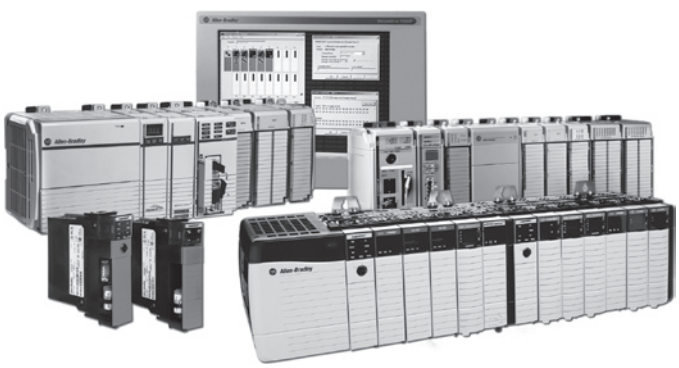
15
Kontroleri ControlLogix

Programabilni logički kontroleri nastavljaju svoju evoluciju uporedo s novim tehnologijama koje dopunjavaju mogućnosti kontrolera. PLC je počeo kao zamena za releje koji su uključivali i isključivali uređaje na izlazima ali su obavljali i funkcije tajmera i brojača. Postepeno su dodavane razne matematičke i logičke funkcije. Da bi zadovoljile rastuće potrebe savremenih industrijskih upravljačkih sistema, vodeće kompanije u oblasti automatike napravile su novu klasu industrijskih kontrolera nazvanih programabilni automatizovani kontroleri, ili PAC (engl. *programmable automation controler*). Po fizičkom izgledu slični su PLC kontrolerima, ali u istom programskom okruženju imaju ugrađene naprednije mogućnosti komunikacije i prikupljanja podataka, kao i mogućnosti obrade signala, upravljanja procesima i mašinske vizije.

Porodica programabilnih automatizovanih kontrolera proizvođača Allen-Bradley sastoji se od sistema ControlLogix, sistema CompactLogix, sistema FlexLogix, kontrolera SoftLogix 5800 i sistema DriveLogix. Suštinsku razliku između PLC i PAC uređaja čini softver. U osnovi, konfiguracija lestvičaste logike ostala je ista, ali je promenjen način adresiranaja naredaba. Naredni odeljci ovog poglavlja opisuju razne primene softvera upravljačke platforme kontrolera Logix. Pretpostavlja se da čitalac poznaje osnovne naredbe lestvičaste logike i funkcije (za bitove, tajmere i brojače) koje su opisane u prethodnim poglavljima ove knjige, pa se zato ne ponavljaju u ovom poglavlju.



Slika 15-1 Programabilni automatizovani kontroler (PAC).
Izvor: Slika je objavljena s dozvolom kompanije Rockwell Automation, Inc.

Deo I
Organizacija memorije i projekta

Ciljevi ovog dela

Pošto pročitate ovaj deo, moći ćete da:

- Opišete organizaciju projekta
- Definišete poslove, programe i rutine
- Identifikujete vrste datoteka za podatke
- Organizujete i upotebite razne vrste datoteka za podatke

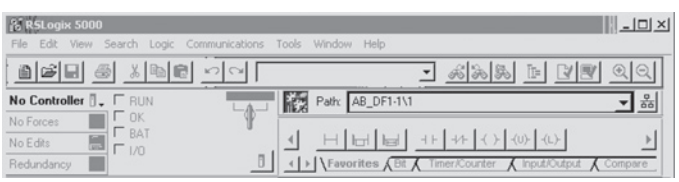
Struktura memorije

Procesori ControlLogix omogućavaju upotrebu fleksibilne strukture memorije. Memorija nije podeljena na fiksne oblasti koje su unapred dodeljene pojedinim vrstama podataka ili bitova U/I stanja. Internu organizaciju memorije kontrolera tipa ControlLogix određuje sam korisnik kada započne nov projekat pomoću softvera RSLogix 5000 (slika 15-2). Ta činjenica omogućava da se podaci s kojima program radi definišu tako da odgovaraju potrebama aplikacije, umesto da se aplikacija prilagođava određenoj strukturi memorije. Sistem ControlLogix (CLX) može činiti bilo šta od samostalnog kontrolera i U/I modula u zajedničkoj šasiji, do vrlo distribuiranog sistema, koji se sastoji od više šasija i mreža koje rade u sadejstvu.

Konfigurisanje

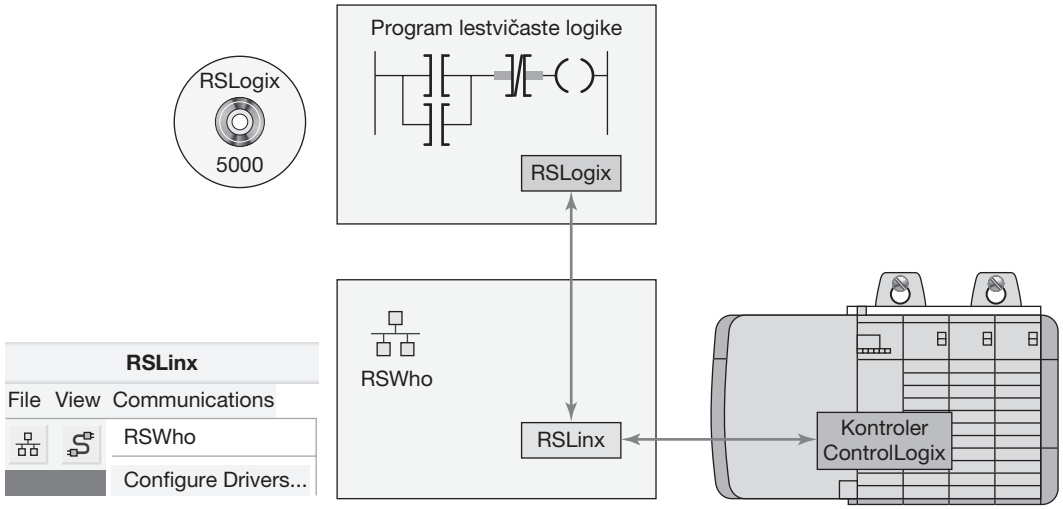
Konfigurisanje modularnog CLX sistema podrazumeva uspostavljanje komunikacione veze između kontrolera i procesa. Softver za programiranje mora da zna koji se CLX hardver koristi kako bi mogao da šalje ili prima podatke. Konfiguracioni podaci opisuju vrstu procesora i U/I modula koji se koriste u sistemu.

Softver za programiranje RSLogix 5000 omogućava zadavanje ili konfigurisanje strukture memorije kontrolera ControlLogix. Komunikacioni softver RSLinx služi za uspostavljanje komunikacione veze između softvera za programiranje RSLogix 5000 i ControlLogix hardvera, kao što je prikazano na slici 15-3. Da biste uspostavili komunikaciju s kontrolerom, treba napraviti odgovarajući upravljački program (drajver) pomoću softvera RSLinx. Drajver radi kao softverski interfejs prema hardverskom uređaju. Softver RSWho je interfejs za pretraživanje mreže koji omogućava prikazivanje u jednom prozoru svih drajvera koji su već konfigurisani u mreži.

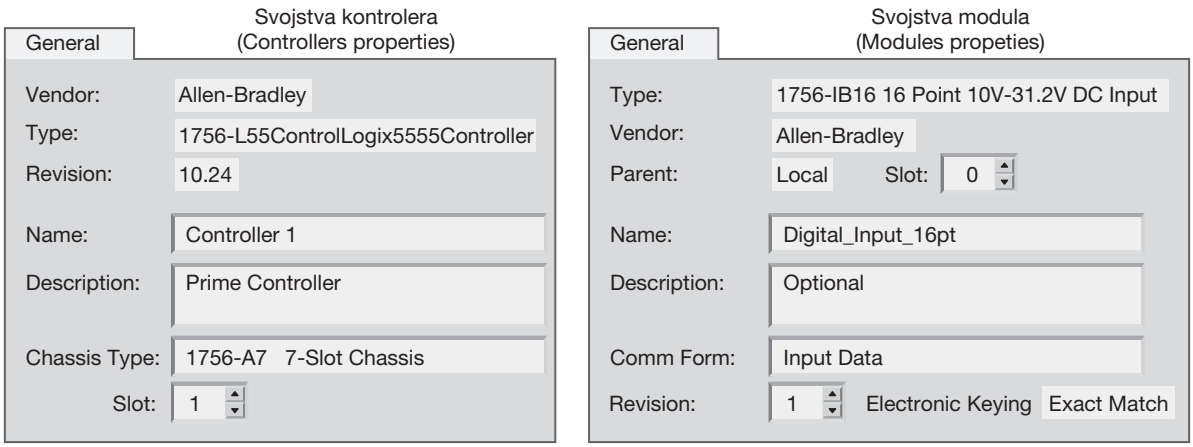


Slika 15-2 Glavni ekran paketa RSLogix 5000.

Slika 15-4 prikazuje primer okvira za dijalog koji kao sastavni deo postupka konfigurisanja sistema omogućava podešavanje svojstava kontrolera i svojstava modula. Prikazane vrednosti parametara su tipični primeri opštih podatka koji su potrebni. Pošto prvo podesite kontroler, u softveru RSLogix 5000 treba podesiti i U/I module. Moduli neće raditi ako nisu ispravno konfigurisani. Softver sadrži sve podatke o hardveru koji su mu potrebni za konfigurisanje svakog modula iz porodice ControlLogix.



Slika 15-3 Softverski paketi RSLinx i RSLogix.



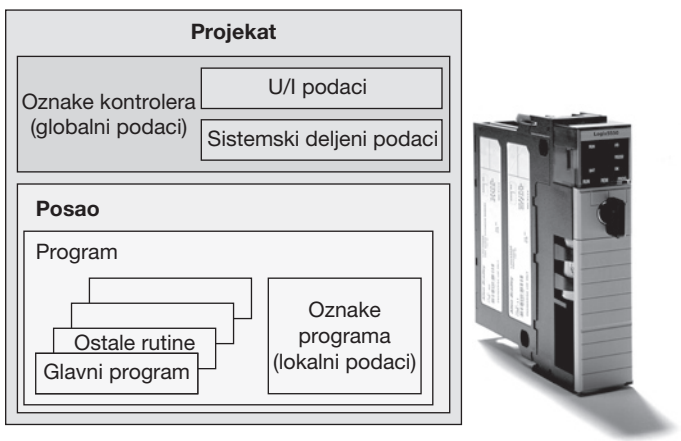
Slika 15-4 Okviri za dijalog koji omogućavaju podešavanje svojstava kontrolera i svojstava modula.

Projekat

RSLogix smešta programske i konfiguracione podatke u datoteku koja se zove projekat. Slika 15-5 prikazuje blok dijagram datoteke projekta. Datoteka projekta sadrži sve podatke koji pripadaju jednom projektu. Glavne komponente u datoteci projekta su poslovi (engl. *tasks*), programi i rutine. U svakom datom trenutku, kontroler može da učita i izvršava samo jedan projekat.

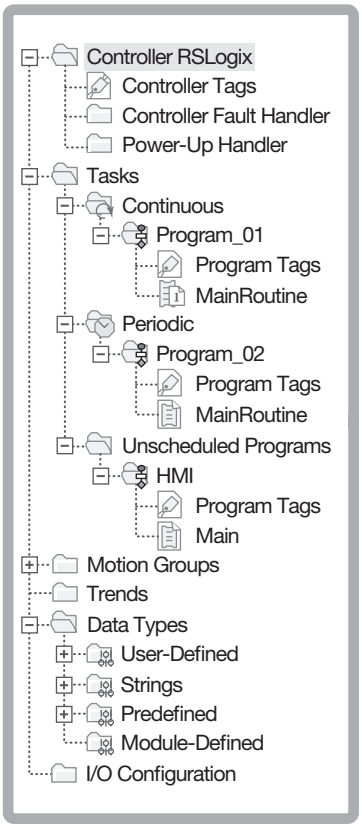
Organizaciono stablo u paketu RSLogix 5000 prikazuje organizaciju projekta u formatu stabla na kojem se vide poslovi, programi, rutine, tipovi podataka, trendovi, U/I konfiguracije i oznake. U svakom direktorijumu grupisane su zajedničke funkcije. Ta struktura pojednostavljuje navigaciju kroz projekat i ukupan prikaz celokupnog projekta.

Svakoj ikonici direktorijuma u projektu prethodi ikonica znaka plus ili znaka minus. Znak + znači da je direktorijum zatvoren. Pritisnite ga mišem da biste otvorili tu granu stabla i prikazali sadržaj direktorijuma. Znak



Slika 15-5 Datoteka programa za kontroler tipa ControlLogix.
Izvor: Slika je objavljena s dozvolom kompanije Rockwell Automation, Inc.

– pokazuje da je direktorijum već otvoren i vidi se njegov sadržaj. Kada pritisnete desni taster miša, otvaraju se razni priručni (kontekstni) meniji čiji sadržaj zavisi od



Slika 15-6 Organizaciono stablo kontrolera.

konteksta. Često ćete otkriti da je to prečica za pristupanje prozoru svojstava ili opcijama na glavnoj paleti menija.

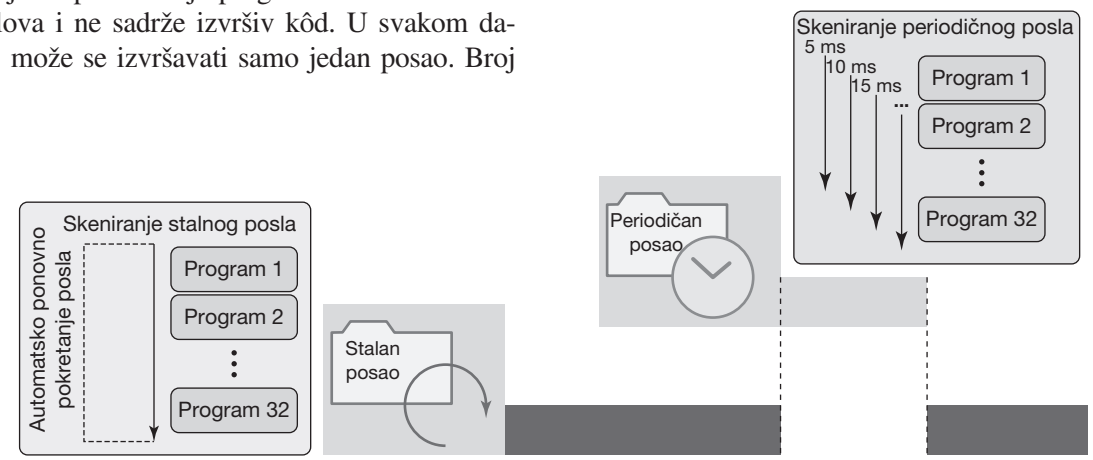
Poslovi

Poslovi (engl. *tasks*) prvi su nivo organizacije u projektu. Jedan posao se sastoji od kolekcije programa za izvršavanje. Kada se izvršava posao, izvršavaju se programi koji mu pripadaju, redosledom kojim su navedeni unutar posla. Lista programa zove se programski raspored. Poslovi omogućavaju raspoređivanje programa u zavisnosti od određenih uslova i ne sadrže izvršiv kôd. U svakom datom trenutku, može se izvršavati samo jedan posao. Broj poslova koji kontroler podržava zavisi od konkretnog kontrolera. Glavne vrste poslova (slika 15-7) jesu sledeće:

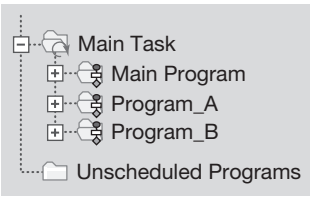
- *Stalni poslovi* se izvršavaju neprestano, ali uvek ih može prekinuti drugi periodičan posao. Stalni poslovi imaju najniži nivo prioriteta. Stalni posao kontrolera ControlLogix sličan je datoteci 2 na platformi SLC 500. U ovom slučaju stalni posao zove se *glavni posao* (engl. *Main Task*).
- *Periodični poslovi* deluju kao prekidi u određenim vremenskim intervalima. Oni prekidaju odvijanje stalnih poslova i izvršavaju se tokom perioda fiksnog trajanja u zadatim vremenskim intervalima.
- *Poslovi za obradu događaja* deluju na isti način kao prekidi. Umesto da se uvek pokreću u određenim vremenskim intervalima, poslovi za obradu događaja pokreću se kada se određeni događaj odigra ili ne odigra.

Programi

Programi su drugi nivo raspoređivanja programskog koda unutar projekta. Funkcija nivoa ispod Main Task jeste da definišu i odrede redosled kojim programi treba da se izvršavaju. Program ne sadrži izvršiv kôd. Rutine unutar programa izvršavaju se redosledom koji zavisi od posla kojem pripadaju u organizaciji kontrolera (slika 15-8). U primeru na slici, na osnovu prikazanog redosleda, vidi se da prvo izvršava Main Program (glavni program), drugi je Program_A, a treći Program_B. Programi koji nisu dodeljeni nijednom poslu (engl. *unscheduled tasks*) nisu raspoređeni za izvršavanje. Neraspoređene programe kontroler preuzima ali ih ne izvršava. Ti programi ostaju neraspoređeni dok ne zatrebaju. U zavisnosti od verzije softvera RSLogix 5000, jedan posao se može sastojati od najviše 100 programa.



Slika 15-7 Stalni i periodični poslovi.



Slika 15-8 Redosled izvršavanja programa.

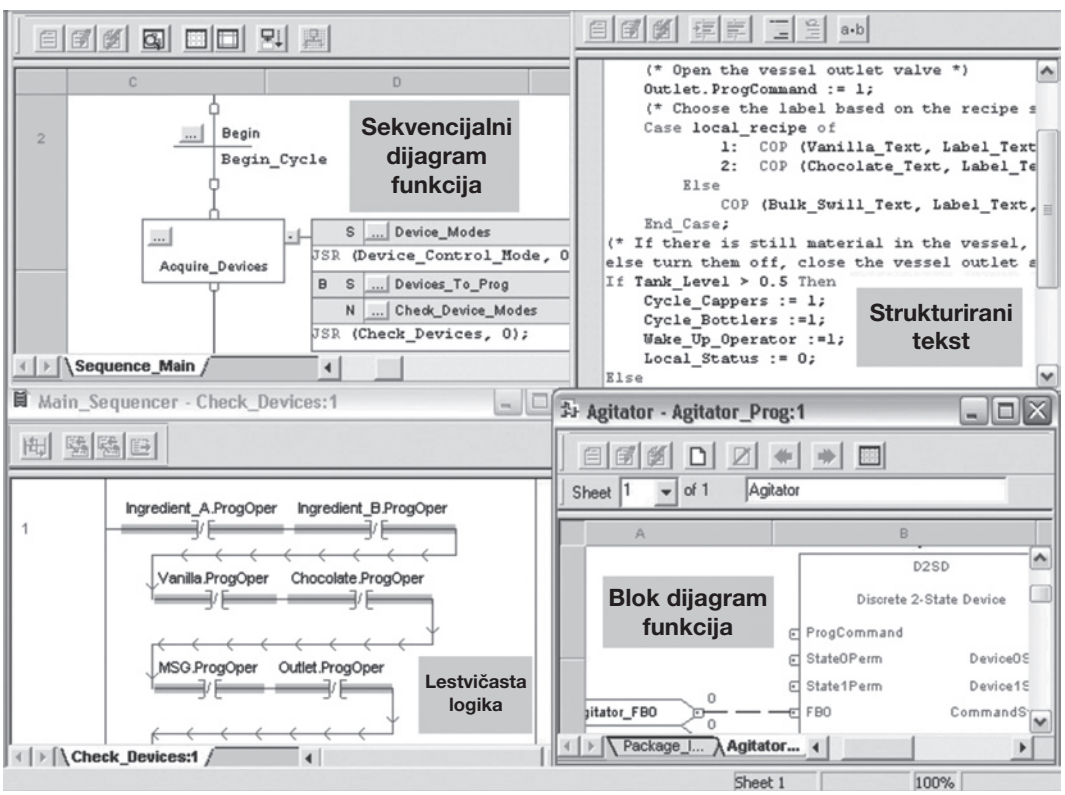
Rutine

Rutine su treći nivo organizovanja programskog koda unutar projekta i sastoje se od izvršivog koda. Svaka rutina sadrži skup logičkih elemenata na određenom programskom jeziku. Kada programer piše rutinu, može je zadati u obliku lestvičaste logike, sekvencijalnog dijagrama funkcija, blok dijagrama funkcija ili strukturiranog teksta (slika 15-9). Svaka rutina mora biti napisana u celini na istom jeziku. Broj rutina po projektu ograničen je samo količinom memorije u kontroleru. Možete praviti biblioteke standardnih rutina koje ćete koristiti na više mašina i u više aplikacija. Rutina se može definisati kao jedna od sledećih vrsta:

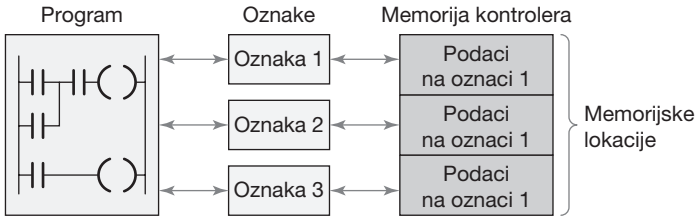
- *Glavna rutina* je ona koja je podešena tako da se prva izvršava kada se pokrene program. Svaki program najčešće ima jednu glavnu rutinu kojoj sledi više podrutina.
- *Podrutina* je rutina koju poziva druga rutina. Podrutine se koriste u obimnim i složenim programima ili u programskim projektima koji zahtevaju upotrebu više od jednog programskog jezika.
- *Rutina za obradu greške* izvršava se kada kontroler naiđe na grešku u programu. Svakom programu možete pridružiti jednu rutinu za obradu greške, ako smatrate da je to potrebno.

Oznake

Za razliku od konvencionalnih kontrolera, ControlLogix koristi adresnu strukturu u obliku oznaka (engl. *tag-based*). Oznake su opisna imena koja imaju smisao u aplikaciji, a ne samo generičke adrese. Oznaka se definiše da bi predstavljala određene podatke i da bi se njome obeležila oblast u memoriji gde se ti podaci nalaze. U aplikacijama razvijenim pomoću softvera RSLogix 5000, nema unapred definisanih tabela za podatke kao što su one za kontroler SLC 500. Kada u programu želite da upotrebite ili učitate određene podatke, treba da zadate ime oznake kojom je obeležena memorijska lokacija gde se ti podaci nalaze, kao što je ilustrovano na slici 15-10. Ta funkcija omogućava da podatke imenujete na osnovu njihovih funkcija u upravljačkom programu, čime se dobija samodokumentujuća logika. Kad god vam zatreba da grupišete podatke, formirajte niz, što je grupa oznaka sličnih tipova.



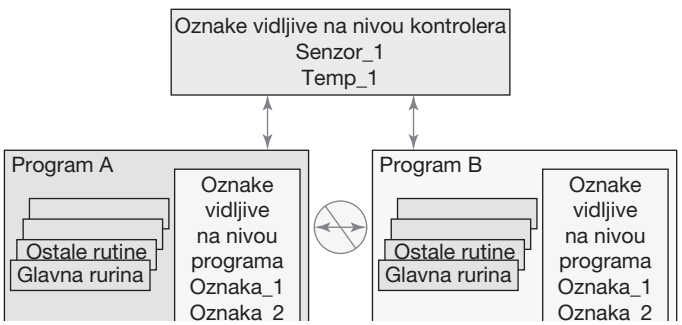
Slika 15-9 Svaka rutina se sastoji od skupa logičkih elemenata na određenom programskom jeziku.



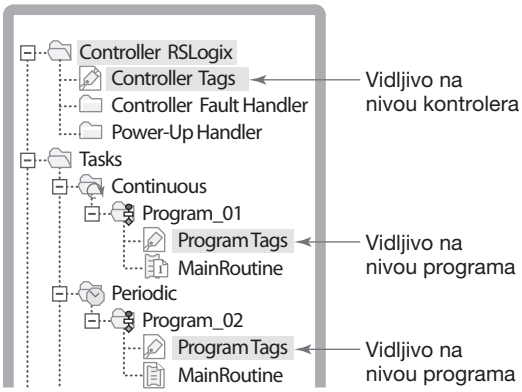
Slika 15-10 Oznake kojima su obeležene memorijske lokacije.

Opseg vidljivosti (engl. *scope*) definiše koji programi imaju pristup određenoj oznaci. Kada definišete oznaku, morate zadati i njen opseg vidljivosti. Oznaka može imati dve vrste opsega vidljivosti: na nivou programa i na nivou kontrolera. Oznaka koja je vidjiva na nivou programa sastoji se od podataka kojima mogu da pristupaju samo rutine koje pripadaju programu (lokalni podaci). Rutine koje pripadaju drugim programima ne mogu da pristupaju oznakama čiji je opseg vidljivosti program i pripadaju drugim programima. Oznaka koja je vidljiva na nivou kontrolera sastoji se od podataka kojima mogu da pristupaju sve rutine kontrolera (globalni podaci). Slika 15-1 prikazuje dva programa, A i B, unutar jednog projekta. Obratite pažnju na to da svaki program ima oznake čiji je opseg vidljivosti program i imaju jednaka imena (Oznaka_1, Oznaka_2, Oznaka_3). Pošto im je opseg vidljivosti program, nisu ni u kakvoj međusobnoj vezi, uprkos tome što imaju ista imena. Podacima čiji je opseg vidljivost program mogu da pristupaju samo rutine koje se nalaze unutar istog programa. Ista imena oznake mogu se pojavljivati u različitim programima kao lokalne promenljive zato što kada definišete oznaku, možete zadati njen opseg vidljivosti.

Opseg vidljivosti oznake morate deklarisati kada definišete oznaku. Slika 15-12 prikazuje oznake vidljive na nivou programa i na nivou kontrolera u obliku u kojem su se one prikazuju na organizacionom stablu kontrolera, u programu kojem pripadaju. U/I oznakama se automatski određuje opseg vidljivosti na nivou kontrolera.



Slika 15-11 Oznake sa opsegom vidljivosti na nivou programa i na nivou kontrolera.

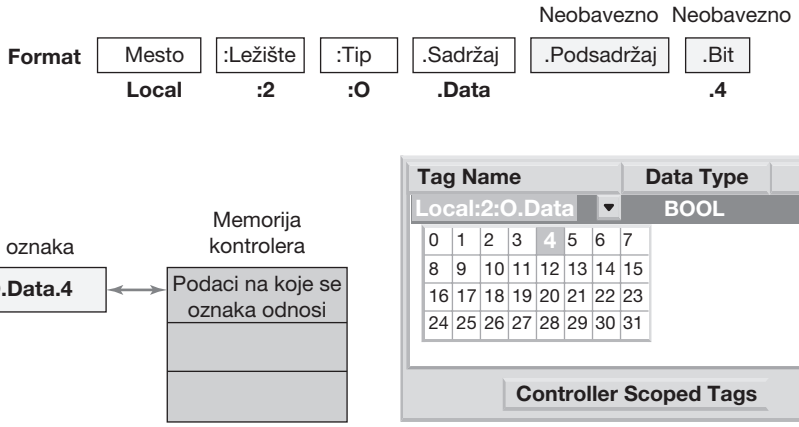


Slika 15-12 Listing oznaka vidljivih na nivou programa i na nivou kontrolera.

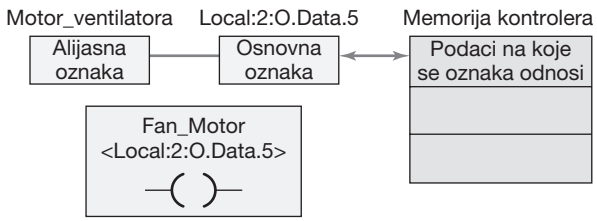
Postoje četiri vrste oznaka: osnovne, alijasi, predajne i prijemne. Vrsta oznake određuje kako će oznaka delovati unutar projekta. Osnovnom oznakom se obeležavaju razne vrste podataka koje koristi logika projekta. Ta oznaka definiše memorijsku lokaciju na kojoj su podaci smešteni. Upotreba osnovnih memorijskih oznaka zavisi od vrste podataka koje oznaka predstavlja. Slika 15-13 prikazuje osnovnu oznaku Local:2:O.Data.4, čiji je format sledeći:

Mesto	Mesto u mreži. LOCAL znači da se uređaj nalazi se u istoj šasiji kao kontroler.
Ležište	Broj ležišta U/I modula u šasiji.
Tip	I = ulazni O = Izlazni C = konfiguracioni S = statusni
Sadržaj	Konkretni podaci koji se tiču U/I modula; zavisi od vrste podataka s kojima modul radi.
Podsadržaj	Podaci koji su specifični za sadržaj.
Bit	Konkretan bit digitalnog modula; zavisi od veličine U/I modula (0–31 na modulu s 32 tačke).

Alijasna oznaka omogućava zadavanje alternativnog imena za postojeću oznaku. To je samo drugo ime za memorijsku lokaciju kojoj je već dodeljeno ime. Alijasna oznaka se može odnositi na osnovnu oznaku, drugi alijas, predajnu ili prijemnu oznaku. Alijasna oznaka se često koristi za definisanje oznake čije ime predstavlja ulaz ili izlaz iz stvarnog sveta. Slika 15-14 prikazuje upotrebu alijasne oznake. Pošto je alijasna oznaka (Motor_ventilatora) vezana za osnovnu oznaku (<local:2:O.Data.5>), svaka akcija koja se odvija nad osnovnom oznakom odvija se i nad njenim alijasom, i obrnuto. Alijasno ime je



Slika 15-13 Osnovna oznaka.



Slika 15-14 Alijasna oznaka koja je pridružena osnovnoj oznaci.

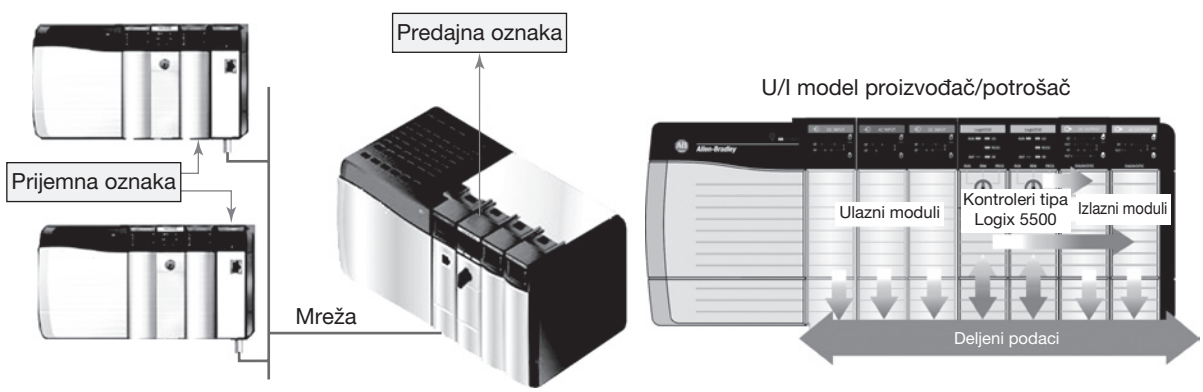
razumljivije i lakše se povezuje s elementom u aplikaciji, dok osnovna oznaka opisuje fizičko mesto tačke u šasiji ControlLogix sistema.

Predajne/prijemne oznake koriste za deljenje podataka između dva ili više umreženih uređaja. Predajna oznaka šalje podatke, dok prijemna oznaka prima podatke. Predajne oznake uvek imaju opseg vidljivosti na nivou kontrolera. Slika 15-15 prikazuje primer kako kontroler može da generiše podatke i šalje ih preko mreže na druga dva kontrolera koji primaju podatke. Na predajnom kontroleru definisana je oznaka predajnog tipa, dok je na prijemnim kontrolerima definisana oznaka s istim imenom, ali prijemnog tipa.

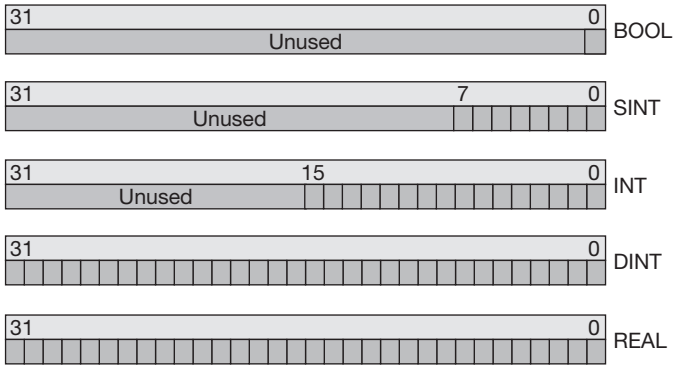
Kada projektujete aplikaciju, treba da je konfigurišete tako da podatke predaje globalno drugim kontrolerima u sistemu preko pozadinske sabirnice, kao i da prima oznake od drugih kontrolera. Ta odlika omogućava da za svaki kontroler odredite podatke koje će slati i podatke koje će primati. Slično tome, više kontrolera se mogu povezati na jedan predajni izvor podataka, što čini nepotrebnim slanje više poruka koje sadrže iste podatke.

Kontroleri tipa Logix rade s 32-bitnim rečima. Vrste podataka koje mogu nositi oznaku osnovnog tipa su BOOL, SINT, INT, DINT i REAL, kao što je ilustrovano na slici 15-16 i opisano u nastavku teksta. Kontroler smešta sve podatke na najmanje 4 bajta ili 32 bita podataka.

- **BOOL** je osnovna oznaka za podatke tipa Boolean koja smešta 1 bit podataka u bit 0 na memorijskoj lokaciji koja zauzima četiri bajta. Ostali bitovi, od 1 do 31 ne koriste se. Vrednost podataka tipa BOOL kreće se u opsegu od 0 do 1, to jest predstavlja stanje uključeno ili isključeno.
- **SINT** (Single Integer) koristi osam bitova memorije a podatke smešta u bitove od 0 do 7. Ti bitovi se ponekad nazivaju donji bajt. Ostali bitovi, od 8 do 31 ne koriste se.



Slika 15-15 Upotreba predajnih/prijemnih oznaka za deljenje podataka.
Izvor: Slika je objavljena s dozvolom kompanije Rockwell Automation, Inc.



Slika 15-16 Vrste podataka na koje se može primeniti osnovna oznaka.

- **INT** (Integer) – osnovna oznaka koja koristi 16 bitova (od 0 do 15), koji se ponekad nazivaju donji bajtovi. Bitovi od 16 do 31 se ne koriste.
- **DINT** (Double Integer) – osnovna oznaka koja koristi 32 bita, tj. sva četiri bajta, a opseg vrednosti je sledeći: od -2^{11} do 2^{11} (od -2.147.483.648 do 2.147.483.647).
- **REAL** je osnovna oznaka koja takođe zauzima 32 bita na memorijskoj lokaciji, a opseg vrednosti je definisan u skladu sa IEEE standardom za arhitekture s pokretnim zarezom.

Strukture

Postoji još jedna klasa podataka koja se zove struktura. Oznaka tipa struktura grupiše razne druge vrste podataka koje treba da se tretiraju kao celina i koriste se za određenu namenu. Slika 15-17 prikazuje primer strukture u paketu RSLogix. Svaki element strukture zove se član strukture, a svaki član može imati različit tip podataka.

Kontroler ControlLogix podržava tri vrste struktura: unapred definisane strukture, strukture za module i strukture koje definiše korisnik. Kontroler generiše unapred definisane strukture za tajmere, brojače, poruke i PID vrste. Slika 15-18 prikazuje primer unapred definisane strukture za brojačku naredbu, a članovi strukture su podešena

	Name	Data Types		Style	Description
	PRE	DINT		Decimal	
	ACC	DINT		Decimal	
	EN	BOOL		Decimal	
	TT	BOOL			
Članovi	DN	BOOL	Tipovi podataka		
	FS	BOOL		Decimal	
	LS	BOOL		Decimal	
	OV	BOOL		Decimal	
	ER	BOOL			

Slika 15-17 Oznaka tipa struktura.

Data type : COUNTER

Name Counter

Description

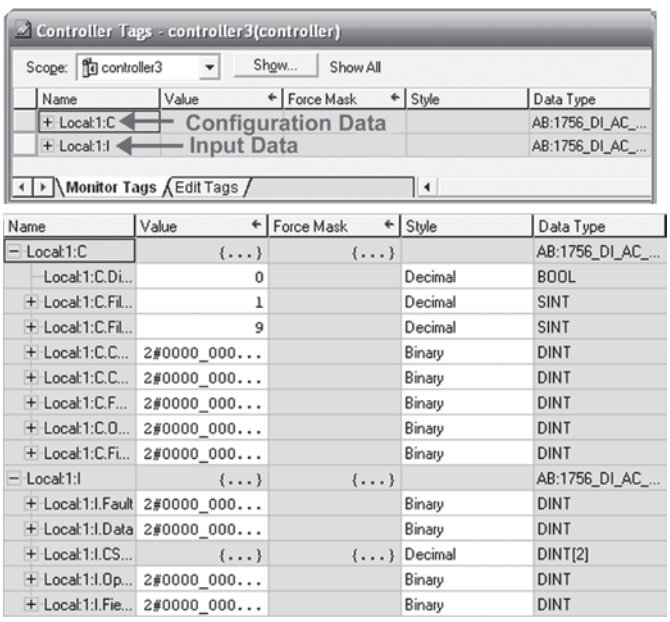
Members **Data type size : 12 byte(s)**

	Name	Data Type	Style	Description
	PRE	DINT	Decimal	
	ACC	DINT	Decimal	
	CU	BOOL	Decimal	
	CD	BOOL	Decimal	
	DN	BOOL	Decimal	
	OV	BOOL	Decimal	
	UN	BOOL	Decimal	

Slika 15-18 Unapred definisana struktura.

vrednost brojača, akumulirana vrednost i statusni bitovi naredbe.

Strukture za module prave se automatski kada se u sistemu konfigurišu U/I moduli. Kada dodate ulazni ili izlazni modul, skupu oznaka kontrolera automatski se dodaje više definisanih oznaka. Slika 15-19 prikazuje dve oznake (Local:1:C i Local:1:1) koje su generisane nakon dodavanja u sistem digitalnog ulaznog modula. Oznake te vrste generišu se radi čuvanja izlaznih, ulaznih i konfiguracionih podataka o modulu. Ulazne oznake obeležene s Data sadrže bitove koji predstavljaju ulaze modula. Oznake obeležene s Configuration određuju karakteristike i način rada modula. Ime Local znači da se te oznake nalaze



Slika 15-19 Struktura za digitalni ulazni modul.

Name: **Tank** Size: 16 byte(s)

Description: Generic Storage Tank Data Type

	Name	Data Type	Style	Description
	Level	INT	Decimal	Stores the Level in Inches
	Pressure	DINT	Decimal	Stores the Pressure in PSIG
	Temp	REAL	Float	The Temperature in F
	Agitator_Speed	DINT	Decimal	Speed in RPM
*				

Slika 15-20 Korisnička struktura koja predstavlja rezervoar.

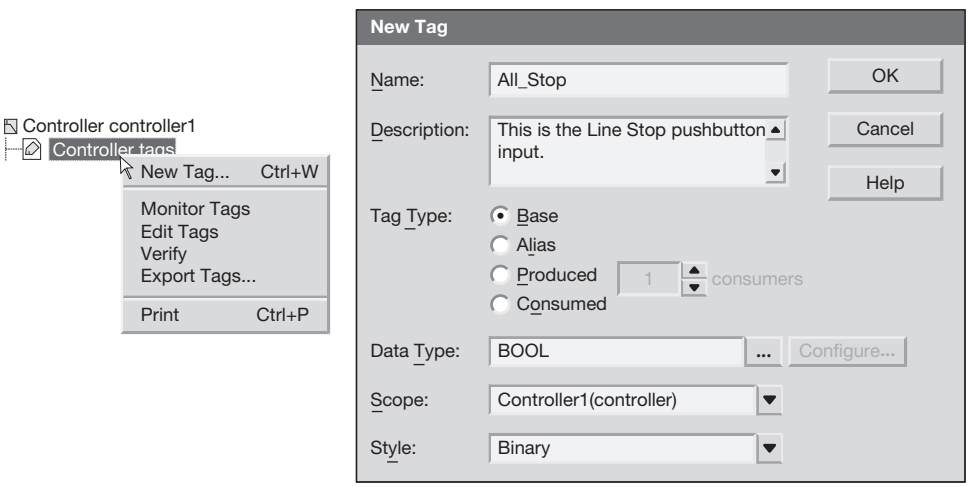
u istom reku kao procesor. 1 znači da je modul utaknut u ležište 1 u šasiji. Slova I i C pokazuju da li su u pitanju ulazni podaci ili konfiguracioni podaci.

Struktura koju definiše korisnik dopunjava unapred definisane strukture jer omogućava definisanje namenskih struktura radi čuvanja i obrade određenih podataka kao grupe. Slika 15-20 prikazuje korisničku strukturu koja sadrži podatke o rezervoaru. Svi podaci koji se odnose na rezervoar čuvaju se kao jedna celina. U fazi projektovanja programa, programer definiše generičku namensku strukturu koja sadrži sve aspekte što opisuju rezervoar. Svaki član strukture ima smisleno ime i definiše se sa odgovarajućim tipom podataka i stilom, kao što je REAL (brojevi s pokretnim zarezom) za temperaturu i DINT (decimalne vrednosti) za brzinu mešača izraženu kao broj obrtaja u minutu. Osoblje koje je zaduženo za instaliranje i održavanje sistema lako može da pronađe sve podatke koji se odnose na rad rezervoara pošto se svi podaci čuvaju kao jedna celina.

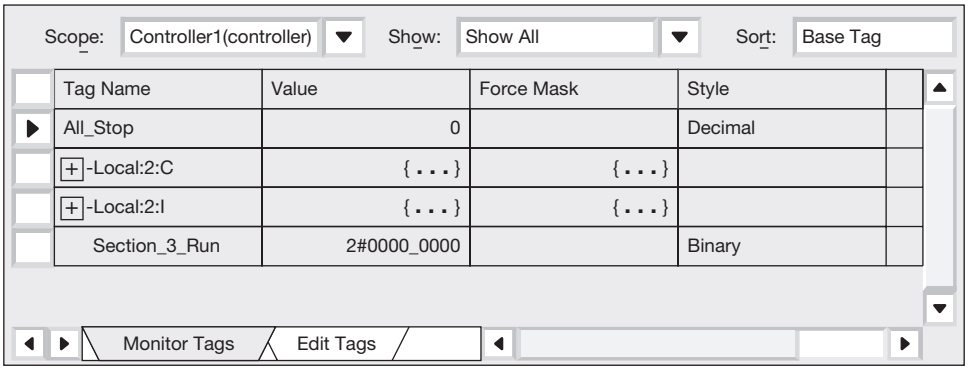
Definisanje oznaka

Oznake možete definisati na više načina. Možete ih definisati u editoru za oznake pre nego što unesete program, upisati imena oznaka u program, ili možete upisati znakove pitanja (?) na mestu imena oznaka, a oznake dodati kasnije. Slika 15-21 prikazuje primer okvira za dijalog New Tag u kojem je definisana osnovna oznaka čiji je opseg vidljivosti kontroler. Kada definišete oznake, treba da zadate sledeće podatke:

- *Ime oznake*, koje mora da počinje alfabetskim znakom ili podvlakom (_). Imena mogu sadržati samo alfabetske znake, cifre ili podvlake, a mogu sadržati najviše 40 znakova. Ne mogu sadržati više susednih podvlaka unutar ili na kraju imena, ne pravi se razlika između malih i velikih slova i nisu dozvoljeni razmaci u imenu oznake.
- *Opis oznake*, koji nije obavezan, može sadržati najviše 120 znakova.
- *Tip oznake*: osnovna, alijasna, predajna ili prijemna.
- *Tip podataka*, koji može biti jedan od unapred definisanih ili namenski koji je korisnik definisao.
- Opseg vidljivosti oznake. To može biti kontroler ili jedan od postojećih programa.
- *Stil prikazivanja* koji određuje oblik u kojem će se oznaka prikazivati u softveru za programiranje kontrolera. Softver nudi listu raspoloživih stilova.
- Da li želite ili ne želite da oznaka bude na raspolaganju drugim kontrolerima i broj drugih kontrolera koji mogu da koriste oznaku.



Slika 15-21 Osnovna oznaka čiji je opseg vidljivosti kontroler.



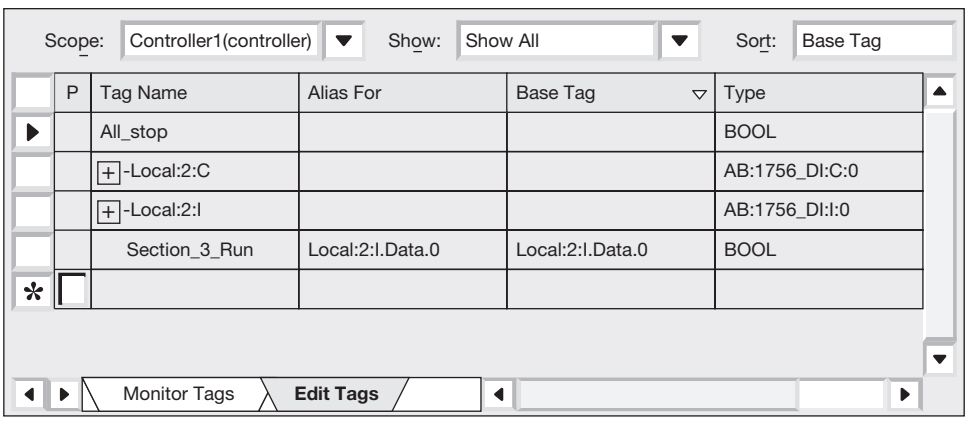
Slika 15-22 Prozor Monitor Tags.

Pregledanje i ažuriranje oznaka

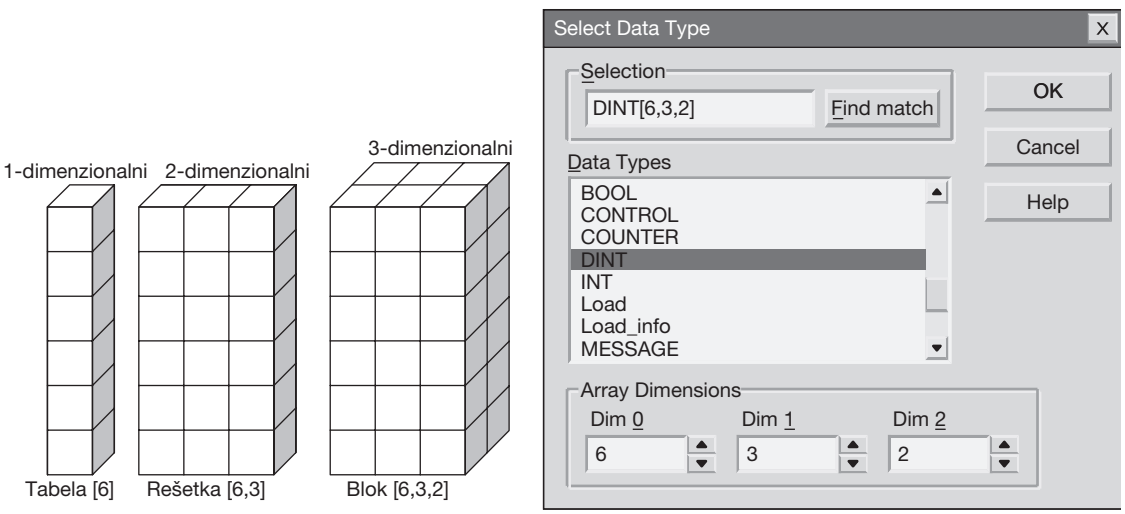
Pošto definišete oznake, možete pregledati i pratiti njihove vrednosti u prozoru Monitor Tags (slika 15-22). Kada izaberete jezičak Monitor Tags, prikazuju se tekuće vrednosti u oznakama.. Kolona Force Mask omogućava da nametnete stanje ulazima i izlazima kada tražite greške. Prozor Edit Tag (slika 15-23) omogućava definisanje novih oznaka i menjanje postojećih. Izaberite jezičak Edit Tags ako treba da dodate nove oznake ili da izmenite postojeće.

Nizovi

Mnogim upravljačkim programima protrebna je mogućnost smeštanja blokova podataka u memoriju u obliku tabela kojima se može pristupati u trenutku izvršavanja programa. Niz (engl. *array*) jeste vrsta oznake koja sadrži



Slika 15-23 Prozor Edit Tags.



Slika 15-24 Vrste nizova.

blok s više elemenata podataka. Svaki element niza mora biti istog tipa podataka (npr. BOOL, SINT ili INT). Niz zauzima blok susednih lokacija u memoriji. Nizovi su slični tabelama podataka. Upotreba nizova pruža najbolje iskorišćenje propusne moći procesora kontrolera ControlLogix. Budući da su nizovi grupe oznaka za iste podatke koji zauzimaju susedne lokacije u memoriji, ta činjenica omogućava efikasno učitavanje velikih količina podataka. Nizovi mogu imati jednu, dve ili tri dimenzije (slika 15-24) kako bi predstavili podatke koje treba da sadrže.

Jedna oznaka unutar niza predstavlja jedan element niza. Taj element može biti prost tip podataka ili struktura. Indeks elemenata niza počinje od 0 i raste do ukupnog broja elemenata manje jedan. Slika 15-25 prikazuje primer rasporeda memorije za jednodimenzionalni niz koji je definisan tako da sadrži pet temperatura. Ime oznake je Temp, a niz se sastoji od pet elemenata numerisanih od 0 do 4.

Niz - Temp
Tip podataka - INT[5]

Temp[0]	297
Temp[1]	200
Temp[2]	180
Temp[3]	120
Temp[4]	100

Slika 15-25 Raspored memorije za jednodimenzionalni niz.

PITANJA KOJA SE ODNOSE NA GRADIVO IZ DELA I

1. Uporedite konfiguraciju memorije kontrolera ControlLogix 5000 s onom u kontroleru SLC 500.
2. Šta sadrži projekat?
3. Navedite četiri programske funkcije koje se mogu obaviti pomoću organizacionog stabla programa.
4. Objasnite funkciju poslova u projektu.
5. Navedite tri glavne vrste poslova.
6. Koje vrste poslova deluju kao prekidi u određenim vremenskim intervalima?
7. Objasnite funkciju programa unutar projekta.
8. Objasnite funkciju rutina unutar projekta.
9. Koja se rutina konfiguriše tako da se prva izvršava?
10. Navedite četiri vrste programskih jezika koji omogućavaju programiranje kontrolera ControlLogix 5000.
11. Čemu služe oznake?
12. Uporedite opsege vidljivosti oznaka na nivou programa i na nivou kontrolera.
13. Navedite vrstu oznake za svaku od sledećih namena:
    a. Definisanje alternativnog imena za oznaku.
    b. Deljenje podataka u mreži.
    c. Čuvanje raznih vrsta podataka.
14. Koja je razlika između predajne i prijemne oznake?
15. Navedite pet osnovnih vrsta podataka koje se mogu obeležiti oznakama.
16. Navedite vrstu podataka koju biste upotrebili za sledeće:
    a. Smeštanje 32-bitnog podatka u memoriju.
    b. Stanje prekidača uključen/isključen.
    c. Smeštanje 16-bitnog podatka u memoriju.
    d. Smeštanje 8-bitnog podatka u memoriju.
17. Opišite od čega se sastoji unapred definisana struktura.
18. Opišite od čega se sastoji struktura za modul.
19. Opišite od čega se sastoji struktura koju korisnik definiše.
20. Objasnite dva načina definisanja oznaka.
21. Kada definišete oznake, koja ograničenja važe za imena oznaka?
22. Šta znači stil prikazivanja oznake?
23. Napišite primer niza koji omogućava čuvanje četiri brzine.