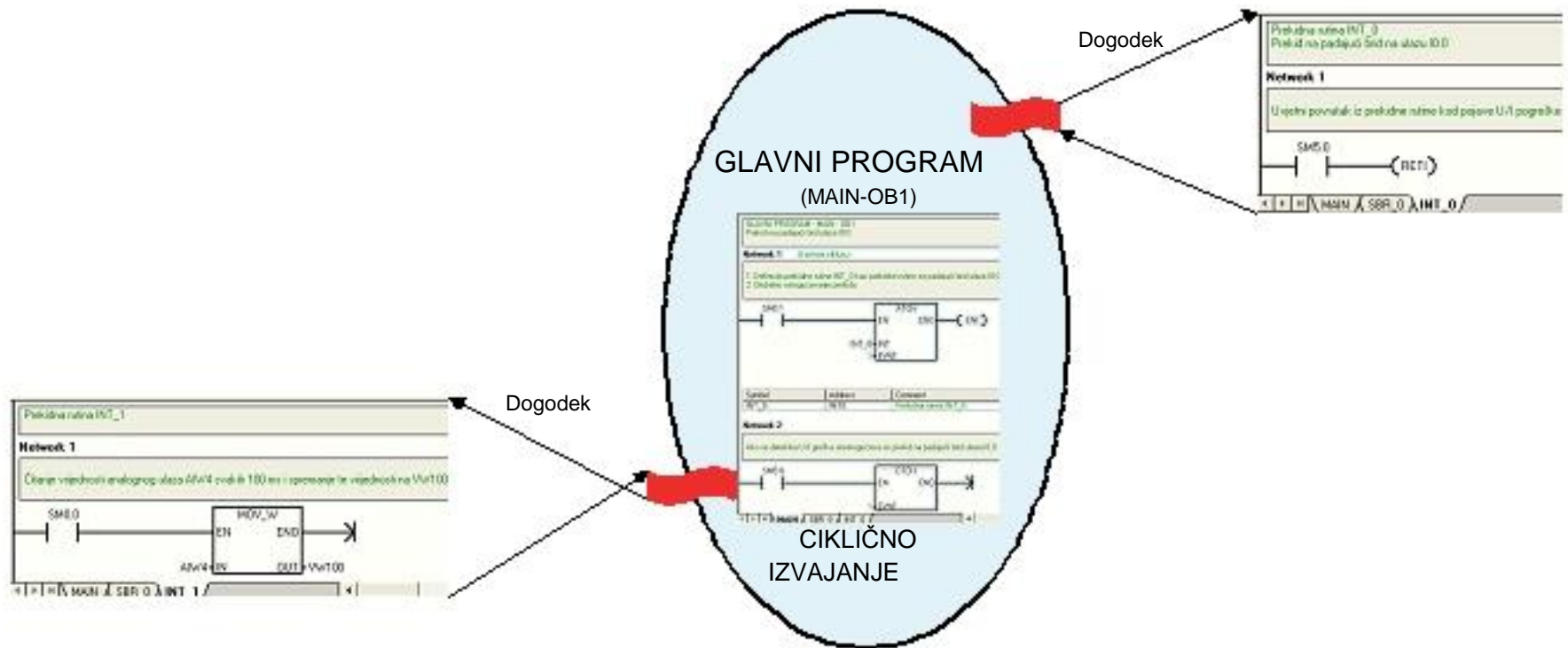


Prekinitveni način delovanja PLK

- Glavni program (OB1; MAIN) se izvaja ciklično
- Prekinitev začasno ustavi izvajanje glavnega programa in zažene izvajanje prekinitvene rutine



- Prekinitvena rutina je podprogram, ki se izvrši, ko se zgodi prekinitev in se v njej običajno izvede obdelava prekinitve, ki jo je sprožila
- PLK S7-200 lahko obdeluje maksimalno 128 prekinitvenih rutin

Opis delovanja PLK S7-200 s prekinitvenimi rutinami

- Prekinitvena rutina se izvršuje odvisno od prirejenih ji notranjih ali zunanjih dogodkov
- Ko se izvrši zadnji ukaz v prekinitveni rutini, se kontrola izvajanja programa vrača glavnemu programu
- Iz rutine se lahko pride tudi s pomočjo ukaza za pogojno vrnitev iz prekinitve
- Prekinitvene rutine morajo biti optimizirane za izvršitev posebnih opravil, ki se hitro izvršijo, nakar se kontrola izvajanja ponovno vrne glavnemu programu

Tipi prekinitev pri PLK S7-200

- S7-200 podpira naslednje tipove prekinitev:
 - Prekinitve komunikacijskega porta: S7-200 generira dogodke, ki omogočajo uporabniškemu programu upravljanje s komunikacijskim portom
 - V/I prekinitve: S7-200 generira dogodke za različne spremembe stanja posameznih V/I (ti dogodki omogočajo uporabniškemu programu reakcijo na hitre števec, impulzne izhode ter rastoči ali padajoči rob signala na vhodu)
 - Časovno bazirane prekinitve: S7-200 generira dogodke, ki omogočajo uporabniškemu programu, da deluje v določenih časovnih intervalih

■ Tabela tipov prekinitev pri PLK S7-200:

| Številka dogodka prekinitev | Opis prekinitev | Prioriteta skupine | Prioriteta v skupini |
|-----------------------------|----------------------------------|------------------------------|----------------------|
| 8 | Port 0: Sprejem znaka | | 0 |
| 9 | Port 0: Prenos končan | | 0 |
| 23 | Port 0: Sprejem sporočila končan | | 0 |
| 24 | Port 1: Sprejem sporočila končan | Komunikacijske (najvišja) | 1 |
| 25 | Port 1: Sprejem znaka | | 1 |
| 26 | Port 1: Prenos končan | | 1 |
| 19 | PTO 0 končal prekinitev | | 0 |
| 20 | PTO 1 končal prekinitev | | 1 |

| Številka dogodka prekinitve | Opis prekinitve | Prioriteta skupine | Prioriteta v skupini |
|-----------------------------|----------------------|--------------------|----------------------|
| 0 | Rastoči rob I0.0 | | 2 |
| 2 | Rastoči rob I0.1 | | 3 |
| 4 | Rastoči rob I0.2 | | 4 |
| 6 | Rastoči rob I0.3 | | 5 |
| 1 | Padajoči rob I0.0 | | 6 |
| 3 | Padajoči rob I0.1 | | 7 |
| 5 | Padajoči rob I0.2 | | 8 |
| 7 | Padajoči rob I0.3 | | 9 |
| 12 | HSC0 CV = PV | | 10 |
| 27 | HSC0 Sprememba smeri | | 11 |
| 28 | HSC0 Zunanji reset | Diskretne | 12 |
| 13 | HSC1 CV = PV | (srednja) | 13 |
| 14 | HSC1 Sprememba smeri | | 14 |
| 15 | HSC1 Zunanji reset | | 15 |
| 16 | HSC2 CV = PV | | 16 |
| 17 | HSC2 Sprememba smeri | | 17 |
| 18 | HSC2 Zunanji reset | | 18 |
| 32 | HSC3 CV = PV | | 19 |
| 29 | HSC4 CV = PV | | 20 |
| 30 | HSC4 Sprememba smeri | | 21 |
| 31 | HSC4 Zunanji reset | | 22 |
| 33 | HSC5 CV = PV | | 23 |

| Številka dogodka prekinitve | Opis prekinitve | Prioriteta skupine | Prioriteta v skupini |
|-----------------------------|-------------------------------|--------------------|----------------------|
| 10 | Časovna prekinitvev 0 | | 0 |
| 11 | Časovna prekinitvev 1 | Časovne | 1 |
| 21 | Timer T32 CT = PT prekinitvev | (najnižja) | 2 |
| 22 | Timer T96 CT = PT prekinitvev | | 3 |

Ukaz za delo s prekinitvami pri PLK S7-200:

■ Omogočanje in onemogočanje prekinitvev

- Ukaz za omogočanje prekinitvev (ENI) globalno omogoča obdelavo vseh pridruženih prekinitvenih dogodkov
- Ukaz za onemogočanje prekinitvev (DISI) globalno onemogoča obdelavo vseh prekinitvenih dogodkov
- Pri prehodu v RUN način so prekinitve inicialno onemogočene

- V RUN načinu se lahko prekinitve omogočijo z uporabo ENI ukaza
- Izvršitev DISI ukaza ustavi obdelavo prekinitve; aktivni prekinitveni dogodki bodo postavljeni v čakalno vrsto

■ Pogojna vrnitev iz prekinitve

- Ukaz za pogojno vrnitev iz prekinitve (CRETI) se lahko uporabi za vrnitev iz prekinitve glede na stanje pogoja v programski logiki

■ Prirejanje prekinitvev

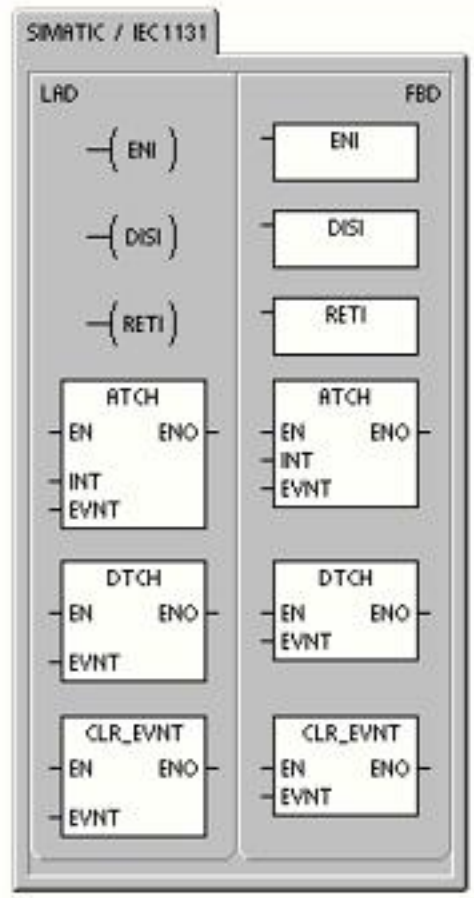
- Ukaz za prirejanje prekinitvev (ATCH) poveže prekinitveni dogodek EVNT s številko prekinitvene rutine INT in omogoči prekinitveni dogodek

■ Razdruževanje prekinitev

- Ukaz za razdruževanje prekinitev (DTCH) razdruži prekinitveni dogodek EVNT od vseh prekinitvenih rutin in onemogoči prekinitveni dogodek

■ Brisanje prekinitvenega dogodka

- Ukaz za brisanje prekinitvenega dogodka pobriše prekinitvene dogodke tipa EVNT iz čakalne vrste prekinitev
- Ta ukaz se uporablja za brisanje neželenih prekinitvenih dogodkov iz čakalne vrste prekinitev



■ Delo z ukazi za prireditev in razdružitev prekinitev

- Preden lahko neko prekinitveno rutino pokličemo moramo izvršiti prireditev prekinitvenega dogodka programskemu izseku, ki se bo izvršil, ko se bo zgodil ta izbrani dogodek
- Za ta namen uporabimo ukaz za prireditev prekinitve, ki ji pripada izbrani prekinitveni dogodek (določen s številko prekinitvenega dogodka) in programski izsek (določen s številko prekinitvene rutine)
- Možno je prirediti več prekinitvenih dogodkov eni prekinitveni rutini, enega dogodka pa ni možno istočasno prirediti večjemu številu prekinitvenih rutin
- Ko priredimo prekinitveni dogodek prekinitveni rutini, je ta prekinitev tudi avtomatsko omogočena
- Če so vse prekinitve onemogočene z ukazom za globalno onemogočitev prekinitev, se vsak pojav prekinitvenega dogodka postavi v čakalno vrsto prekinitev, vse dokler ne bodo prekinitve ponovno omogočene, z uporabo ukaza za globalno omogočitev prekinitev, ali dokler ne pride do preлива v čakalni vrsti prekinitev

- Možno je tudi posamično onemogočiti prekinitveni dogodek, s prekinitvijo povezave med prekinitvenim dogodkom in prekinitveno rutino z ukazom za združitev prekinitve, ki prekinev vrača v neaktivno stanje

■ **Prioritete prekinitev in čakalna vrsta prekinitev**

- Prekinitve se obdelujejo v S7-200 po principu "prva, ki se zgodi - se prva tudi obdela" znotraj posamezne prioritete skupine
- V enem trenutku se lahko izvaja samo ena uporabniška prekinitvena rutina
- Ko se enkrat začne izvajanje neke prekinitvene rutine, se ta rutina izvede do konca. Le-te se ne da prekiniti z nobeno drugo prekinitveno rutino, tudi ne z rutino višje prioritete
- Prekinitve, ki se zgodijo v trenutku izvajanja neke prekinitvene rutine, bodo postavljene v čakalno vrsto prekinitev in bodo izvedene kasneje

Postopek izvajanja prekinitev v S7-200

■ Postopek izvajanja prekinitev pri PLK S7-200 je sestavljen iz naslednjih korakov:

1. Pre-Int 1 – strojna oprema da zahtevo za izvajanje prekinitev s postavitvijo podatkov na sklad
2. Pre-Int 2 - sistemski program označi prekinitveni dogodek in ga da v čakalno vrsto prekinitev
3. Pre-Int 3 - sistemski program s signalnimi zastavicami da znak, da se lahko začne z izvajanjem prekinitev
4. Pre-Int 4 - sistemski program izvede zamenjavo vlog
5. Pre-Int 5 - sistemski program odstrani prekinitveni dogodek iz čakalne vrste prekinitev
6. Pre-Int 6 - sistemski program pripravi sklad prekinitvenih bitov in registre akumulatorja

7. Pre-Int 7 – sistemski program postavi zahtevo za vrnitev in pokliče uporabniško prekinitveno rutino
8. Post-Int 1 - sistemski program preveri preliv čakalne vrste prekinitev
9. Post-Int 2 - sistemski program preveri signalne zastavice, preko katerih vidi, ali je potrebno izvršiti neko drugo prekinitev
10. Post-Int 3 - sistemski program izvede zamenjavo vlog in vrne kontrolo glavnemu programu

Diagram sekvence splošnega izvajanja uporabniške prekinitvene rutine

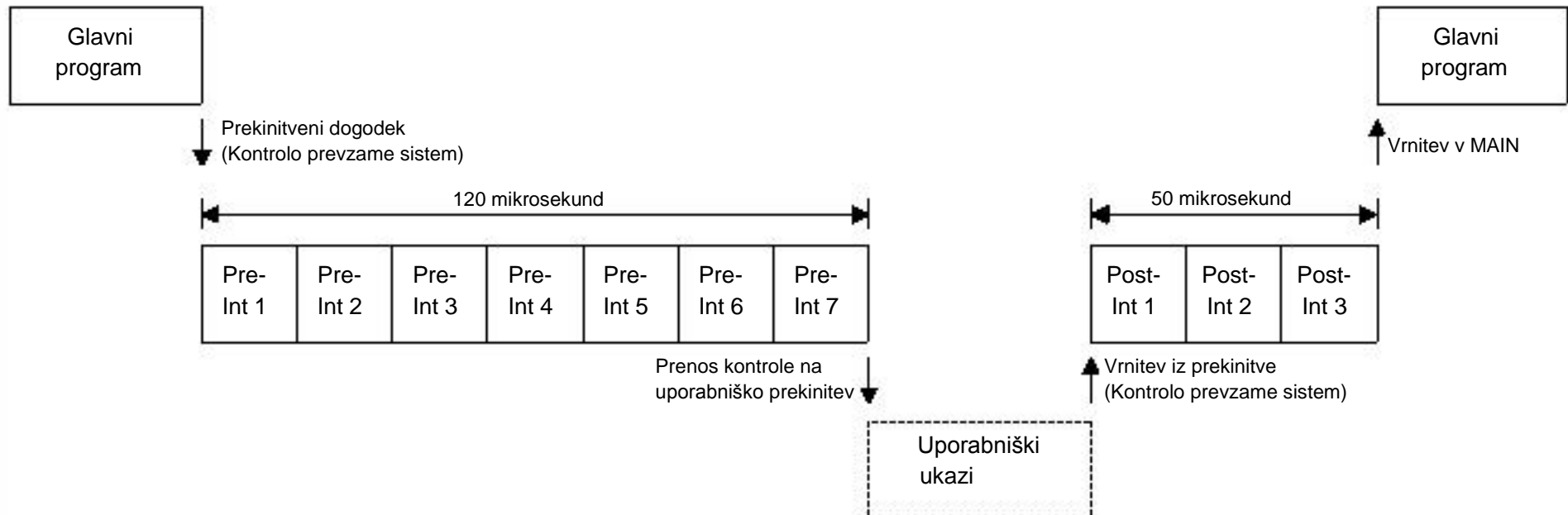
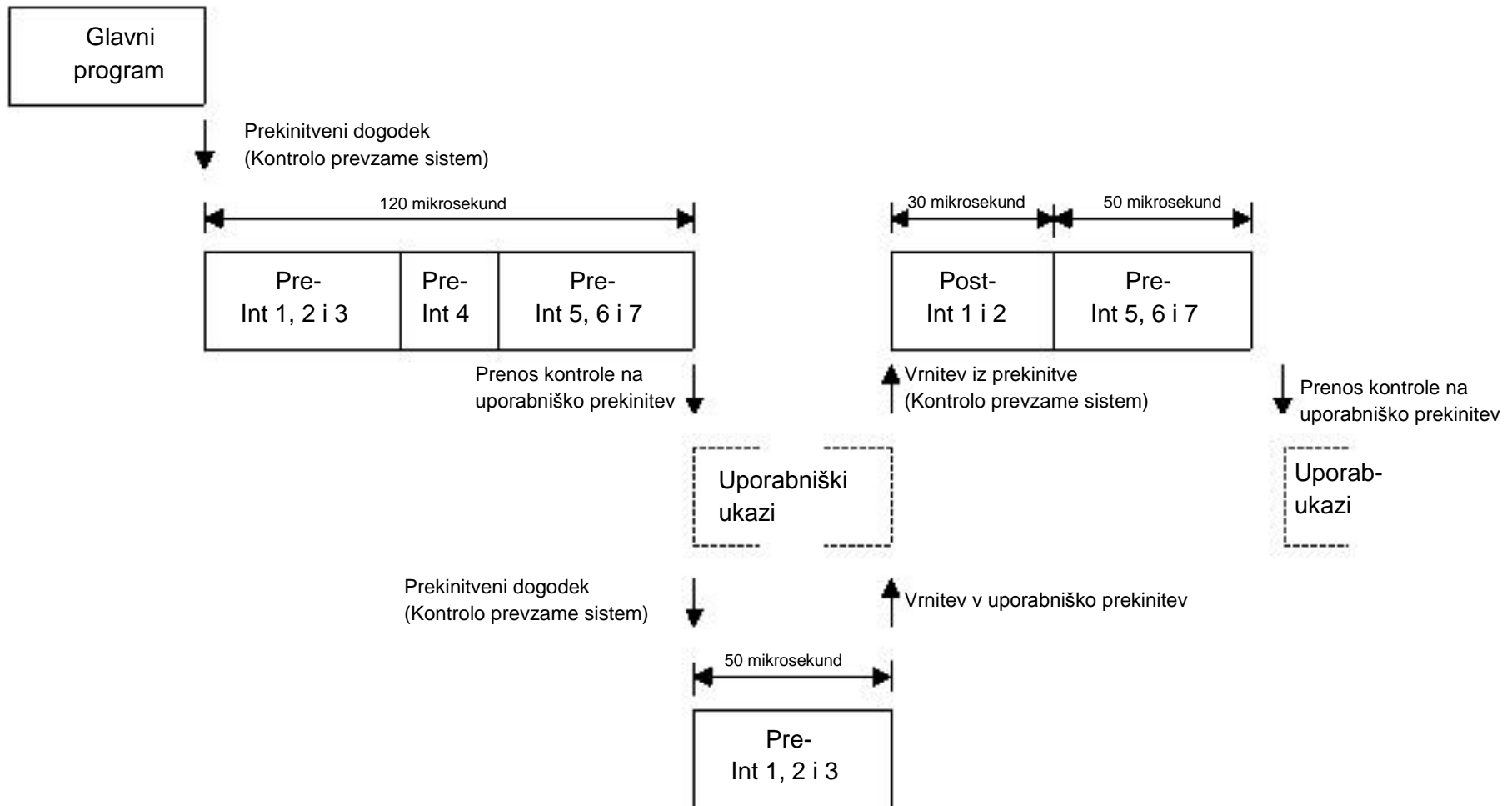
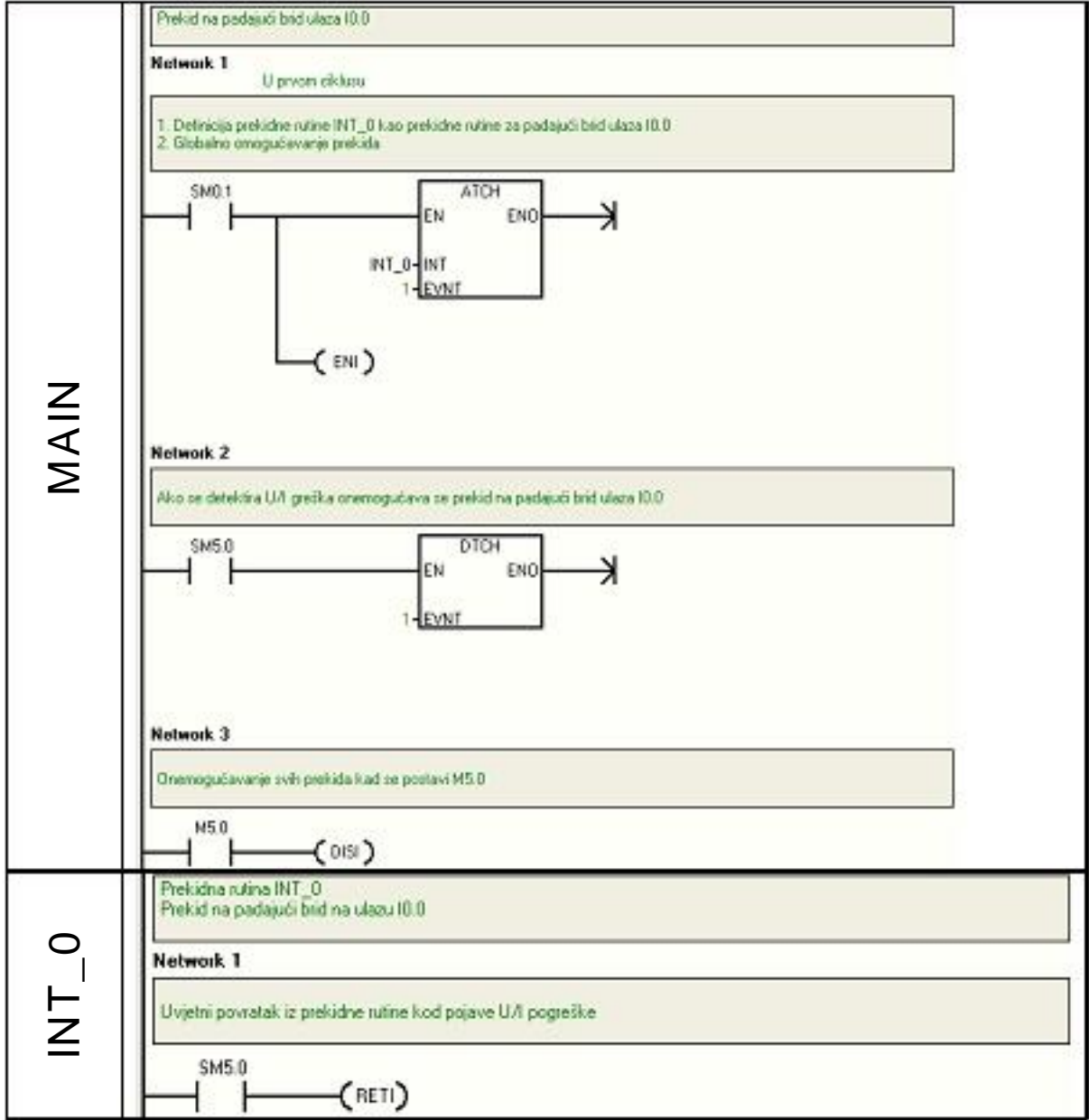
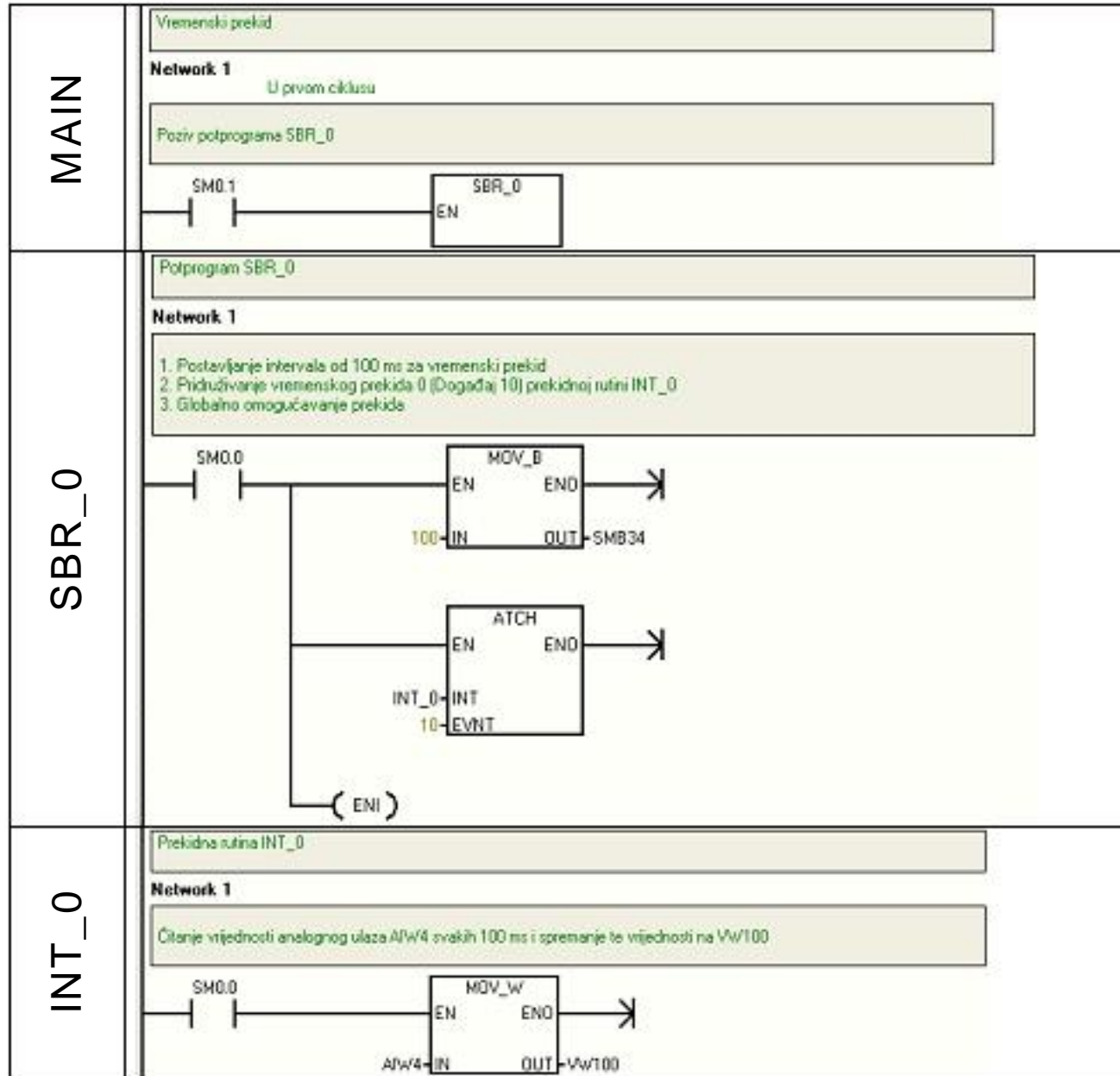


Diagram sekvence izvajanja uporabniške prekinitvene rutine v primeru, ko se nova prekinitev pojavi v trenutku izvajanja tekoče prekinitvene rutine





V/I prekinitav

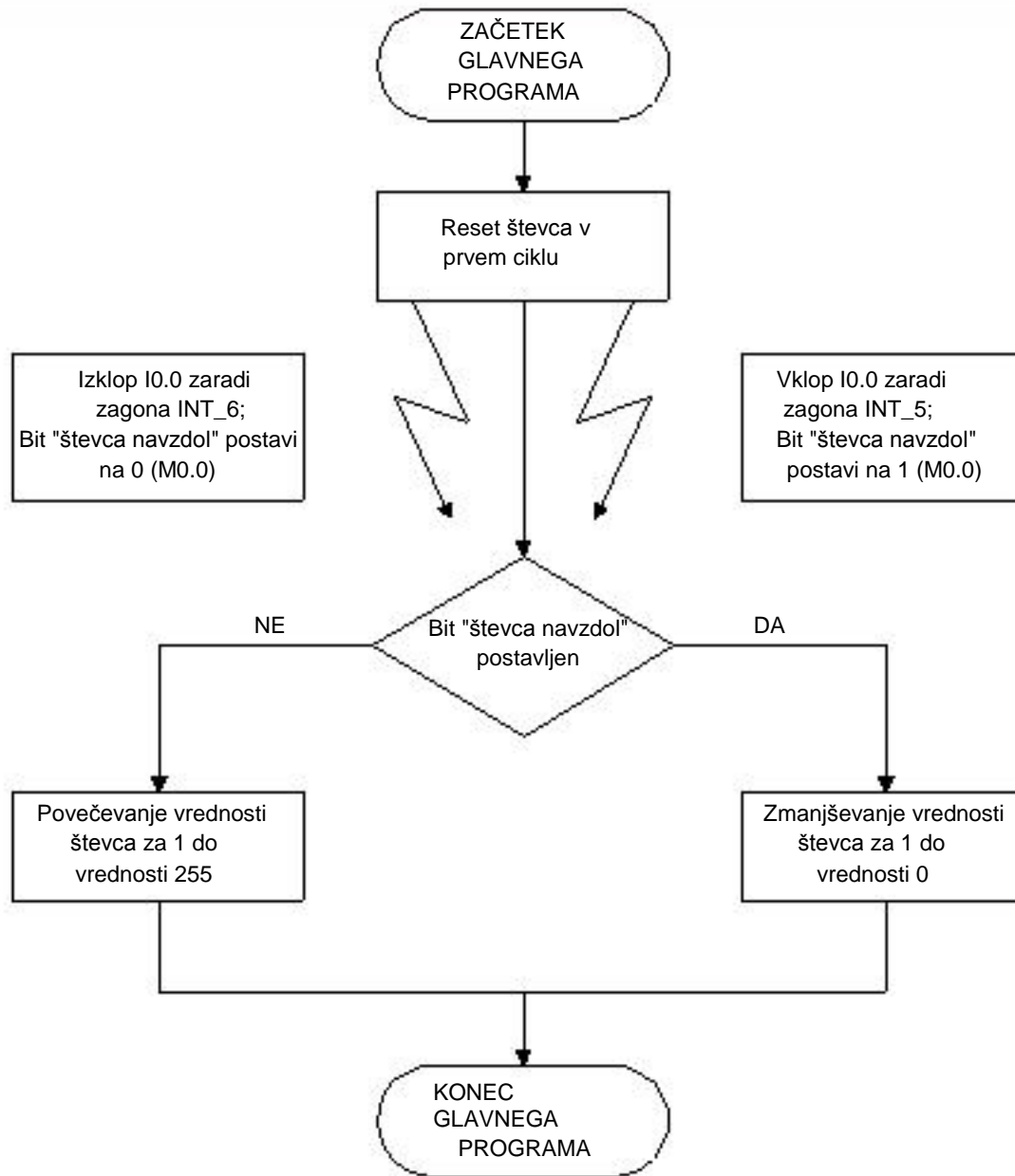


Časovna prekinitev

■ Primer 1: V/I prekinitve

- V tem primeru se izvaja štetje od 0 do 255, odvisno od vhoda I0.0
- Če je vhod I0.0 postavljen, šteje program navzdol, če ni, pa šteje navzgor
- Tako kot se vhod postavlja na 1 in 0, poklicani V/I prekinitveni rutini postavljata na 1 in 0 pomnilniški bit M0.0 "števec navzdol"

■ Diagram poteka programa je prikazan na naslednji sliki



■ Implementacija programa v PLK je podana na naslednjih slikah

| | Symbol | Address | Comment |
|---|----------------|---------|---|
| 1 | Uvijek_ON | SM0.0 | Ovaj bit je uvijek uključen |
| 2 | Prvi_ciklus_ON | SM0.1 | Ovaj bit je uključen samo u prvom ciklusu |
| 3 | Takt_1s | SM0.5 | Takt koji je uključen 0.5 s i isključen 0.5 s, za ciklus od 1 s |
| 4 | Dolje_Bit | M0.0 | Ovaj "Brojač dolje" memorijski bit određuje smjer brojanja |
| 5 | | | |

U/I PREKIDI

Network 1 Inicijalizacija, postavljanje vrijednosti brojenja na 0, pridruživanje i omogućavanje prekida

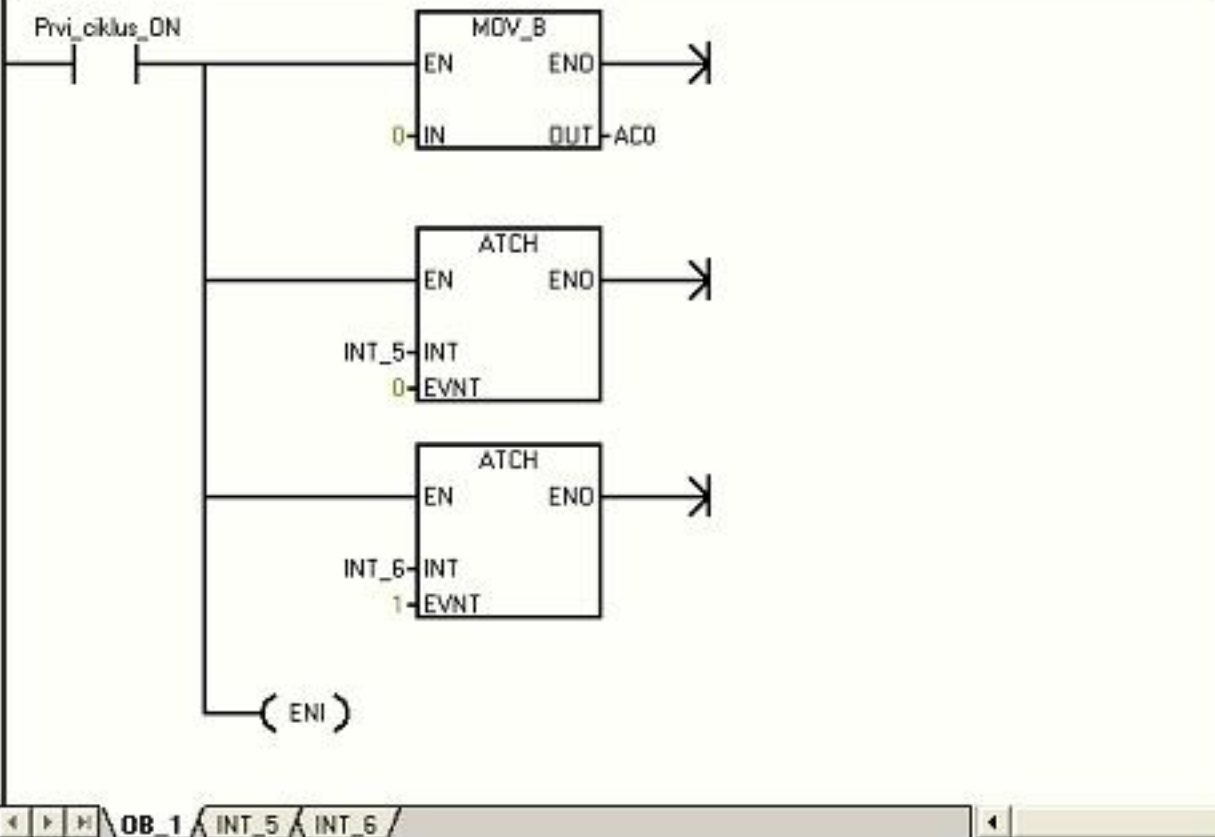
U prvom ciklusu skeniranja:

Spremanje konstante 0 u akumulator ACO kako bi se vrijednost brojenja postavila na 0.

Pridruživanje prekidnog događaja 0 prekidnoj rutini INT_5 kako bi se izvršavala na rastući brid na ulazu I0.0.

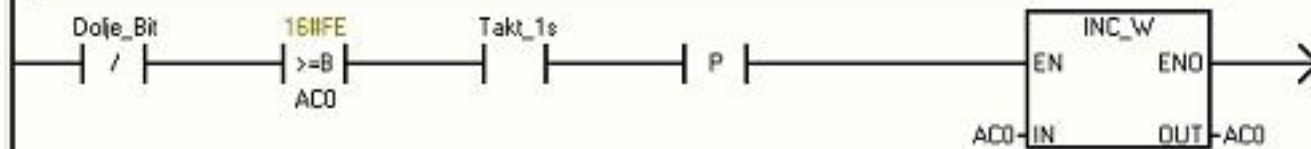
Pridruživanje prekidnog događaja 1 prekidnoj rutini INT_6 kako bi se izvršavala na padajući brid na ulazu I0.0.

Omožucavanje svih prekida



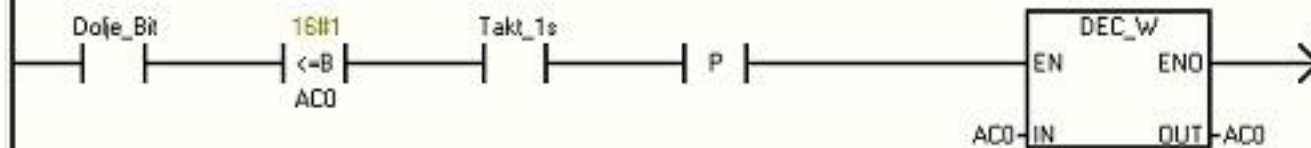
Network 2 Brojenje prema gore

Ako je vrijednost brojanja ≤ 254 , kad se pojavi takt vrijednost brojanja se poveća za jedan



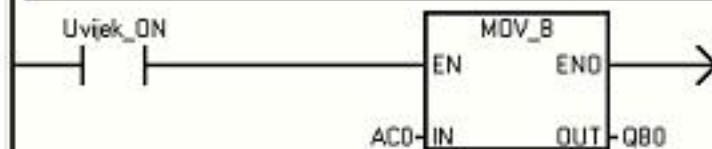
Network 3 Brojenje prema dođe

Ako je vrijednost brojanja > 1 , kad se pojavi takt vrijednost brojanja se smanji za jedan



Network 4

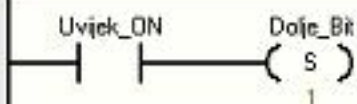
U svakom ciklusu skeniranja:
Postavljanje trenutne vrijednosti brojanja na izlazni bajt Q80



PREKIDNA RUTINA 5
INT_5

Network 1

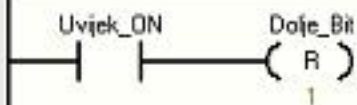
Postavljanje moda brojenja prema dole
Postavlja se "Brojač dole" memorijski bit (M0.0)



PREKIDNA RUTINA 6
INT_5

Network 1

Postavljanje moda brojenja prema gore
Briše se "Brojač dole" memorijski bit (M0.0)

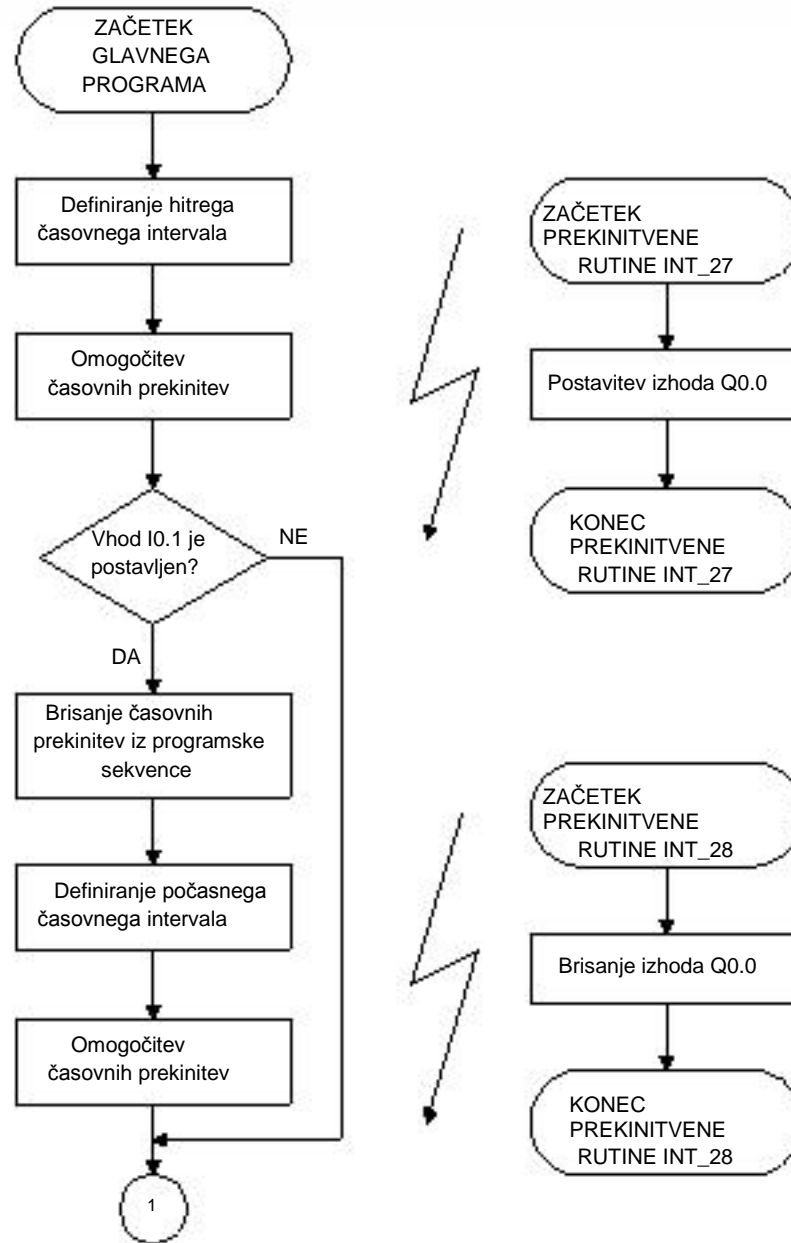


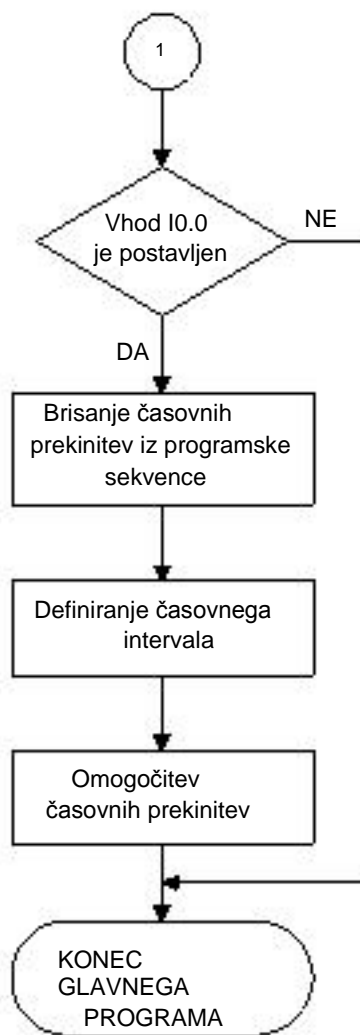
■ Primer 2: Časovna prekinitev

■ Časovne prekinitev so uporabljene za izvedbo utripanja

■ Vklop vhoda I0.1 zmanjša frekvenco utripanja za polovico, medtem ko vklop vhoda I0.0 vrne originalno frekvenco utripanja

■ Diagram poteka programa je prikazan na naslednji sliki





■ Implementacija programa v PLK je podana na naslednjih slikah

| | Symbol | Address | Comment |
|---|-------------------|---------|---|
| 1 | Uvijek_ON | SM0.0 | Ovaj bit je uvijek uključen |
| 2 | Prvi_Ciklus_ON | SM0.1 | Ovaj bit je uključen samo u prvom ciklusu skeniranja |
| 3 | Vijeme_0_Interval | SMB34 | Vremenski interval (5 ms do 255 ms: 1-ms povećanje) za vremenski prekid 0 |
| 4 | Vijeme_1_Interval | SMB35 | Vremenski interval (5 ms do 255 ms: 1-ms povećanje) za vremenski prekid 1 |
| 5 | | | |

VREMENSKI PREKIDI

Network 1 Inicijalizacija, definiranje vremena, pridruživanje i omogućavanje prekida

U prvom ciklusu skeniranja

Definiranje prekida

INT_27 Višemerni prekid 0
Višemerni interval = 50 ms (SMB34)
Prekidni događaj = 10

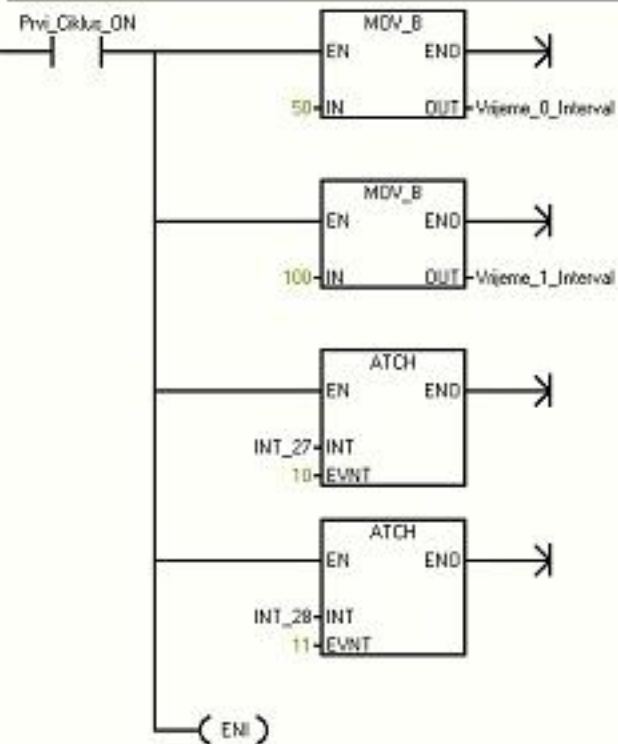
INT_28 Višemerni prekid 1
Višemerni interval = 100 ms (SMB35)
Prekidni događaj = 11

Pridruživanje prekida

Događaj 10 - INT_27

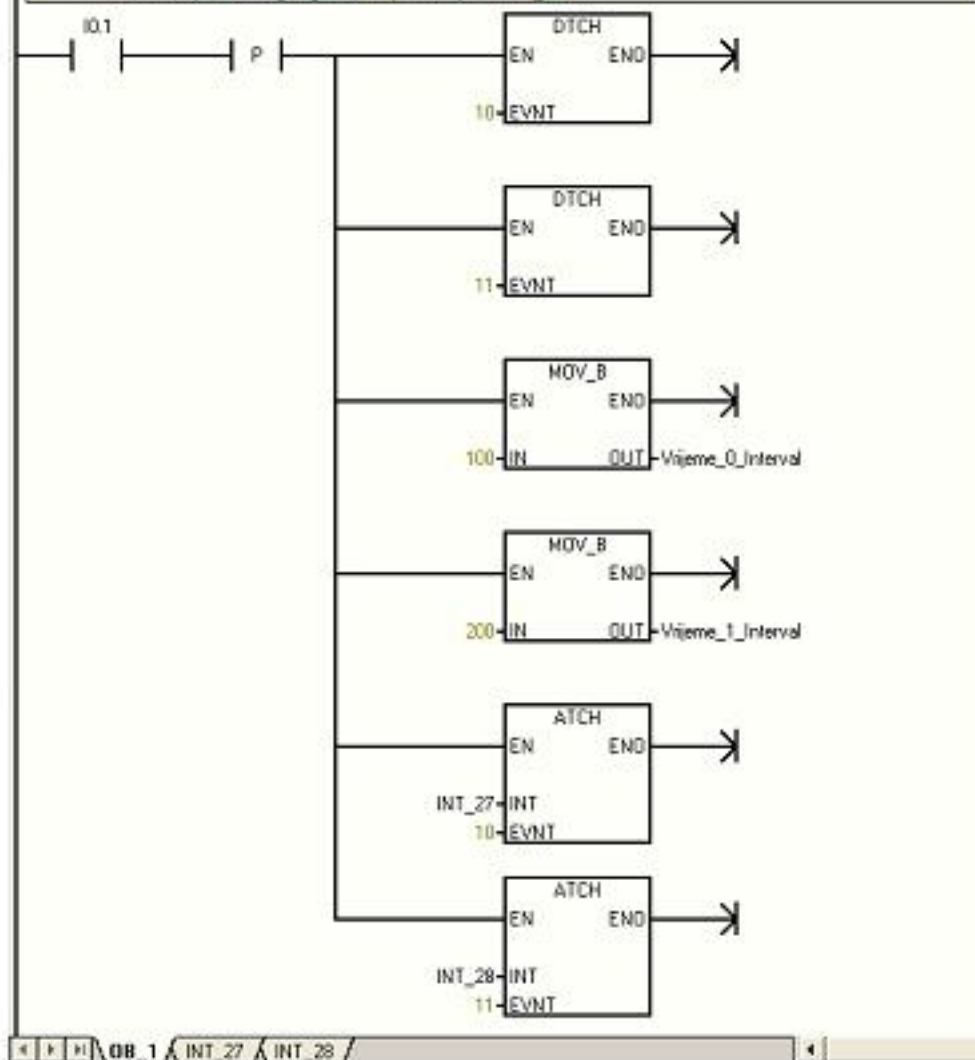
Događaj 11 - INT_28

Omožučavanje svih prekida



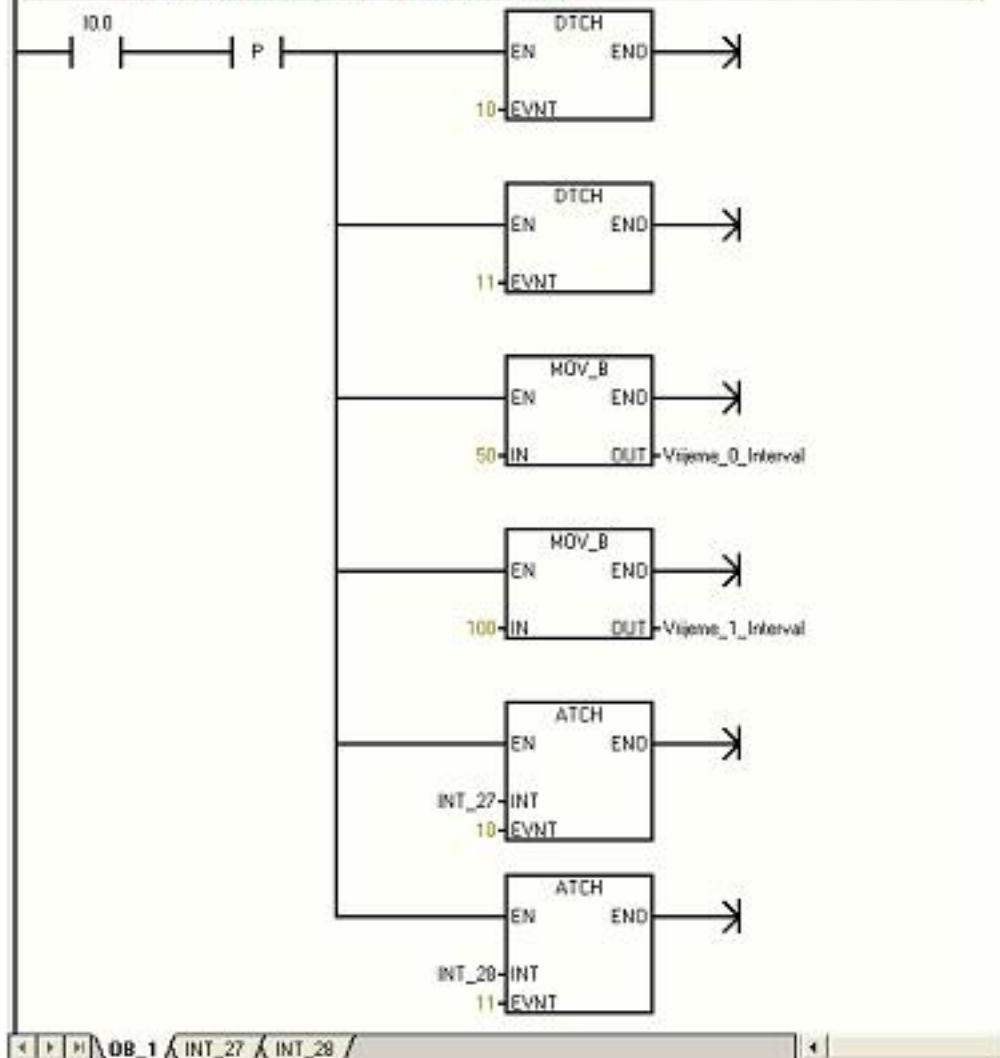
Network 2 Promjena intervala vremenskog prekida

Na pozitivan brrd na ulazu I0.1:
Dijeljenje prekidnog događaja 10 od INT_27
Dijeljenje prekidnog događaja 11 od INT_28
Vremenski interval = 100 ms (SMB34) za vremenski prekid 0
Vremenski interval = 200 ms (SMB35) za vremenski prekid 1
Pridruživanje prekidnog događaja 10 prekidnoj rutini INT_27
Pridruživanje prekidnog događaja 11 prekidnoj rutini INT_28



Network 3 Promjena intervala vremenskog prekida na prethodnu vrijednost

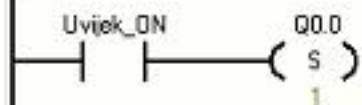
Na pozitivan brd na ulazu I0.0 vraćaju se stare postavke:
Odjeljivanje prekidnog događaja 10 od INT_27
Odjeljivanje prekidnog događaja 11 od INT_28
Vremenski interval = 50 ms (SMB34) za vremenski prekid 0
Vremenski interval = 100 ms (SMB35) za vremenski prekid 1
Pridruživanje prekidnog događaja 10 prekidnoj rutini INT_27
Pridruživanje prekidnog događaja 11 prekidnoj rutini INT_28



PREKIDNA RUTINA 27
INT_27

Network 1

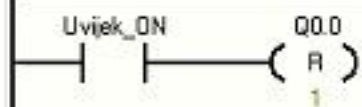
Postavljanje izlaza Q0.0



PREKIDNA RUTINA 28
INT_28

Network 1

Brisanje izlaza Q0.0



Priprava za laboratorijsko vajo:

■ Izdelaj sistem za naslednjo nalogo:

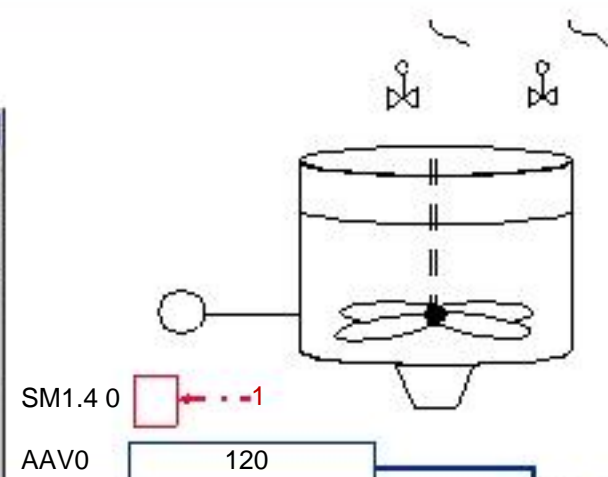
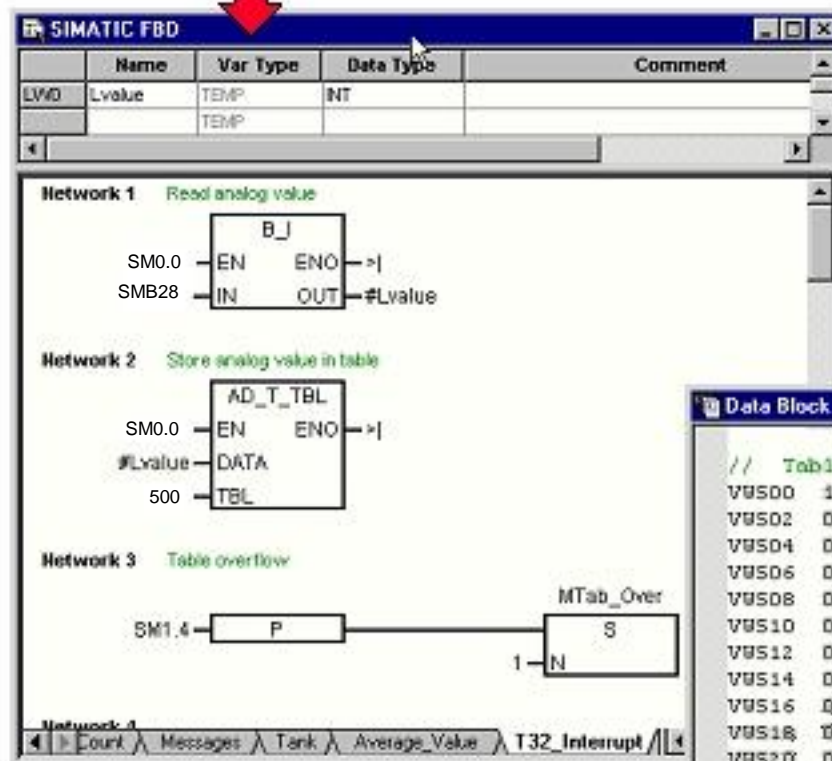
- V merilnem ciklu se vsebina rezervoarja meri vsakih 10 sekund in se shranjuje v tabelo dokler se ne izmeri 10 merilnih vrednosti
- Analogna vrednost za vsebino rezervoarja se simulira s potenciometrom 0 10 vrednosti v tabeli je enako prelivu. Ko se opravijo meritve, se izračuna srednja vrednost teh 10 analognih vrednosti iz tabele. Nato se ponavlja enak merilni cikel

■ Potrebno je:

- V prekinitveni rutini 0 vsakih 10 sekund odčitati analogno vrednost in jo shraniti v tabelo
- Timer T32 aktivirati v podprogramu "Rezervoar". Ta timer mora generirati prekinitveni dogodek 21 vsakih 10 sekund ter ponastaviti in omogočiti prekinitvene dogodke
- V podprogramu "Srednja vrednost", ko se pojavi preliv tabele, se izračuna srednja vrednost 10 analognih vrednosti. Ta vrednost se potem shrani v spremenljivko VW300

Dogodek 21

Tabela z ukazi



```
Data Block
// Table to store 10 Analog values
V#500 1Q // Table with 10 Elements
V#502 0 // Number used Elements
V#504 0 // Element 1
V#506 0 // Element 2
V#508 0 // Element 3
V#510 0 // Element 4
V#512 0 // Element 5
V#514 0 // Element 6
V#516 0 // Element 7
V#518 0 // Element 8
V#520 0 // Element 9
V#522 0 // Element 10
V#524 0 // Average value
```

SM1.4

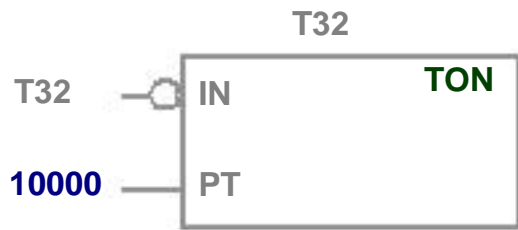
Ta bit se postavi, ko "Add to Table" ukaz poskuša povzročiti preliv tabele z elementi (>10).

Struktura tabele je naslednja:

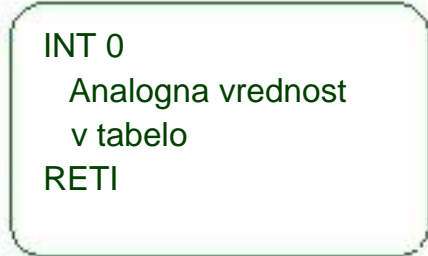
1. vrednost
2. vrednost
3. do n-te vrednosti

število elementov v tabeli
številka zapisa
podatki

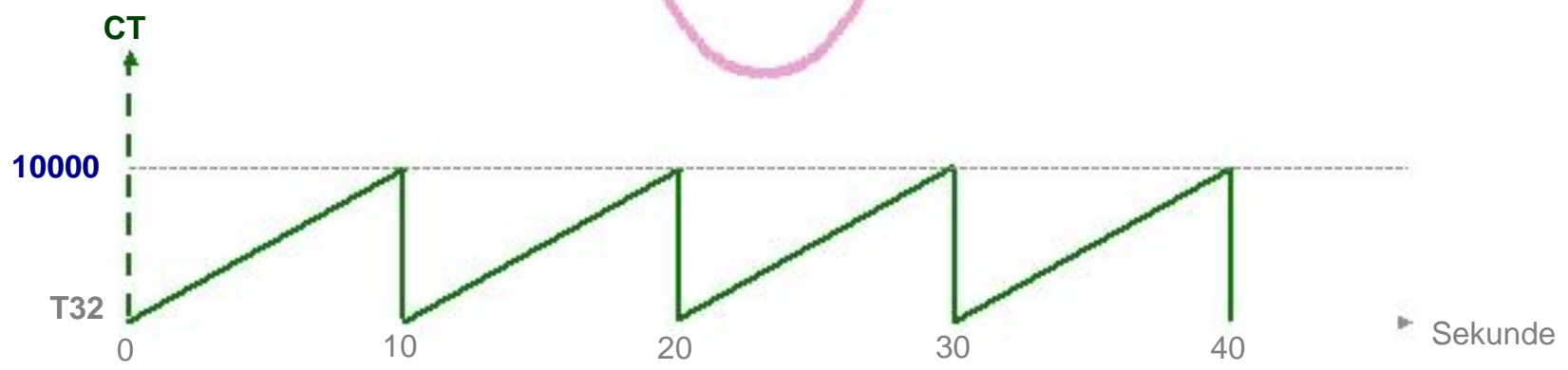
Prekinitev cikličnega programa



| Dogodek | Opis |
|---------|----------------|
| 21 | CT = PT |



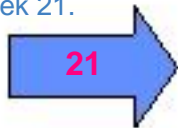
dokler ni prekinjen s prekinitvenim dogodkom



Ukazi za prekinitve



Vsakih 10000 ms = 10 s bo generiran prekinitveni dogodek 21.



Ukaz za prireditev prekinitve (ATCH) poveže prekinitveni dogodek (21) s prekinitveno rutino INT_0 (T32_Interrupt).

Network 1 Attach Interrupt 21 Time T32 CT=PT

| Name | Var Type | Data Type | Comment |
|------|----------|-----------|---------|
| EN | IN | BOOL | |
| LWD | Ltime | INT | |

Network 1 logic: ATCH block with EN (SM0.1), ENO (ENO), INT (T32_Interrupt), and EVNT (21).

Network 2 Enable interrupt

Network 2 logic: ENI block with SM0.1 as input.

Network 3 Interrupt 21 if CT=PT

Network 3 logic: T32 block with IN (T32), TON (T32), and PT (#Ltime).

FOR...NEXT zanka

- **FOR** ukaz povzroča ponavljajoče se izvajanje ukazov med FOR in NEXT
- Potrebno je specificirati začetne vrednosti naslednjih spremenljivk:
 - Tekočo vrednost števca "INDEX"
 - Začetno vrednost "INITIAL"
 - Končno vrednost "FINAL"
- Če ima INITIAL vrednost 1 in FINAL vrednost 10, kot v primeru, se ukazi med FOR in NEXT izvršijo 10-krat. Vrednost števca "INDEX" se samodejno povečuje za 1 po vsaki ponovitvi zanke
- **NEXT** ukaz označuje konec telesa programske zanke

FOR / NEXT zanka

The screenshot displays the SIMATIC Manager interface. At the top, a table lists variables:

| Name | Var Type | Data Type | Comment |
|--------|----------|-----------|---------|
| EN | IN | BOOL | |
| Laddr | IN | DWORD | |
| Lvalue | OUT | INT | |
| Lindex | TEMP | INT | |
| Lnum | TEMP | INT | |

Below the table is a ladder logic network (Network 3) titled "For / Next Loop". It features a FOR instruction block with inputs EN (SM0.0), INDX (#Lindex), INIT (+1), and FINAL (#Lnum). The FOR block is connected to an ADD_I block, which has inputs IN1 (#Laddr) and IN2 (#Lvalue), and output ENO (#Lvalue). This output is connected to another ADD_I block, which has inputs IN1 (+2) and IN2 (#Laddr), and output ENO (#Laddr). The network concludes with a NEXT instruction block.

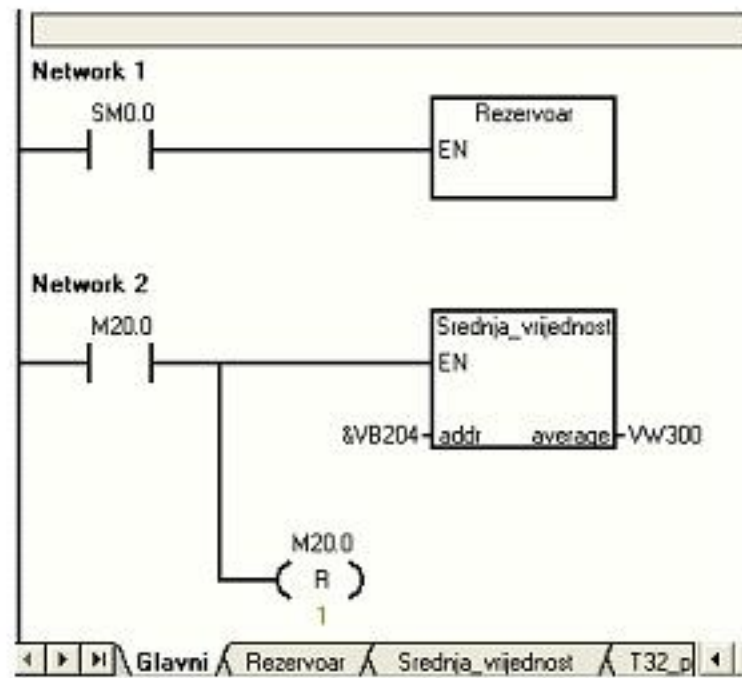
On the right, a "Data Block" window shows a table of values:

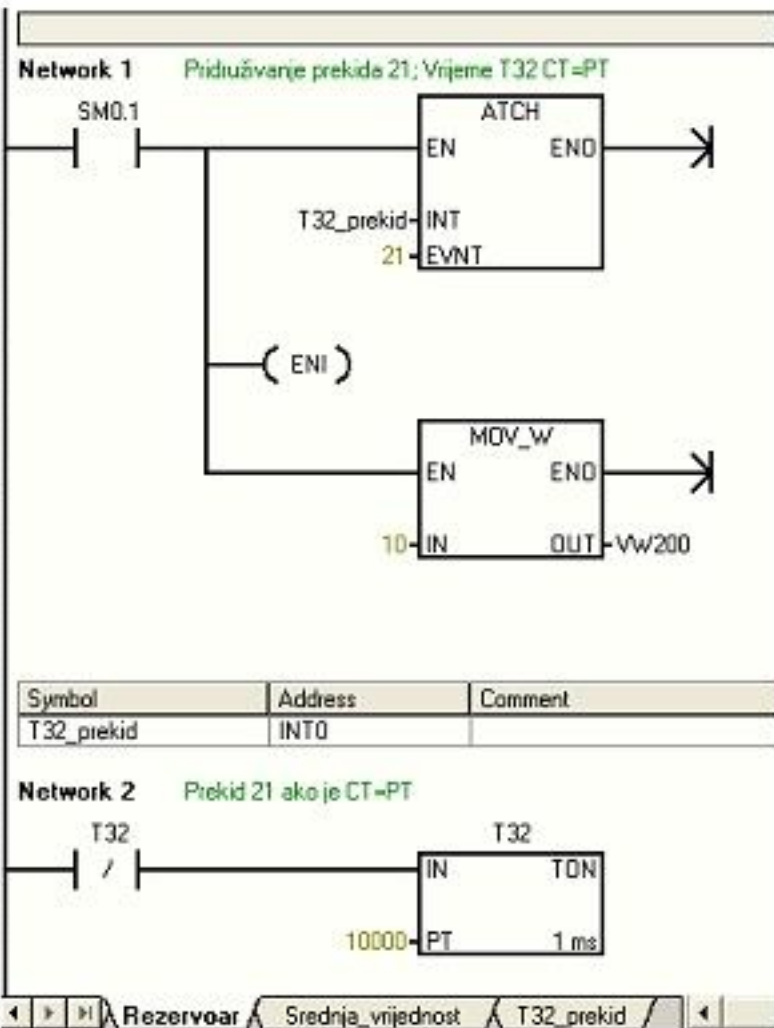
| Address | Value | Comment |
|---------|-------|------------------------|
| VB500 | 10 | Table with 10 Elements |
| VB502 | 0 | Number used Elements |
| VB504 | 0 | Element 1 |
| VB506 | 0 | Element 2 |
| VB508 | 0 | Element 3 |
| VB510 | 0 | Element 4 |
| VB512 | 0 | Element 5 |
| VB514 | 0 | Element 6 |
| VB516 | 0 | Element 7 |
| VB518 | 0 | Element 8 |
| VB520 | 0 | Element 9 |
| VB522 | 0 | Element 10 |
| VB524 | 0 | Average value |

At the bottom right, a callout box shows the "Average_Value" function block with inputs EN (MTab_Over) and Laddr Lverage (&VB500), and output VAV.

- &VB500 posreduje naslov VB500 podprogramu
- # Laddr vsebuje naslov VB500
- *Laddr vzame vsebino z naslova VB500

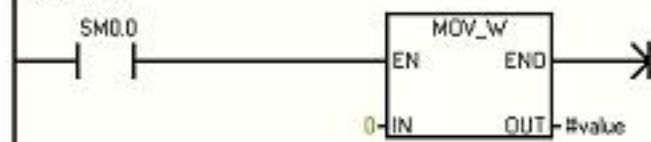
Rešitev:



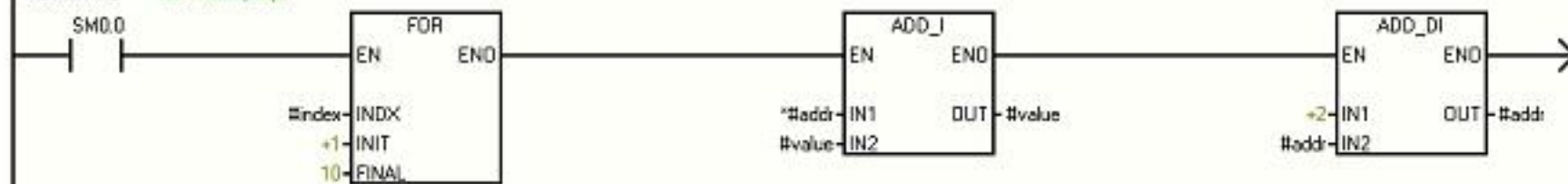


| | Symbol | Var Type | Data Type | Comment |
|-----|---------|----------|-----------|---------|
| | EN | IN | BOOL | |
| LDD | addr | IN | DWORD | |
| | | IN_OUT | | |
| LW4 | average | OUT | INT | |
| LW6 | index | TEMP | INT | |
| LW8 | value | TEMP | INT | |

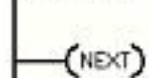
Network 1



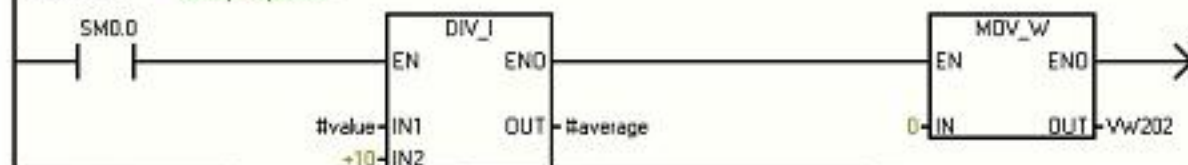
Network 2 For / Next petlja



Network 3 Kraj petlje



Network 4 Srednja vrijednost



| | Symbol | Var Type | Data Type |
|-----|--------|----------|-----------|
| LW0 | temp | TEMP | INT |

