ikone  uključimo numerički display. Kojeg kada prevučemo na ekran dobijamo rezultat kao na prethodnoj slici.

I kliknemo dva puta levim klikom dobijamo sledeći ekran. Gde imamo ista podešavanja kao i kod numeričkog ulaza.



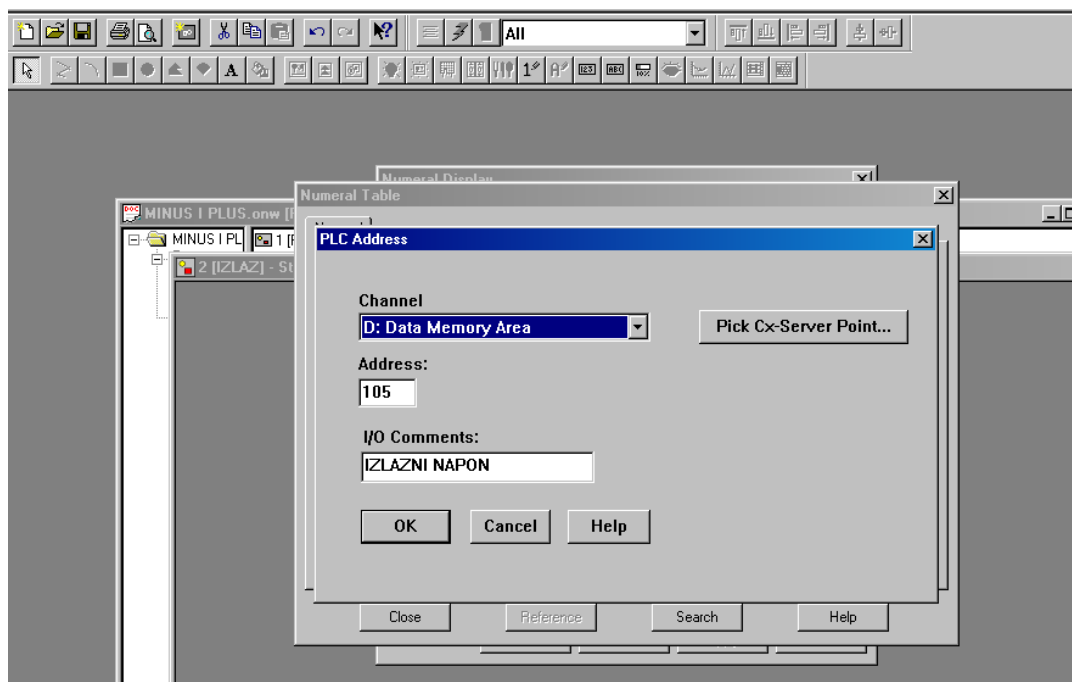
Slika 40. Podešavanja u numeričkom displeju

Ulazimo u tabelu:



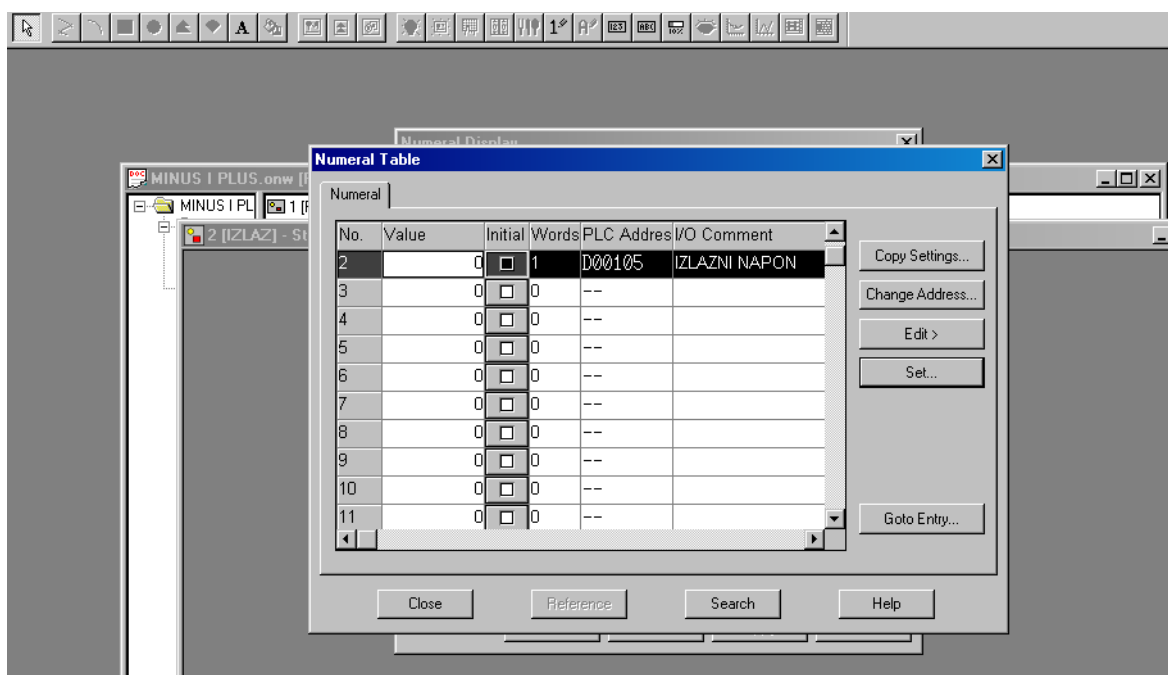
Slika 41. Ulaz u numeričku tabelu

I preko Set podešavamo našu vrednost:



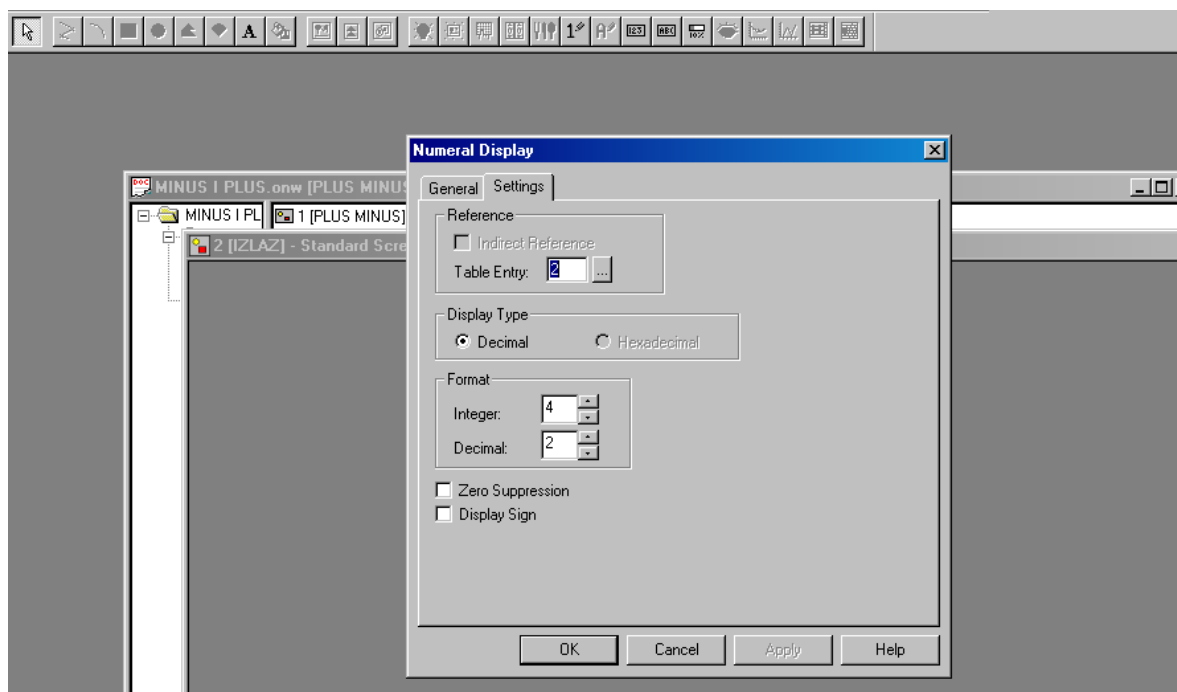
Slika 42. Povezivanje podatka numeričke tabele

Na kraju ispunjena tabela izgleda ovako:



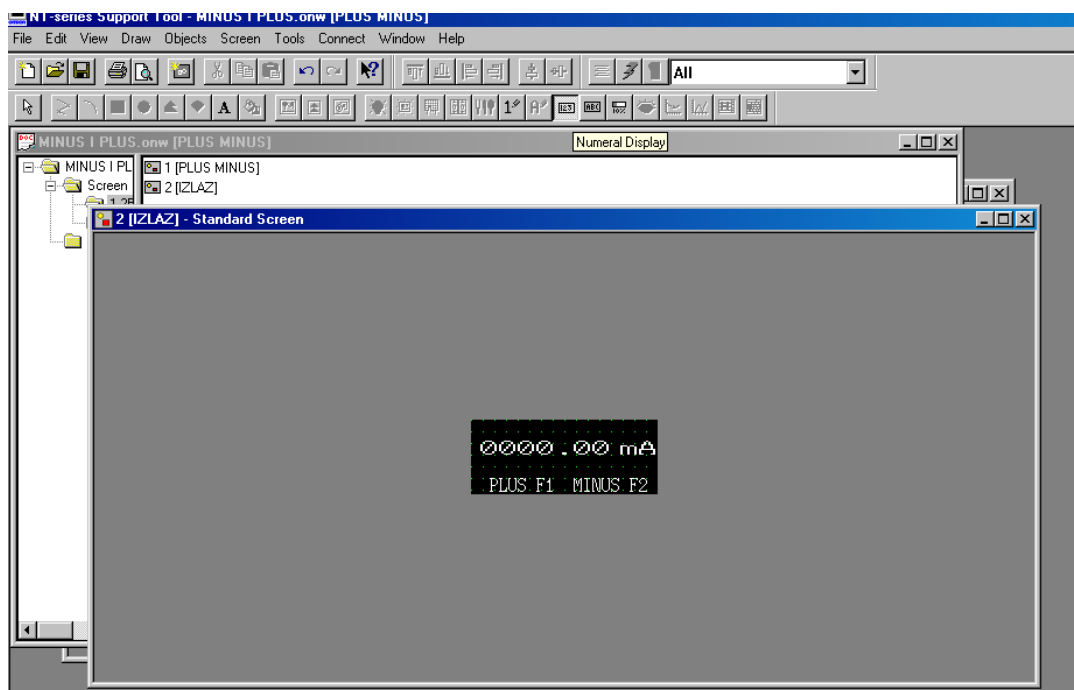
Slika 43. Izgled popunjene numeričke tabele

Izlaskom iz nje (Close) vraća se u meni za podešavanje (u ovom primeru podešen je prikaz ekranu od 4 numerička karaktera sa 2 decimalna mesta):



Slika 44. Podešavanje prikaza numeričkog podatka

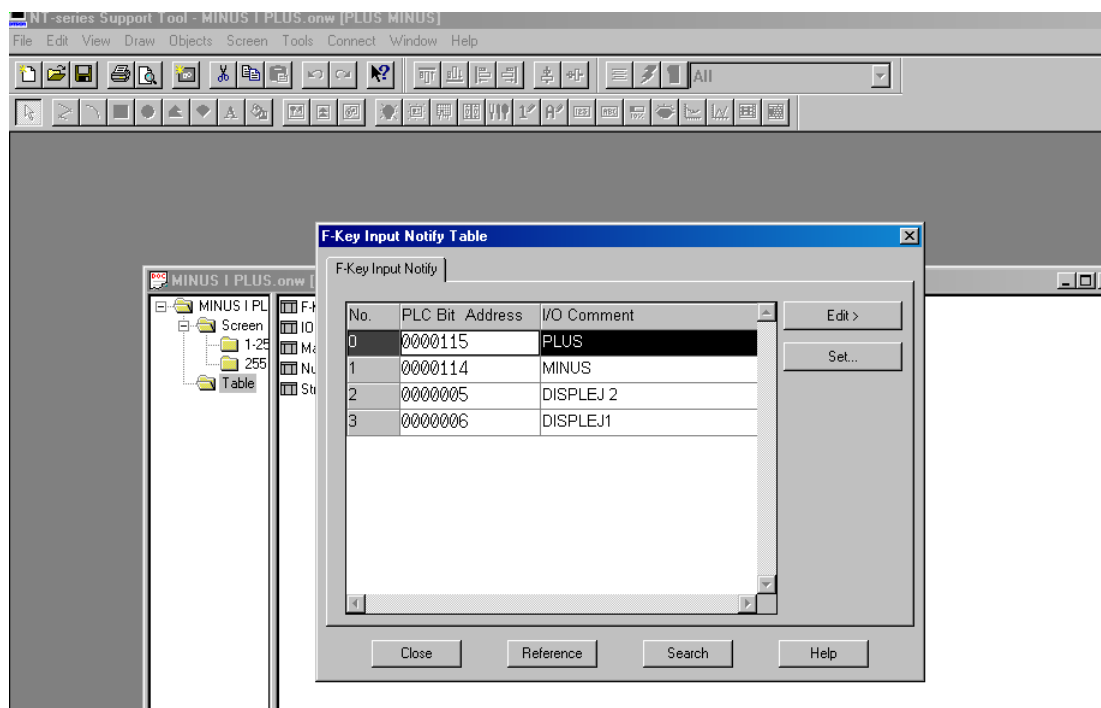
Krajnji izgled ekrana prikazan je na sledećoj slici.



Slika 45. Krajnji izgled ekrana za prikaz numeričkog podatka

Rešenje postavljenog zadatka može biti različito, kako za dva opisana primera kao i za sve ostalo (uvek ima nekoliko različitih pravaca = putanja), tako je u ovom primeru više korišćeno PLC programiranje da bi se dovršio zadatak. Neko drugo rešenje bi bilo: korišćenje meni ekrana, ili korišćenje simbolnih ekrana, i slično...

6.1.3. Podešavanje funkcijskih tastera



Slika 46. Podešavanje funkcijskih tastera

Kada smo završili izgled ekrana treba omogućiti korišćenje funkcijskih tastera. Ulazimo u tabelu i podešavamo vrednosti za funkcijske tastere prema tabeli i prethodnoj slici:

No. u F-Key	Funkcijski tasteri na PT-u
0	F1
1	F2
2	F3
3	F4

Pri tome koristimo PLC adrese za pozivanje sa programibilnog terminala iz ulazno/izlaznih (I/U) memorijskih lokacija.

Na kraju treba prebaciti ekrane i podešavanje u PT. Opis prebacivanja dat je posebno u okviru poglavlja 7.

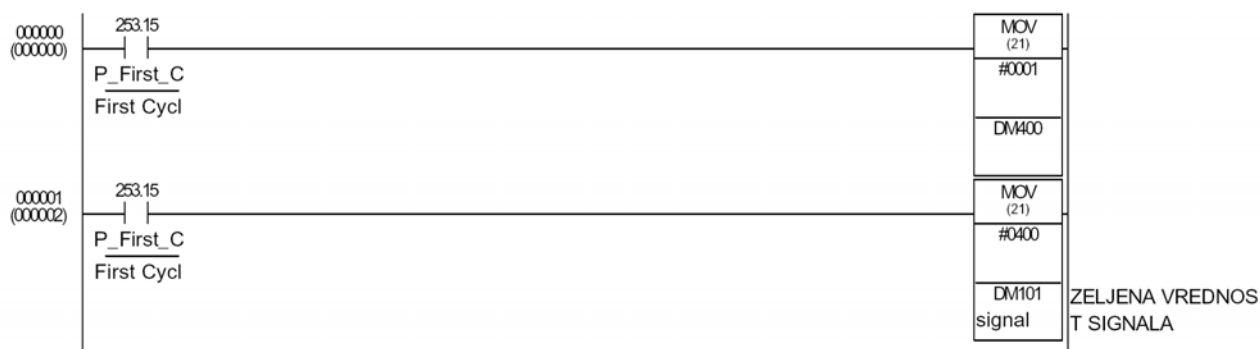
6.1.4. Lestvičasti dijagram

Način pisanja lestvičastog dijagrama, ovde nije opisan pošto je on obrađen u ranijim vežbanjima. Navedimo samo specifičnosti koje se odnose na poziv pojedinih ekrana iz PLC programa. Ekran se upisuje upisom njihovog rednog broja 1 i 2 u memorijsko područje Control Area DM400. Tako da u prvom skenu treba pozvati početni ekran, u našem primeru 1 upisivanjem jedinice u DM400 (vidi 0 red na slici 47).

Tabela simbola je:

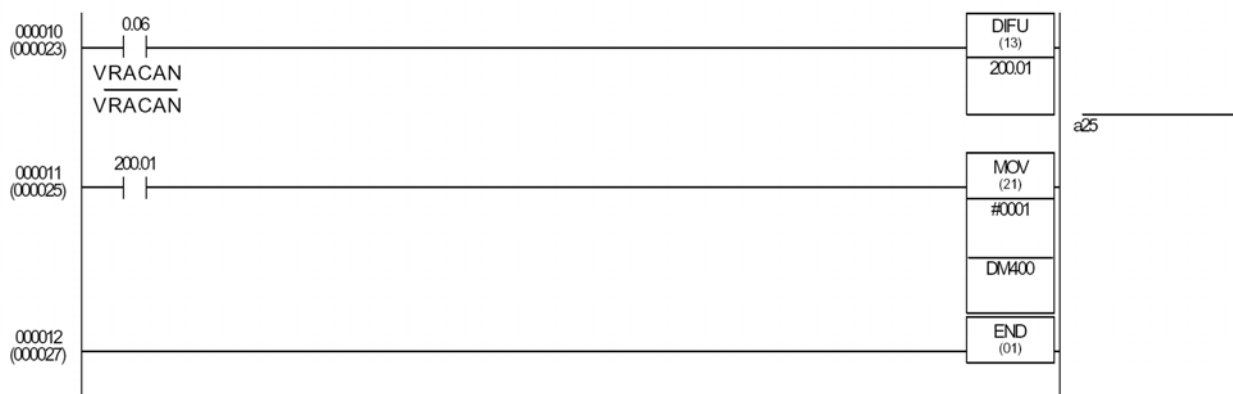
Name	Type	Address / ... Rack Lo... Usa... Comment
DOLE	BOOL	150.01 DETEKCIJA OPADAJU...
GORE	BOOL	150.00 DETEKCIJA OPADAJU...
IZLAZ	CHANN...	236 ADRESE POMOCNOG ...
minus	BOOL	1.14 taster za smanjenje izlaz...
NAPON	BOOL	0.05 ZA PREBACIVANJE N...
plus	BOOL	1.15 taster za povecavanje vr...
realna	REAL	DM105 REALNA VREDNOST ...
signal	UINT	DM101 ZELJENA VREDNOST ...
VRACANJE	BOOL	0.06 VRACANJE NA EKRAN...

Lestvičasti dijagram dat je na sledećim slikama:



Slika 47. Lestvičasti dijagram zadatka iz primera 1

Slika 48. Nastavak lestvičastog dijagrama zadatka iz primera 1



Slika 49. Nastavak lestvičastog dijagrama zadatka iz primera 1

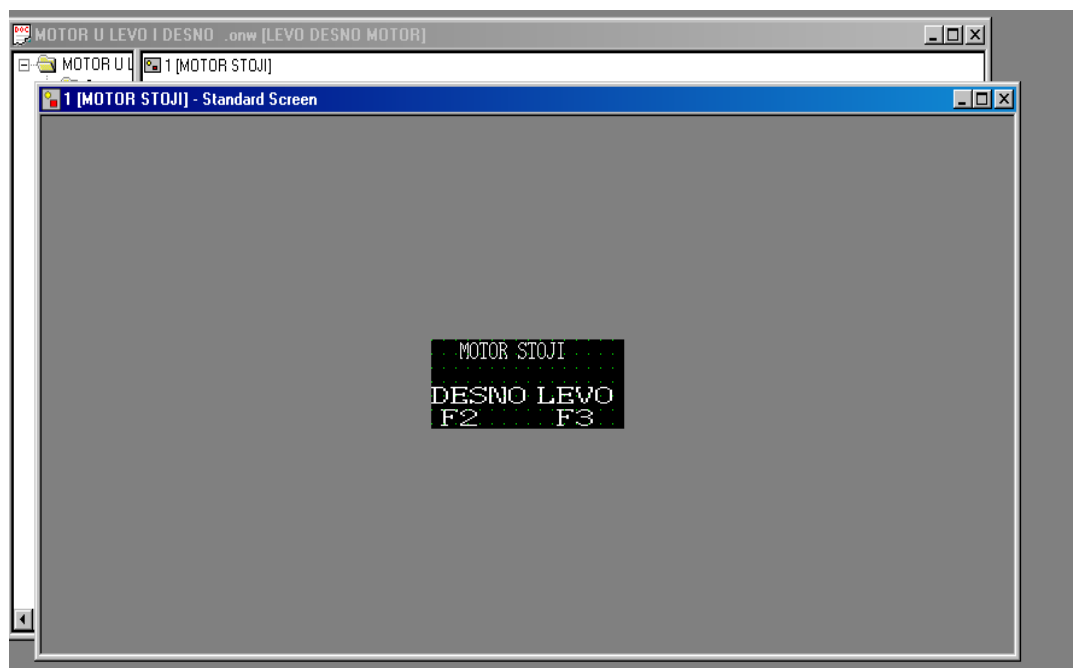
6.2. Upravljanje asinhronim motorom - zadatak 2

Izvesti pokretanje asinhronog motora u levo i desno i zaustavljanje preko terminala NT11S.

6.2.1. Programiranje terminala

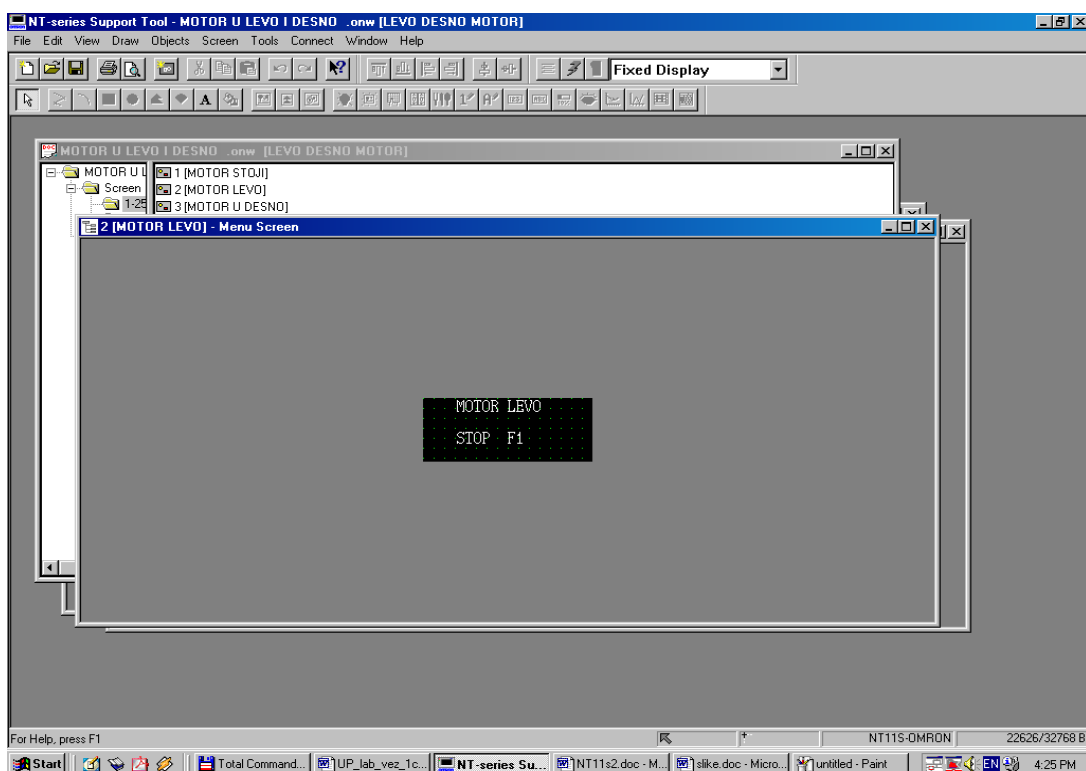
U paleti sa alatima aktiviramo prečicu za TEXT 

Duplim klikom na ekran pojavljuje nam se dijalog box za pisanje teksta na ekran. Uređujući displej dobijamo sledeći ekran.

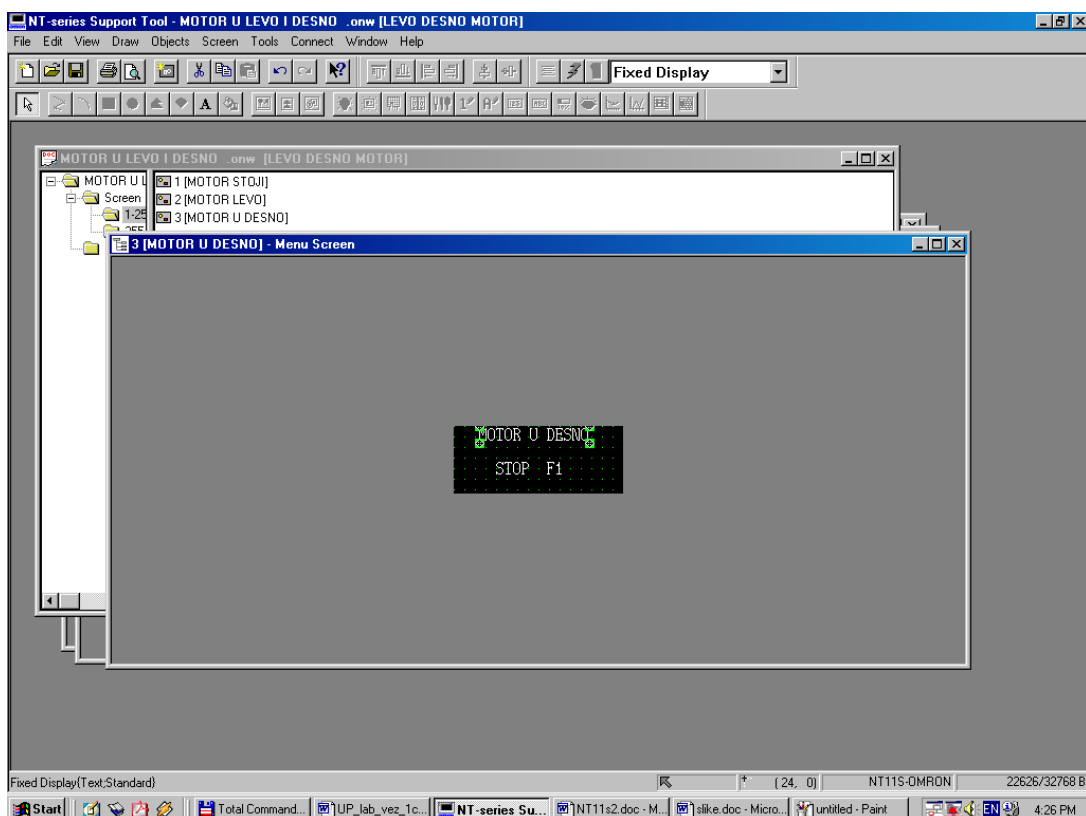


Slika 50. Izgled ekrana MOTOR STOJI

Da bi se omogućilo da se na displeju vidi na koju se stranu motor okreće, napravljena su i sledeća dva ekrana MOTOR LEVO I MOTOR DESNO:



Slika 51. Izgled ekrana MOTOR LEVO

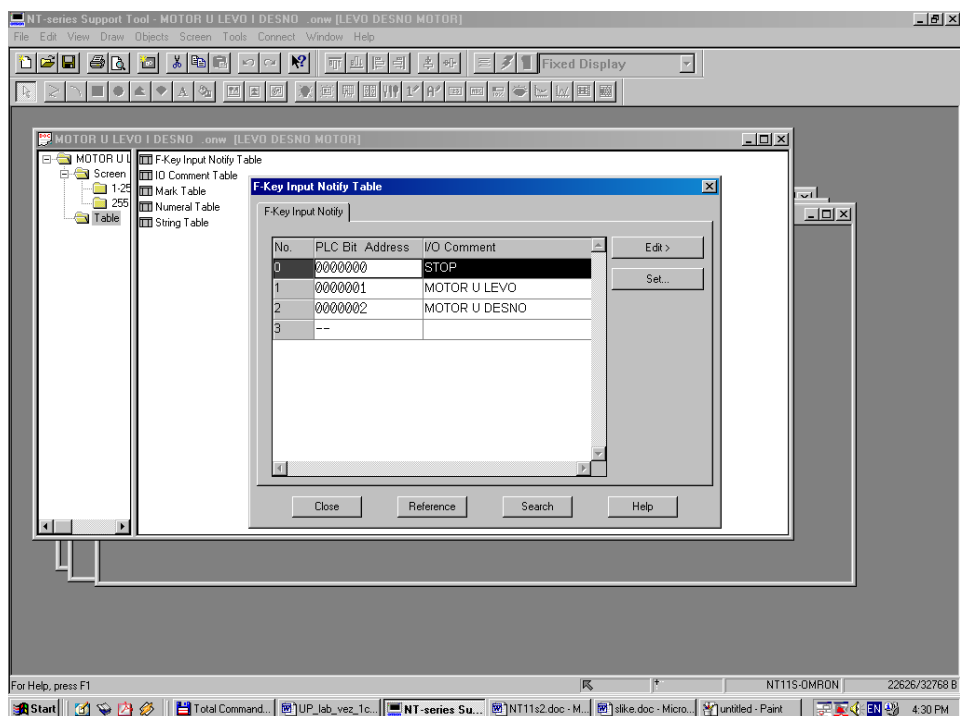


Slika 52. Izgled ekrana MOTOR DESNO

Nakon kreiranja izgleda ekrana treba omogućiti korišćenje funkcijskih tastera, ulaskom u tabelu funkcijskih tastera. U njemu podešavaju se vrednosti za funkcijske tastere prema tabeli:

No. u F-Key	Funkcijski tasteri u PT-u
0	F1
1	F2
2	F3
3	F4

Pri tome koristimo PLC adrese za pozivanje sa programibilnog terminala iz njegovih ulazno/izlaznih (I/U) memorijskih lokacija, prema sledećoj slici, odnosno lestvičastom dijagramu:



Slika 53. Definisane funkcionalnih tastera

6.2.2. Lestvičasti dijagram

Lestvičasti dijagram je dat na slici 54.

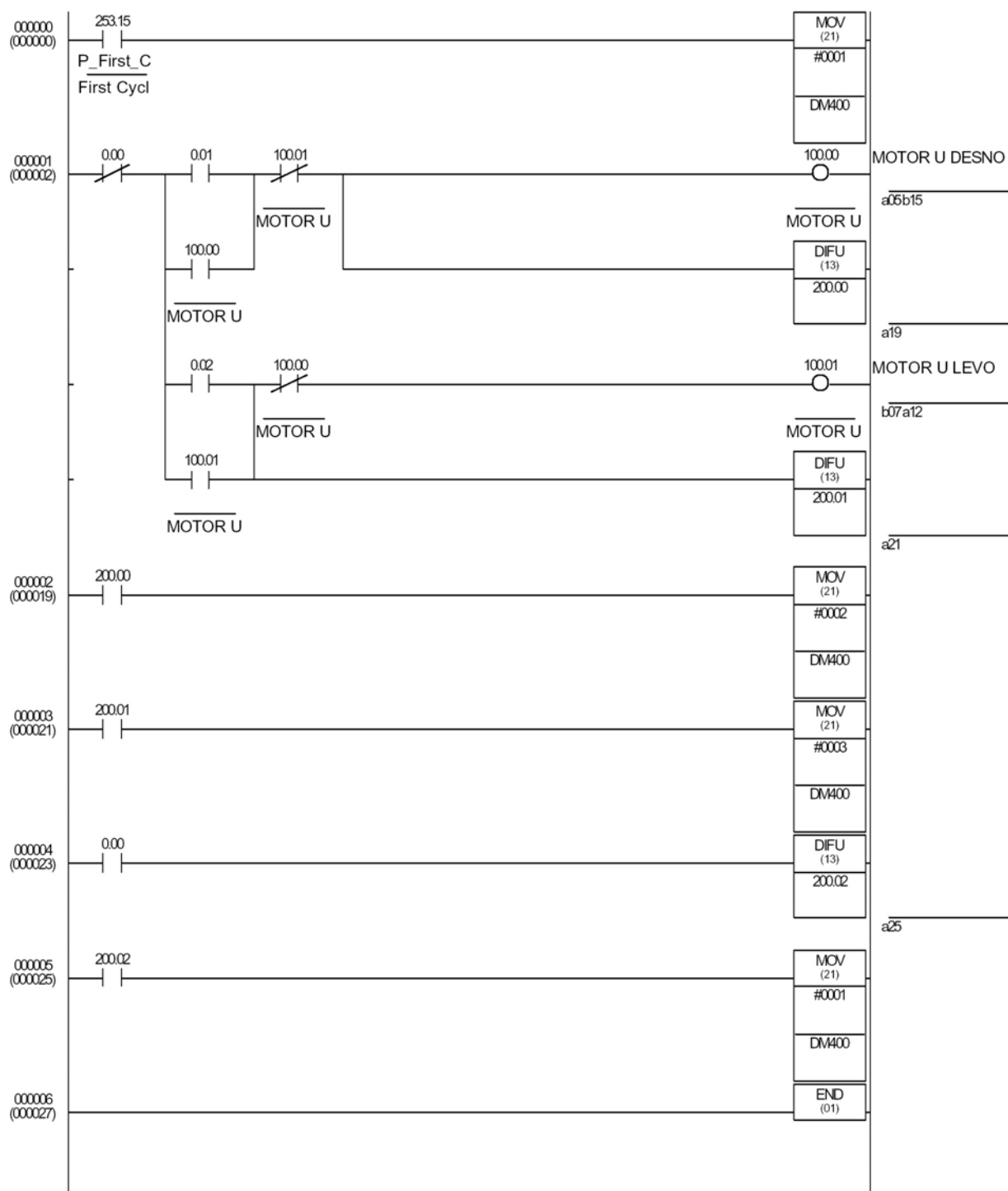
6.3. Primer alarmnog ekrana – zadatak 3

Izvesti alarmni ekran za prikazivanje stanja nužnog stopa i spremnosti za pogon dva frekventna pretvarača koji gone transportnu traku iz dvadesete laboratorijske vežbe.

6.3.1. Programiranje terminala

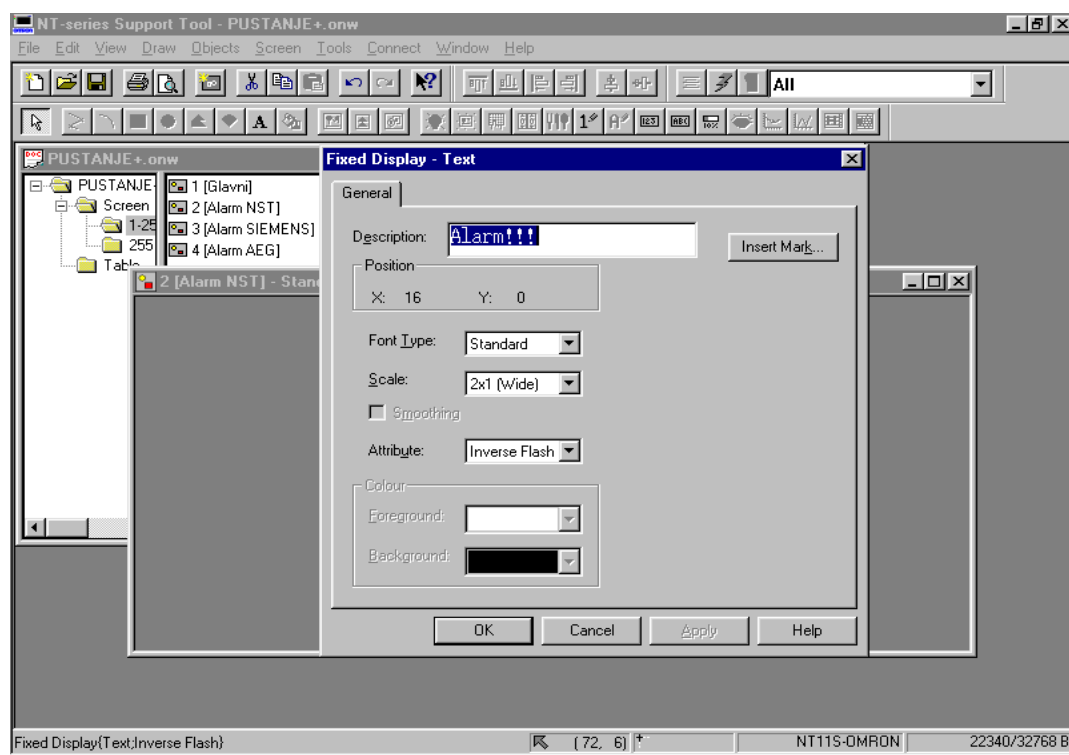
Lestvičasti dijagram i opis upravljanja sistemom transportne trake dati su u posebnom opisu, pa ovde nisu ponovljeni. Problem se svodi na to da pri pritisku na nužni stop i/ili prekidu dojavnih kontakata na frekventnim pretvaračima treba pozvati odgovarajući ekran koji jasno, naprimer uočljivim tekstom koji blinka treba da dojavljuje alarmno stanje. Alarm se kvitira pritiskom na neki od tastera PT-a naprimer F3.

Tekstualno polje u alarmnom ekranu, prema tome možemo urediti na način prikazan na slici 55.



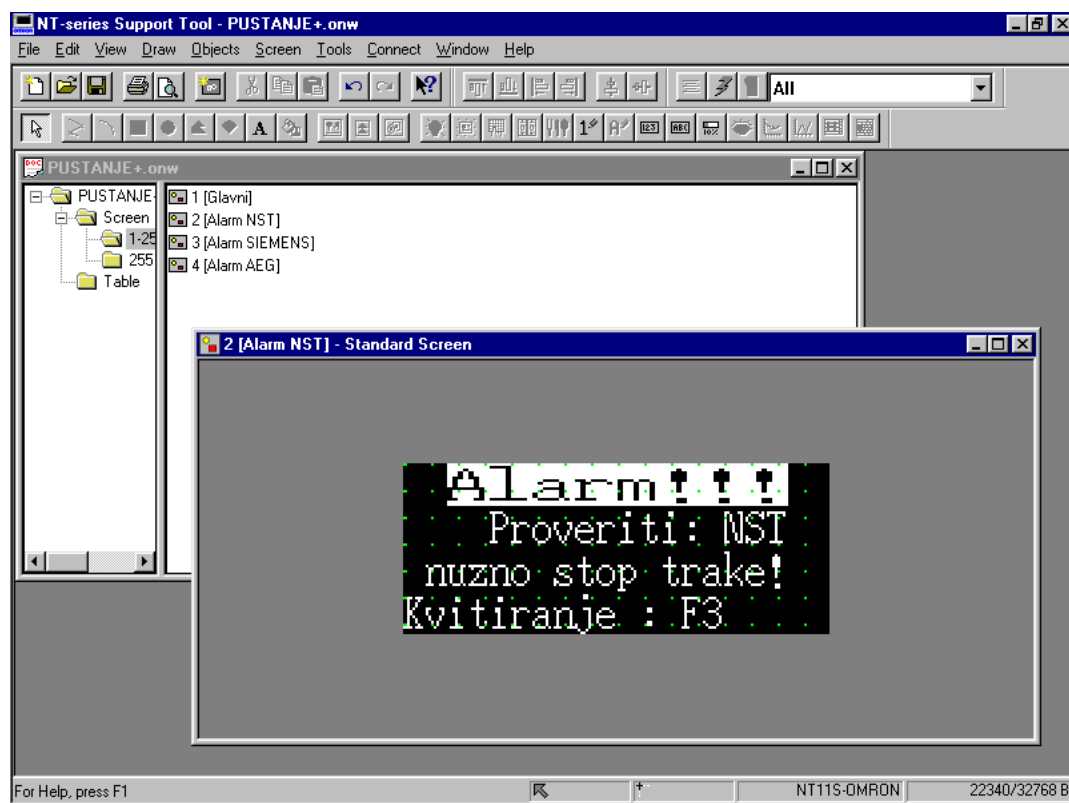
Slika 54. Lestvičasti dijagrama zadatka iz primera 2

U ovom slučaju ćemo promeniti osobine teksta (Attribute). Osobine možemo podesiti na: standardno (Standard), suprotno (Inverse), nastupati, bljeskati (Flash) i suprotno nastupati, bljeskati (Inverse Flash). Ove poslednje tri opcije se mogu i posebno simulirati, da bi se pogledalo kako bi to izgledalo na ekranu sa prečicom  (OFF) i  (ON) u paleti sa alatima.



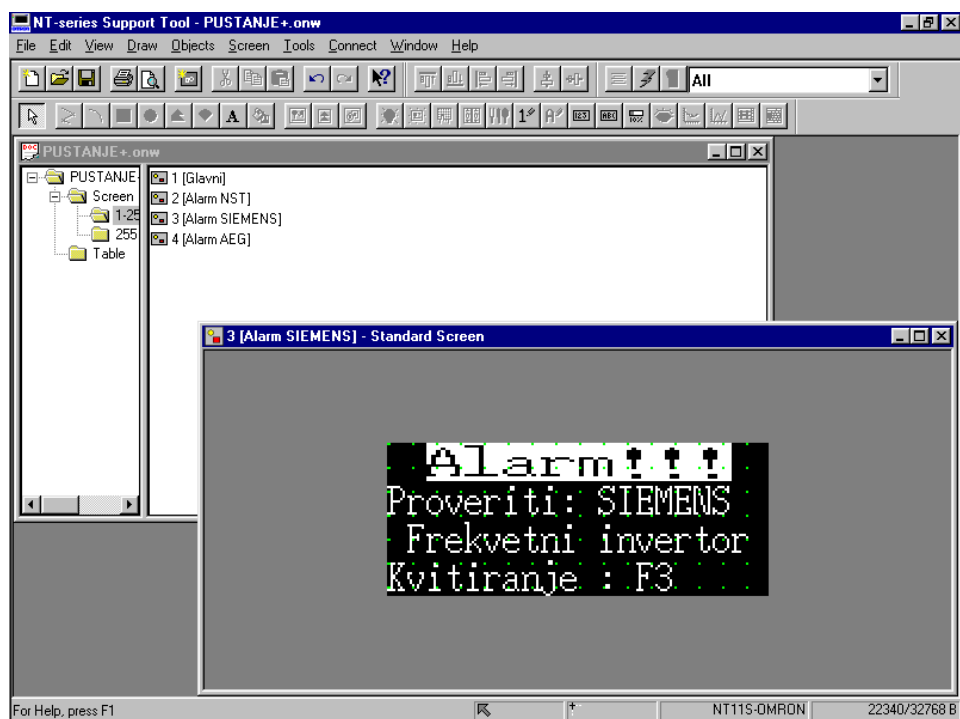
Slika 55. Podešavanje atributa tekstualnog polja u ekranu nužnog stopa NST

Nakon upisivanja ostalog dela teksta ekran za nužni stop trake izgleda na sledeći način.



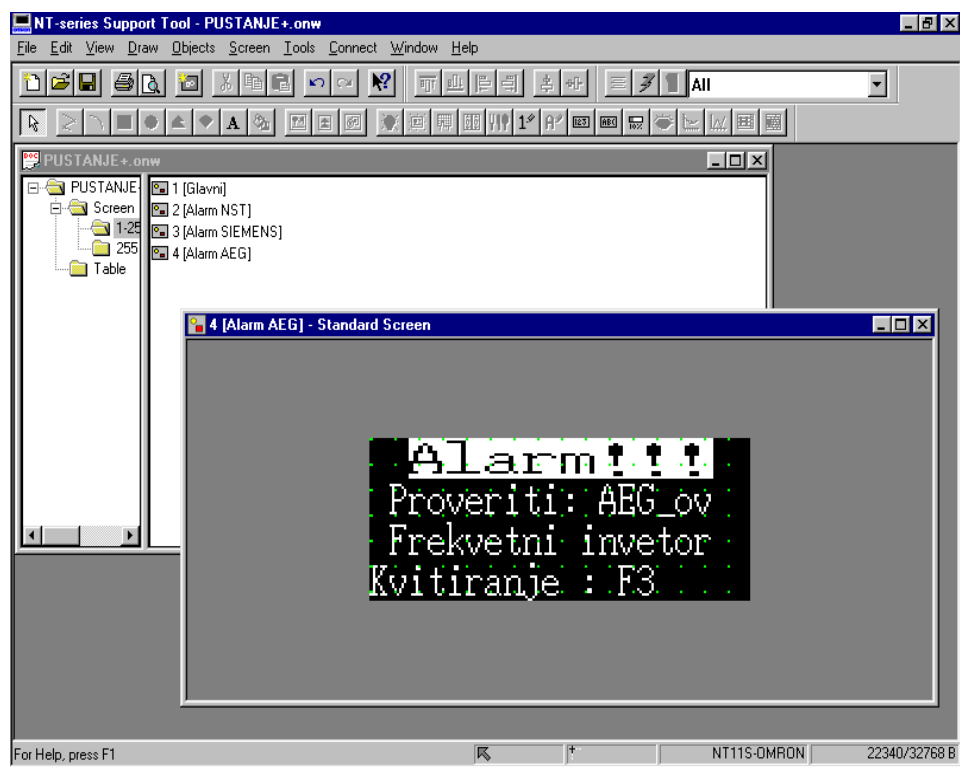
Slika 56. Alarmni ekran nužnog stopa NST

Sledeći je alarmni ekran za Siemens frekventni pretvarač.



Slika 57. Alarmni ekran Siemens-ovog frekventnog pretvarača

Alarmni ekran za AEG-ov frekventni pretvarač.

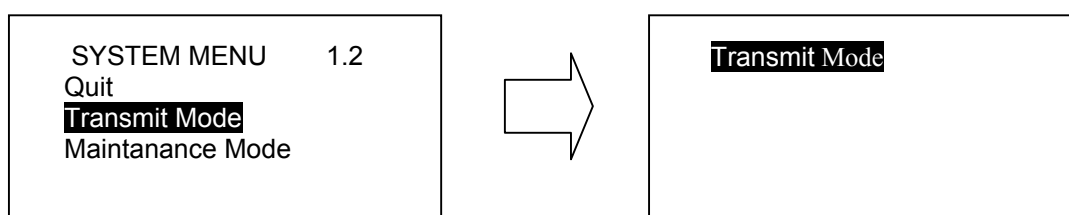


Slika 58. Alarmni ekran AEG-ovog frekventnog pretvarača

7. Prebacivanje projekta u programabilni terminal

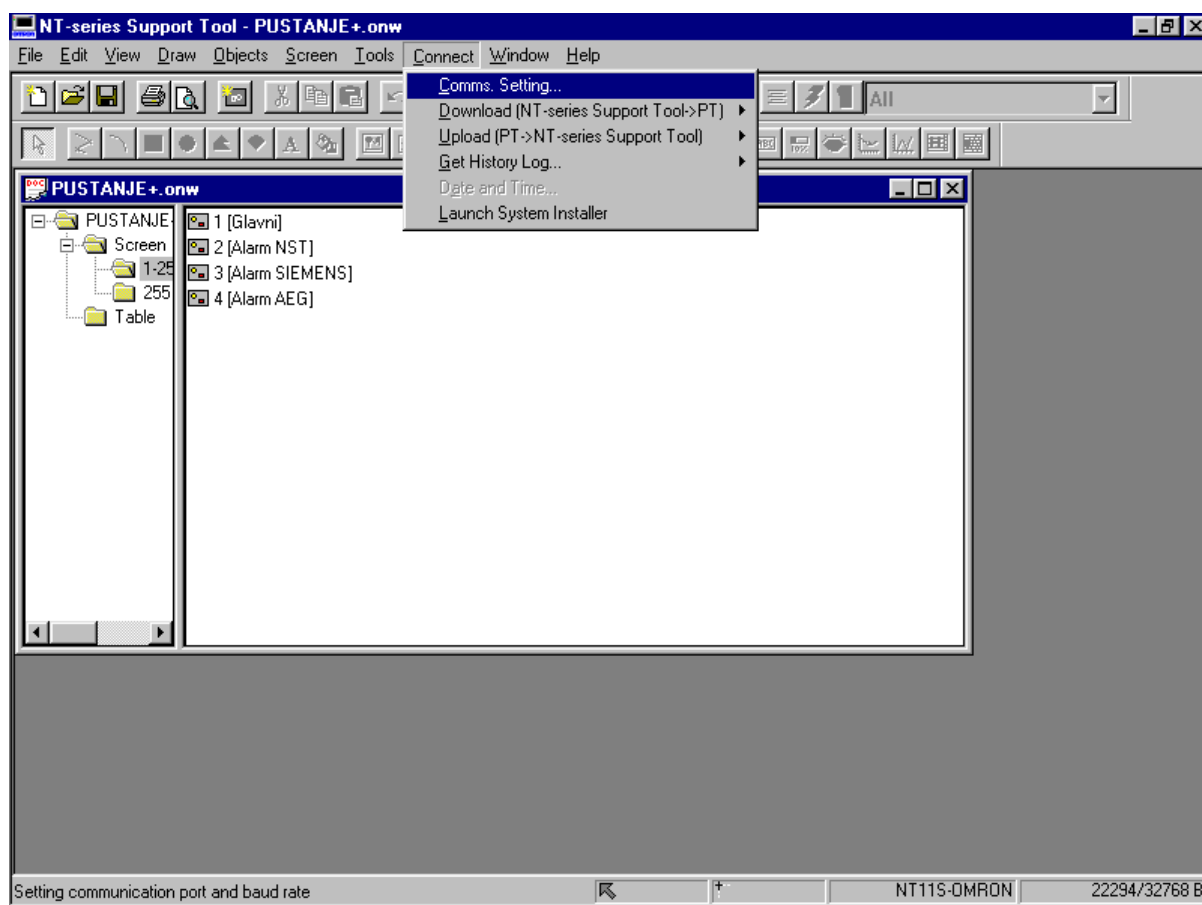
Na početku treba spomenuti da pre prebacivanja projekta u programabilni terminal treba proveriti da li je upisan sistemski program u memoriju terminala. Ako nije treba ga instalirati na način objašnjen u 4. poglavlju.

Kada se terminal priključi na napajanje treba bilo koja dva funkcijska tastera (F1-F4) pritisnuti istovremeno i tada se ulazi u glavni meni programabilnog terminala. Zatim treba izabrati opciju za prebacivanje podataka Transmit Mode. U opciju se ulazi sa ENTER, a izlazi sa bilo koja dva zajedno pritisnuta funkcijska tastera (RESET).



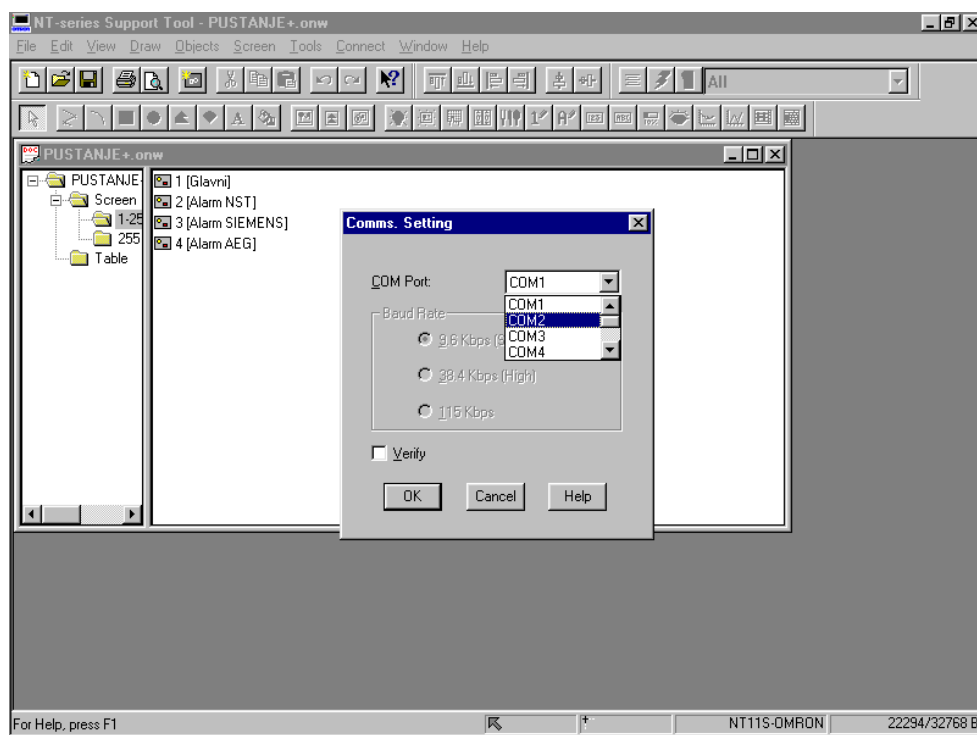
Računar i programabilni terminal treba spojiti preko serijskog RS-232C kabla.

Sledeći korak je određivanje komunikacionog porta koji će se koristiti za prenos podataka. To možemo uraditi preko menija Connect / Comms Setting.



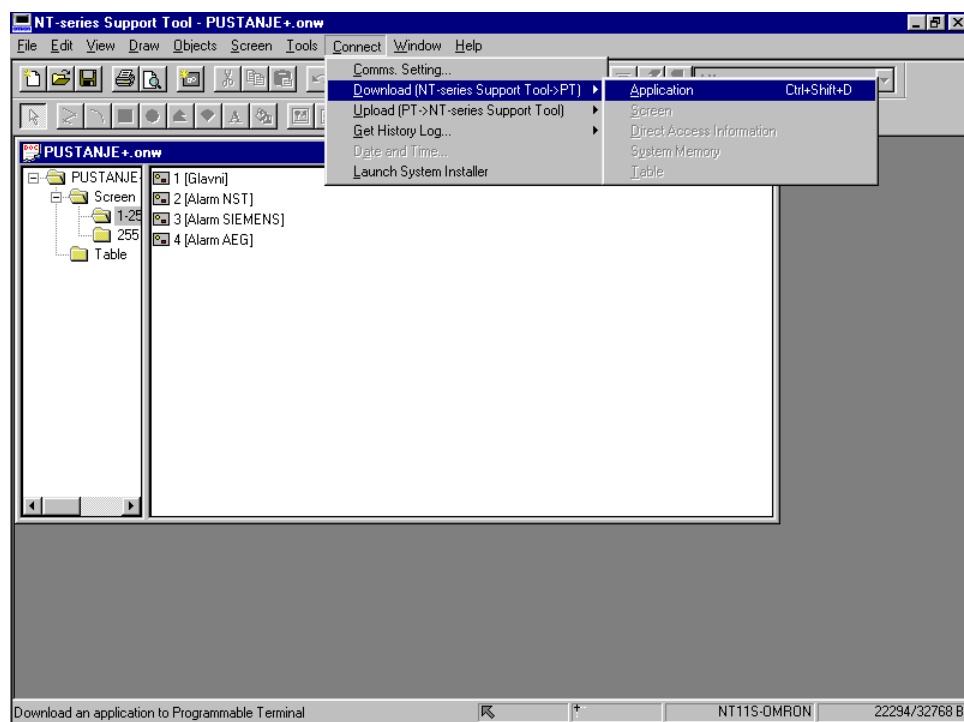
Slika 59. Podešavanje komunikacionog porta za prenos programa u PT

Preduslov za prebacivanje projekta u terminal je tačan izbor komunikacionog porta računara.



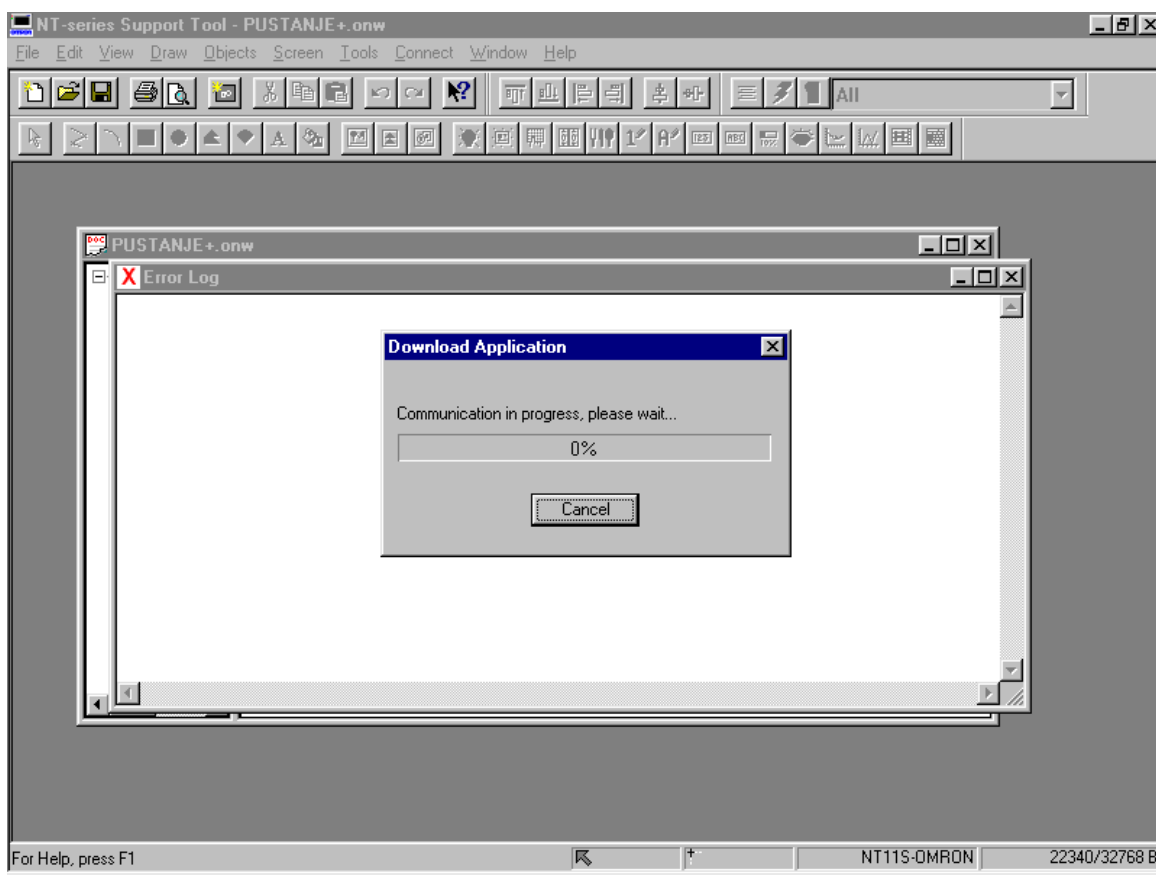
Slika 60. Izbor komunikacionog porta

Sledeći korak je prebacivanje podataka u terminal. Biramo opciju Download (NT-series Support Tool -> PT) / Application.



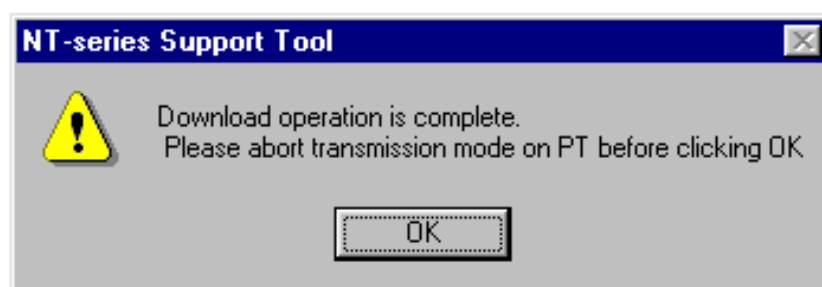
Slika 61. Prvi korak prenosa programa u PT

Nakon startovanja dobijamo poruku da je komunikacija računara i terminala u toku.



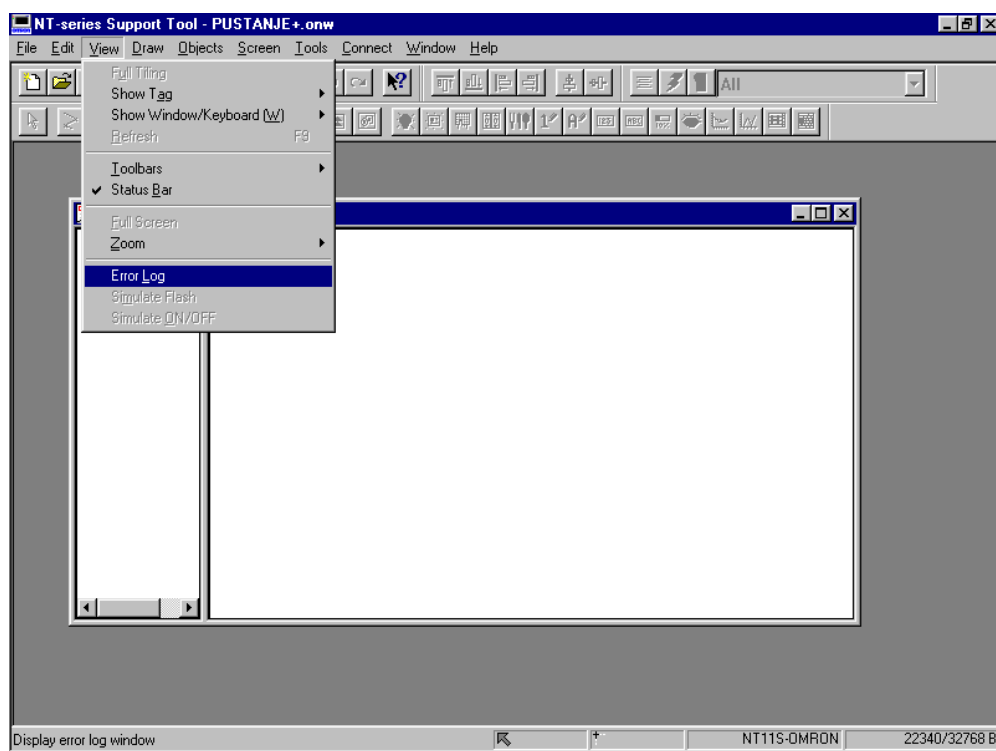
Slika 62. Prenos programa u PT

Kada je prebacivanje podataka završeno treba se vratiti u glavni meni pritiskom bilo koja dva funkcijska tastera (F1-F4) i odspojiti RS-232C serijski kabl. Nakon prebacivanja podataka pritisnemo OK.



Slika 63. Poruka o uspešnom prenosu programa u PT

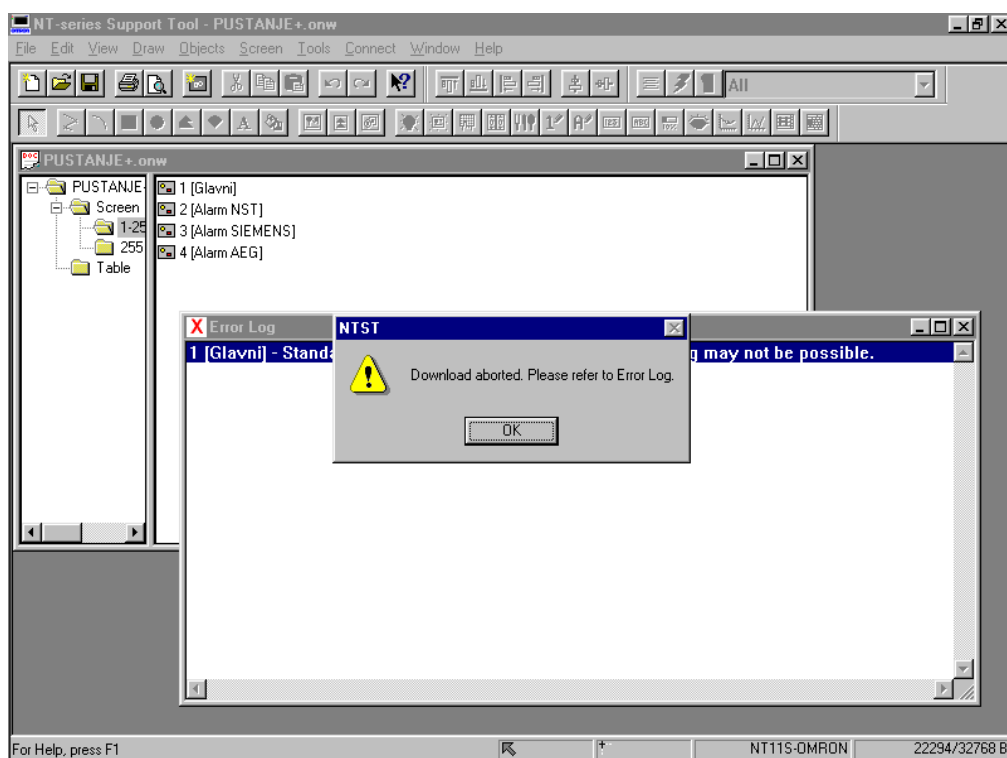
Ali pre toga da napomenem koje su greške u radu sa prebacivanjem podataka tj. postoji opcija koja nam omogućuje da vidimo gde je greška ako se pojavi obaveštenje u vezi nje.



Slika 64. Izbor za pregled izveštaja o prenosu

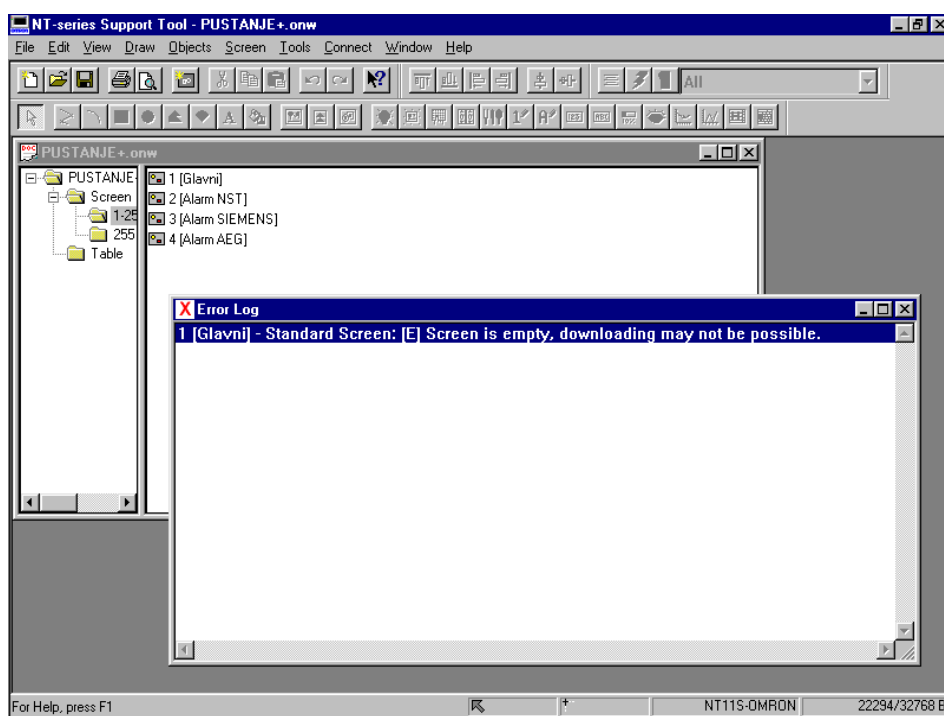
A greške koje se najčešće javljaju , prilikom prebacivanja podataka, su sledeće:

- kad smo zaboravili da stavimo text na ekran a pokušavamo da prebacimo ekran u memoriju PT:



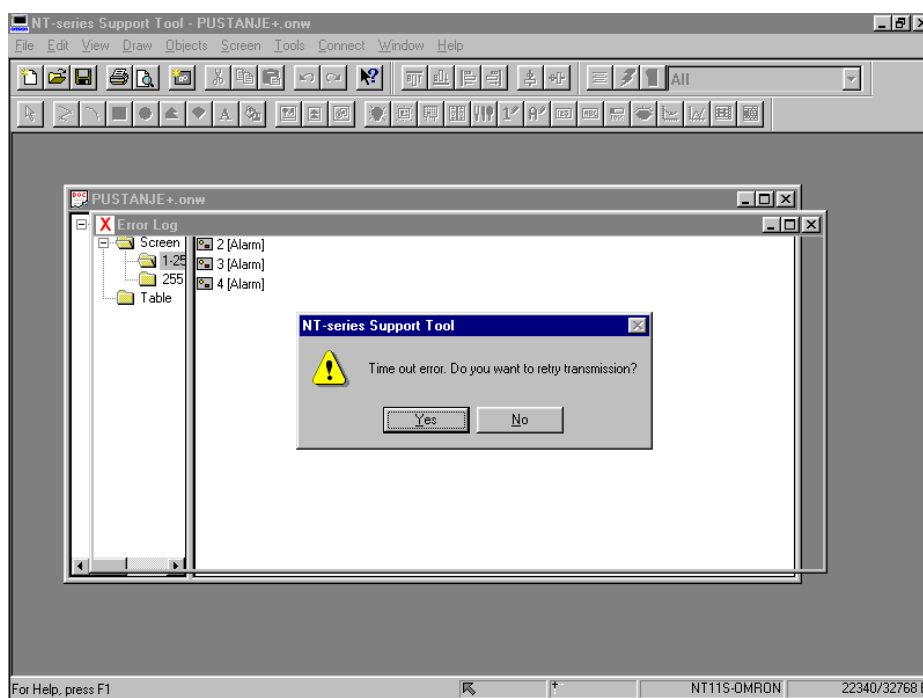
Slika 65. Poruka o prekinutom prenosu

Kada pogledamo u greške (Error Log) tu pročitamo šta je glavni problem što nije uspelo prebacivanje:



Slika 66. Izveštaj o razlogu prekida prenosa zbog nedefinisanog glavnog ekrana

- A vrlo česta greška je da nismo postavili serijski kabl ili da nismo odredili dobar port pa će program stalno javljati ovog tipa upozorenja sve dok to ne izmenimo:



Slika 67. Izveštaj o razlogu prekida prenosa zbog neuspostavljanja veze sa PT-om

OSNOVI PRIMENE *SCADA* SISTEMA ZA SUPERVIZIJSKU KONTROLU I PRIKUPLJANJE PODATAKA

1. Cilj labotatorijske vežbe:

- Upoznati se sa načinom funkcionisanja i primene *SCADA* sistema za supervizijsku kontrolu i prikupljanje podataka "CX-Supervisor" firme "Omron".
- Odvežbati priložene primere projekata *SCADA* sistema.
- Odvežbati samostalno dodatno zadate primere projekata *SCADA* sistema.

2. Laboratorijska vežba se sastoji od:

- PC računar Pentium III sa instaliranim softverom CX-Supervisor

3. Uvod

Sistem procesnog upravljanja čine proces kojim se upravlja, upravljačka oprema i čovek-operater.

Proces kojim se upravlja formira se od krupnih mehaničkih jedinica i njihovih međuspojeva, koji omogućuju skladištenje, obradu, izmenu i reciklažu materije i energije. Primeri su rezervoari, kondenzni sudovi, izmenjivači toplote, grejači, pumpe, elektromotori, transformatori, generatori, kotlovi, reaktorske posude i slično. Integracija ovih jedinica u veće jedinice ili postrojenje, izvodi se povezinjanjem krupnih elemenata cevovodima, ventilima, energetskim kablovima, reduktorima, kontaktorima itd.

Osnovni resursi proizvodnog procesa su: uloženi kapital, oprema za proizvodnju, sirovine, informacije i ljudski resursi. Pravilnim korišćenjem i organizovanjem raspoloživih resursa mogu se zadovoljiti pooštreni proizvodni zahtevi današnjice koji se odnose na:

- poboljšanje kvaliteta proizvoda;
- povećanje efektivnosti proizvodnje (bolje korišćenje kapaciteta, manje otpada);
- povećanje pouzdanosti i sigurnosti proizvodnog procesa;
- zaštita okoline;
- manja potrošnja energije i sirovina po jedinici proizvoda itd.

Podizanje nivoa organizovanosti i racionalnosti proizvodnog procesa postiže se upravljanjem. Sa stanovišta tematike koja će se u daljem razmatrati pod upravljanjem ćemo podrazumevati proces preslikavanja informacija o stanju procesa iz prostora merenja u prostor akcija.

Upravljanje $u(t)$ je funkcionalno dejstvo na proizvodni proces, koji se bira iz nekog skupa mogućih upravljanja U . Izbor odgovarajuće funkcije upravljanja i modifikacija upravljačkih parametara, vrši se na osnovu informacija o stanjima i promenama u proizvodnom procesu i okruženju. Za pravilno donošenje bilo kakve upravljačke odluke znači potrebno je uspostaviti sistem za prikupljanje, obradu i pravovremeno pretvaranje rezultata obrade u akcije. U savremenim tehničkim rešenjima proces prikupljanja, prezentacije i analiza informacija kao i izbor upravljačkih akcija obavlja se u upravljačkom računarskom sistemu. Na slici 2.1 prikazana je skica informacionih veza između proizvodnog procesa i čoveka - operatera.

transceiveri). Podaci se učitavaju i obrađuju a dobijeni rezultate prikazuju i/ili memorišu na nekoj od pomoćnih memorija. Istorijski gledano, prvi mini računari primenjeni na proizvodne procese imali su isključivo informacione funkcije, čiji je cilj permanentno praćenje procesa, detekcija alarmantnih situacija i eventualno detaljnije sagledavanje arhiva procesa na bazi off-line analiza prikupljenih podataka. Međukorak ka upravljačkim funkcijama su tzv. vodiči operatera, koji daju preporuke za manuelno upravljanje.

Kod delimično ili potpuno automatizovanih sistema automatskog upravljanja teško je jasno razdvojiti informacione i upravljačke funkcije, obzirom da su upravljačke funkcije uvek jednim svojim delom informacione.

U pogledu upravljačkih funkcija praksu je potvrdila značaj i aktuelnost koncepta hierarhijskog upravljanja složenim sistemima. Naime, upravljačke funkcije upravljačkih računarskih sistema uređene su hierarhijski, od prostijih ka složenijim, tako da obrazuju višenivovsku strukturu koja je standardno prihvaćeni u praksi.

U tabeli 4.1 dat je hierarhijski sistem skup upravljačkih funkcija upravljačkih računarskih sistema. Posebni komentari se odnose na cilj upravljanja (upravljanje stacionarnim stanjem ili dinamikom prelaznih procesa), strukturu sistema upravljanja (u otvorenoj ili zatvorenoj sprezi) i strukturu odgovarajućeg upravljačkog računarskog sistema.

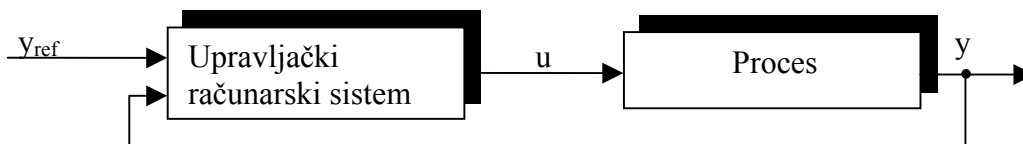
Ni vo	Funkcija upravljačko računarskog sistema	Cilj upravljanja		Struktura sistema upravljanja	Struktura upravljačkog računarskog sistema
		Stacionarno stanje	Dinamika prelaznih procesa		
5		Upravljanje složenim tehnološkim kompleksima - interakcija sa okolinom i ekonomski aspekti		Programska ili u povratnoj sprezi	Složena računarska mreža
4		Upravljanje složenim tehnološkim kompleksima - koordinacijom funkcija		Programska ili u povratnoj sprezi	Lokalna računarska mreža
3	Supervizijsko upravljanje			Programska ili u povratnoj vezi sa malom dinamikom	Mikro ili mini računari (PC)
2	Direktno digitalno upravljanje		Servo ili regulacioni problem	U povratnoj vezi	Standardni regulator ili PLC
1	Logičko upravljanje	Relizacijom prekidačkih funkcija direktno ili indirektno upravljanje stacionarnim stanjem ili dinamikom		Programsko i/ili u povratnoj vezi	PLC

Tabela 4.1. - Upravljačke funkcije upravljačkih računarskih sistema

Logičko upravljanje se ostvaruje na najnižem hierarhijskom nivou realizacijom prekidačkih (Bulovih) funkcija, kombinovanih sa kašnjenjem i drugim elementarnim funkcijama potrebnim za realizaciju logičkog upravljanja. Upravljački računarski sistemi koji obavljaju ove funkcije poznati su pod nazivom Programabilni logički kontroleri - PLC. Mogu se primeniti u otvorenoj ili zatvorenoj

sprezi. U zatvorenoj sprezi realizuju upravljanje redosledom operacija (sequencing control) ili tzv. "paketno" upravljanje (batch control).

U sistemima sa direktnim digitalnim upravljanjem upravljački računarski sistem generiše upravljačke signale u realnom vremenu na bazi merenja analognih i/ili digitalnih promenljiva procesa, i nalazi se unutar petlje upravljanja. Pouzdani sistemi sa direktnim digitalnim upravljanjem postali su ostvarljivi tek kada je tehnologija omogućila dovoljnu brzinu računara. Na slici 2 prikazana je blok šema sistema sa direktnim digitalnim upravljanjem.



Slika 4.2. - Struktura sistema direktnog digitalnog upravljanja

Referentna vrednost se definiše, u opštem slučaju, van povratne petlje, od strane operatera ili nadređenog nivoa upravljanja. Rezultati merenja se takođe dovode u upravljački računarski sistem u kom se realizuje algoritam upravljanja, koji, kao rezultat generiše u realnom vremenu upravljački signal tj. ulaz u proces. Primena upravljačkih računarskih sistema u diskretnom digitalom upravljanju proizvodnim procesima pruža široke mogućnosti za izbor algoritma.

Osnovna uloga supervizijskog upravljanja je postavljanje referentnih vrednosti za regulacione petlje sa direktnim digitalnim upravljanjem, na bazi tehničkih merenja podataka (izlazi i ulazi u proces, kao i promenljive kao što su željene osobine proizvoda, cena energije itd.), i zadatih kriterijuma. Postavljanje fiksnih referentnih vrednosti spada u klasu sistema sa programskim upravljanjem. Formalno, problem određivanja referentnih vrednosti rešava se korišćenjem statičkih modela sistema koji opisuju njegovo stacionarno stanje. Korelacije referentnih vrednosti vrše se u realnom vremenu. Upravljački računarski sistemi za realizaciju ovih zadataka moraju realizovati brzo i tačno matematičke operacije.

Na nivou koordinacije i planiranja usklađuje se rad podprocesa u okviru većih celina i planira se dugoročna upravljačka strategija na bazi globalnih kriterijuma (koji tada obuhvataju i ekonomske aspekte) i analize interakcije sa okolinom.

Zadaci koordinacionog nivoa su:

- praćenje materijala;
- upravljanje redosledom obavljanja funkcija (sekvenciranje);
- koordinacija supervizorskih funkcija na višem nivou;
- usklađivanje rada podsistema na nižem nivou;
- upravljanje skladištima.

Funkcije nivoa planiranja su sledeće:

- definicija opštih zadataka celine i pojedinih pogona;
- analiza ekonomskih faktora (porudžbine, cene, prodaja itd.) i iznalaženje optimalnih rešenja (maksimizacija profita);
- koordinacija sa drugim upravljačko - informacionim sistemima;
- izveštavanje o globalnim efektima strategije upravljanja.

Na ovim "višim" nivoima koriste se računarski sistemi većih memorijskih kapaciteta i aritmetičke snage.

5. SCADA sistemi

Sistem za supervizijsku kontrolu i prikupljanje podataka uobičajeno nosi naziv *SCADA* što je zapravo akronim engleskog originalnog naziva "*Supervisory Control And Data Acquisition*" (u prevodu "*supervizijsko upravljanje i prikupljanje podataka*"). *SCADA* u najširem smislu predstavlja kompletan sistem čije su osnovne funkcije sadržane u njegovom nazivu, dakle sistem koji omogućava nadzor (eng. *monitoring*) i upravljanje nad delom ili celinom proizvodnog procesa. Preciznije, ovakav sistem u sebe uključuje, s jedne strane, potrebnu mernu, reglacionu i izvršnu opremu sa svim pratećim elementima: energetikom, sistemima besprekidnog napajanja (UPS - Uninterruptable Power Supply), komunikacionim sistemima, upravljačkim platformama (komandni pult, ekran koji reaguje na dodir itd.), računarske sisteme bazirane na različitim platformama, koji vrše finalno prikupljanje, obradu i prezentovanje podataka, regulacione, savetodavne i druge funkcije.

Po svojoj strukturi *SCADA* predstavljaju hibridne sisteme u koje je inkorporiran kako tehnološki hardver (merna, servo i procesna oprema), tako i uobičajeni računarski hardver u užem smislu. Nagli razvoj digitalne tehnologije učinio je da je 80% merne i regulacione tehnike danas bazirano na mikro i mini računarskim sistemima, tako da savremene *SCADA*-e predstavljaju najvećim delom zapravo lokalne računarske mreže koji obuhvataju računare veoma različite po svojoj strukturi, mogućnostima i funkcijama. Kompleksnost ovakvih sistema je ogromna i u skladu je sa njihovom ulogom, jer dobro projektovan, izveden i podešen *SCADA* sistem treba da donese višestruke koristi:

- značajne uštede u proizvodnji, kroz ekonomičnije korišćenje resursa i smanjenje škarta;
- poboljšanje kvaliteta proizvoda;
- dobro praćenje stanja tehnološke opreme i smanjenje troškova njegovog održavanja kroz funkcije upozorenja i dijagnostike;
- skraćenje trajanja prekida i minimizaciju gubitaka usled havarija;
- tačnije praćenje obima proizvodnje i učinka, kao i lakše planiranje proizvodnje i rukovođenje zalihama;
- lakše i znatno brže prilagođenje proizvodnje zahtevima tržišta za izmenjenim, poboljšanim ili čak novim proizvodima.

6. Funkcionalna klasifikacija računara u okviru *SCADA* sistema

SCADA sistemi u većini primena sadrže po nekoliko desetina računarskih sistema čija se klasifikacija po funkcijama može izvršiti na sledeći način:

- inteligentna merna oprema;
- akvizicioni računari;
- reglacioni sistemi;
- centralni računarski sistem (nadzorni računarski sistem u užem smislu, men - machine interface - MMI), i
- mrežni računari (serveri).

6.1. Inteligentna merna oprema

Savremena merna oprema već uobičajeno sadrži mikrokontrolerski ili mikroracunarski sistem, koji osim nominalnih funkcija (merenje, preračunavanje, filtriranje, prikazivanje, promena skala, konvertovanje, totalizacija, dijagnostika) može da obavlja i neke jednostavnije akvizicione ili regulacione funkcije. Mogućnost komunikacije preko nekog serijskog priključka je praktično uvek

prisutna. Ovakvi uređaji ne sadrže masovne eksterne memorije (hard, *CD* niti floppy dražjove), komunikaciju sa korisnikom obavljaju preko nekoliko tastera i alfanumeričkih displeja i pakuju se u kućišta sa visokim stepenom zaštite predviđenim za montiranje na zid ili u elektro-orman. Odlikuje ih visoka robustnost, budući da su predviđeni za ugradnju u neposrednoj blizini mernih mesta, dakle u ekstremnim ambijentalnim uslovima. Zavisno od važnosti veličine koju mere, mogu biti priključeni na sisteme besprekidnog napajanja, kako bi se obezbedio kontinuitet (ili neophodnu vremensku autonomiju) nadzornih i/ili upravljačkih funkcija.

6.2. Akvizicioni računari

Osnovna funkcija akvizicionih računara u *SCADA* sistemima je informaciona, tj. prikupljanje i pamćenje veće količine podataka. Mada je to uobičajena funkcija mnogih računarskih sistema, akvizicioni računarski sistemi kakvi se koriste u industriji, pa i u *SCADA* sistemima, su po mnogo čemu specifični. Kao i u slučaju drugih računara za industrijsku upotrebu, od ovih računara se očekuje robustnost u svakom pogledu, što ima direktnog uticaja na njihovu arhitekturu. Drugim rečima odsustvo mehaničkih, elektromehaničkih delova (diskova i tastature), koji su ujedno i najosetljiviji, je uobičajeno za ove računare. Kako je njihova primarna funkcija prikupljanje podataka, ali ne i njihovo prezentovanje, praktično ne raspolažu hardverom za komunikaciju sa operaterom, a korisnički interfejs je sužen na svega nekoliko tastera i alfanumerički displej. S druge strane, ovi računari baterijski podržanim memorijskim kapacitetima, obično dovoljnim da podatke prikupljane sa vremenskom bazom od nekoliko sekundi prate nedeljama ili mesecima unazad. Kako su izvori podataka sami merni i indikacioni uređaji, čije je analogne merne signale potrebno digitalizovati, akvizicioni računari o kojima je reč, raspolažu potrebnim brojem analogno - digitanih (A/D) konvertorskih kanala. Pošto broj mernih signala može varirati od svega nekoliko do nekoliko stotina po jednom akvizicionom računaru, njihova arhitektura je uobičajeno modularna, a projektant određuje broj i tip ulaznih modula zavisno od broja i vrste mernih signala.

Obzirom na ogromne ulazne mogućnosti, akvizicioni računari mogu biti korišćeni samo kao inteligentne ulazne jedinice velikog kapaciteta, koje komuniciraju putem mreža sa drugim računarima u *SCADA* sistemu. U ovom slučaju akvizicioni računari zapravo rade u realnom vremenu i prosleđuju arhivirane podatke odmah. U ovakvim primenama broj akvizicionih računara se prilagođava broju mernih signala i koncentraciji merne opreme. Obzirom na značajan pad cena ovakve opreme, važan kriterijum postaje smanjenje troškova kabliranja, pa se akvizicioni računari postavljaju na mestima koncentrisanja mernih signala u samom proizvodnom procesu.

Potpuna funkcija akvizicionih računara dolazi do izražaja u primenama u kojima priroda tehnološkog procesa ne zahteva očitavanje prikupljenih podataka u realnom vremenu. Po pravilu se radi o tehnološkim celinama koje nemaju direktan uticaj na praćeni proizvodni proces, ili o dislociranim delovima postrojenja i/ili mernim stanicama, često bez ljudse posade; ponekad je u pitanju i prikupljanje podataka za formiranje statičkih modela u svrhu implementiranja funkcija supervizijskog upravljanja i koordinacije. U ovakvim primenama vremenske baze prikupljanja podataka su često vrlo spore (minuti, sati), a podaci se prenose ostalim *SCADA* računarima na sat, dnevno ili čak nedeljno; ne retko komunikacija nije vremenski sinhronizovana, već se podaci prosleđuju ostalim računarima tek po nastanku nekog značajnog događaja, kao što bi moglo biti registrovano merenje izvan dozvoljenih granica, pojava prirodnog fenomena npr. padavina, ispad uređaja i slično. Za ovakve primene akvizicioni računari se opremaju hardverom i softverom za ekonomisanje sopstvenom potrošnjom, baterijskim izvorom napajanja, kao i neophodnim komunikacionim interfejsima u slučaju dislociranih objekata (radio GSM ili žičanim modemima).

Budući da su usko specijalizovani, akvizicioni računari po pravilu ne zahtevaju nikakav aplikativni softver. Eventualno konfigurisanje definiše vremenske baze prikupljanja podataka, način prikupljanja podataka (stalno ili po nastanku događaja), neke parametre komunikacije sa ostalim računarima u sistemu i slično.

6.3. Regulacioni sistemi

Regulacioni sistem (upravljački sistem u užem smislu) predstavljaju centralni deo *SCADA* sistema, osim ukoliko je *SCADA* sistem sveden na čisto informacionu funkciju, pa u njemu ne postoji upravljački deo. Računari upravljačkog dela *SCADA* sistema su "odgovorni" za upravljanje procesom i od kvaliteta te funkcije direktno zavisi uspešnost čitavog sistema u ekonomisanju proizvodnjom.

Upravljačke funkcije *SCADA* sistema mogu biti u širokom opsegu: od daljinskog izvršenja manuelnih komandi unetih preko centralnih računara *SCADA*-e, do automatizovanog uvođenja dela ili celine proizvodnog procesa. U prvom slučaju je zapravo u pitanju ručno daljinsko upravljanje i to obično predstavlja fazno(etapno) rešenje *SCADA* sistema u pokušaju da se stigne do sistema sa što većim učešćem "prave" automatike. Kada je u pitanju automatizovano vođenje procesa, što je obično krajnji cilj, u praksi se mogu uočiti dve dominantne vrste upravljačkih funkcija: logičke i funkcije direktnog digitalnog upravljanja (DDC - Direct Digital Control).

Pojava jeftine i pouzdane digitalne tehnologije otvorila je vrata razvoju kompletno novih struktura regulacionih sistema. Čitav niz teorijski dobro poznatih koncepcija postao je praktično izvodljiv: prediktivni, adaptivni, robusni, fazi (fuzzy) regulatori, regulacioni sistemi bazirani na primeni ekspertnih sistema i veštačke inteligencije. U *SCADA* sistemu postalo je moguće da se u jedan računar ugradi veliki broj regulacionih krugova; obezbedi uključivanje uticaja raznih veličina, i tako obezbedi formiranje kompleksnih multivarijabilnih regulatora visoke performanse. Uvođenje ovih struktura premešta problematiku projektovanja i izrade kvalitetnih *SCADA* sistema iz domena digitalne tehnologije u domen teorije i prakse automatskog upravljanja. Prikupljanje informacije od strane *SCADA* sistema, mogu u velikoj meri, uz korišćenje savremenih moćnih CAD (Computer Aided Design) alata da se iskoriste za identifikaciju poremećaja, formiranje matematičkih modela, otkrivanje nepravilnosti u radu izvršnih organa itd. Implementacija ovih rešenja u *SCADA* sistemima svodi se na projektovanje softvera bez ikakvih zahteva u domenu tehnološkog ili računarskog hardvera.

Logičko upravljanje i jednostavniji regulacioni sudovi otvaraju se u okviru *SCADA* sistema sa programabilnim logičkim kontrolerima PLC (Programmable Logic Controller) ili u specijalizovanim radnim stanicama (Field Station). Ovi računari podležu veoma oštrim zahtevima u pogledu robustnosti i srednjeg vremena rada pre prvog otkaza (MTBF - Mid Time Before Failure) u odnosu na akvizicione računare.

6.4. Samostalni regulatori

Samostalni regulatori su upravljački računari koji u sebi sadrže analogno/digitalne (A/D) i digitalno/analogne (D/A) konvertore potrebne za direktno sprezanje sa mernom i izvršnom opremom i kao takvi mogu izvršavati regulacione funkcije nezavisno od *SCADA* sistema. Tipične podfunkcije ovih uređaja su:

- mogućnost ručnog vođenja procesa;
- ručno ili automatizovano (autotuning) podešavanje promenljivih parametara regulatora;
- aktiviranje modifikacija osnovnog algoritma;
- definisanje profila referentnog (setpoint) signala, ograničenje njegovog rasta i slično.

Funkcije se mogu uobičajeno aktivirati preko jednostavnog korisničkog interfejsa, na prednjoj masici uređaja, ili preko centralnog računara korišćenjem serijske komunikacije. Uređaji su obično predviđeni za ugradnju u kontrolne table ili elektro-ormane. Izvode se u različitim tehnologijama, obično uz korišćenje mikrokontrolera, mikroprocesora ili signal procesora (DSP - Digital Signal Processor). Uobičajeno se ne mogu programirati već samo konfigurisati za konkretnu aplikaciju.

6.5. Programabilni logički kontroleri

Ovaj tip upravljačkih računara je veoma čest u *SCADA* sistemima, pre svega zbog njihove fleksibilnosti. Za razliku od dosada primenjenih računarskih komponenti *SCADA* sistema, ovaj tip računara je programabilan, što je osnova njegove fleksibilnosti. U hardverskom pogledu najvažnija je modularnost, što omogućava da ovakav računar bude opremljen potrebnim brojem dodatnih modula različitih tipova:

- ulazni moduli (analogni, digitalni, brojački, specijalizovane za posebne merne davače kao što su Pt100, termoparovi i slično), koji vrše analognu - digitalnu konverziju raznovrsnih mernih i indikatorskih signala;
- izlazni moduli koji generišu potrebne komandne signale kroz digitalno - analognu konverziju;
- komunikacioni moduli koji služe za podršku serijske komunikacije sa drugim računarima u mreži;
- regulacioni moduli koji se efektivno ponašaju kao samostalni regulatori;
- specijalizovani moduli za priključivanje pisača, ručnih terminala, diskova i sl.

Shodno primeni, projektant odlučuje o broju i tipu dodatnih modula koji se ugrađuju uz postojeću centralnu jedinicu PLC-a, čine da je ovakav tip računara izuzetno fleksibilan i u hardverskom smislu.

Jednom hardverski konfigurisan, PLC može da izvršava različite funkcije:

- regulacione, bilo korišćenjem specijalizovanih modula ili kroz softversku implementaciju, korišćenjem standardnih ulaznih i izlaznih modula;
- funkcije logičkog upravljanja, korišćenjem ulaznih i izlaznih modula;
- akvizicione funkcije, korišćenjem ulaznih modula;
- komunikacione funkcije, korišćenjem odgovarajućih modula, kada PLC može imati čitav niz uloga (master, koncentrator, slave);

Kakva će uloga biti PLC-a u pojedinoj primeni, zavisi od izbora modula i aplikativnog softvera.

PLC uređaji se izrađuju u različitim klasama koji se obično određuju na bazi broja ulaza i izlaza koji se za male uređaje kreće od nekoliko desetina, kod srednjih do hiljadu a kod najjačih do nekoliko hiljada ali uz veliku brzinu obrade.

Bez obzira na klasu PLC-a značajan problem je izrada kvalitetnog aplikativnog softvera. Da bi mogućnosti ovih računara došle do izražaja, projektant i izvođač moraju da poseduju znanja iz oblasti automatskog upravljanja i programiranja, ali i da poseduju dovoljno znanja o tehnološkom procesu u kome se računar primenjuje, da razumeju i do kraja razrade sve koncepte logičkog i direktnog digitalnog upravljanja, da predvide sve havarijske situacije koji se mogu detektovati, da ugrade funkcije dijagnostike izvršne i merne opreme, i konačno da uklope tako isprogramirani uređaj u celinu *SCADA* sistema.

Programiranje PLC uređaja se vrši kombinacijom lestvičastog dijagrama i funkcionalnih blokova.

Budući da PLC računari nemaju korisničke interfejsse koji bi omogućili preprogramiranje, uobičajeno se uz njih od strane proizvođača isporučuju programski paketi na raznim standardnim platformama, najčešće na popularnim PC platformama koji su neka vrsta razvojnog okruženja, u kome se unosi, testira i prevodi izvorni kod. Jednom formiran program se upisuje u baterijski podržanu memoriju PLC računara posredstvom serijske komunikacije, odakle se po potrebi može i pročitati, uz poštovanje odgovarajućih sigurnosnih procedura. Uobičajena mogućnost ovih razvojnih paketa je i praćenje izvršavanja programa na PLC računaru, što predstavlja neku vrstu alata za otkrivanje semantičkih grešaka.

6.6. Specijalizovane radne stanice

Samostalni regulatori i PLC računari shvaćeni su kao dve dijametralno suprotne koncepcije. Samostalni regulatori su računari sa fiksiranim regulacionim funkcijama, i nisu podložni nikakvom dodatnom programiranju, osim eventualno konfigurisanju, što im značajno smanjuje fleksibilnost. S druge strane PLC uređaji pružaju potpunu slobodu programiranja bilo kakvog upravljačkog koncepta, ali zato zahtevaju da programer odlično poznaje regulacionu problematiku; u suprotnom upravljački sistem može biti neupotrebljiv. Kao kompromis ove dve koncepcije pojavljuju se računarski sistemi koji na neki način objedinjuju dobre strane obe koncepcije. Ove specijalizovane radne stanice zapravo nude osnovne mogućnosti programiranja PLC-a, ali sa dodatnim hardverom (npr. frekventni regulator) i sistemskim softverom koji već podržava razne regulacione funkcije i kocepte u mnogobrojnim varijantama, pa je na programeru jednostavniji zadatak izbora pogodne koncepcije regulacije, umesto njegovog kompletnog programiranja.

Specijalizovane radne stanice (Field Point) predstavljaju modularne, distribuirane sisteme koji komuniciraju sa okruženjem preko svojih ulano/izlaznih modula i poseduju veoma snažnu softversku podršku. Rade samostalno ili u sprezi sa PLC računarom. Softversku podršku radne stanice podržava real - time operacioni sistem koji podržava rad real - time programa za prihvatanje i obradu i realizaciju upravljačkih zadataka (npr. Lab VIEW). Razvoj aplikacija se vrši na PLC platformi. U toku razvoja uspostavlja se snažna interakcija između specijalizovane radne stanice i PLC -a. Pri tom programu se odvijaju u radnoj stanici, a PC pruža uglavnom grafičku podršku. Nakon završetka razvoja prekida se komunikacija između PC-a i radne stanice

6.7. Centralni računarski sistem

Centralni računarski sistem *SCADA* postavlja mrežu računara sa klasičnim korisničkim interfejsom: grafiku visoke rezolucije, tastatura, miš, štampač itd. Uobičajeno je smešten izvan buke tehnološkog procesa, u klimatizovanom prostoru. Predstavlja vizuelno najupečatljiviji deo *SCADA* sistema, a informacije o procesu prezentuje u formi animiranih prikaza visoke rezolucije, lista, trendova, istorijskih i statističkih grafikona i sl. Osnovne upravljačke funkcije obuhvataju supervizijski i nivo koordinacije na skali hierarhijskih nivoa upravljačkih sistema (samo u retkim slučajevima i neke funkcije iz nivoa planiranja), dok se informacione funkcije svrstavaju u nekoliko kategorija: prikaz i arhiviranje merenih i izvedenih veličina, beleženje i arhiviranje događaja, beleženje i arhiviranje alarma. U složenim aplikacijama kombinuju se sa ekspertskim sistemima, u kojima se korišćenjem algoritama estimacije i testiranja hipoteza pokušava da izvrši samootkrivanje mogućih kvarova opreme i tako spreči veći zastoj ili havarije.

Popularnost PC računara nije mimošla ovu oblast, pa tako svi renomirani svetski proizvođači industrijskih računara imaju verzije *SCADA* softvera za PC računare, iako se svi slažu u tome da PC računar nije pogodan za ovu primenu ni po svom hardveru, ni pouzdanosti, niti po operativnim sistemima koji iza njega stoje, ali ga konkurentnim čini njegova cena i raspoloživost.

Što se *SCADA* softvera tiče, pod njim se podrazumeva najčešće softverski paket koji omogućava konfigurisanje i aktiviranje aplikacije. Pod konfigurisanjem se podrazumeva formiranje kompleksne baze podataka, sa podacima koji se prikupljaju posredstvom mreže od merne opreme, akvizicionih i upravljačkih računara, načinima njihovog arhiviranja i prezentovanja, alarmima, događajima, mogućim komandama operatera i strukturi računarske mreže. Kreiranje potrebne baze zahteva detaljno poznavanje tehnološkog procesa, koncepcije upravljanja koje su u njemu primenjene, strukture računarske mreže koja realizuje *SCADA* sistem i samog konfiguracionog softvera u okviru *SCADA* programskog paketa. Kreirana baza podataka u kombinaciji sa aktivacionim softverom iz *SCADA* paketa predstavljaju zapravo ono što se u užem smislu podrazumeva pod *SCADA* softverom.

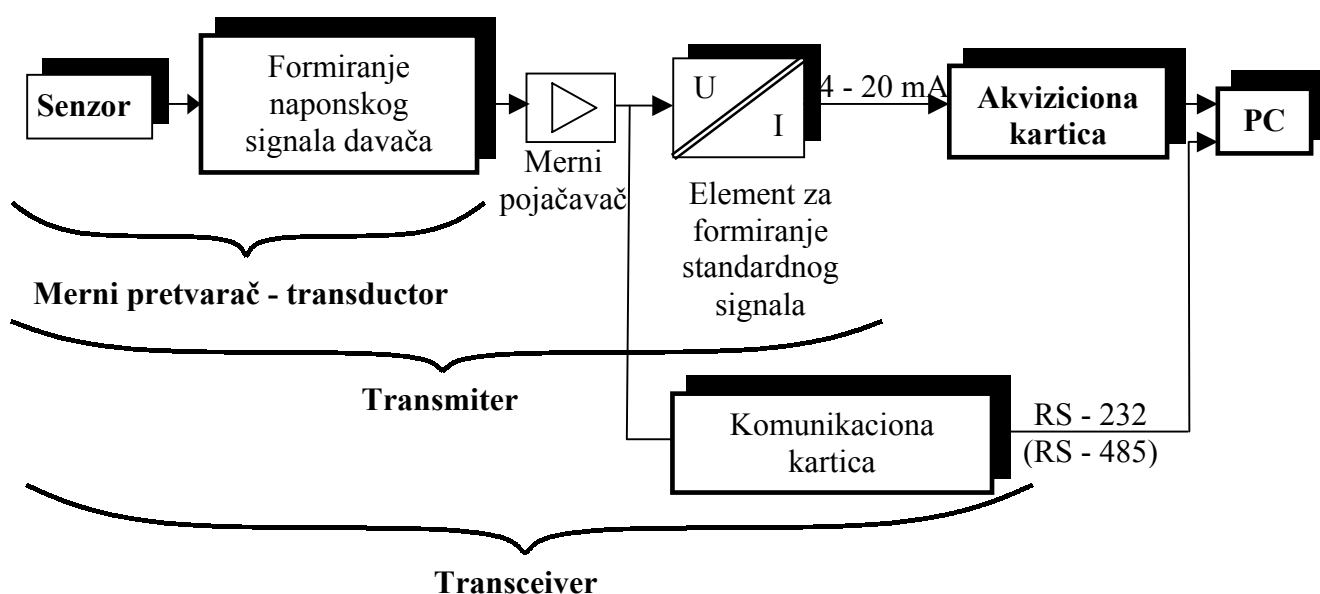
Svi proizvođači upravljačke računarske opreme imaju konvencijalno dostupne *SCADA* softvere za raznovrsne platforme. Nažalost, do danas ne postoji ozbiljan prostor u domenu standardizacije ovih proizvoda, a problemi koji iz toga proističu najčešće se odnose na kompatibilnost uređaja različitih proizvođača, prenosivost softvera i proširivanje sistema. Drugim rečima da bi se uređaji različitih proizvođača spojili u jedinstven *SCADA* sistem, najčešće je potrebno imati mnoštvo specijano pisanih softverskih komunikacionih drajvera, koji značajno podižu i ovako visoke cene ovih paketa. Konačno jednom formiran *SCADA* sistem nije lako unapređivati u delu centralnog računarskog sistema, jer to po pravilu znači značajne tehničke i finansijske napore, koji su ponekad uporedivi sa onima potrebnim za uvođenje potpuno novog centralnog računarskog dela *SCADA* sistema.

7. SCADA konfiguracije

Konfiguracija *SCADA* sistema je veoma različita i zavisi od prostornog rasporeda tehnološkog procesa, upravljačkog zadatka itd., i kreće se od jednoručarskog sistema do tzv. *WASCAD* sistema.

Najjednostavnija konfiguracija pod *SCADA* sistemima svodi se na sistem koji čine sa jedne strane prekidači, davači, releji itd. a sa druge strane PC računar koji preko svoje akviziciono-upravljačke kartice prima podatke, obrađuje ih, formira informacije o nadziranom procesu, i određuje potrebne upravljačke akcije, ako je to na ovom nivou primene predviđeno. Programaska podrška u ovom modelu *SCADA* sistema svodi se na podršku prihvata signala na PC magistrali.

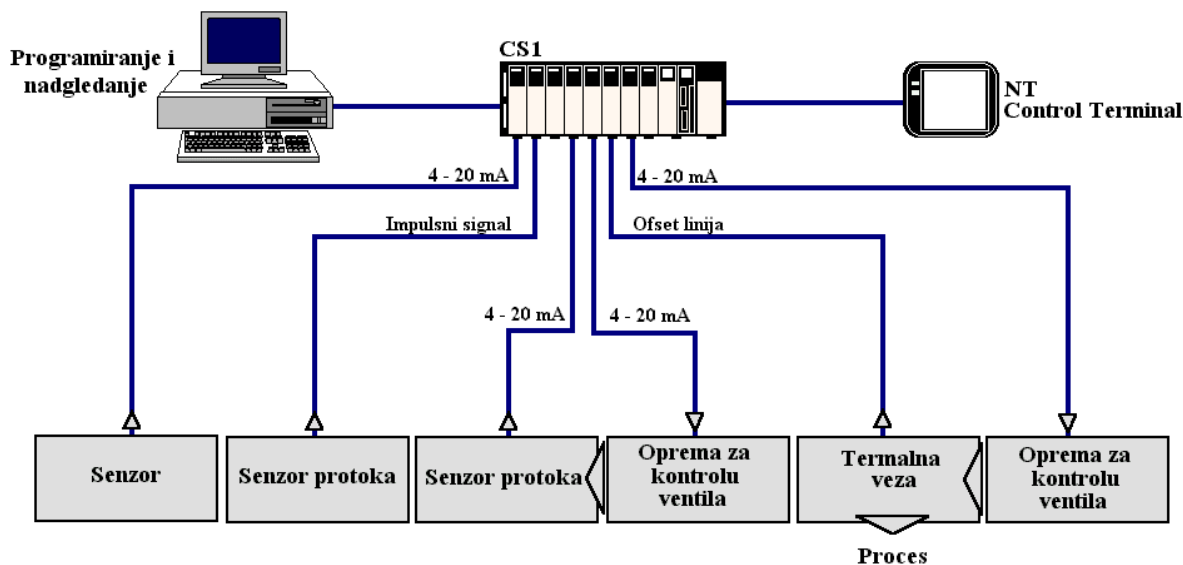
Akvizionarna kartica mora biti osposobljena za prihvatanje i izdavanje analognih i digitalnih veličina. Ulazne digitalne veličine se uglavnom svode na prihvatanje beznaponskih kontakata, a izlazne na relej ili tranzistor relativno male snage. Analogne ulazne veličine se primaju direktno sa mernog davača (transducer - senzor), ili iz sklopa koji zajednički čine merni davač i element za prilagođavanje (kondicioniranje) signala (transmitter) ili iz sklopa koji signal sa mernog davača priprema za direktan prihvatanje od strane PC računara preko neke od standardnih serijskih komunikacionih veza (tranceiver). Pod kondicioniranjem signala u ovom slučaju podrazumeva se filtriranje, pojačanje, linearizacija, baferovanje sample/hold, prigušenje itd. signala sa mernog davača. Na slici 6.1 prikazana je šema pripreme analognih signala za konfiguraciju *SCADA* sistema sa akvizicionom karticom.



Slika 7.1. - Blok šema prihvata analognog signala preko akvizicione kartice.

U jednostavne strukture spada i konfiguracija koja se formira od jednog PLC-a i PC računara. Na slici 7.2. prikazana je jedna ovakva struktura sa sensorima, aktuatorima i kontrolnim upravljačkim panelom. Pri ovakvoj konfiguraciji pretpostavlja se da PC ima sve uobičajene periferije kao što su: štampač, miš, bar kod čitači itd.

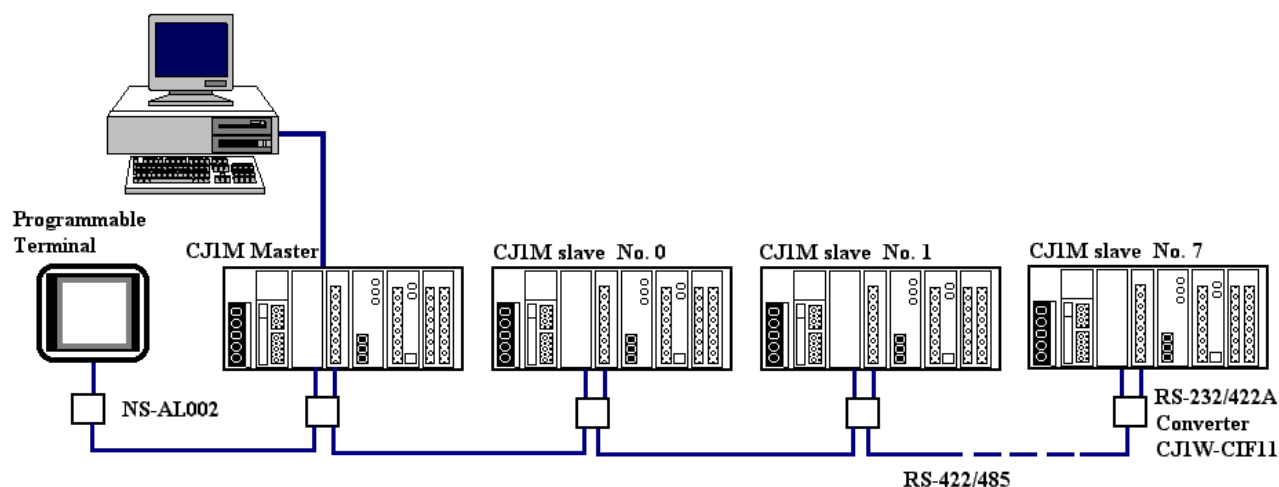
Kompleksniji primer *SCADA* sistema je komunikacionom vezom povezani skup upravljačkih (PLC) računara koji komuniciraju sa jednim PC računarom. PLC računari su pri tom raspoređeni u prostoru u okruženju tehnološkog procesa (fizička distribucija) i raspolažu sa onoliko softverskih potencijala sa koliko mogu obaviti svi upravljačko-akvizicioni zadaci lokalne sredine (logička distribucija). PLC računari se međusobno sprežu preko komunikacione, najčešće serijske veze. U ovakvoj vezi jedan od računara mora preuzeti ulogu glavnog (master), dok drugi moraju imati podređenu ulogu (slave).



Slika 7.2 - Konfiguracija za SCADA sistem od jednog PLC-a i PC-a.

Fizičkom i logičkom raspodelom i kvalitetnom komunikacijom ostvaruje se osnovno načelo distribuiranog upravljanja: u svaku lokalnu sredinu locirati onoliko hardvera i softvera koliko je tamo neophodno, a da se preko obezbeđene razmene informacija obezbedi horizontalna i vertikalna komunikacija između elemenata sistema.

Jedna relativno jednostavna konfiguracija sa više PLC računara je prikazana na slici 7.3. Akvizicija podataka se ostvaruje preko PC računara ili preko lokalnih upravljačkih panela.

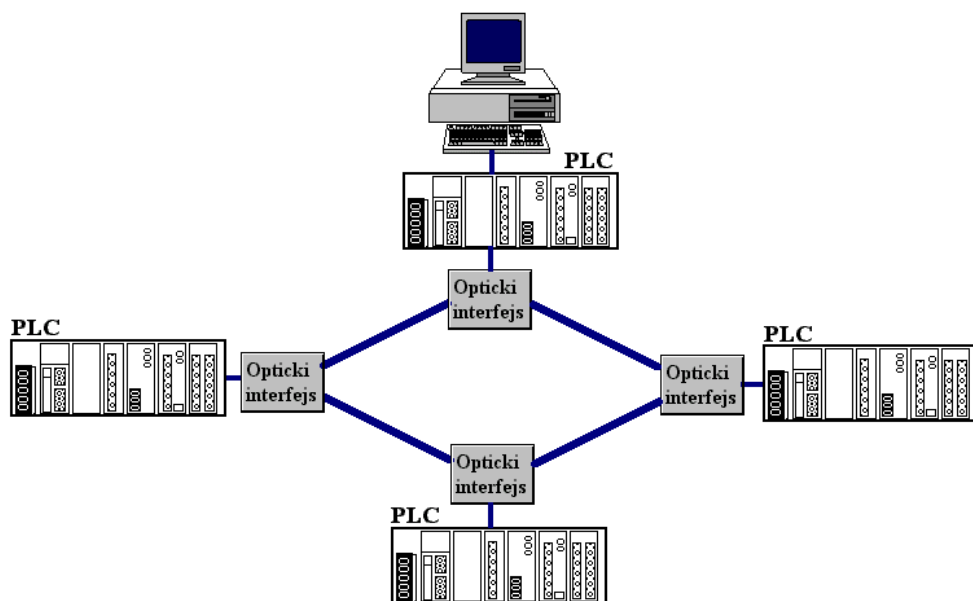


Slika 7.3.- Masteri / Slave odnos između više PLC računara preko serijske veze.

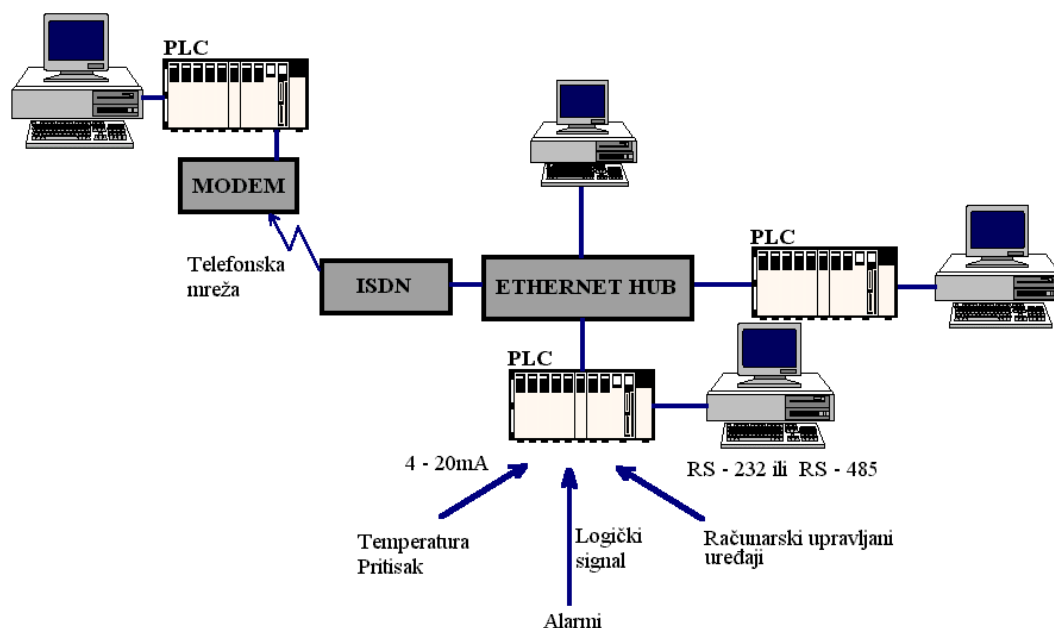
Na slici 7.4. prikazana je sprega više PLC računara vezanih u prstenastu strukturu sa dupliranim komunikacionim vezama. Između optičkih interfejsa postavljena je u ovoj strukturi optička i komunikaciona veza.

Veza između optičkih interfejsa i PLC-a je žičana. Udvostručivanjem veza postiže se visoki nivo pouzdanosti komunikacija.

Uključivanjem dostignuća savremenih komunikacionih rešenja tj. veza preko modema, žične i bezžilne telefonske mreže i sprega sa klasičnim računarskim mrežama omogućava proširenje SCADA konfiguracije i na široke geografske prostore. Na slici 7.5. prikazane su neke mogućnosti prenosa merenih podataka na velike udaljenosti.



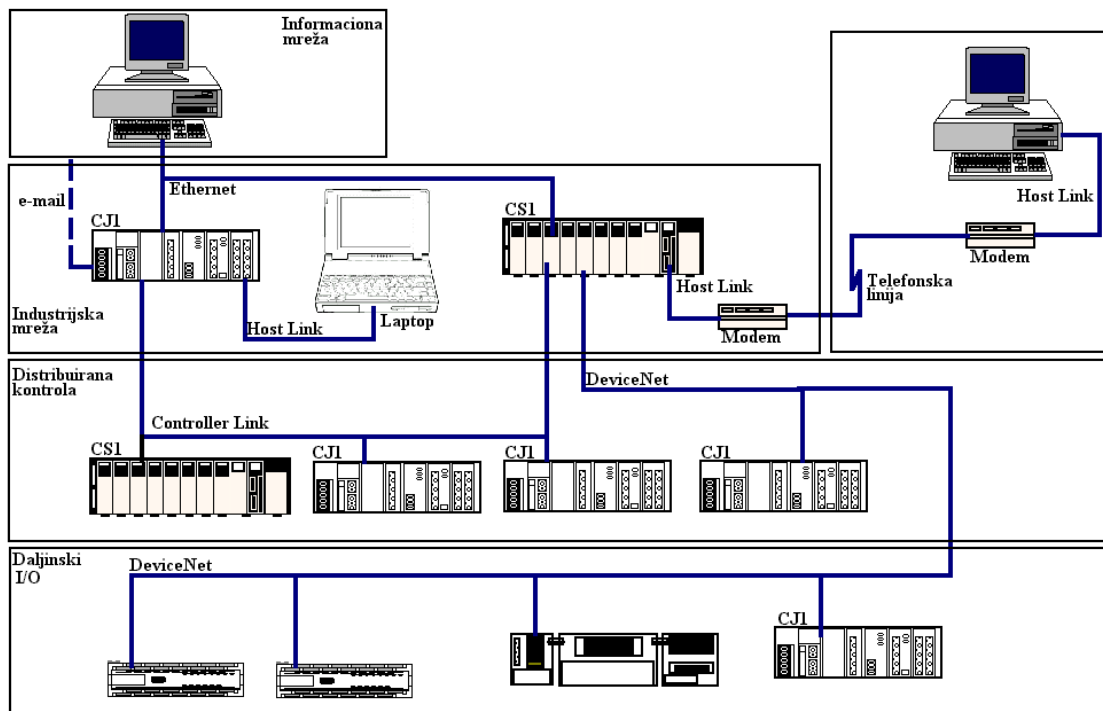
Slika 7.4. - Veza više PLC uređaja preko optičke komunikacije



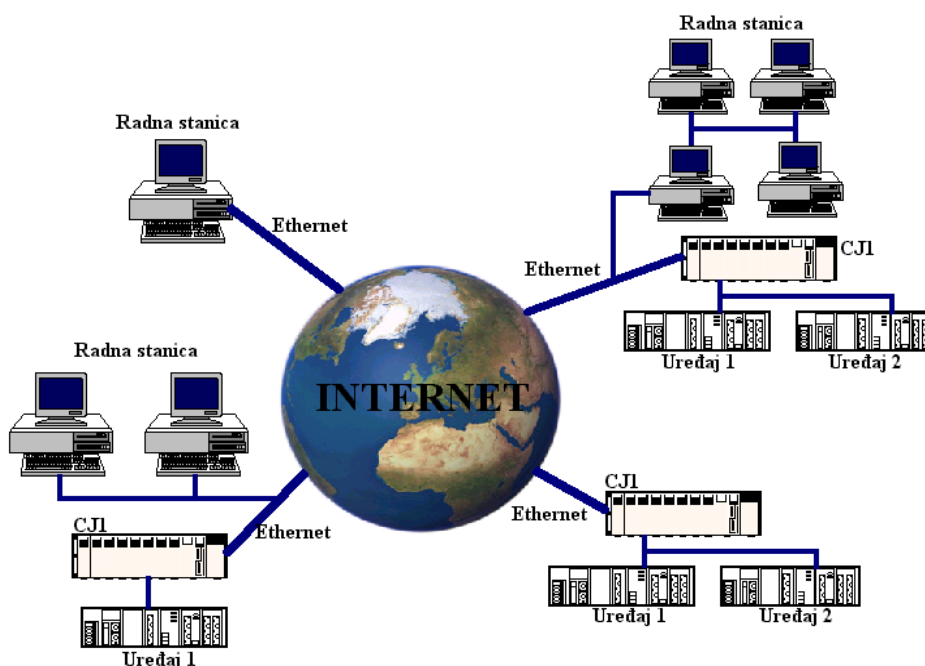
Slika 7.5. - Veza od mernog podatka do Ethernet mreže.

Komunikacije između elemenata *SCADA* sistema se mogu ostvariti preko različitih prenosnih medija i protokola. Na slici 7.6. prikazana je jedna konfiguracija prenosa informacija od ulazno/izlaznih jedinica direktno spregnutih na senzore i aktuatore u procesu preko različitih medija i protokolarnih rešenja u lokalnim računarskim mrežama.

Prenosom procesnih podataka do Ethernet mreže ostvaruje se mogućnost za formiranje najsloženije geografski distribuirane *SCADA* konfiguracije koja se veoma često zove WASCAD (Wide Area *SCADA*). Ulazkom u INTERNET mrežu WASCAD sistem postaje potpuno otvoren za sve klijente. Na slici 7.7. prikazan je koncept WASCAD-a.



Slika 7.6. - Složena upravljačko-informaciona konfiguracija SCADA sistema.



Slika 7.7. - WASCAD konfiguracija.

Što je mreža složenija to ona sve više i više dobija karakter informacione a ne upravljačke mreže. Naime zbog mogućih prekida komunikacija najčešće se izbegava formiranje povratnih veza na osnovu informacija dobijenih preko bežičnih komunikacionih veza.

8. Funkcije i karakteristike SCADA sistema

Računarski sistemi za upravljanje i akviziciju procesnih podataka su u pogonu u mnogim tehnološkim procesima već dosta dugo. Kroz praksu i iskustvo utvrđen je zajednički skup osnovnih funkcija SCADA sistema. Trend razvoja SCADA sistema je u standardizaciji tj. u stvaranju mogućnosti da se prilagodi potrebama i varijantama različitih tehnoloških procesa.

SCADA sistem je tako koncipiran hardver i softver koji obezbeđuje fleksibilan skup funkcija za nadzor, upravljanje i analizu procesa. Efikasnost rada SCADA sistema postiže se funkcionalnom integracijom sledećih podсистema:

Hardverski podсистem - obuhvata u najširem smislu celokupan hardver koji se instalira za potrebe nadzora i upravljanja proizvodnim procesom.

Softverski podсистem - obuhvata celokupan softver uključujući i pomoćne programe kojima se kontroliše rad celokupnog hardvera. Sastoji se od niza programskih paketa instaliranih na pojedinim elementima sistema kao što su: sami SCADA programi, programi za analizu rada sistema i ispitivanje kvaliteta, programi a lokalno upravljanje PLC-ima, komunikacioni programi, operativni sistem računara.

Komunikacioni podсистem - obuhvata softver i hardver za povezivanje elemenata nadzorno-upravljačkog sistema sa tehnološkim procesom i između hardverskih i softverskih podсистema.

Tehnološki podсистem - je proizvodni proces kojim se želi automatski upravljati i koji se želi nadzirati.

Osnovne pretpostavke koje SCADA sistem mora da ispuni su:

- pružanje brzih i tačnih informacija o svim relevantnim podacima o objektima nad kojima se vrši nadzor i upravljanje;
- visok nivo pouzdanosti rada sistema, što znači da primenjena rešenja sadrže zaštite i redundantna rešenja na raznim nivoima sistema;
- velik komfor u radu operatera, tako da je ceo sistem za nadzor i upravljanje veoma jednostavan za rad i korišćenje.

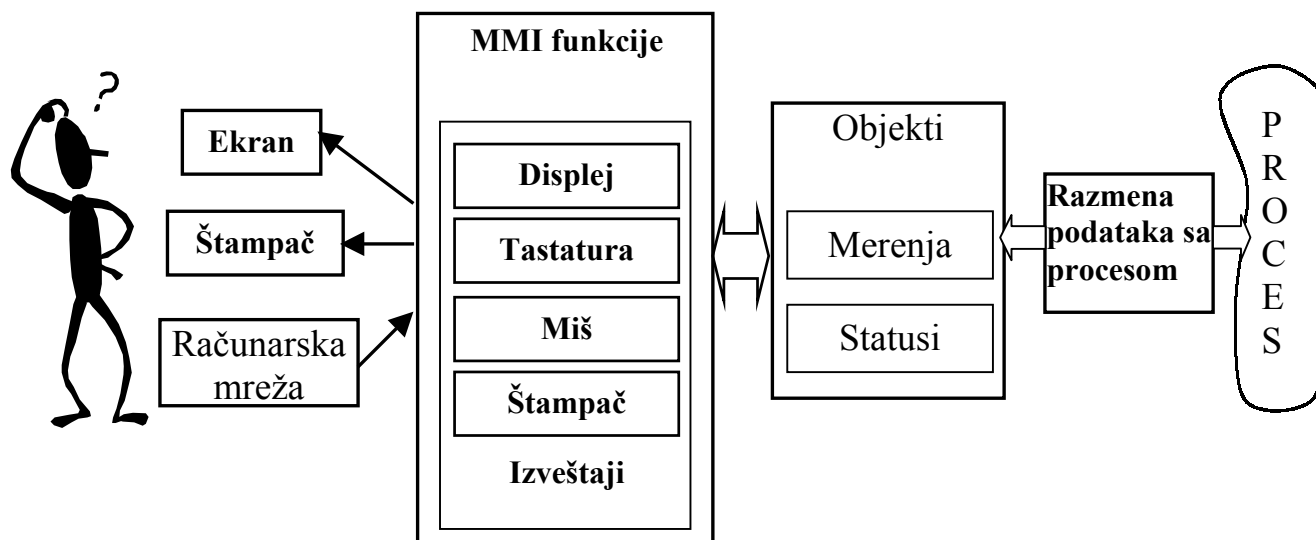
Razvojni koncepti SCADA sistema se zasnivaju na ispunjavanju sledećih zahteva:

- otvorenost arhitekture;
- selektivni pristup informacijama i
- komunikacija čovek - mašina.

Osnovna karakteristika otvorene arhitekture je modularnost sistema. Otvorena arhitektura dopušta vertikalna i horizontalna proširenja sistema. Vertikalno proširenje sistema znači da se u nadzorno - upravljački sistem može pridružiti različiti broj i različiti tipovi uređaja (PLC itd.), da se mogu dodavati novi ili ukidati postojeći. Horizontalno proširenje znači da se isti SCADA sistem može primeniti na različite tehnološke procese. Najveća prednost otvorene arhitekture je mogućnost dodavanja novih modula kao i komunikacionih protokola.

Selektivni pristup informacijama podrazumeva da više tipova korisnika koristi dati sistem, kao što su dispečeri, tehnolozi održavanja, menadžment i sl. Svaki tip korisnika ima različite tehnološke potrebe za podacima, nivo znanja i iskustvo. SCADA sistem mora biti tako fleksibilan da svakom tipu korisnika omogućava da na jednostavan i brz način dođe do podataka koji su mu od interesa. Korisnik ne želi da gubi vreme na traženje i prikaz informacija koje ga ne interesuju. Korisnik ne treba da bude okupiran računarom, već svojim poslom. SCADA mu služi kao alat kojim on operiše svojim tehnološkim procesom pomoću računara.

Komunikacija čovek mašina - MMI (povezuje upravljački sistem sa čovekom operaterom). Računari i softver u *SCADA* sistemu su samo alat, koji ne može da zameni prosuđivanje, inicijativu i kreativnost korisnika. Komunikacija čovek - mašina treba da obezbedi korisniku punu kontrolu nad sistemom, a ne sistema nad korisnikom. *SCADA* sistemi postižu ovaj cilj obezbeđujući korisniku mogućnost praćenja tehnološkog procesa preko interaktivnih prikaza na ekranim računara. Kretanje kroz program se obezbeđuje sistemom menija, funkcionalnim tasterima itd. *SCADA* korisniku treba da pruži kontrolu tehnološkog procesa, nadzor, hronologiju događaja, rukovanje alarmima i kvarovima, komunikaciju sa PLC - uređajima, off-line analize, izveštaje za razne vrste korisnika itd. Blok šema veze operatera i *SCADA* sistema prikazana je na slici 8.1.



Slika 8.1. - Blok šema veze operatera i *SCADA* sistema

Svoju funkciju *SCADA* sistem ostvaruje putem razmene podataka između različitih softverskih modula.

Prvi softverski modul obezbeđuje razmenu informacija između procesa i *SCADA* sistema. Sledeći softverski modul sadži programe za razvoj aplikacija za svaki pojedini objekat. Ovaj deo programa može biti manje ili više složen u zavisnosti od objekata i detaljnosti upravljačko - akvizicionih zahteva. Ovaj modul ažurira bazu podataka.

Razmena podataka između softverskih modula se rešava na više načina. Osnovni zahtev je da se ispune uslovi real-time obrade. Protok podataka se ostvaruje ili cikličkim ispitivanjem svih stanja ili se realizuje po principu event-driven tj. inicijalizacijom od strane odgovarajućeg događaja.

SCADA sistem konstantno prikuplja podatke iz procesa u realnom vremenu u određenom ciklusu skeniranja programabilno podesivom, smešta ih u bazu podataka i procesira ih saglasno zahtevima korisnika definisano u projektnoj dokumentaciji sistema. Autorizovani operateri mogu menjati određene parametre procesa upravljanja, i obavljati određene proračune, pregledati i štampati podatke, uključivati i isključivati real - time informacije iz PLC - ova, interaktivno menjati razne opcije, granice i sl.

8.1. Osnovne karakteristike SCADA sistema

Upotrebljivost nekog *SCADA* sistema, i upoređivanje efikasnosti različitih *SCADA* sistema može se izvršiti na osnovu analize sledećih karakteristika efikasnosti:

Funkcionalnost *SCADA* sistema određuje se izborom opcija pri projektovanju. Viši nivo funkcionalnosti se postiže projektovanjem po modulima uz primenu struktuiranog i objektnog programiranja uz primenu relacionih baza podataka.

Pouzdanost *SCADA* sistema ogleda se u pouzdanosti opreme, pouzdanosti podataka i pouzdanosti napajanja električnom energijom.

Ukoliko se npr. primenjuje više računara mrežno spregnutih, tada se otkaz jedne ili više radnih stanica može nadoknaditi radom na nekoj drugoj radnoj stanici u sistemu. Oprema koja se koristi mora biti kvalitetna. Podaci se mogu gubiti zbog kvara disk jedinica. Ako se striktno poštuju procedure za spašavanje podataka, tada se veliki broj podataka može restaurirati u vrlo kratkom vremenu nakon nastanka kvara. Pouzdanost podataka se povećava i primenom backup fajlova. Od brisanja ili nedozvoljene promene podaci se mogu štititi sistemom logičke kontrole. Korisnik u ovom slučaju se upozorava na nelogičnosti ili mu se određene akcija zabranjuje, u zavisnosti od tehnologije rada.

Sistem besprekidnog napajanja obezbeđuje sistem od nepoželjnih efekata iznenadnog nestanka napona.

Upotrebljivost se ogleda u lakoći rukovanja i mogućnosti prilagođavanja zahtevima upravljanog procesa. Bitna olakšanja u primeni daju različite pomoći "help", više opcija za istu operaciju, preglednost prikaza tehnološkog procesa, animacije događaja itd.

Porukama upozorenja i selektivnim pristupom pojedinim tipovima korisnika zabranjuju se određene opcije primene na nekom nivou. Promene i akcije se moraju unositi u poseban fajl hronologije događaja.

Efikasnost *SCADA* sistema se ocenjuje na bazi brzine izvršavanja ciklusa skeniranja i brzine komunikacije između elemenata sistema. Evidentno je da je ovako definisana efikasnost funkcija kvaliteta primenjenih računara i PLC uređaja.

Fleksibilnost *SCADA* sistema se ogleda u mogućnosti da korisnik bez posebnih poteškoća može uraditi sve potrebne dopune.

Otvorenost za dalja softverska i hardverska proširenja i povezivanje sa drugim sistemima je takođe bitna karakteristika.

Održavanje *SCADA* sistema je:

- redovno i periodično;
- održavanje u smislu eventualnih dorada i promena usled izmena u tehnološkom procesu.

Nakon instaliranja sistema treba odmah napraviti takvu organizaciju održavanja koje povećava pouzdanost sistema i otkaze i kvarove svodi na minimum.

Podrška proizvođača se ogleda u obezbeđenosti obuke kadrova, dokumentaciji, mogućnosti konsultantskih usluga itd.

Dobra dokumentovanost se ogleda u kvalitetu priručnika, pomoći "help"-u, opisu baze podataka itd.

9. Baze podataka

Baze podataka predstavljaju osnovu na kojoj počiva ceo informacioni sistem *SCADA*-e. U bazi podataka čuvaju se: podaci iz procesa, parametri sistema, ručno uneti podaci, razni izvedeni podaci, arhive i sl. Efikasnost sistema zavisi obezbeđenosti brzog i efikasnog pristupa bazama podataka.

Pri početku razvoja neke primene mora se brižljivo izvršiti analiza skupa podataka koji se nadzire, tehnološki zahtevi, ostali zahtevi korisnika, kako bi se definisala efikasna baza podataka.

Baza podataka SCADA sistema uglavnom sadži tri osnovna dela:

- procesne baze - baze realnog vremena;
- teleinformacione baze;
- relacione baze podataka.

U procesnu bazu stižu sirovi podaci iz procesa i tu počinje njihova obrada. Procesna baza ima više delova:

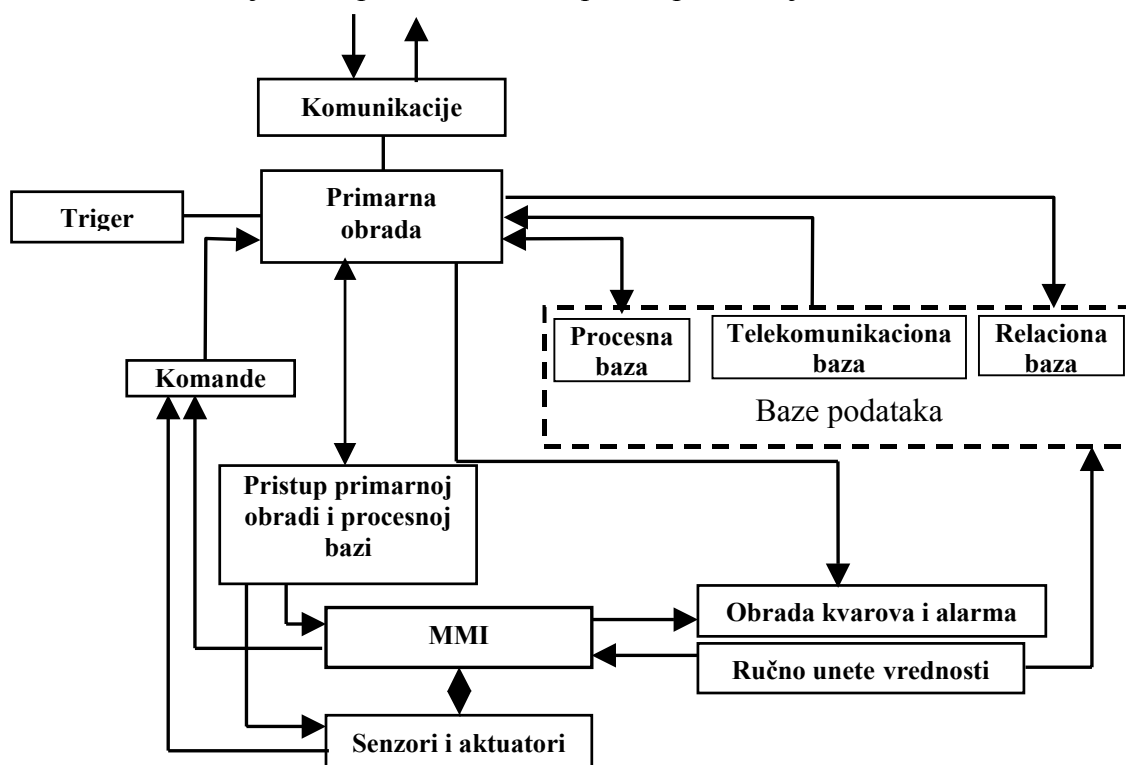
- podaci iz procesa i deo rezervisan za podatke koji stižu iz procesa;
- parametri i deo procesne baze koji je rezervisan za podatke koji stižu iz procesa;
- ručno uneti podaci: postoje dve funkcije. Jedna je unos podataka koji se ne zahvataju iz procesa već se direktno unose kako bi se postigla puna observabilnost tehnološkog procesa, dok je druga funkcija isključivanje/uključivanje određenih podataka iz procesa;
- izvedeni podaci: su deo procesne baze rezervisan za podatke koji predstavljaju izvedena merenja po nekom algoritmu ili definisanoj zakonitosti.

Sa procesnom bazom tesno je vezan program za primarnu obradu. Podaci stižu u računar posredstvom komunikacije. Sirovi podaci se smeštaju u procesnu bazu podataka, obrađuju komunikaciju i počinje njihova primarna obrada. Primarna obrada sirove podatka pretvara u inženjerske jedinice, promeravajući pri tom granice alarma, granice ručno unetih vrednosti, vrši proračun izvedenih merenja i ako je to potrebno aktivira alarmni sistem.

Teleinformaciona baza sadži podatke i parametre potrebne za izvršenje komunikacije. Baza se sastoji od osnovnih podataka u vezi PLC-ova, vrsti komunikacije, komunikacionim protokolima itd.

Procesna baza je rezidentna u memoriji računara. Za razliku od nje, relaciona baza se nalazi na disku računara i tu beleži svaki ciklus merenja kao jedan zapis. Nad ovom bazom se rade sve off-line obrade, pretraživanja, izveštaji, trendovi i sl. Ovoj bazi se pridodaju podaci o hronologiji događaja, hronologiji alarma i kvarova, posebni podaci koje korisnik želi da analizira itd.

Veza komunikacija, baza podataka, MMI i proces prikazan je na slici 9.1.



Slika 9.1. - Veza između delova SCADA sistema

9.1. Obrada parametara tehnoloških procesa

Stanje neke promenljive u tehnoloskom procesu karakterišu ili digitalne ili kontinualne merne veličine. *SCADA* sistemi ove informacije dobijaju od PLC uređaja preko komunikacionih linija. Vizuelizacija u *SCADA* sistemima ove podatke treba da prikaže korisniku u što je moguće verodostojnijoj formi.

9.2. Dojave

O velikom broju događaja koje se odvijaju u tehnološkim procesima informacije dobijamo putem dojava. Dojave imaju samo dva stanja npr. oznaka 1 se pridružuje prisustvu plamena u gorioniku, a 0 da plamena nema. Dojave mogu biti pojedinačne ili grupne.

Za vrednovanje pojedinačne dojava nije potrebno saznanje o još nekoj dojavi. Npr. prisustvo plamena se samo po sebi može ustanoviti iz dojava. Verodostojnost dojava *SCADA* sistem ne može ustanoviti. Ako je neka pojava vitalna za rad tehnološkog procesa tada se dojava udvostručuje. Pri tom nije celishodno da obe dojave označavaju istu informaciju. Npr. celishodna je da su dve dojave o istoj promeni komplementarne tj. ako jednu pojavu prva dojava označava sa 1 tada druga dojava ovu pojavu treba da označava sa 0-om. Verodostojnost u ovom slučaju potvrđuje različitost dojava.

Kod nekih tehnoloških procesa informaciju o stanju mogu dati samo dve dojave (par dojava). Npr kod rezervoara stanje se može oceniti iz dve dojave na osnovu sledeće tablice istine:

Nivo ispod minimuma	Nivo iznad minimuma	Nivo
0	0	Nivo između minimuma i maksimuma
0	1	Nivo iznad maksimuma
1	0	Nivo ispod maksimuma
1	1	Stanje neverodostojno

Iz primera se vidi da se verodostojnost informacije može oceniti i na osnovu ove dve dojave. Pri tom treba naglasiti da se ne može iz dojava utvrditi nastanak greške u senzoru.

Retko ali se ipak može sresti situacija pri kojoj više dojava označuju isto stanje. Npr. kod izlaza iz neke transportne trake materija se može usmeriti u tri pravca. Pri tom samo se jedno stanje može podesiti. U ovom slučaju samo jedna dojava može imati jediničnu vrednost. Ako se pojave dve jedinice tada sigurno imamo grešku u sistemu senzora.

Merenjem se dobijaju informacije o temperaturi, pritisku, protoku itd. Klasični senzori daju signal od 4 do 20mA. Na ulazu A/D konvertora strujni signal se preko otpornika prevodi u naponski, npr. na otporniku od 100 oma pri struji od 20mA imaćemo napon od 2 V. Ovaj napon se konvertuje u (10 - 16)-to bitni binarni broj koji je srazmeran ulaznom naponu. U *SCADA* sistemu se ne prikazuje ovaj binarni broj, već se on mora pretvoriti u inženjersku jedinicu. Ovaj postupak se često sreće pod nazivom skaliranje. Skaliranje u nekim slučajevima može biti i nelinearno.

Ako je izlaz iz senzora manje od 4mA ili veći od 20mA to sigurno znači da je u sistemu sa senzorom nastala neka greška.

Ako je veza između senzora preko interne magistrale (npr. PROFIBUS) tada se verodostojnost primljenog podatka takođe potvrđuje kontrolom da li je veći ili manji od nekih vrednosti.

U realnim tehnološkim procesima promene ne nastaju naglo. Ako se između dva uzorkovanja uoči nagla promena to obično upućuje na prisustvo greške.

Ako se u prenosnom sistemu izgubi neki podatak tada se pristigla vrednost mora proceniti. U velikom broju slučajeva usvaja se prethodna vrednost za narednu.

Tehnološke veličine se menjaju između dve unapred određene vrednosti. Ako merena vrednost ukazuje na prekoračenje ovog opsega tada govorimo o alarmnom stanju. Često se razmatra i brzina promene tj. vrši se analiza trenda. O nastanku alarmnog stanja mora se izvestiti korisnik bilo zvučnim bilo vizuelnim znakom. Neka alarmna stanja prate i automatska isključenja.

9.3. Zadaci obrade procesnih signala

Dojave i rezultate merenja se obrađuju na više načina od kojih se najčešće sreću:

a) Poruke

Pri nastanku neke dojave korisnik se obaveštava nekom tekstualnom porukom. Poruka opisuje razlog dostavljanja poruke. Poželjno je na ekranu imati posebno područje za ispis poruke. Poruke ne moraju uvek obaveštavati samo o nastanku alarmnog stanja. Poruke se ispisuju i pri nastanku i pri prestanku nekog stanja.

Pri projektovanju sistema treba izvršiti analizu koja stanja treba smatrati alarmnim. Ako se pojavi veliki broj alarmnih poruka tada će operater teško odrediti da li je stanje stvarno opasno.

Alarmna stanja se moraju arhivirati. Arhivira se vreme nastanka i prestanka alarmnog stanja. Arhiva se čuva nekoliko dana, a po potrebi i duže. SCADA sistem mora obezbediti i mogućnost pretraživanja i izlistavanja alarmnih stanja.

b) Izračunavanje stanja

Veoma često se neke veličine stanja ne mere već određuju putem nekog algoritma na osnovu izmerenih stanja.

c) Arhiviranje

SCADA sistemi obavezno moraju obezbediti arhiviranje stanja. Arhiviraju se dojave i rezultati merenja. Na osnovu arhiviranih podataka vrši se analiza odvijanja tehnološkog procesa. Dužine čuvanja podataka i učestalost uzimanja podataka za arhivu zavisi od zahteva korisnika. Pri arhiviranju često se koriste neke tehnike kompresije podataka.

Post - mortem arhiva se koristi za utvrđivanje razloga nekog zastoja ili havarije. Arhiva sadrži sve dojave i merene podatke. Dužina arhive je određena. Stari podaci se pokrivanju novim.

Upis u arhivu se vezuje na neki uslov. Pri normalnom radu uvek se zamenjuju stari podaci sa novim u arhivi. Kad nastane neko nepoželjno stanje tada se zadržava stanje u ovoj arhivi.

Pravilnim korišćenjem ove arhive u off-line postupku može se izvršiti simuliranje stanja pre havarije.

d) Određivanje srednjih časovnih, dnevnih itd. vrednosti.

Veoma često se pri normalnom radu vrši usrednjavanje vrednosti tj. određivanje časovnih, dnevnih itd. srednjih vrednosti. Veoma često se u odnosu na srednju vrednost izračunava i minimalno i maksimalno odstupanje.

Veoma često se beleže i vrednosti po smenama.

e) Određivanje dužine trajanja pogonskog stanja.

Na osnovu merenih vrednosti mogu se odrediti i trajanja pogonskog stanja neke pumpe npr. koja se naizmenično uključuje ili isključuje.

9.4. Određivanje nivoa pristupa SCADA sistemu

Velikim sistemom u principu nikad ne upravlja samo jedan operater. U slučaju istovremenog rada više operatera postavlja se problem nivoa kompetentnosti.

Pri tome ne mora svaki operater imati pristup svakom parametru ili imati pravo da menja neki od parametara. Svaka promena parametara se pri tome mora arhivirati sa naznakom ko je izveo promenu parametra.

Poruke ne moraju da se pojave na svakom ekranu. Raspodelom poruka na određene ekrane izbegava se nagomilavanje informacija.

10. Usluge SCADA sistema

Ako se pri projektovanju nekog sistema za upravljanje i vizuelizaciju poklope zahtevi i mogućnosti SCADA sistema tada se problem projektovanja relativno lako može rešiti. Ako ne postoje sve funkcije u unapred predviđenim objektima SCADA softvera tada se dopune moraju napisati na jednom od programskih jezika (npr. Visual BASIC).

Primena sistema za vizuelizaciju prolazi kroz dve faze:

- Razvoj i,
- Primena

Razvojna opcija SCADA sistema obezbeđuje grafički dizajn, animaciju i spregu identifikatora i objekata tehnološkog procesa.

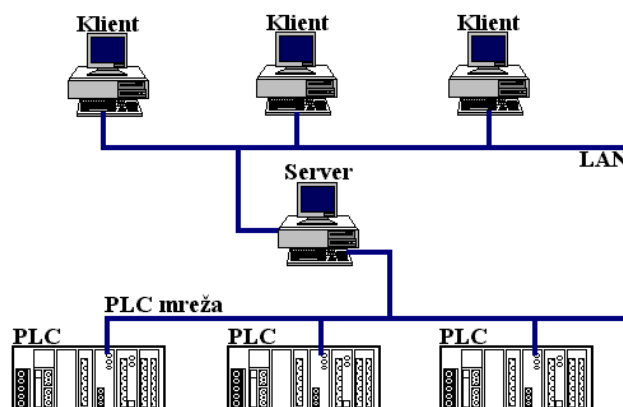
10.1. Komunikacija između sistema za vizuelizaciju i tehnološkog procesa

Osnovni zadatak sistema za vizuelizaciju je da sa određenom učestanošću učita stanja upravljačkog tehnološkog procesa. Ovi podaci se pripremaju u PLC uređajima. PLC uređaji do danas nemaju zajedničku komunikacionu platformu. SCADA-e zbog toga moraju "prepoznati" velik broj različitih PLC uređaja. PLC uređaji i SCADA se moraju tako odabrati da imaju mogućnost komunikacije.

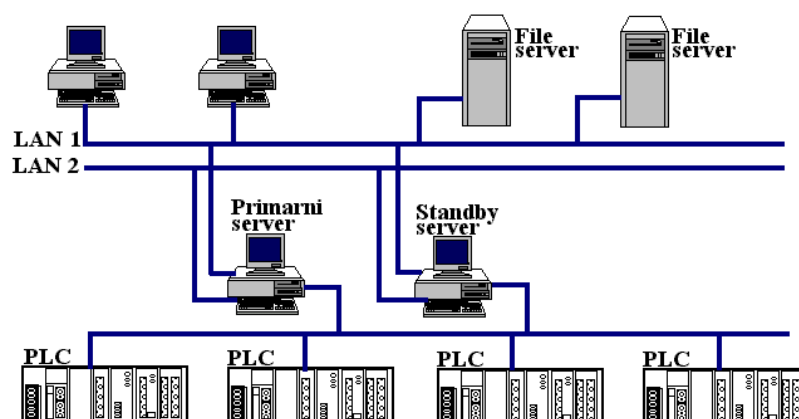
PLC uređaji uglavnom komuniciraju preko RS232C ili RS485 linija sa brzinom od 19200 bauda. To znači da u strukturi samo jedan računar (PC) može imati ulogu master-a, dok PLC uređaji moraju imati ulogu slave-a. Kod složenijih aplikacija moguće je primeniti i multi-master princip.

10.2. Sprege server-klijent

Nadzor nad jednim tehnološkim procesom često obavlja istovremeno više operatera. Zbog toga SCADA mora da deluje istovremeno u tzv. server-klijent sistemu.



Slika 10.1. - Sistem sever - klijent.



Slika 10.2. - Redundantni sistem sa više servera.

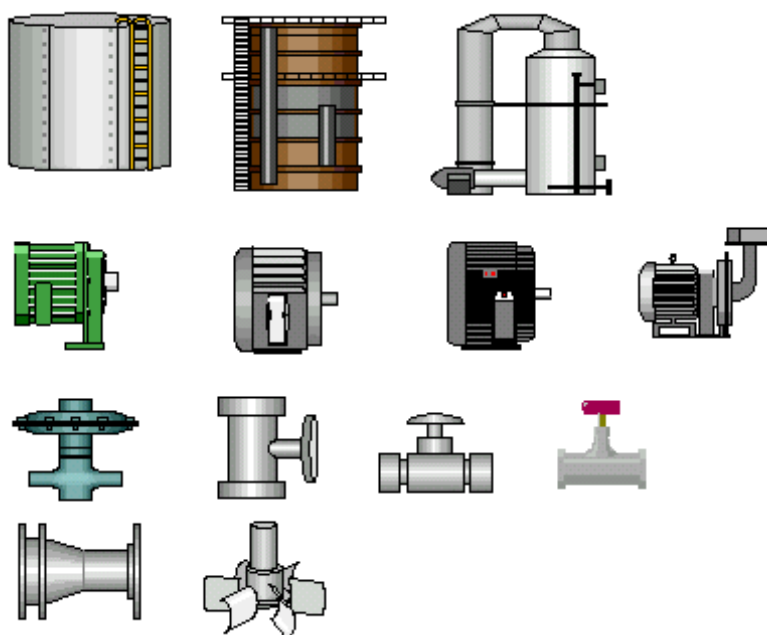
Server je u stalnoj vezi sa PLC uređajima i poseduje bazu podataka i vrši primarnu obradu. Klijenti preko brze komunikacione linije primaju informacije. Radne stanice servera i klijenta su u principu iste.

SCADA sistemi podržavaju rad servera samo sa određenim brojem objekata. Razlog za ova ograničenja leži u usaglašavanju oprečnih zahteva velike baze podataka i malog vremena uzorkovanja. Ako je tehnološki proces složen tada se mora primeniti sistem raspodele zadatka.

Savremene tendencije teže da izbegnu instaliranje programa za vizuelizaciju na klijente, već preferiraju spregu preko interneta (intraneta).

11. Formiranje tehnoloških šema

Najveća količina informacija pregledno se može prikazati na ekranu primenom grafičkog rešenja. Tehnološki procesi se zbog toga prikazuju na ekranima u vidu tehnoloških šema.



Slika 11.1. - Detalj iz biblioteke šemotehničkih skica.

Celokupna tehnologija se veoma retko može prikazati samo na jednom ekranu. Zbog toga se ukupna tehnološka šema razlaže na segmente. Jedna globalna slika uvek sadrži osnovne detalje svih segmenata. Ako se želi imati uvid u neki detalj tada se poziva neki segment. Segmenti se mogu formirati i u više nivoa.

Tehnološka šema se počinje stvarati sa jednim stanjem procesa. Razvoj se vrši primenom nekog editora. Crtanje bitno olakšava činjenicu da *SCADA* programi poseduju biblioteku šemotehničkih skica tipičnih elemenata tehnoloških šema (rezervoar, motor, ventil itd.).

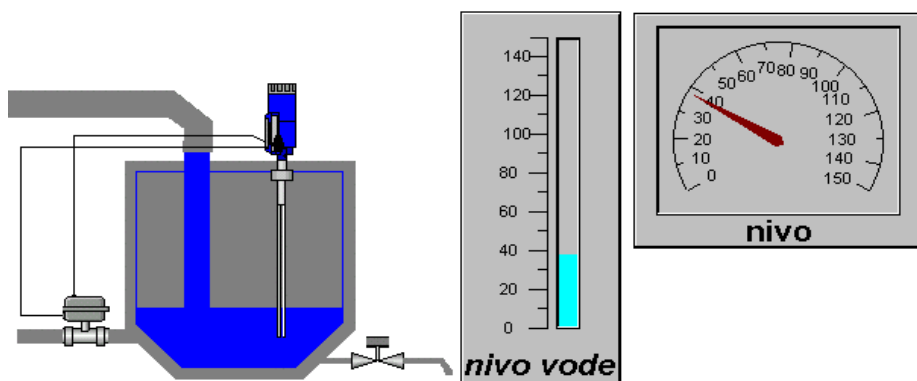
Da bi se dobilo u verodostojnosti i ubrzala izrada tehnoloških šema često se koriste i digitalne fotografije tehnološke opreme.

Nepotrebna slika daje malo informacija o stanju procesa. Animacijom slika se oživljava na osnovu aktuelnih informacija iz procesa.

U animaciji značajnu ulogu ima i boja. Isti element tehnološke šeme se boji u zavisnosti od stanja drugom bojom. Npr. pumpa se oboji zelenom bojom pri normalnom radu, a crvenom ako ne radi ili je u kvaru. Ako se želi razdvojiti stanje isključenosti od kvara, tada se koriste tri boje. Neka stanja se mogu veoma uspešno ilustrovati trepćućim slikama.

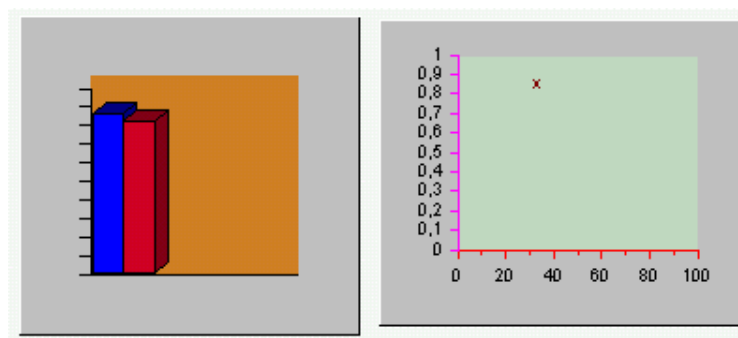
Ako se u sliku unose promenljivi tekstovi tada kažemo da primenjujemo tekstualnu animaciju. Svakom objektu u šemi možemo pridružiti neke tekstove drugih sadržaja ili boja.

Izmerena vrednost neke procesne veličine može se prikazati korišćenjem virtuelnog instrumenta, ili menjanjem odnosa obojenosti različitim bojama, npr. u rezervoaru visina vodenog stuba oboji se plavom bojom, a vazduh iznad belom, ili se vidi zid rezervoara.



Slika 11.2. - Neke od mogućnosti formiranja virtuelnih instrumenata.

Biblioteka *SCADA*-e sadrži u principu i bogati izbor različitih formi prekidača, potencijometara, tastera itd. Ovim elementima prikazujemo stanja diskretnih ili podesivih tehnoloških veličina.



Slika 11.3. - Dijagrami trenda promene nekih procesnih veličina.

Pozicija jednog dela tehnološke šeme se može obrtati ili translatorno pomerati. Ova mogućnost se veoma uspešno može koristiti u animaciji stanja.

Ako se u tehnološkom procesu pojavi neka promena tada se na ekranu može generisati pripadajuća promena slike npr. vrata se nacrtaju zatvorena, a kada krajnji prekidač dostavlja informaciju o promeni stanja tada se vrata nacrtaju u otvorenom stanju.

Promene neke merene fizičke veličine crtaju se na dijagramima trenda. Za dijagrame odvaja se poseban deo na ekranu koji se poziva po potrebi. Crtanje dijagrama se vrši na osnovu informacija iz baze podataka.

Ako se u biblioteci progama ne nađe dovoljno elemenata, tada se slika, tj. animacija, mora programirati primenom seta naredbi koje poseduje *SCADA* program. Naredbe u principu imaju formu koja je poznata iz već postojećih viših programskih jezika (C, C++, Pascal, itd.).

12. Šta je CX-Supervisor?

CX-Supervisor je *SCADA* (Supervisory Control and Data Acquisition) paket koji kreira grafički korisnički interfejs i radi u sklopu sa CX-serverom na kontroli i nadgledanju opreme u pogonima. Omogućava jasan, precizan i nedvosmislen prikaz informacija korišćenjem grafičkih objekata, animacija, grafika i dijagrama. Lakoća korišćenja i intuitivni dizajn dozvoljava *SCADA* dizajneru da brzo konfiguriše, testira i ispravlja projekat.

CX-Supervisor se sastoji od dva odvojena Windows programa: razvojno okruženje i izvršni modul. *SCADA* aplikacije se kreiraju i testiraju u razvojnom okruženju a isporučuju se kao finalni proizvod pomoću izvršnog modula.

Izvršni modul služi samo za izvršavanje prethodno kreiranih aplikacija i pomoću njega se ne mogu kreirati nove aplikacije.

Sadržaj glavnog prozora

Glavni prozor CX-Supervisora sadrži deset oblasti: Titlebar, Menubar, Toolbar, Control Bar, Text Bar, Graphic Object Bar, Alignment Bar, project Area, Status Bar i palette.

Titlebar

Na glavnom prozoru pokazuje "CX-Supervisor" i sadrži standardne Windows opcije.

Menubar (Glavni meni)

Sadrži CX-Supervisor menije koji su organizovani prema funkciji. Tu je takođe dodatak osetljiv miš meni kojim se prilazi pomoću desnog tastera miša na radni prostor.

Toolbar

Toolbar je obično ispod glavnog menija i sadrži dugmade koje predstavljaju short cut-ove za učestaliye korišćenje meni opcija. Kada je pokazivač miša pozicioniran na dugme njegovo ime ili namena se ispisuje.

Control Bar

Ova linija sadrži glavne opcije korišćenja i ime objekta.

Text Bar

Sadrži opcije obrade teksta.

Graphic Object Bar

Sadrži grafičke objekte koji se mogu dodati na stranu.

Alignment Bar

Linija za svrstavanje sadrži opcije za svrstavanje označenih objekata zajedno ili u rešetku.

Project Area

Ovo je prostor između Toolbar-a i Status Bar-a gde se vrše sve aktivnosti. Prozori koji se otvaraju u prostoru projekta uključuju: stranice, radni prostor, paletu, itd.

Paleta

Vidljiva je kada je stranica aktivna. Ona sadrži boje stilove linija i šare. Klikom na različite delove palete dozvoljava selekciju veličine linije, oblike, boje itd. Prikazivanje palete može biti omogućeno selektovanjem Palette iz menija View.

Status Bar

Status Bar je postavljen na dnu glavnog prozora i prikazuje poruke vodeći računa o tekućem statusu aktivnosti u CX-Supervisoru, na primer, selekcijom jednog objekta na stranici prouzrokuje prikazivanje njegovih koordinata.

12.1. Opšti pregled projekta

CX-Supervisor aplikacija sastavljena je od jedne stranice ili više stranica linkovanih zajedno. Stranice mogu sadržati pasivne ili aktivne crteže, tekst ili animacije i može biti grupisan zajedno u logički formirani projekat. Projekti se grade i testiraju u razvojnom okruženju i pokreću kao samostalne aplikacije u CX-Supervisor izvršnom okruženju.

Samo jedan projekat može biti otvoren za uređivanje u CX-Supervisoru. Pokušaj otvaranja drugog projekta će prouzrokovati zatvaranje tekućeg i eventualno snimanje promena i otvaranje drugog projekta.

Veze između informacija u projektu su prikazane u Project Information dijalog prozoru.

Opis stranica

Sistem i rukovanje na stranicama i objektima unutar projekata daje osnovu CX-Supervisor aplikaciji. Svaki projekat sadrži jednu ili više stranica. Stranice u projektu omogućuju vizualni prikaz u saglasnosti sa prikazom objekata koji su sadržani na svakoj strani obezbeđujući grafičko predstavljanje sistema koji se nadgleda.

Menjanje osobine stranica

Stranice imaju određene atribute ili svojstva koji definišu izgled na ekranu u izvršnom okruženju.

Menjanje ovih osobina:

- Iz File menija odaberemo Page Properties
- Upisi naslov stranice u polje Page Title
- Dodaj opis ako je potreban u polje Page
- Pomoću FullSize ili Center odaberi način na koji želiš da stranica bude prikazana
- Upiši koordinate gornjeg levog ugla stranice u polja Top i Left. Ove granice moraju biti u pixelima i moraju biti pozitivni celi brojevi između 0 i 2000
- Upiši dimenzije visine i širine stranice u polja Height i Width. Ako prethodno odaberete Full Size opciju tada su ova polja automatski popunjena.
- Odaberi vrstu ivice u polju Border

- Odaberi način prikazivanja stranice u polju Display Mode-u
- Podesi i manje važne osobine.

Objekti

Stranice u Scadi se sastoje od objekata. Primeri grafičkih objekata uključuju elipse, poligone i tekst. Kontrolni objekti mogu menjati ili prikazivati stanje tačaka ili izraza. Najčešće se njihove osobine mogu podesiti dvoklikom na levi taster miša.

Pritiskom na desni taster miša dobijamo meni sa najčešće korišćenim prečicama.

Animacije

Animacije se aktiviraju pomoću vrednosti ili stanja jedne ili više tačaka. Primer ako je jedna celobrojna tačka povezana sa jednim objektom; kad je vrednost tačke "0" boja objekta je crna. Posle nekog intervala vrednost tačke se menja na vrednost "5" koji menja boju objekta u crveno. Jedna animacija može biti aktivirana kao rezultat promene jedne tačke ili kao rezultat pisanja programa (**Script**).

Animacija se kreira korišćenjem **Animation Editor**-a u razvojnom okruženju.

Pisanje programa (Scripts)

Script je jednostavan programski jezik koji se koristi za manipulisanje tačkama. Program definiše akcije jednog objekta, stranice ili projekta. Skripte se kreiraju pomoću **Script Editor**-a.

Svaki objekat može imati akcije koje mu omogućavaju da reaguje na promene u okruženju. Akcije se mogu dodavati iz Animation Editor-a.

Tačke (Points)

Tačkama nazivamo promenljive koje koristimo za smeštaj podataka. Ovi podaci mogu biti sačuvani kao jedan od 4 tipa: logički, celobrojni, realni ili tekstualni. Tekst i tačke povezane na PLC imaju svoju dužinu - one mogu imati više od jednog elementa (tako da se ceo blok podataka može odjednom učitati sa PLC-a radi postizanja optimalne brzine komunikacije). Podaci mogu biti primljeni ili predati sa raznih spoljašnjih izvora (windows aplikacije ili najčešće PLC). Podaci takođe mogu biti postavljeni unutrašnjim script komandama, definisanim akcijama animacije. U **Point Editor**-u je predviđena dozvola za stvaranje i modifikovanje baze tačaka. Obezbeđuje kontrolu za dodavanje, modifikovanje i brisanje tačaka i takođe omogućuje kontrolu konfiguracije PLC-ova koji su korišćeni u projektu. Takođe omogućava kontrolu za filtriranje i sortiranje tačaka tako da baza može da se formira da obezbedi brzo i lako upravljanje preko njih. Postoje 3 grupe tačaka koje su uvek pristupne. Ostale grupe mogu se dodati tokom pravljenja nove tačke jednostavnim upisivanjem imena nove grupe u "group" combobox u Add Point dijalogu. Ove nove tačke se dodavaju u grupu Default.

Sistemske tačke

Sistemske tačke su unapred definisane tačke koji su automatski kreirani kada se otvori novi projekat. One se ne mogu menjati ili brisati ali se njihove osobine mogu koristiti.

13. Data Logging (Logovanje podataka)

Ovo poglavlje se odnosi na mogućnosti logovanja podataka (data logging) uključujući biranje i konfigurisanje podataka koji će biti logovani, logovanje, gledanje i prenošenje zapisanih podataka.

Šta je Data Logging?

Pojam Data Logger (logovanje podataka) je mogućnost definisanja više tačaka i izraza tokom razvoja projekta, čije vrednosti mogu biti snimljene dok se CX-Supervisor projekat izvršava. Ove događaje možemo gledati u toku izvršavanja ili ih možemo zabeležiti za buduće procenjivanje. Ovako prikupljeni podatci takođe mogu biti preneseni u druge programe kao što je Microsoft Excel.

Snimljeni događaji su smešteni u jedan ili više Data Set-ova koji se mogu videti korišćenjem Data Log Viewer-a. Događaje za snimanje je definisao korisnik i više Data Set-ova može biti korišćeno za zapisivanje specijalnih oblasti ili tipova događaja, svaki automatski u toku rada ili kao posebno označen od strane korisnika.

Kao pomoć, da bi videli određeni broj događaja u izdvojenom Data Set-u mogu se dalje razvijati u grupe podataka (Groups). Određene oblasti podataka mogu se smestiti u te grupe koji se tada mogu videti svaki posebno ili sa ostalim grupama unutar Data Set-a.

Glavni elementi u oblasti Data Logger CX-Supervisor-a:

Data Logger Editor - koristi se za podešavanje parametara logovanja.

Data Logging at Runtime - logovanje podataka.

Data Log Viewer - koristi se za gledanje logovanih podataka.

Remote Data Viewer - omogućuje gledanje odvojeno od dela koji se izvršava.

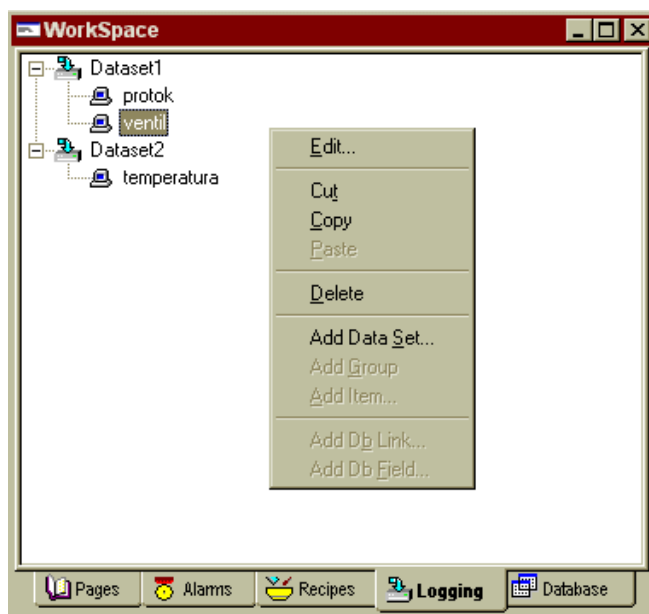
Data Log Export facility - omogućuje prenos u druge programe (MS Excel).

Script Functions - daje potpunu kontrolu nad logovanim procesom.

13.1. Data Log Editor

Podešavanje Data Setova parametara logovanja

Data Set-ovi, grupisani podaci i Item-ovi (članovi) za logovanje su pravljani iz Logging Tab-a u Workspace editoru. Item-ovi (članovi) se edituju korišćenjem menija koji se dobija pritiskom desnog dugmeta miša. Opcije koje su pristupačne sa menija su sledeće:



- Edit - omogućuje da parametri selektovanih Data Set-ova , Group-a ili Item-a budu editovani.
- Cut , Copy , Paste - omogućuje da selektovani item bude isečen ili kopiran i tada stavljen u Data Set ili Group.
- Delete - briše selektovani item.
- Add Data Set , Add Group , Add Item - omogućuje dodavanje novog Data Set-a, Group-a ili Item-a u selektovani deo.

Funkcionalnost Item-a je određena pomoću njihovog izraza. Ako je njihov izraz ime jedne tačke (point), Item će biti isti tip kao što je tačka koja je konfigurisana u Point editoru: u memoriji, ulaz, izlaz ili ulaz/izlaz. Ako izraz sadrži proračun, konstante i/ili više tačaka, item je prikazan kao skript proračun.

Adding / Editing Data Set Properties

Prozor **Add / Modify Data Set** se otvara kada se bilo koja **Add data Set** ili **Edit** (postojećeg Data Set -a) opcija selektuje iz menija.

Data Set Properties (Data Set osobine)

Polje Data Set Name omogućuje da se Data Set pozove sa punim opisom u Workspace view-u i takođe iz skript jezika. Takođe se oblikuje prefiks za ime fajla. Dozvoljeno je najviše 26 karaktera.

Polje Period omogućuje upisivanje perioda u satima, danima ili mesecima i definiše vremenski period grupisanja podataka u posebne fajlove. Na primer, ako je Period postavljen na 1 mesec svaki logovani podatak za tekući mesec se snima u isti fajl (data file), čak i kada je sistem zaustavljen i restartovan. Na primer kada se proces logovanja ne završi pre mesečne granice, fajl podataka se zatvara i počinje se novi koji dobija dosledno ime u skladu sa novim mesecom. Ovo omogućava lako arhiviranje i backup logovanih podataka. Kreiranje novog fajla podataka je uvek određeno prirodnim granicama; za mesec - prvog u mesecu, za dane - 24 sata i za sate - svaki sat.

No. Of Files to Keep određuje maksimalni broj fajlova podataka koji treba da se čuvaju. Upisan broj fajlova se čuva sve dok se ne kreira novi fajl, tada se najstariji izbriše. Na primer, za čuvanje podataka procesa za dve godine možete čuvati 24 fajla za vremenski period od 1 meseca. Imena fajlova se generišu automatski (zasnovano na imenu Data Set-a dodavanjem sufiksa) tako što se dodaje sufiks na ime Data Set-a, koji se povećava uvek za jedan kada se novi fajl generiše.

Ako je Keep all Files polje označeno nijedan fajl neće biti izbrisan.

Start Logging on Application Startup polje upravlja logovanjem Data Set-a. Kada je ovo polje označeno i ako se sistem startuje svi Item-ovi u Data Set-u počinju sa logovanjem; isto tako zaustavlja logovanje kada se sistem zaustavi. Ako ova opcija nije selektovana logovanje mora da se kontroliše sa skript funkcijom Start/Stop Logging na zahtev korisnika.

Editing Item Properties (Promena osobina itema)

Prozor **Add/Modify Item** se otvara kada se dodaje novi item ili se menja neki postojeći.

Item Properties (Osobine Item-a)

Ove opcije omogućuju upisivanje imena item-a i pomoćnih izraza. Možemo izabrati tip podataka i ako želimo možemo promeniti vrednost deadband-a.

Polje Item Name omogućava dodavanje potpunog naziva item-a. Naziv treba da bude kratak ali takođe i jasan. Dozvoljena je dužina od najviše 26 karaktera.

Polje Expression definiše imena tačaka ili izraza koji se loguju. Dugme **Browse** omogućava lako selektovanje tačke.

Polje Data Type određuje tip podataka koji je logovan. Selektujte Boolean, Integer ili Real. Za

tačke ovo je automatski preneseno iz baze podataka tačaka (point database). Za izraze tip mora biti izabran.

Polje Deadband omogućava procentualno podešavanje granice koja se primenjuje za logovanje

upisanog izraza. Kada se vrednost menja unutar granica Deadband-a vrednosti se ignorišu. Ovo je naročito korisno kada na merene veličine utiču smetnje. Odmah nakon što vrednost pređe granice deadband-a nova vrednost će se logovati. Ovo može puno da smanji veličinu fajla i obezbeđuje lakšu analizu.

Sample Rate

Opcija brzine uzorkovanja se koristi za određivanje kako će se podaci logovati, selektovanjem jednog od dva načina: on change ili on interval. Po default-u je on interval.

Kada je 'On Change' opcija izabrana podatak će biti logovan samo kada se vrednost promeni. Ovo obezbeđuje da se sve informacije loguju - čak i one promene koje se javljaju brže nego što je maksimalna brzina uzorkovanja, i takođe obezbeđuje da neaktivni periodi ne generišu duplikate podataka.

'On Interval' opcija brzine uzorkovanja obezbeđuje da se podaci loguju u određenim vremenskim intervalima. Potrebno je upisati interval kao broj ili sa liste odabrati oblik: npr. '5 seconds'. Pomoću ove metode će se vrednost zapisivati u svakom intervalu. Ako se promena kratkotrajno javlja između dva uzorkovanja te promene se ne beleže, što može biti od koristi.

Scale (Grafik)

Polje za podešavanje opsega omogućuje postavljanje granica i imena grafika izabranog item-a.

U polje Minimum value upisati minimalnu vrednost koja treba da se ispiše na grafiku za odgovarajući item.

U polje Maximum value upisati maksimalnu vrednost koja treba da se ispiše na grafiku za odgovarajući item.

Scale label je polje za tekst koje sadrži šta će se ispisati na Y - osi grafika.

Opcije koje se mogu obaviti nad Dataset-ovima i item-ima iz pop-up menija su sledeće:

Copy , Cut , Paste Options (Opcije copy , cut i paste)

Opcije Copy, Cut i Paste prate windows metod rada i omogućuju da selektovani item bude kopiran ili isečen iz jednog Data Set Group-a i prebačen u drugi. Imena item-ova unutar jednog Data Set-a moraju biti jedinstvena, kada se kopirani item vraća u isti Data Set u kojem to ime već postoji na ime se dodaje numerički sufiks.

Novi ili već postojeći Item se može preimenovati, parametri se mogu menjati pomoću obeležavanja itema i biranjem Edit opcije iz pop-up menija.

Adding/Renaming Groups

Dodavanje ili preimenovanje grupa u Data Set-u prati windows metod rada. Nove grupe se dodaju biranjem Add Group opcije iz pop-up menija. Nova grupa se automatski postavlja na određenu poziciju unutar Data Set-a i dobija nedostajući broj grupe. Nove ili postojeće grupe se mogu preimenovati obeležavanjem grupe i biranjem Edit opcije iz pop-up menija. Grupe se brišu sa Delete.

14. Logovanje podataka

Logovanje podataka se vrši u skladu sa osobinama item-ova koji su smešteni u Data Set-u. Sistem može da sadrži više Data Set-ova. Svaki Data Set će imati sopstvenu grupu fajlova kada je sistem u izvršnom modu. Količina podataka koja je smeštena u jedan Data Set fajl je određena dužinom vremenskog perioda koji je izabran pri kreiranju Data Set-a.

Svi log fajlovi su smešteni u direktorijum " Data Logging " koji je odvojen od ostalih fajlova CX-Supervisor-a. Direktorijum se kreira automatski kada se projekat pokreće prvi put. Ovaj poddirektorijum se koristi kao zaštita protiv slučajnog brisanja važnih fajlova. Na primer ako se projekat u CX-Supervisor-u zove **c:\CX-Supervisor\Project** tada se kreira direktorijum koji se zove **c:\CX-Supervisor\Project\Data Logging** za smeštanje log fajlova.

Svaki Data Set ima sopstvenu grupu fajlova. Sledeća pravila se koriste prilikom kreiranja fajlova radi njihove lakše identifikacije. Imena fajlova se kreiraju od imena DataSet-a, a dodaje im se vreme i datum kreiranja.

Data Log fajlovi su označeni sa ekstenzijom **.dlv** "**Data Log Values**" i format imena fajla je:

<ime DataSet-a><vreme i datum>.dlv

Vreme zapisivanja je oblika:

[YYYY MM DD HH]

gde su:

YYYY = godina u obliku 1999

MM = mesec u obliku 01 (Januar)

DD = dan u mesecu u obliku 01 - 31

HH = sati u obliku 00 - 23

Primer:

DataSet 1 brzina[2003012015].dlv

Novi log fajl se kreira kada istekne vreme prethodnog. Vremenski period se određuje kada se kreira Data Set. Minimalni vremenski period je jedan sat.

Napomena:

- Nemojte preimenovati nijedan fajl dok su oni u direktorijumu Data Logging jer File Management Sistem neće ispravno raditi !
- Dodatne fajlove nemojte kopirati u Data Logging direktorijum jer mehanizam za automatsko čišćenje može izbrisati sve log fajlove.

Kada se Data Set pokreće kreira se log fajl za čuvanje logovanih item-ova, taj fajl je 'Live' ili 'Active'. (Uvek je samo jedan 'Live' u Data Set-u). Kada istekne vremenski period za log fajl, novi fajl se kreira sa novom vremenskom oznakom. Tada novi fajl posaje 'Live' a prethodni 'Dead'. Prekoračenjem vremenskog intervala može nastati mnogo 'Dead' fajlova ali samo jedan 'Live'. Ako je granica postavljena na neki broj toliko će se 'Dead' fajlova sačuvati, novi će se kreirati a najstariji će se automatski izbrisati. Oređivanje broja 'Dead' fajlova se vrši pri kreiranju Data Set-a upisivanjem broja u 'Number of Files to Keep'. Ovo je uvek omogućeno kada checkbox 'Keep all files' nije obeležen.

Napomena: Uvek će biti jedan 'Live' fajl za svaki Data Set.

Na primer: Ako je broj fajlova za čuvanje postavljen na 24 , tada će sistem sačuvati 24 'Dead' fajlova i jedan 'Live' tj. ukupno 25 fajlova.

Data Log fajl se generiše tokom trajanja jednog projekta i koristi se u narednom periodu ako je još uvek 'Live'. Na primer ako je sistem uključen u toku dana i isključi se preko noći, tada će Data Set sa trajanjem od 7 dana koristiti isti fajl za svih 7 dana. Ako se Data Set item promeni, briše ili se dodaje novi nije opravdano koristiti postojeći fajl. Kada se sistem startuje i Data Set fajl se otvori CX-Supervisor proverava da li ima puno razlike u tekućem Data Set-u i u Data Set-u smeštenom u fajlu. Ako se detektuje neka razlika sistem će obeležiti postojeći fajl kao invalid (sa greškom) tako što zagrade "[]" promeni na "{}{}" i kreira novi fajl.

Na primer ako se fajl **Rad[2003032922].dlv** detektuje kao invalid (sa greškom) biće preimenovan na **Rad{{2003032922}}.dlv**. Novi **Rad[2003032922].dlv** fajl će se kreirati za logovanje novih podataka. Ovo se najviše javlja tokom razvoja neke aplikacije kada se Data Item-ovi kontinualno ispravljaju i testiraju.

Invalid fajlovi nisu deo Data Set fajlova i ne mogu biti izbrisani ili pronađeni pomoću File Management pretraživača 'next' ili 'previous'. Podaci zapisani u tim fajlovima nisu izgubljeni i mogu

se videti korišćenjem Data Log Viewer-a kao što se mogu i preneti iz njih korišćenjem Export Log-a. Invalid fajlovi se mogu izbrisati standardnim windows komandama.

Primeri vremenskih perioda Data Set-a

Sledeći primeri pokazuju kako se specifični vremenski periodi koriste (Hours(sati), Days(dani), Months(meseci)) dok je sistem pokrenut.

1. Data Set period = Hours

Ako je Data Set nazvan CV500 sa počekom logovanja u 12:30 7. marta 1999. sa vremenskim periodom od 6 sati kreiraće se sledeći fajl:

CV500[1999030712].dlv

Ako se sistem ostavi uključen u Data Logging direktorijum će biti zapisani sledeći fajlovi:

CV500[1999030712].dlv

CV500[1999030718].dlv

CV500[1999030800].dlv

CV500[1999030806].dlv itd.

Prvi fajl će sadržati samo 5,5 sati podataka jer je minimalna rezolucija jedan sat i fajl je napravljen na pola sata, svi naredni fajlovi će sadržati celih 6 sati podataka.

2. Data Set period = Days

Ako Data Set koji se zove Posao počinje sa logovanjem u 23:00 29. decembra 2002. sa vremenskim periodom od jednog dana kreiraće se sledeći fajl:

Posao[2002122923].dlv

U Data Logging direktorijum će se zapisati sledeći fajlovi:

Posao[2002122923].dlv

Posao[2002123000].dlv

Posao[2002123100].dlv

Posao[2003010100].dlv

itd.

Prvi fajl će sadržati samo 1 sat podataka, svi naredni fajlovi sadrže svih 24 sata od podataka.

3. Data Set period = Months

Ako Data Set koji se zove Smene počinje sa logovanjem u 22:00 29. februara 2003. sa vremenskim periodom od 2 meseca kreiraće se sledeći fajl:

Smene[2003022922].dlv

Ako je sistem ostavljen da se izvršava Data Logging direktorijum će nastaviti sa zapisivanjem sledećih fajlova:

Smene[2003040100].dlv

Smene[2003060100].dlv

itd.

Prvi fajl će sadržati samo 1 mesec i 2 sata podataka, svi naredni fajlovi sadrže punih 2 meseca podataka.

14.1. Data Records (Zapisivanje podataka)

Postoje dva načina zapisivanja podataka koji su smešteni u Data Log fajlovima; za svaki logovani item to su: 'Events' (događaji) i 'Actual Data' (stvarni podaci).

Event / Break Records

Event zapisivanje sastoji se od sledećih polja:

Event (dogadjaj), Date(datum događaja), Time(vreme događaja)
Event = Open | Close | Clear | Start | Stop (vrsta događaja)

Data Records

Zapisivanje podataka se sastoji od sledećih polja:

Type(tip), Date(datum), Time(vreme), Milliseconds(milisekunde),

Data(podatak)

Type = Normal | Error(da li je podatak ispravno zabeležen ili ne)

Data = Boolean | Integer | Real(tip podatka)

14.2. Data Log Viewer Component (pregled logovanih podataka)

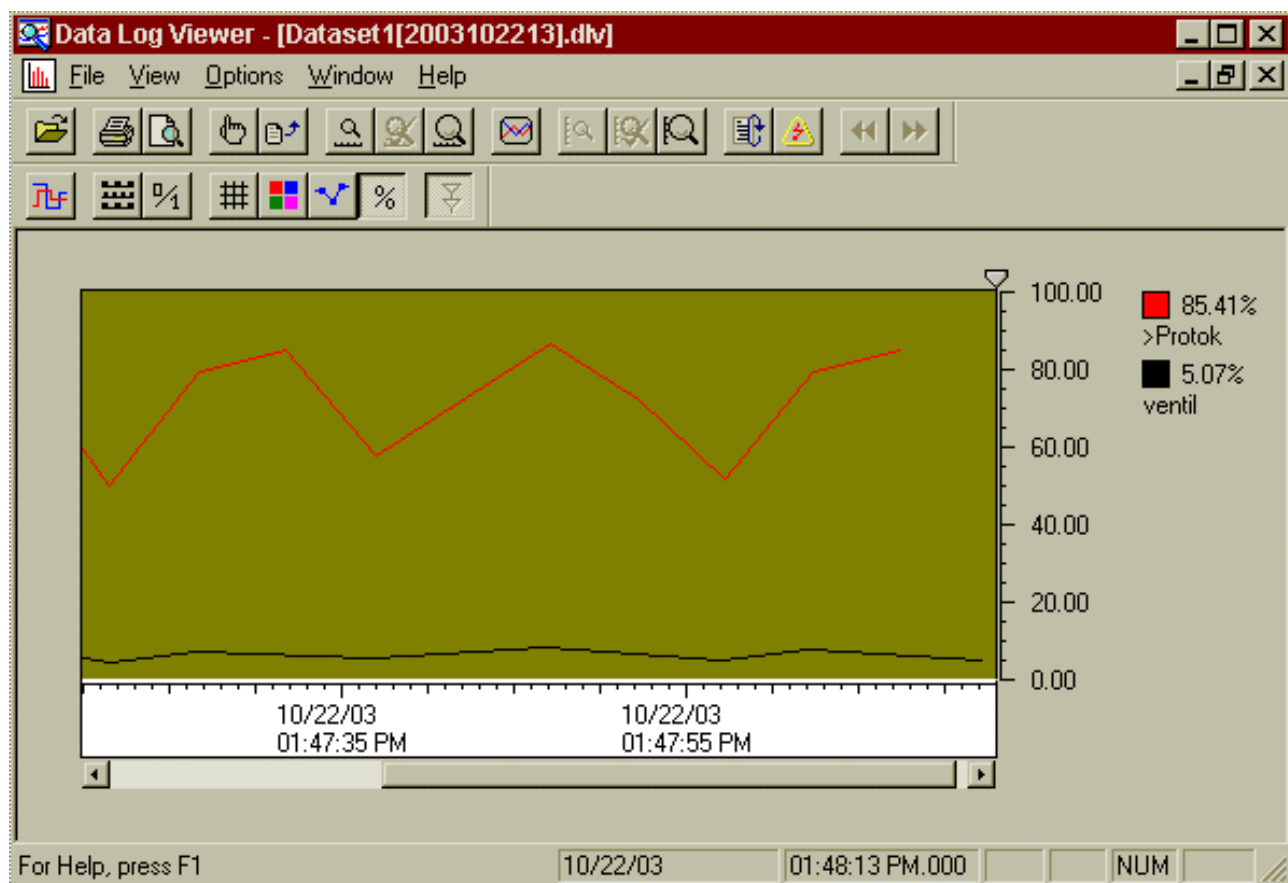
Data Log Viewer se može pozvati na sledeći način:

- selektovanjem Data Log Viewer-a sa pop-up menija (ako je dozvoljen) u toku izvršavanja aplikacije.
- iz skript funkcije ' Open Log View ', a zatvara se sa ' Close Log View ' skript funkcijom.
- iz nekog spoljašnjeg programa.

Osobine:

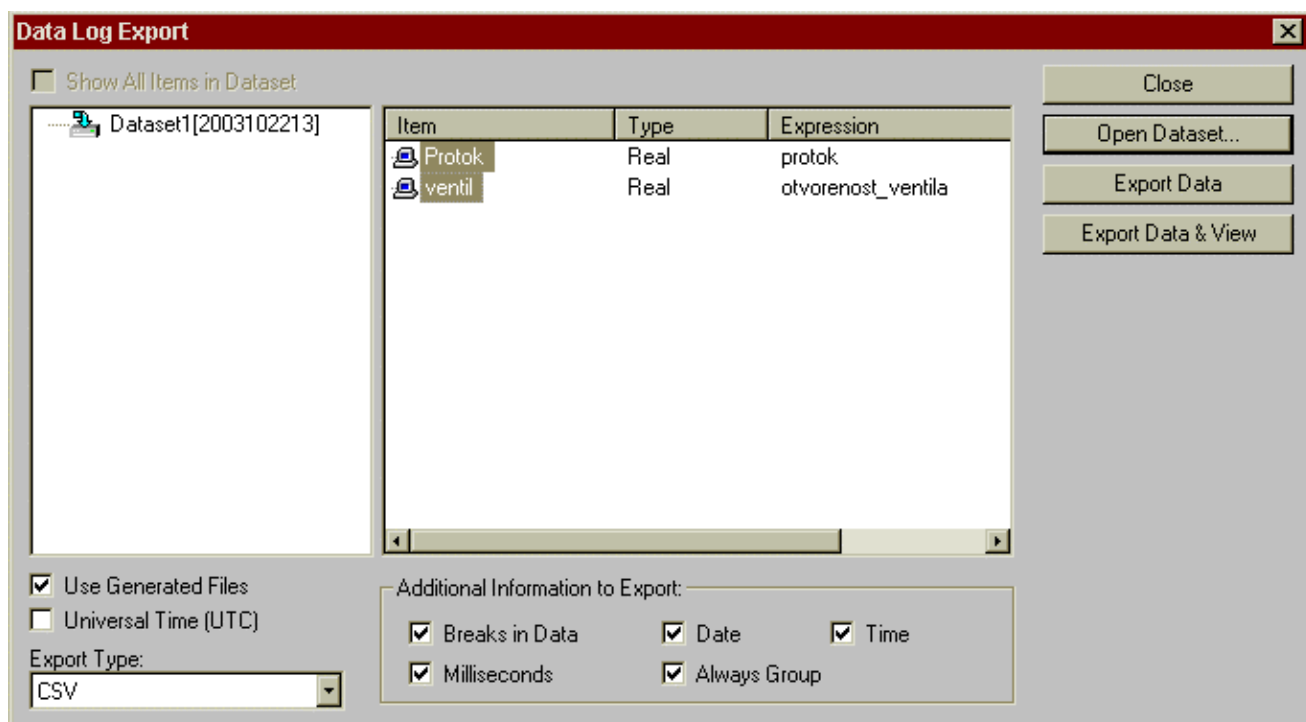
- Data Set prozor ima ime koje je slično kao ime u skript funkciji ili je Data Set odabran od strane korisnika.
- Odjednom se mogu crtati do 10 analognih linija i 150 digitalnih linija. Menjanjem skript parametara može se koristiti automatska selekcija za prikazivanje linija promenljivih.
- prozor pokazuje sledeće:
 - poslednji fajl podataka za taj Data Set
 - x-osa je postavljena na isti vremenski period kao što je i Data Set
 - y-osa postavljena je na celu vrednost skale za prvu liniju
- selektovane analogne linije prikazane su precrtano
- tačke logičkog tipa prikazane su u odvojenom delu
- na x-osi ispisan je stvaran datum i vreme kao lokalno vreme na logovanom sistemu.
- selektovana linija ima znak ">" ispred imena.
- Grafik pokazuje minimalnu vrednost, maksimalnu vrednost i ime grafika za odabranu liniju sa pokazivačem koji se nalazi u sredini.
- kursor omogućava očitavanje podataka sa grafika. Pritiskom na Ctrl+strelice omogućeno je kretanje među logovanim podacima.
- Uvećanje i smanjenje razmere sa ZoomIn i ZoomOut, i za x i y osu.
- 'Previous' i 'Next' dugmadi pozivaju data fajlove prethodnog ili sledećeg vremenskog perioda.
- 'Refresh' ponovo poziva fajl sa diska.
- 'Live' opcija se koristi za prikazivanje trenutnog stanja.
- podatak se može pretvoriti u CSV fajlove ili u tekst

Data Log Viewer izgleda ovako:



14.2.1. Snimanje podataka u nekom od standardnih formata

Podatke je moguće snimiti pomoću ExportLog funkcije iz CX-Supervizor menija, pomoću skript funkcije ili iz Data Log Viewer-a.



Prilikom snimanja podataka moguće je odrediti više parametara:

-use generated files

Ako je ova opcija označena, tada se imena fajlova automatski preuzimaju od već postojećih .dlv fajlova, a u drugom slučaju pojaviće se prozor 'Save File As' i može mu se promeniti ime.

-export type

Bira se tip snimljenog fajla: - CSV(comma-separated values) - podaci odvojeni zarezom
- tekstualni.

-additional information to export

Snima i dodatne informacije: pauza, datum, vreme i milisekunde.

14.2.2. Kreiranje CSV fajlova

CSV je format koji se direktno prepoznaje, otvara i pokazuje sa programima kao što je Excel. Moguće je exportovati jedan ili više itema u isti fajl. Fajl će sadržati kolone za svaki selektovani item i oni će popuniti Data i Time polja. Ako je više selektovano "on interval" item-ova sa različitim vremenskim intervalima, tada će se item-ovi sa istim vremenskim intervalima grupisati zajedno u jedan fajl.

Primer:

Ako su item -ovi "I1" "I2" "I3" "I4" "I5" i "I6" izabrani iz Group1 DataSet-a MyDataSet i item-ovi "I1" "I3" "I6" obuhvataju iste intervale i "I2" i "I5" takođe pripadaju istim intervalima tada će nastati sledeći fajlovi:

MyDataSet 2003011210 Group1.csv; sadrži I1, I3 i I6

MyDataSet 2003011210 Group1_1.csv; sadrži I2 i I5

MyDataSet 2003011210 I4.csv; sadrži I4

Svi "on change" item-ovi će imati svoje sopstveno ime fajla koje nastaje bez obzira na to da li je izabrano više itema, jer nije moguće odrediti vremenski interval sa ovakvim tipovima item-ova.

14.2.3. Kreiranje tekstualnih fajlova

Obzirom na to da samo .csv fajlovi sadrže ograničenu količinu informacije (Break, Date, Time, Millisecond i Value), tekstualni .txt sadrže sve informacije smeštene u izabrane grupe/item-ove (Groups/Items); izraz, labela, 'deadband' itd. Pravila za selekciju više njih su ista kao što je opisano u prethodnom poglavlju, sa izuzetkom 'on change' item-ova koji se mogu zajedno grupisati.

14.3. Povezivanje sa već postojećim bazama podataka

Moguće je zapisivati podatke direktno u već postojeću bazu podataka. Ovo omogućuje direktno smeštanje podataka u raznim standardnim formatima, obezbeđujući korišćenje uobičajenih alata za laku analizu. Baze podataka moraju biti prvo kreirane korišćenjem specifičnog softvera po vašem izboru. Na primer "MS-Access".

Postupak pravljenja veze sa bazom podataka u CX-Supervisor-u:

- kreirati jedan prazan "file dsn" u Windows Control panelu(ODBC).
- kreirajte vezu ("connection") sa kreiranim dsn fajlom u Workspace Database editoru.
- Dodajte "Recordset" (Obavezno stavite "pessimistic" lock metod !), a u njemu sva potrebna polja (Fields) sa "field property" postavljenim na 'value' i isključenim "read on open".

- Kreirajte vezu sa bazom u "Workspace Logging" editoru selektovanjem 'Add Db Link....' iz pop-up menija da bi otvorili prozor Add Database Link.
- kreirajte Field Link za svako polje za logovanje tako što selektujete 'Add Db Field' iz pop-up menija da bi otvorili prozor Add Field Link.

Moguće je podesiti sledeće parametre zapisivanja podataka u bazu:

Brzina uzorkovanja (Sample Rate)

- Vremenski interval između dva zapisa u bazu. Ovo važi samo ako nije uključena opcija zapisa vrednosti na svaku promenu ("On change").

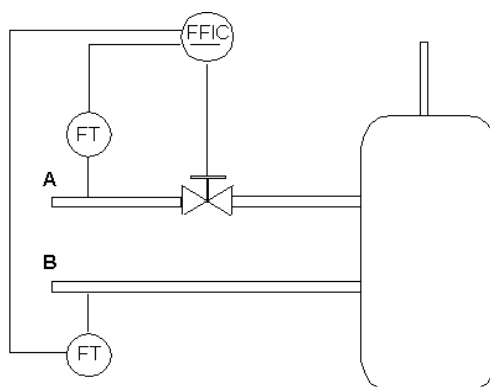
Start Logging on Application Startup

- Kada ova opcija nije uključena, logovanje se mora započeti i zaustaviti skript komandama. Kada je označena, tada se logovanje sa svim povezanim poljima startuje automatski čim je aplikacija pokrenuta.

15. Primeri

15.1. Primer 1.

Nacrtati regulaciju odnosa protoka u **CX-Supervisor**-u na osnovu sledećeg crteža.



Slika 15.1. - Regulacija odnosa protoka.

Na slici 15.1 je prikazana regulacija odnosa, gde se protok B meša sa protokom A u određenom odnosu, da bi došlo do odgovarajuće reakcije. Formula po kojoj se određuje otvorenost ventila da bi postigli željeni odnos je:

$$\frac{P'_A}{P_B} = \text{ODNOS}$$

$$P'_A = \text{ODNOS} \cdot P_B$$

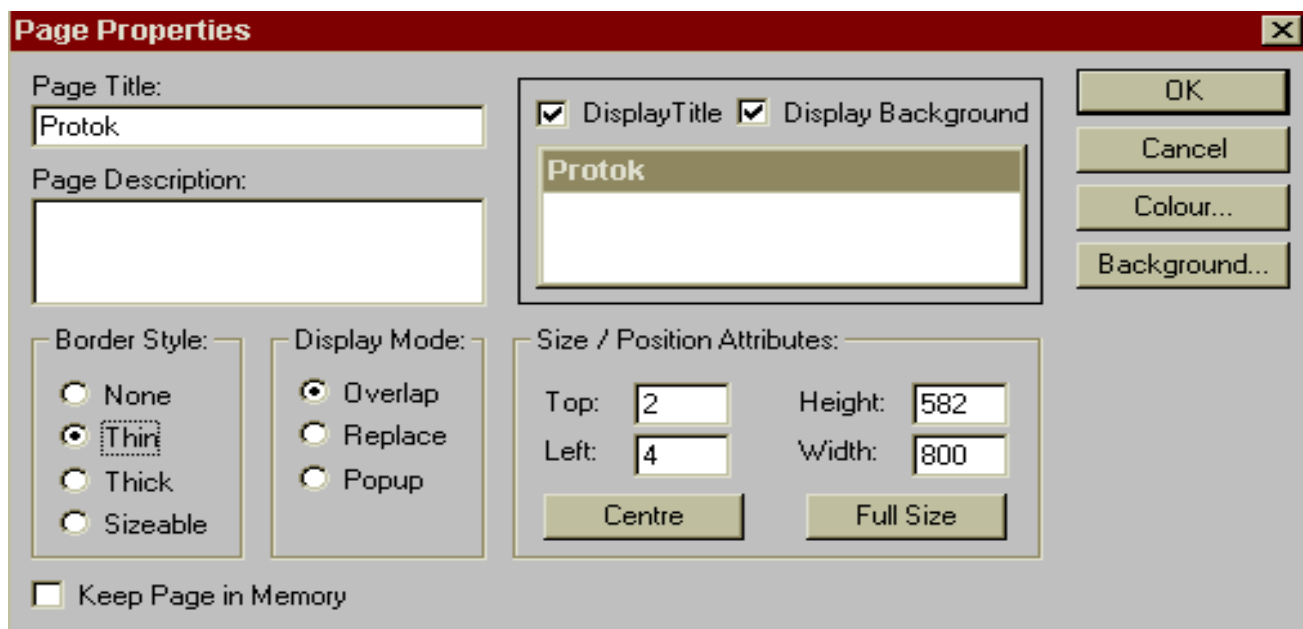
$$P_A \cdot V = \text{ODNOS} \cdot P_B$$

Otvorenost ventila :

$$V = \frac{\text{ODNOS} \cdot P_B}{P_A}$$

1. Otvorite novi projekat u **CX- Supervisor**-u. Napišite ime projekta i zadajte željenu putanju gde hoćete da ga snimate. Nakon toga pojaviće se nova prazna stranica, koju sada treba da podesite. Pre podešavanja izgleda nove stranice, snimite je u **File - Save Page As...** Ovo označava snimanje nove stranice (**Page**).

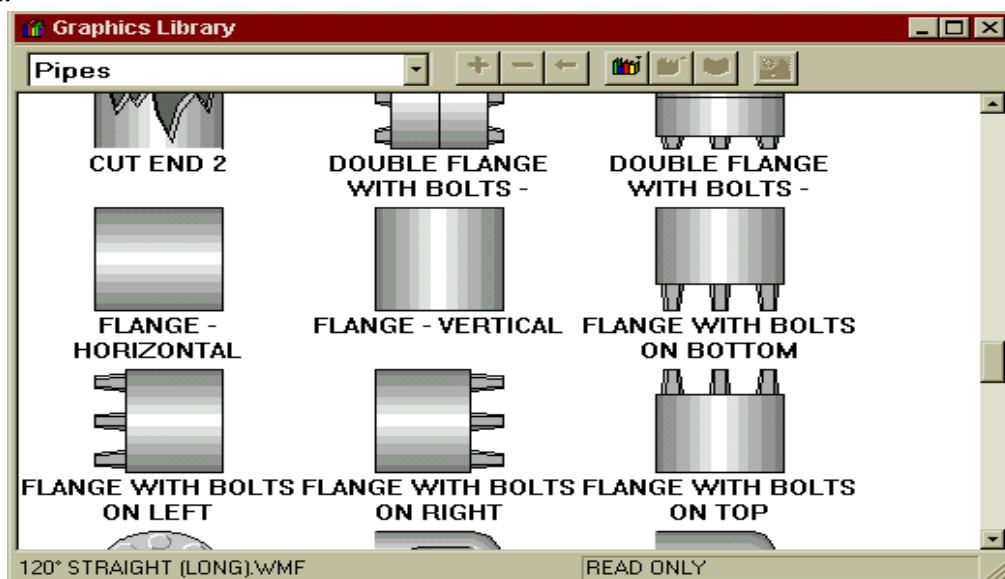
2. Dvoklikom na snimljenu stranicu otvoriće se prozor **Page Properties** gde se podešava osnovni izgled nove stranice. Ovde možete upisati ime, opis stranice, izgled, veličinu, boju itd.



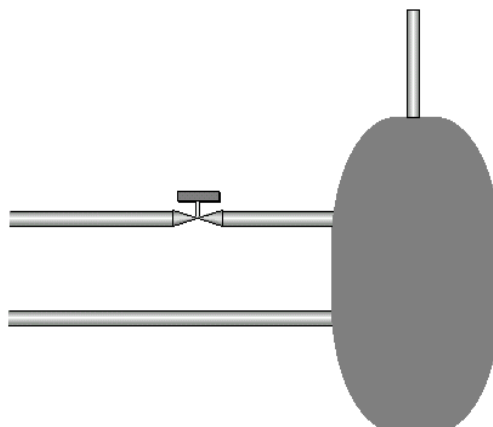
3. Sada možete početi sa crtanjem. Nacrtajte rezervoar na osnovu slike 1 koristeći **Round Rectangle** dugme iz **Graphic Object Bar** menija. Crtež se ne snima isto kao što i stranica nego kao projekat, i to u **Project - Save**, ili klikom na ikonicu za snimanje.



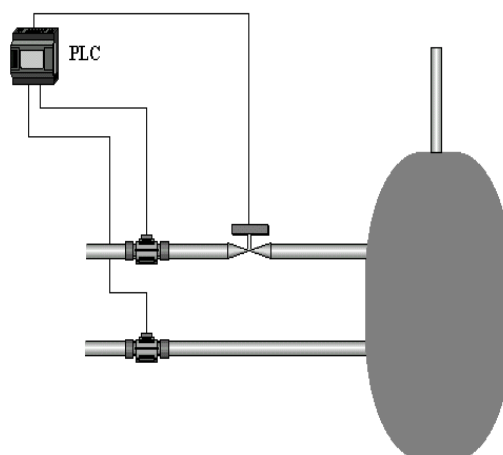
4. Desnim klikom na pozadinu aktivnog prozora pojaviće se **Pop-up Menu** odakle biramo **Graphic Library**. U ovoj biblioteci ima gotovih slika (pod grupe) koje možemo koristiti u našim projektima.



Izaberite odgovarajuće cevi i ventil za dalje crtanje i prevucite selektovani objekat na vašu stranicu. Ove slike možete da menjate, u ovom slučaju da ih oblikujete kako to najviše odgovara za ovaj slučaj.



5. Sada treba da se dočrtaju merni pretvarači protoka kao i uređaj kojim se reguliše njihov odnos. U ovom slučaju uređaj za regulaciju može biti PLC. Sve elemente biramo iz **Graphic Library** biblioteke. Zatim ih povežemo. Komentare pored pojedinih elemenata pišemo na taj način što biramo ikonu *text* iz **Graphic Object Bar** menija.



Kada se sa ovim završi najveći deo crtanja je urađen.

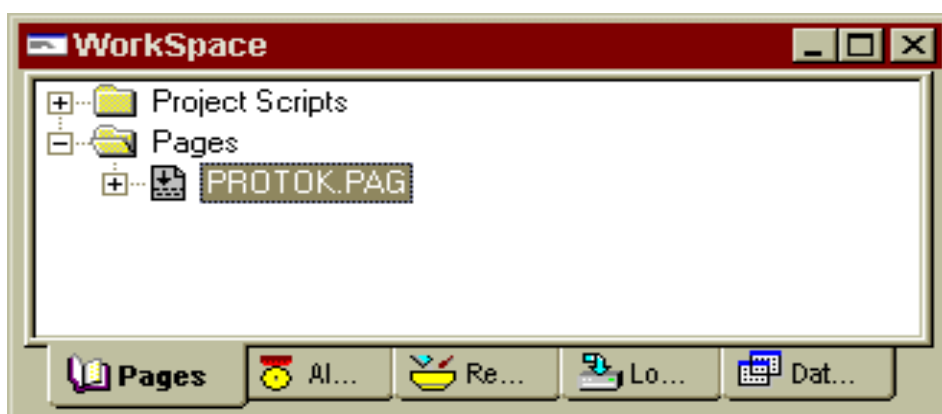
6. Sada se pristupa pisanju programskog dela.

Svakoj promenljivoj u zadatku treba dodeliti jednu tačku (**Point**). **Point Editor** se nalazi u **Toolbar**-u. Jednim klikom na ikonicu pojaviće se prozor **Point Editor**-a. Ovde su sadržane tačke koje su unapred definišane (All Groups, Default, System Points, Unused Points) ali možemo i mi da dodajemo što se automatski smešta u **Default**.

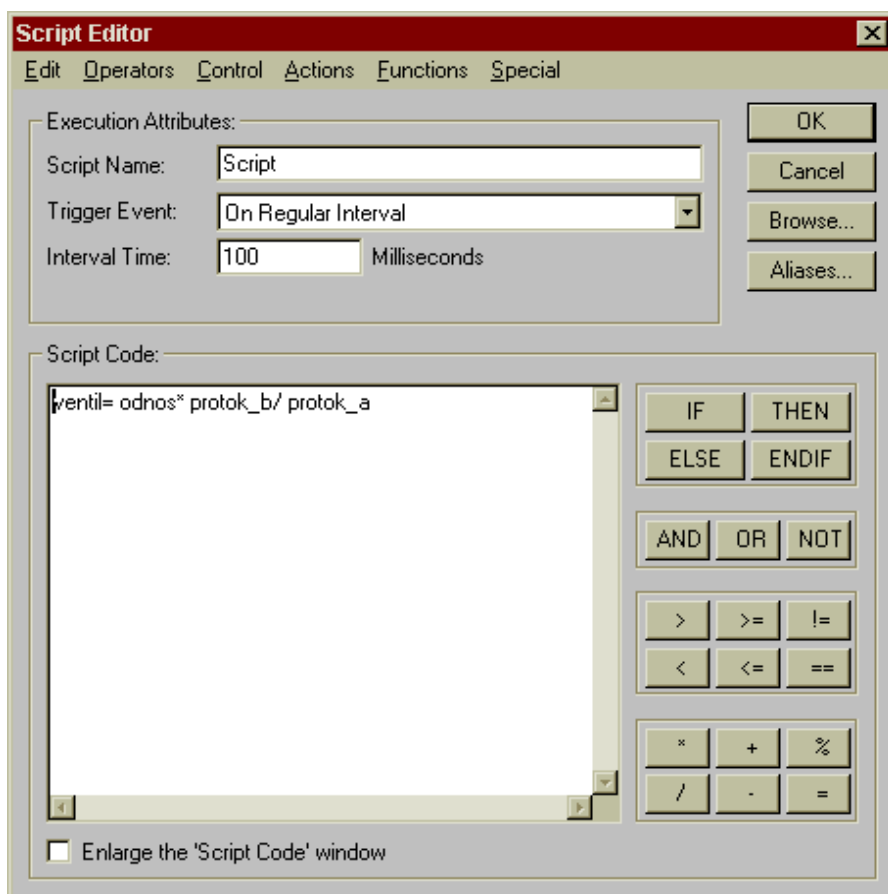


Dodavanje novih tačaka se radi jednim klikom na *plus* znak u **Point Editor**-u. Tada će se pojaviti prozor koji nas pita za osobine nove tačke. Ovde treba upisati ime tačke, opis, tip (logički, celobrojni, realni ili tekstualni), vrednost, gde se nalazi ta tačka (memoriji, ulaz, izlaz) itd. U ovom slučaju potrebne tačke su: *protok_a*, *protok_b*, *odnos* i *ventil*. Od kojih su *protok_a*, *protok_b* i *odnos* realne promenljive i nalaze se u memoriji, dok je *ventil* celobrojna i takođe se nalazi u memoriji. Minimalne i maksimalne vrednosti za *protok_a*, *protok_b* i *odnos* nema potrebe da upisujemo jer nam ponuđeno odgovara, ali kod *ventil* tačke potrebno je podesiti u granice od 0 do 100.

7. Deo za programiranje se piše u **Workspace** prozoru, čija se ikona nalazi na **Toolbar**-u.



U **WorkSpace**-u su sadržani svi elementi koji su nacrtani u projektu (**Pages**). Pisanje programa se radi u **Project Scripts**-u. Desnim klikom na **Project Scripts** biramo **New Project Script**..

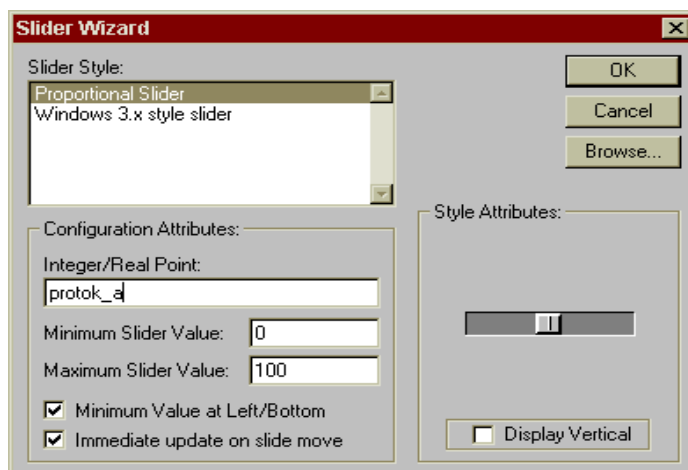


Otvoriće nam se prozor gde se piše program: **Script Editor**. Ovde možemo da biramo kako će se izvršavati naš program: **On Condition**, **On Regular Interval**, **On Initialisation**, **On Termination**, **On Key Press**, **Subroutine**. Nama je potrebno da se program izvršava u jednakim vremenskim intervalima tako da biramo **On Regular Interval** i podesimo da se izvršava svakih 100 mili sekundi. Program se sastoji iz jedne jednačine:

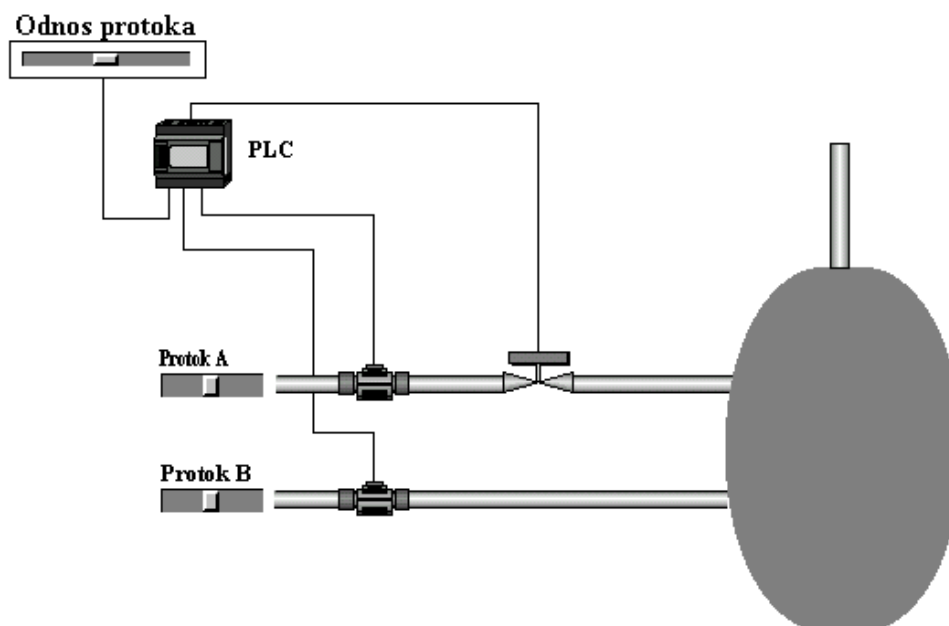
$$V = \frac{ODNOS \cdot P_B}{P_A}$$

koju upisujemo u **Script Code**.

8. Treba još rešiti problem zadavanja različite vrednosti za protoke A i B, kao i njihov odnos. Da bi protok mogli menjati treba da izaberemo ikonu **Slider** iz **Graphic Object Bar**-a. Dvoklikom na taj element pojaviće se prozor **Slider Wizard**, gde povezujemo tačku (Point) protok A sa **Slider**-om i podešavamo granice za taj protok. Isto tako uradimo i za protok B.

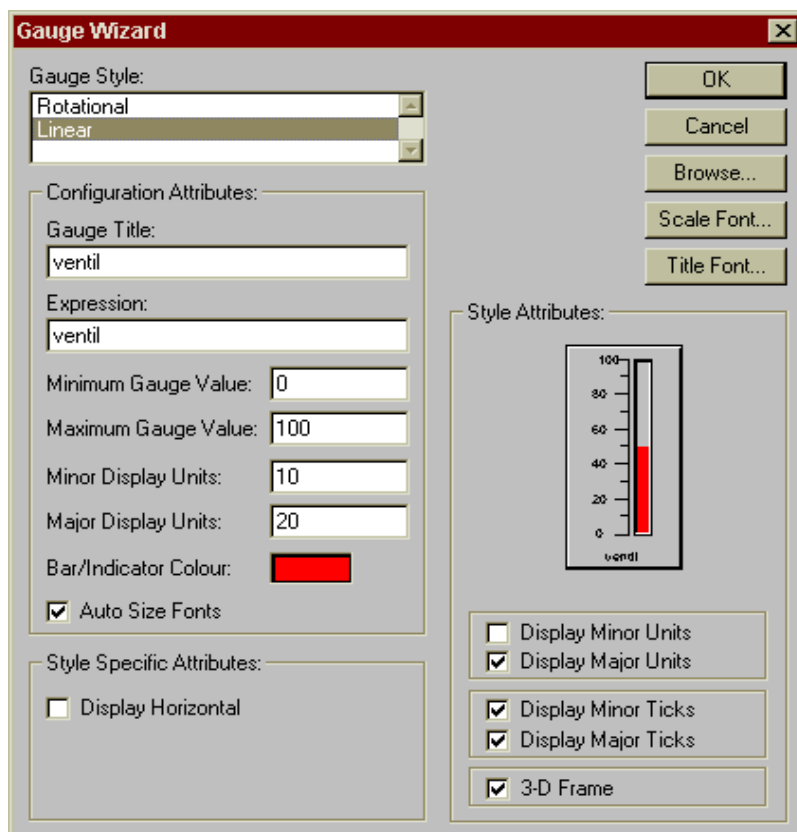


Konstantnu vrednost odnosa protoka takođe treba zadati. U ovom slučaju i to je rešeno sa **Slider**-om.



Ovako napravljen projekat već radi, iako njegovim startovanjem nećemo videti nikakve promene na ekranu.

9. Da bi mogli pratiti promene otvaranja i zatvaranja ventila možemo dodati pokazivač **Linear Gauge**, koji se nalazi u **Graphic Object Bar**-u. Dvoklikom na nacrtani **Linear Gauge** otvoriće se prozor **Gauge Wizard**, gde terba da upišemo koju veličinu želimo da prikazuje (**Expression**), što je u ovom slučaju *ventil*. Možemo podesiti minimalni i maksimalni opseg prikazivanja, što se ovde stavlja od 0 do 100, boju prikazivanja itd.

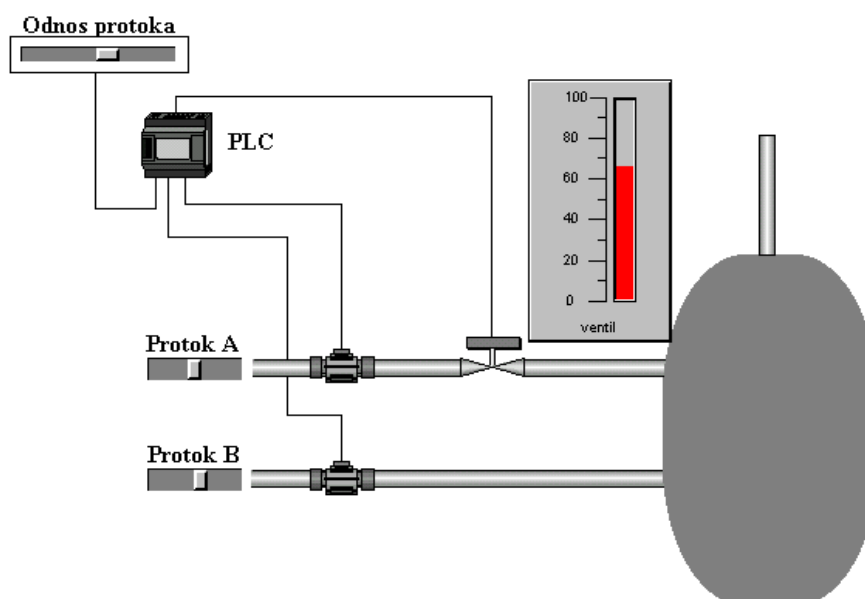


U zadatku se vidi da je protok B uvek isti i da se samo protok A reguliše otvaranjem i zatvaranjem ventila. Da bi znali kada je odnos dobar možemo staviti sigurnosni alarm koji nam javlja da je protok A premali. **Alarm Editor** se nalazi u **Toolbar**-u. Klikom na ikonu otvoriće se prozor, gde se dodaju alarmi ako pritisnemo *plus*. Novi prozor (**Modify Alarm**) daje mogućnosti za različite vrste alarma. Ovde upisujemo ime alarma, prioritet, opis, tip, na osnovu čega da daje alarm i šta da ispisuje ako dođe do toga.

Ovaj alarm ako se aktivira izbacuje jedan prozor (**Acknowledge Box**) gde nas upozorava da je premali protok A. Prioritet alarma je srednji (**Medium**), tip je **Simple** i uslov po kojem ispisuje alarm (**Expression**) je sledeće :

$$\frac{\text{protok_a}}{\text{protok_b}} \cdot 100 < \text{odnos}$$

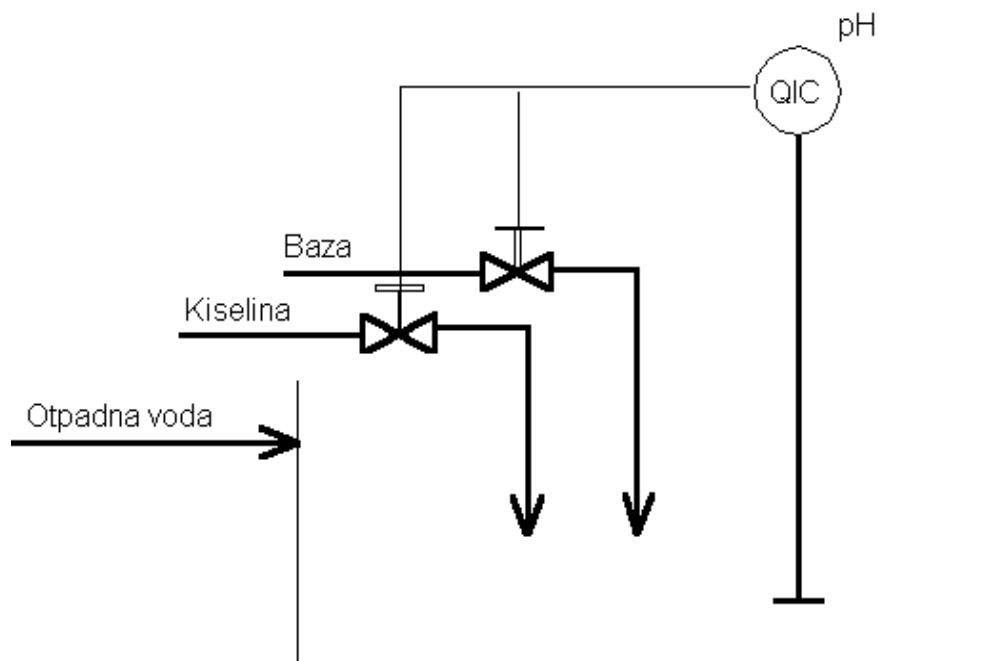
10. Kada se i ovo završi možemo pokrenuti projekat pritisnuvši dugme **Run the project** u **Toolbar**-u.



11. Projekat se zatvara sa **Project - Close**.

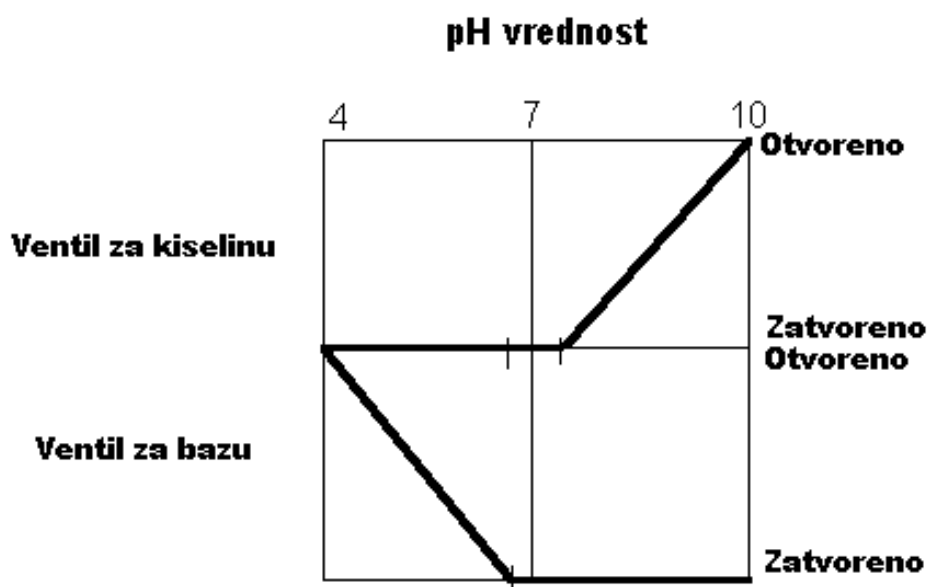
15.2. Primer 2.

Rešiti problem pročišćavanja otpadne vode na osnovu slike 14.2 pomoću **CX-Supervisor-a** dodavanjem baze ili kiseline. Merenjem pH vrednosti otpadne vode određuje se koji ventil treba da se otvori ili zatvori.



Slika 15.2. - Pročišćavanje otpadne vode.

Otvorenost ventila se određuje na osnovu sledećeg upravljačkog signala.



Slika 15.3. - Upravljački signal.

Sa slike 3 vidimo da ventil za kiselinu treba otvoriti ako je pH vrednost između 7.2 i 10, a ventil za bazu ako je pH vrednost između 4 i 6.8. Smatra se da je otpadna voda neutralna ukoliko je pH vrednost između 6.8 i 7.2. Ovaj problem je matematički rešiv na osnovu slike 15.3, pomoću jednačine prave kroz dve tačke.

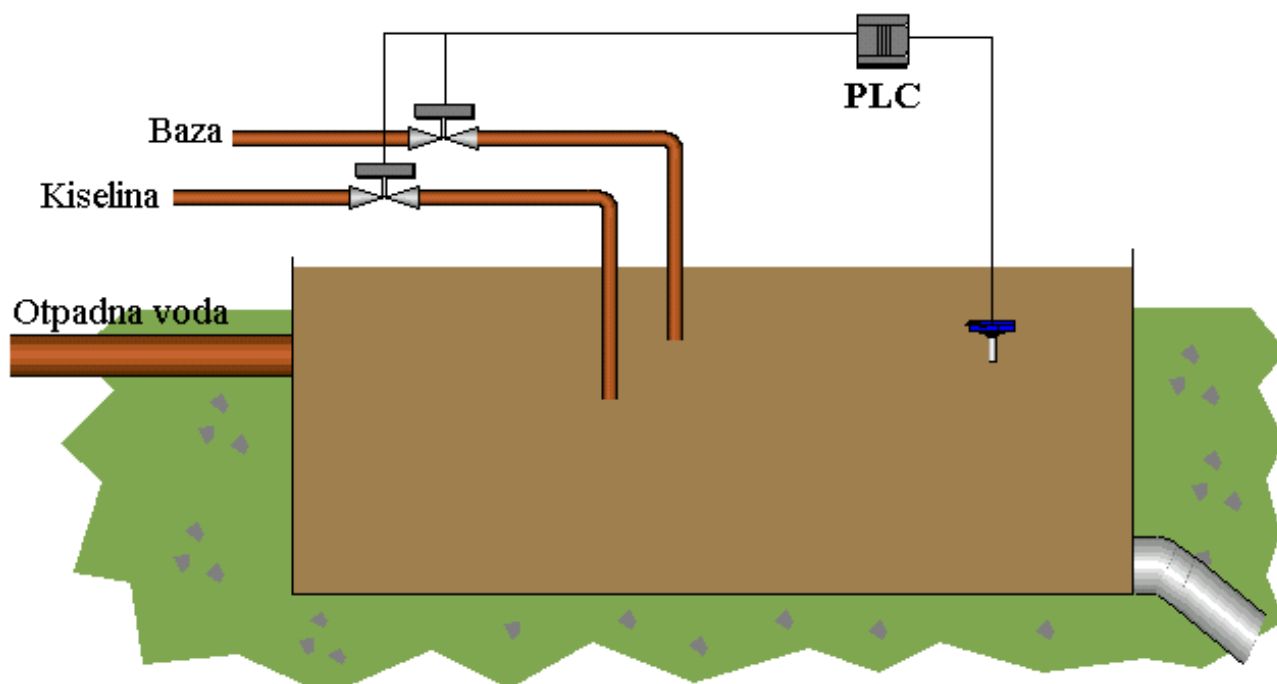
$$baza = 1 - \frac{ph - 4}{2.8}$$

$$kiselina = \frac{ph - 7.2}{2.8}$$

Ove jednačine su nam od pomoći pri pisanju programskog dela.

1. Otvorite novi projekat u **CX-Supervisor-u**. Postupak otvaranja i snimanja je uvek isti kao u prethodnom primeru.

2. Na osnovu slike 14.2 nacrtajte bazen sa otpadnom vodom kao i dovod i odvod te vode. Zatim nacrtajte cevi i ventile koristeći **Graphic Library** i alate iz **Graphic Object Bar** menija. Kao uređaj za regulaciju možemo uzeti PLC. Takođe možemo napisati i komentare pored odgovarajućih cevi.



3. Sada treba definisati sve promenljive tj. tačke (**Point**) u **Point Editor-u**. U ovom slučaju su potrebne tri promenljive, za bazu, kiselinu i pH vrednost. Pritiskom na *plus* dugme u **Point Editor-u** dodajemo nove tačke. Te novo definisane tačke će se smestiti u **Default**. U ovom zadatku potrebne su tri tačke: *baza_ventil*, *kiselina_ventil* i *ph_vrednost*. Sve tačke su realnog tipa i nalaze se u memoriji. Minimalna vrednost za bazu i kiselinu je 0 a maksimalna je 1, dok kod pH vrednosti možemo prihvatiti ponuđenu vrednost.

4. Sada možemo da uradimo programski deo. Otvorimo **Work Space** prozor i desnim klikom na **Project Scripts** biramo **New Project Script...**, koji otvara novi prozor **Script Editor**.

Nama je najpodesnije da se ovaj proces odvija u pravilnim vremenskim intervalima, tako da bismo **On Regular Interval**, koji se izvršava svakih 1000 milisekundi. U **Script Code**-u pišemo program.

Na osnovu analize slike 3 možemo napisati program:

```
kiselina_ventil=0
baza_ventil=0
IF ph_vrednost<=6.8 THEN
    baza_ventil=1-((ph_vrednost-4)/2.8)
ENDIF
IF ph_vrednost>=7.2 THEN
    kiselina_ventil=(ph_vrednost-7.2)/2.8
ENDIF
ph_vrednost=ph_vrednost+baza_ventil-kiselina_ventil
```

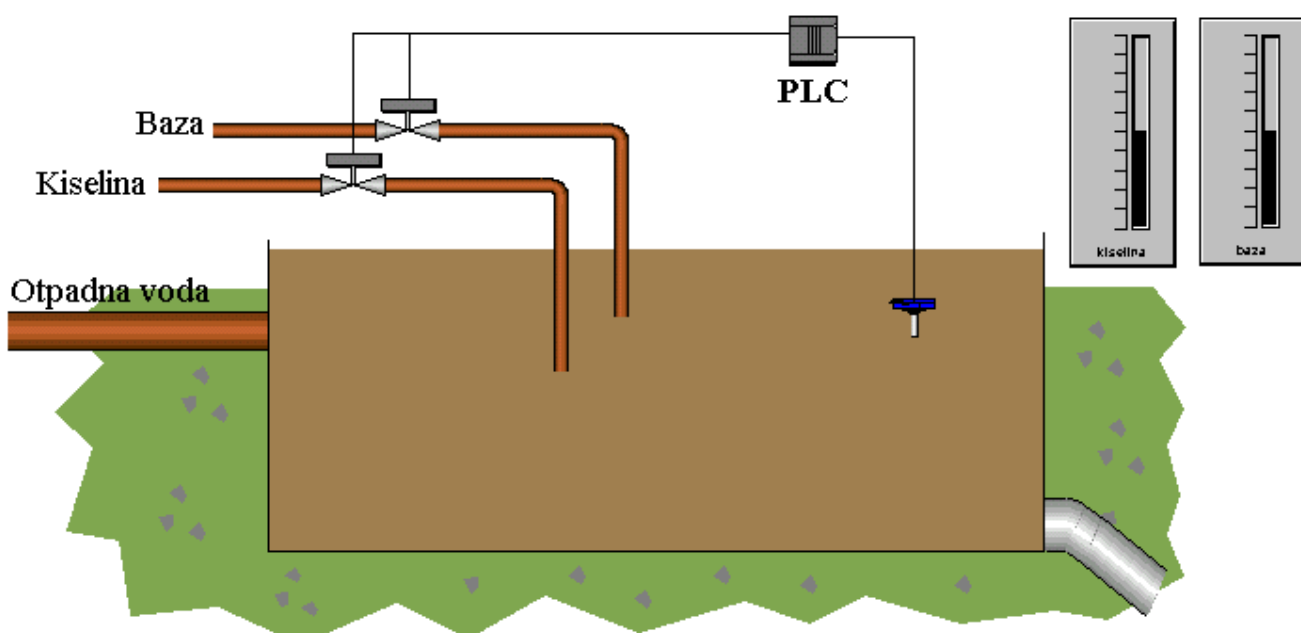
Ovim je urađen veći deo postupka programiranja. Da bi videli promene na ekranu treba da postavimo neke pokazivače za *baza_ventil* i *kiselina_ventil*, dok za *ph_vrednost* treba rešiti da ima uvek različitu vrednost.

Različitu vrednost za *ph_vrednost* rešavamo tako što desnim klikom na **Project Scripts** otvorimo **New Project Script...** I ovaj **Script** treba da se izvršava u jednakim vremenskim intervalima i da se izvršava svakih 10000 mili sekundi. **Script Code** izgleda ovako:

```
ph_vrednost=Rand(6)
ph_vrednost=ph_vrednost+4
```

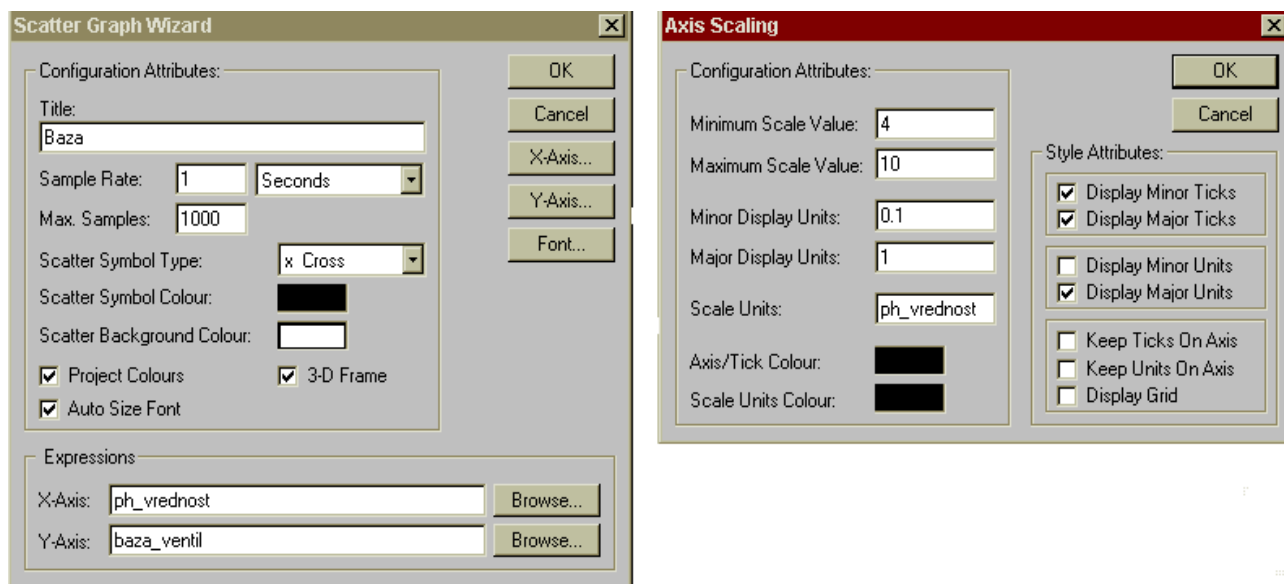
5. Da bi videli promene otvorenosti ventila za *baza_ventil* i *kiselina_ventil* možemo uzeti **Linear Gauge** dugme. Dvoklikom na nacrtani objekat otvoriće se prozor **Gauge Wizard**. U polje **Expression** upisujemo *baza_ventil*, minimalni i maksimalni opseg prikazivanja je od 0 do 1, kao što je definisano u **Point Editor**-u. Isto tako uradimo i za *kiselina_ventil*.

Za sada slika izgleda ovako:

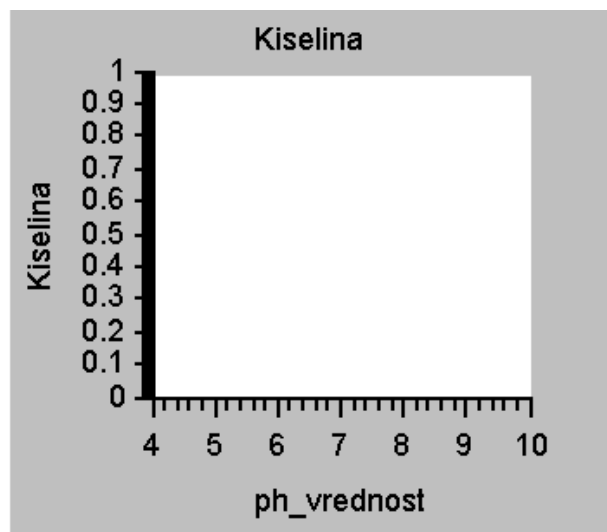
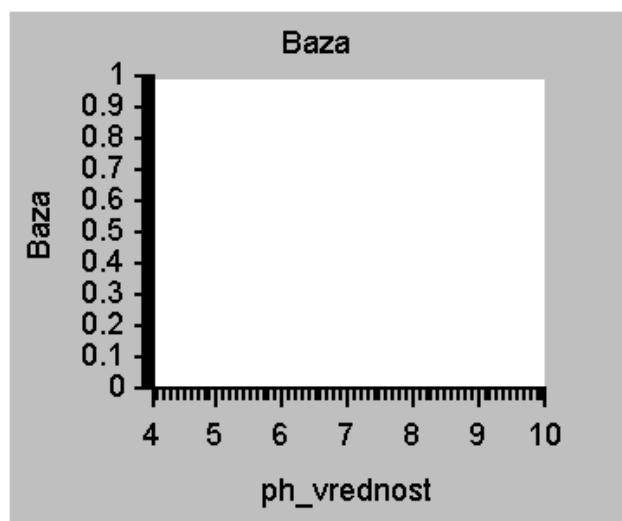


Ovako urađen zadatak funkcioniše, mada mi nemožemo videti kako se zapravo upravlja sa ventilima za kiselinu ili bazu. Promenljivu veličinu za *ph_vrednost* možemo prikazati jednostavno pomoću **Slider**-a, koji je povezan sa promenljivom *ph_vrednost* u granicama od 0 do 10.

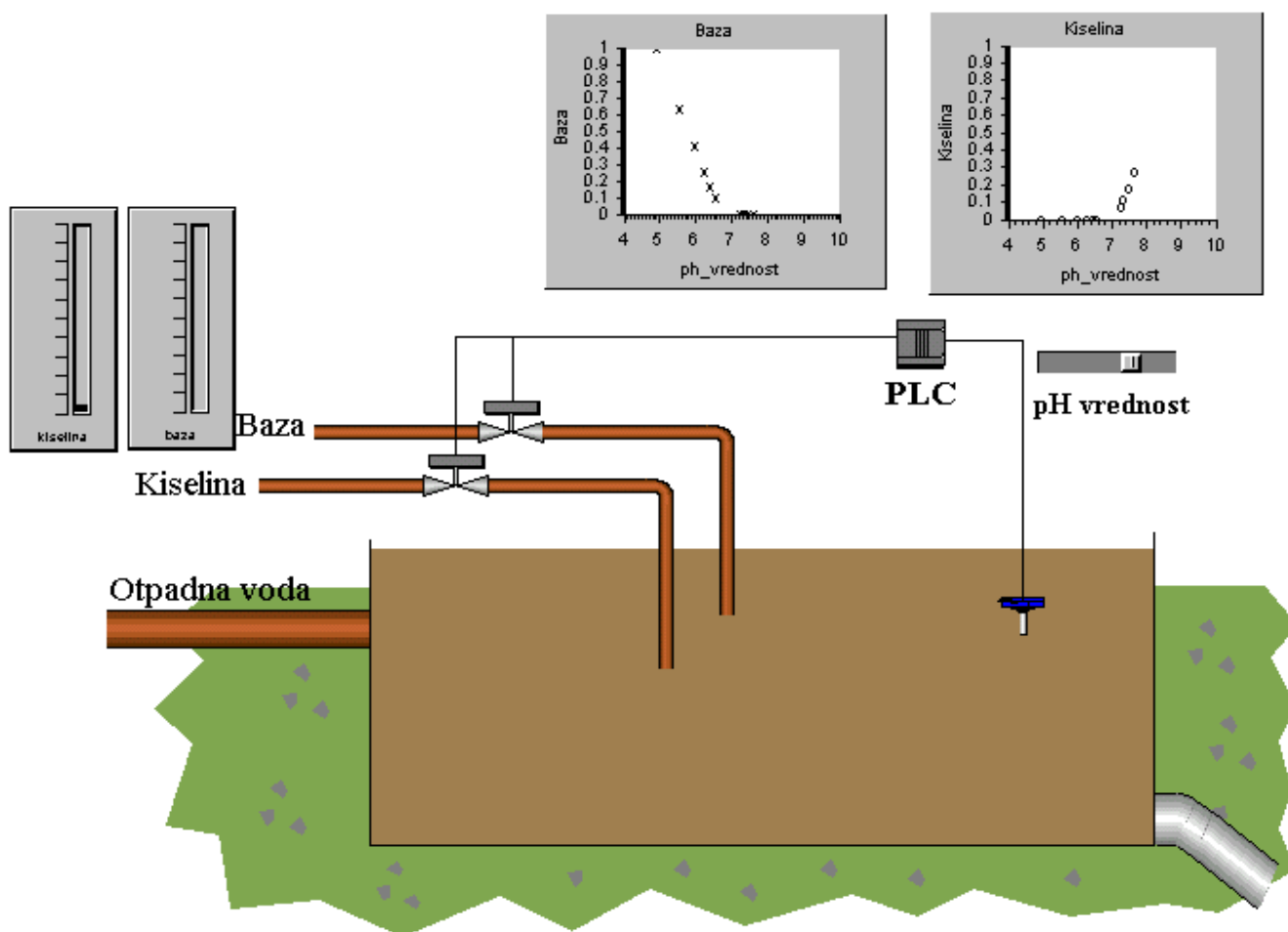
6.U tu svrhu dodajemo dijagrame za iscrtavanje baze i kiseline. Grafici se crtaju isto kao i ostali elementi za prikazivanje. Ovde je korišćena ikona **Scatter Graph** iz **Graphic Object Bar**-a. Dvoklikom na nacrtani element otvara se prozor **Scatter Graph Wizard** gde upisujemo šta da nam prikazuje na x i y osi, u kojim granicama da bude itd.



U ovom slučaju **Sample Rate** podesimo na 1 sekundu, **Max Samples** 1000, za opis (**Expressions**) *ph_vrednost* i *baza_ventil*. Granice prikazivanja podešavamo tako što pritisnemo dugme *Y-Axis...* (kiselina ili baza) ili *X-Axis...* (ph vrednost), u zavisnosti koju ćemo podesiti. Tada se otvara novi prozor koji terbapopuniti. Za kiselinu (*Y-Axis*) **Minimum Scale Value** je 0 **Maximum Scale Value** je 1 **Minor Display Units** je 0.01 **Major Display Units** je 0.1. Za ph vrednost (*X-Axis*) **Minimum Scale Value** je 4 **Maximum Scale Value** je 10 **Minor Display Units** je 0.2 **Major Display Units** je 1. Na isti ovakav način uradimo i za bazu. Dobijena dva dijagrama izgledaju ovako:



7. Krajnja slika izgleda ovako:

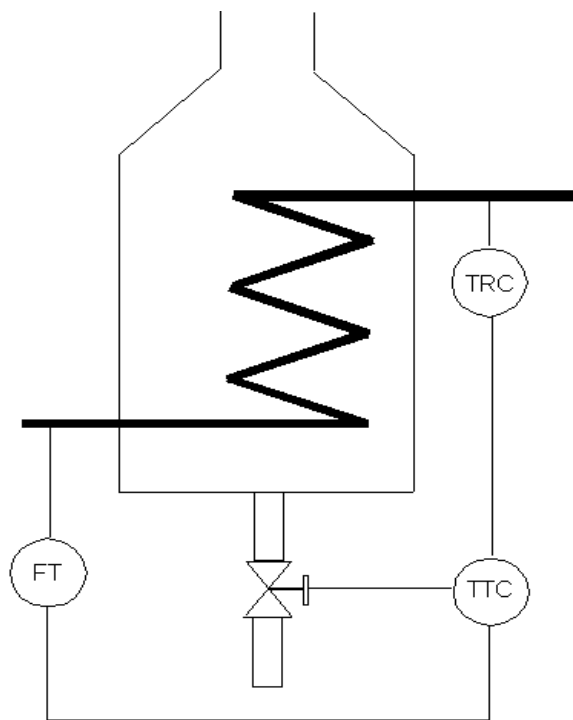


Kada pokrenemo ovaj projekat pritiskom na ikonu **Run the project** u **Toolbar**-u videćemo promene na graficima za bazu i kiselinu iste kao što je zadati upravljački signal.

15.3. Primer 3.

Rešiti problem upravljanja zagrevanja vode pomoću regulacionog kruga temperature u **CX-Supervisor**-u kao što je nacrtano na slici 14.4.

Ulazna veličina je protok koja nije zadata i treba je simulirati po želji. Ostale veličine takođe nisu zadate niti njihov odnos.



Slika 15.4. - Regulacioni krug temperature peći.

Temperatura na ulazu je konstantna što znači da temperatura u kotlu zavisi od otvorenosti ventila i od protoka. Ako otvorenost ventila veća temperatura opada, dok protok raste temperatura raste. Otvorenost ventila zavisi od protoka i od temperature tako što ako protok raste veća je otvorenost ventila, dok ako temperatura raste otvorenost ventila je manja.

Ovakvom analizom slike vezu između promenljivih veličina možemo napisati ovako:

$$\text{temperatura} = \text{const1} * \text{protok} - \text{const2} * \text{otvorenost_ventila}$$

$$\text{otvorenost_ventila} = \text{const3} * \text{protok} + \text{const4} * \text{temperatura}$$

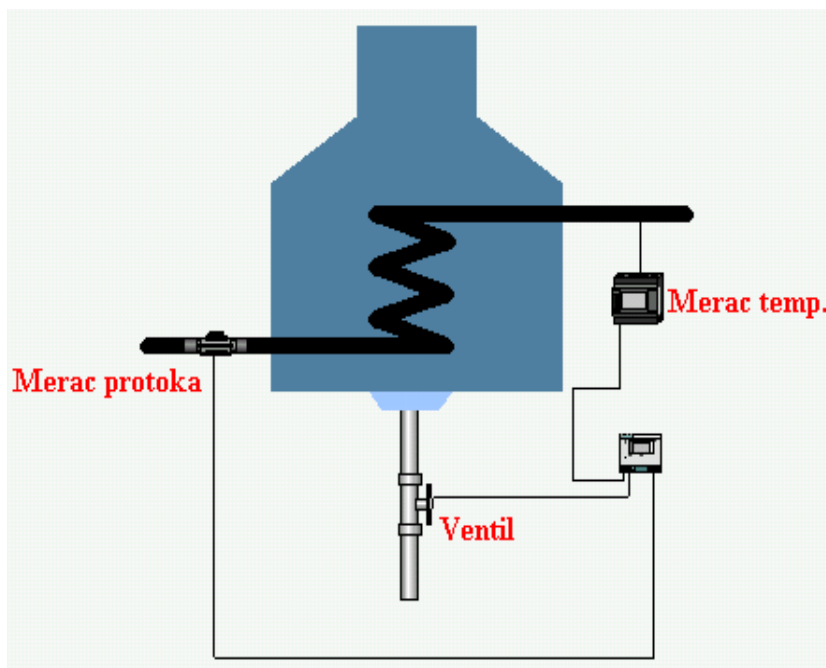
Sada treba rešiti promenljivi ulaz za protok. U slučaju da se uzme konstantan protok sistem će se ustaliti i neće biti potrebe za bilo kakvu regulaciju. Zato stavljamo sinusni ulaz koji izgleda ovako:

$$PROTOK = P_0 + \Delta P \cdot \sin\left(\frac{2 \cdot \pi \cdot t}{T}\right)$$

1. Otvorite novi projekat u **CX-Supervisor**-u. Napišite ime projekta i zadajte željenu putanju gde hoćete da ga snimate. Nakon toga pojaviće se nova prazna stranica, koju sada treba da podesite. Pre podešavanja izgleda nove stranice, snimate je u **File - Save Page As...**

2. Za crtanje ovog projekta potrebni su kotao, grejač, senzor protoka, ventil i senzor temperature. Potrebni elementi za crtanje se nalaze u **Graphic Object Bar** meniju i u **Graphic Library** biblioteci.

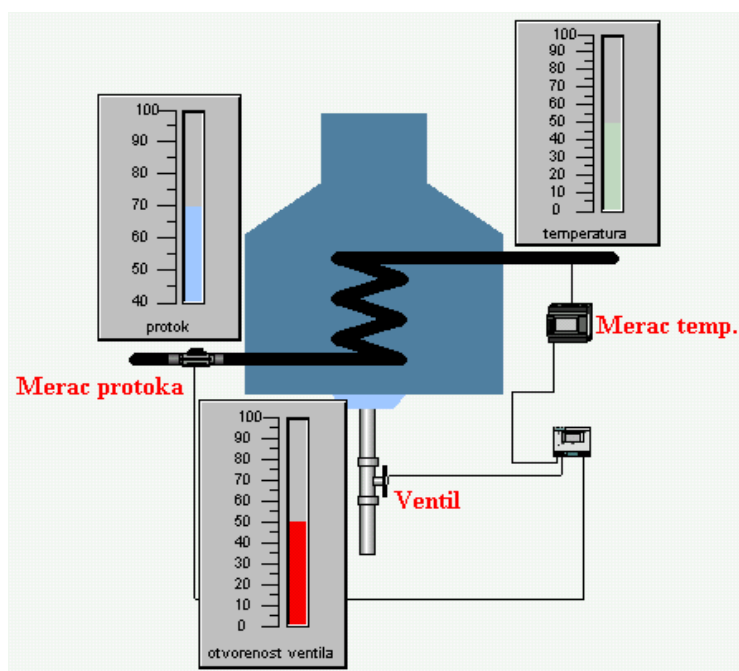
Nacrtana slika izgleda ovako:



3. Sada treba da definišemo tačke (**Points**) u **Point Editor**-u. Potrebne su sledeće tačke: *otvorenost_ventila*, *protok*, *temperatura*, *vreme* i *p*. Tačke *vreme* i *p* potrebni su za simulaciju sinusnog ulaza. Tačka *vreme* je celobrojna a ostale promenljive su realnog tipa.

4. Kada definišemo tačke možemo pisati **Script**. Otvorimo **Work Space** prozor i desnim klikom na **Project Scripts** biramo **New Project Script...**, koji otvara **Script Editor**. Ovaj projekat treba da se izvršava u jednakim vremenskim intervalima, tako i podesimo.

Script Code izgleda ovako:



- deo koji se odnosi na sinusni ulaz:

$$p = 2 * 3.1415926535 * \text{vreme} / 50$$

$$\text{protok} = \sin(p)$$

$$\text{protok} = 0.7 + \text{protok} * 0.2$$

- regulacioni deo:

$$\text{temperatura} = 100 * \text{protok} - 65 * \text{otvorenost_ventila}$$

$$\text{otvorenost_ventila} = 0.6 * \text{protok} + \text{temperatura} / 100$$

$$\text{vreme} = \text{vreme} + 1$$

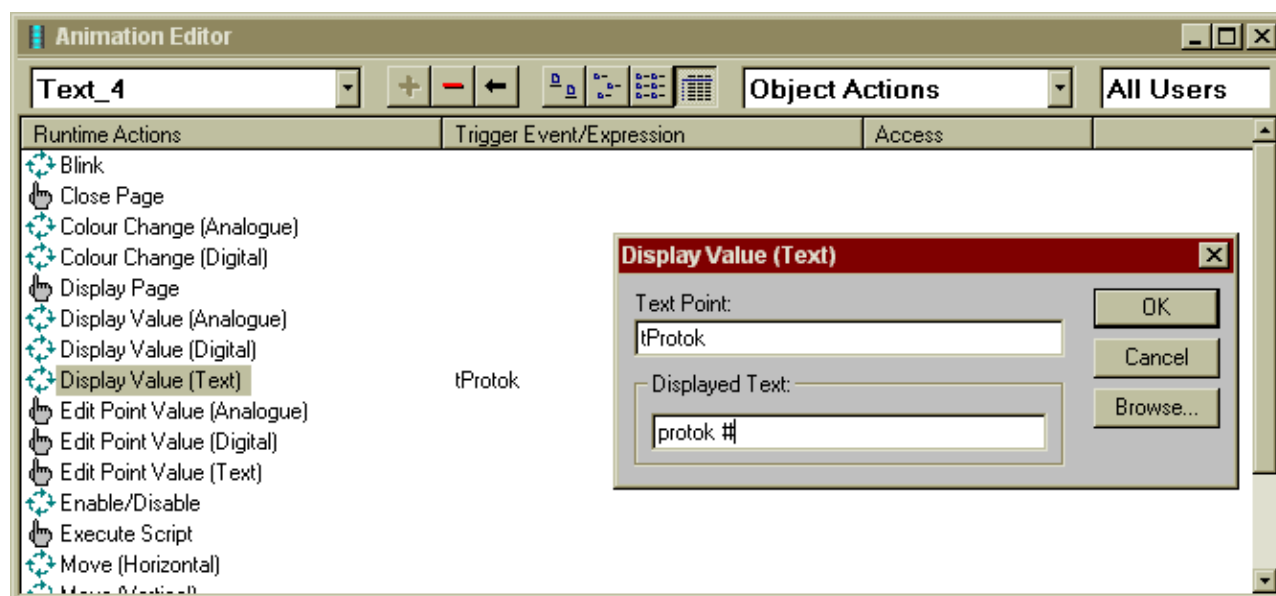
Konstante koje su uzete su sasvim proizvoljne.

5. Za prikazivanje promena temperature, otvorenosti ventila i protoka može se koristiti **Linear Gauge**. Pri popunjavanju ovih objekata treba voditi računa o opsezima prikazivanja.

6. Pokretanjem ovog projekta vide se promene na pokazivačima.

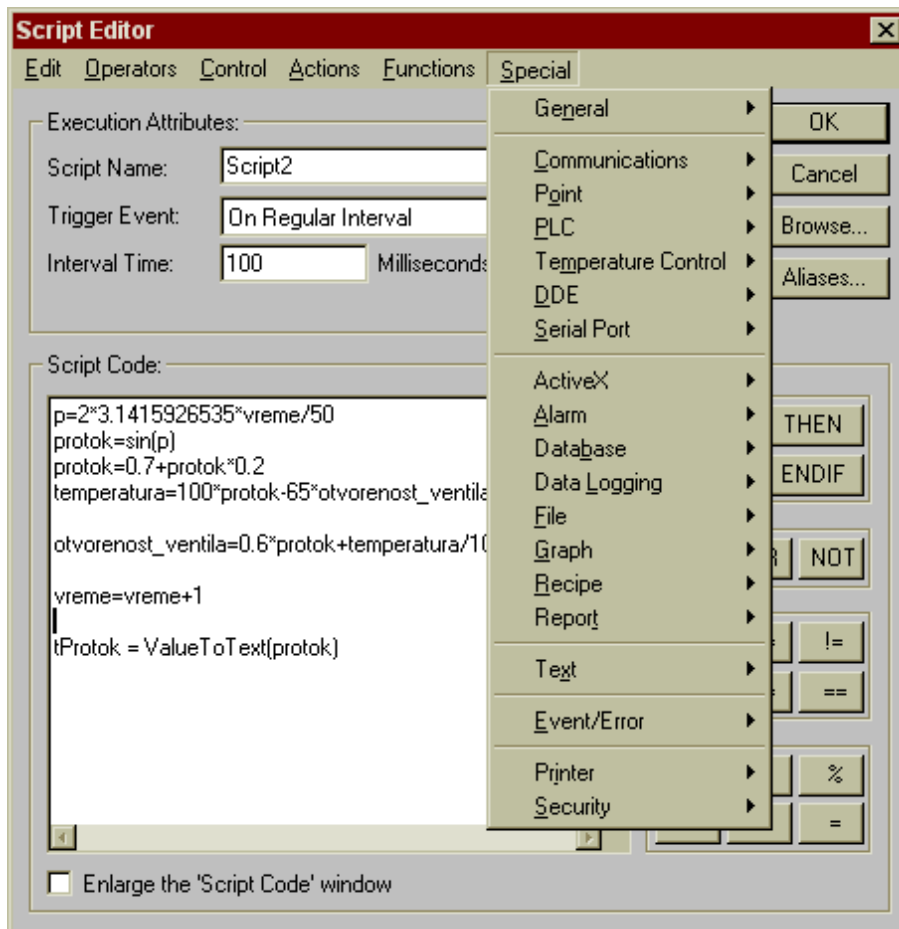
7. Kao novost u ovom zadatku možemo dodati ispisivanje vrednosti protoka kao broj. U tu svrhu koristi se skript funkcija koja se zove **ValueToText** i nalazi se u **Script Editor**-u u specijalnim funkcijama.

Kao prvo treba da napravimo novu tačku u **Point Editor**-u npr. **tProtok** koja je tekstulanog tipa i napisati neki tekst na radnu površinu gde će se vraćati vrednost protoka. Kada napišete neki tekst na otvoreni projekat recimo **protok**, kliknite mišem dva puta na taj tekst. Tada će se otvoriti **Animation Editor** za taj napisani tekst. U ovom prozoru su date neke mogućnosti od kojih treba izabrati dvoklikom **Display Value (Text)** nakon čega će se pojaviti mali prozor u koji upisujemo našu novu tačku **tProtok**.

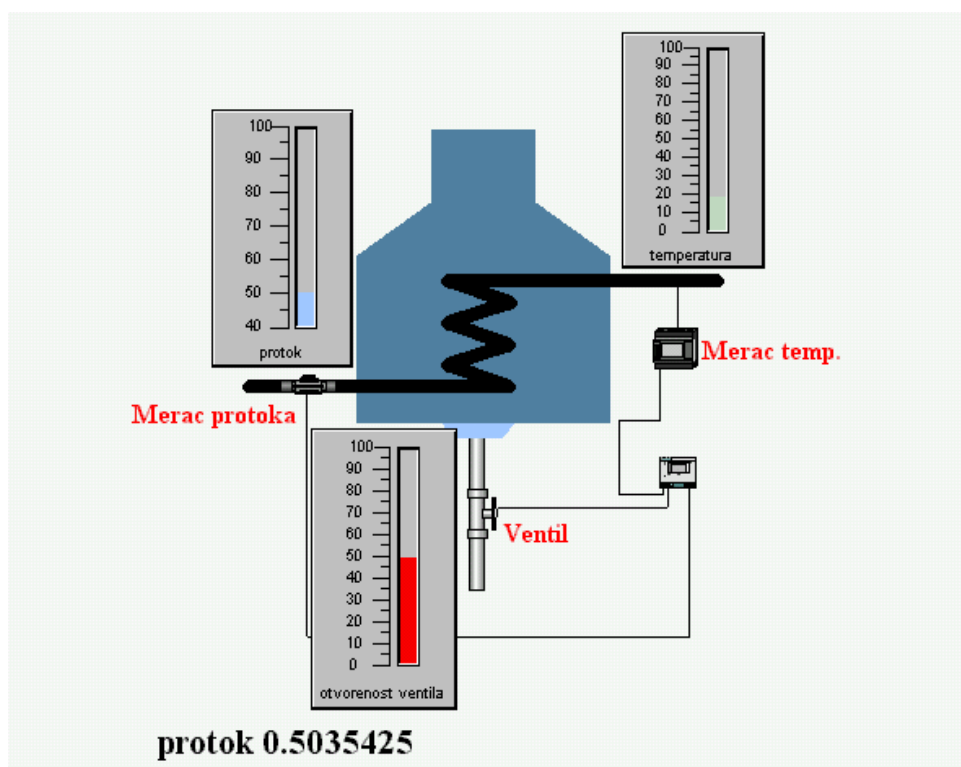


Sada još preostaje pisanje skripta. U već postojeći **Script Editor** prozor dodajemo sledeći red:

```
tProtok = ValueToText(protok)
```



8. Konačna slika samo se malo menja sa time što smo dodali stalno ispisivanje protoka.



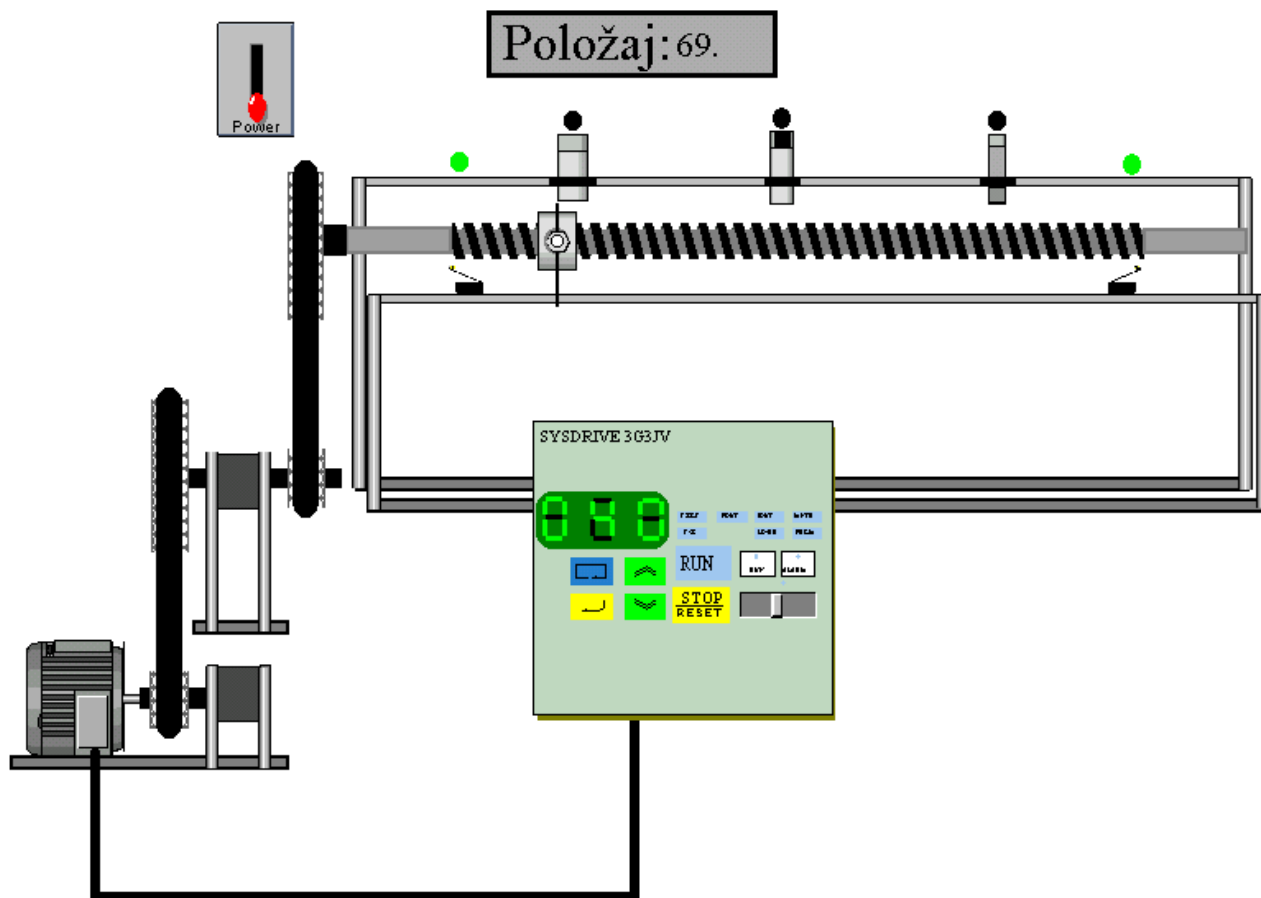
15.4. Primer 4.

Simulirati rad asinhronog motora koji pokreće spiralnu osovinu sa cilindrom. Sistem treba da se ponaša u skladu sa sledećim pravilima:

- Kada se uključi napajanje (pomoću "Power" prekidača) cilindar brzinom1 kreće levo.
- Kada stigne do skroz levog položaja (levi granični prekidač se aktivira) motor menja smer i time se menja smer kretanja cilindra (brzina ostaje ista).
- Posle prolaska ispod prvog senzora brzina se menja na brzinu2.
- Posle prolaska ispod drugog senzora brzina se menja na brzinu3.
- Posle prolaska ispod trećeg senzora brzina se menja na brzinu4.
- Kada stigne do skroz desnog položaja (desni granični prekidač se aktivira) motor menja smer i time se menja smer kretanja cilindra, a brzina se menja u brzinu1.
- Ako u bilo kom trenutku dođe do isključenja napajanja sistem se postavlja u početno stanje (smer kretanja u levo).

Potrebno je simulirati protok vremena, različite brzine kretanja cilindra, reakcije senzora i graničnih prekidača, kontrolu motora pomoću frekventnog pretvarača i menjanje položaja cilindra u zavisnosti od brzine i smera obrtanja motora.

Kreirati objekte koji će pokazivati trenutna stanja svih senzora, položaj, radnu frekvenciju pretvarača i prekidačkih elemenata.



Potrebno je kreirati excel dokument u koji ćemo logovati podatke. U prvi red upišite imena pointa čije vrednosti želite da logujete. Označite ćelije A1 i B1. U meniju "Insert/Name/Define" kreirajte "Name" koji će se zvati "tabela". Zatim kreirajte odgovarajući "file dsn" u Windows Control Panel-u u okviru ODBC podešavanja koji će ukazivati na kreirani excel dokument.

Zatim u okviru "Supervisora" kreirajte vezu prema tom dokumentu (preko dsn fajla) i unesite sva potrebna polja za potrebne pointe, na već navedeni način.

Primer dokumenta koji se dobija logovanjem u vremenskim intervalima od 5 sekundi:

	A	B	C
1	Položaj	Frekvencija	
2	150,00	0,00	
3	137,00	40,00	
4	111,00	40,00	
5	85,00	40,00	
6	59,00	40,00	
7	33,00	40,00	
8	7,00	40,00	
9	19,00	40,00	
10	46,00	40,00	
11	65,00	40,00	
12	93,00	80,00	
13	131,00	80,00	
14	173,00	80,00	
15	215,00	80,00	
16	264,00	120,00	
17	327,00	120,00	
18	387,00	120,00	
19	423,00	40,00	
20	443,00	40,00	
21	464,00	40,00	

Od ovih podataka možete kreirati grafik:

