

CAP. 5 PROBLEME REZOLVATE

Problema 1: Controlul unei macarale

1. Descrierea procesului:

Aplicația constă în controlul unei macarale care trebuie să realizeze cele 2 cicluri de mișcare reprezentate în figura 5.1. Inițial macaraua se găsește în poziția de repaus 1. La apăsarea butonului de pornire, macaraua pornește și se realizează ciclul 1, până se ajunge în poziția de repaus 2, unde rămâne pentru un anumit timp cunoscut (3 s), înainte de a porni ciclul 2; când ajunge în poziția de repaus 1, macaraua se va opri. Un nou ciclu va porni după reapăsarea butonului de pornire.

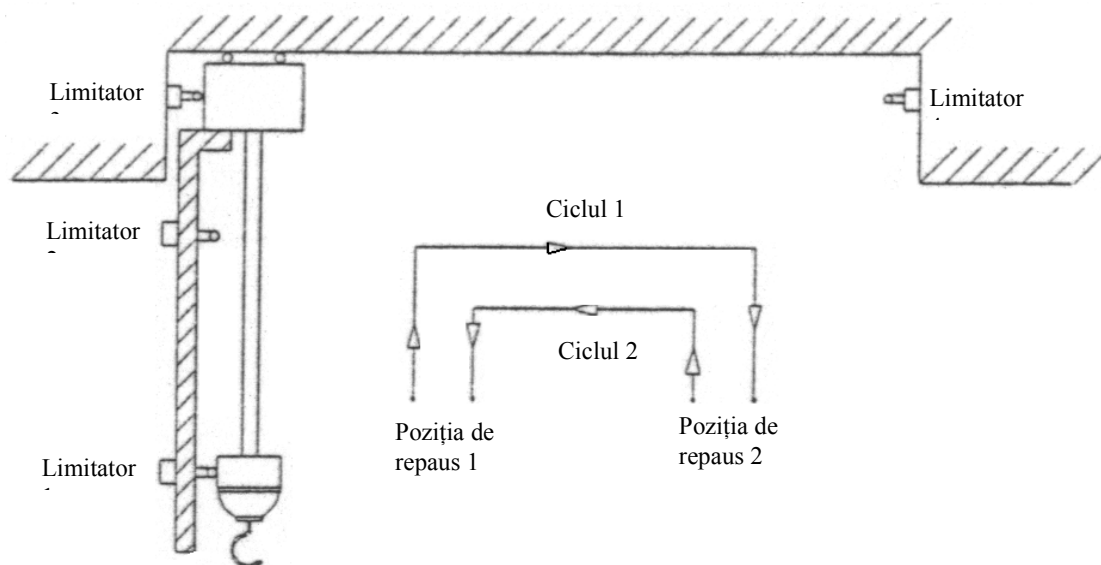


Fig. 5.1 Controlul unei macarale

Elemente de execuție:

- 2 motoare cu 2 senzori de rotație, unul pentru mișcarea orizontală și unul pentru cea verticală

Elemente de măsură:

- 4 limitatoare de cursă
- 1 buton de pornire

2. Soluția de automatizare

❑ Varianța 1: implementarea în mediul ISAGraf

Prima soluție pentru controlul acestei aplicații o reprezintă un automat programabil de tip PEP Smart pentru care s-a dezvoltat un proiect Isagraf ce cuprinde un program principal secvențial.

Dicționarul de variabile globale:

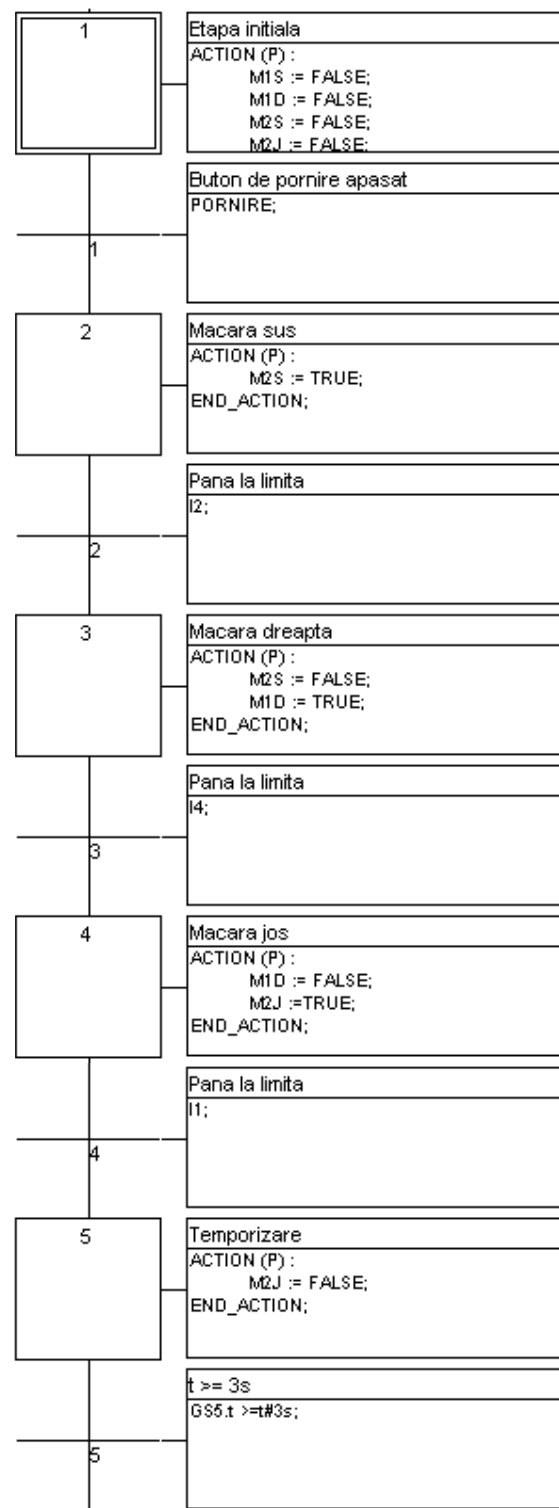
Variabile de intrare booleene:

- Pornire : buton de pornit ciclu
- I1 : limitator jos
- I2 : limitator sus
- I3 : limitator stânga
- I4 : limitator dreapta

Variabile de ieșire booleene:

- M1S : acționare motor 1 stânga
- M1D : acționare motor 1 dreapta
- M2S : acționare motor 2 sus
- M2J : acționare motor 2 jos

Programul principal este prezentat în figura 5.2



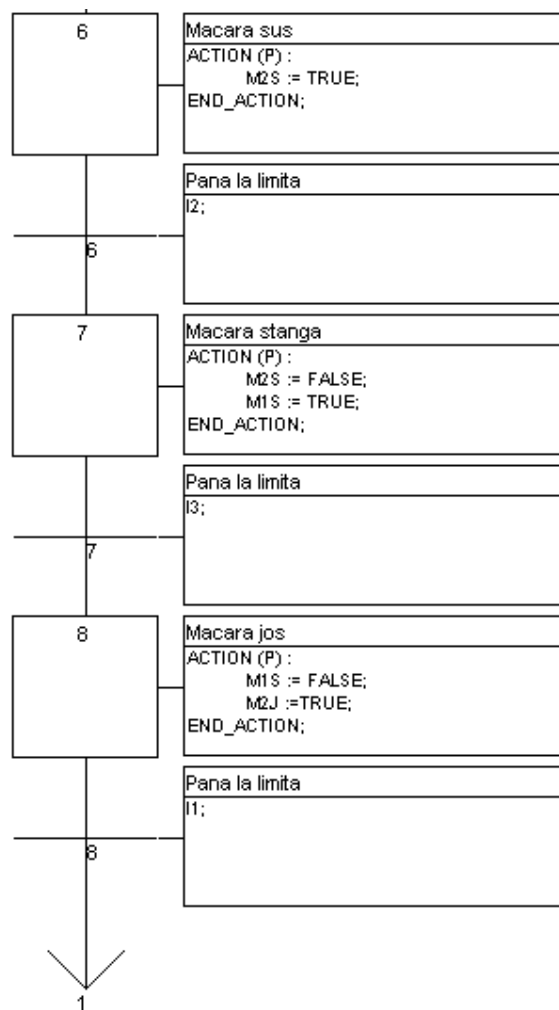


Fig. 5.2 Programul principal

❑ Varianta 2: implementarea in limbajul Ladder Diagram

Pentru controlul acestei aplicații s-a ales un automat programabil de tip Allen Bradley pentru care s-a dezvoltat o diagramă Grafcet (fig 5.3) și un program de tip Ladder Diagram (fig. 5.4).

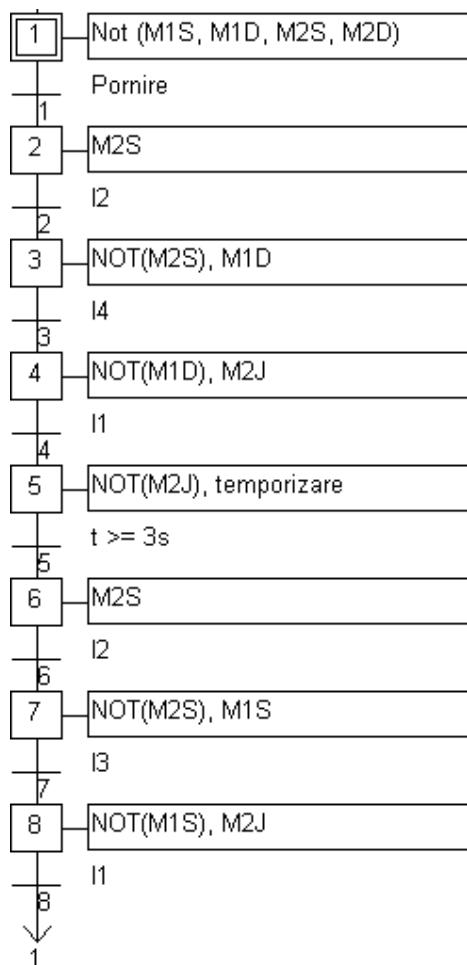


Fig.5.3 Diagrama Grafcet

Asocierea intrărilor și ieșirilor fizice cu biți din regiștrii de intrare / ieșire este prezentată în tabelul 5.1:

Tabelul 5.1

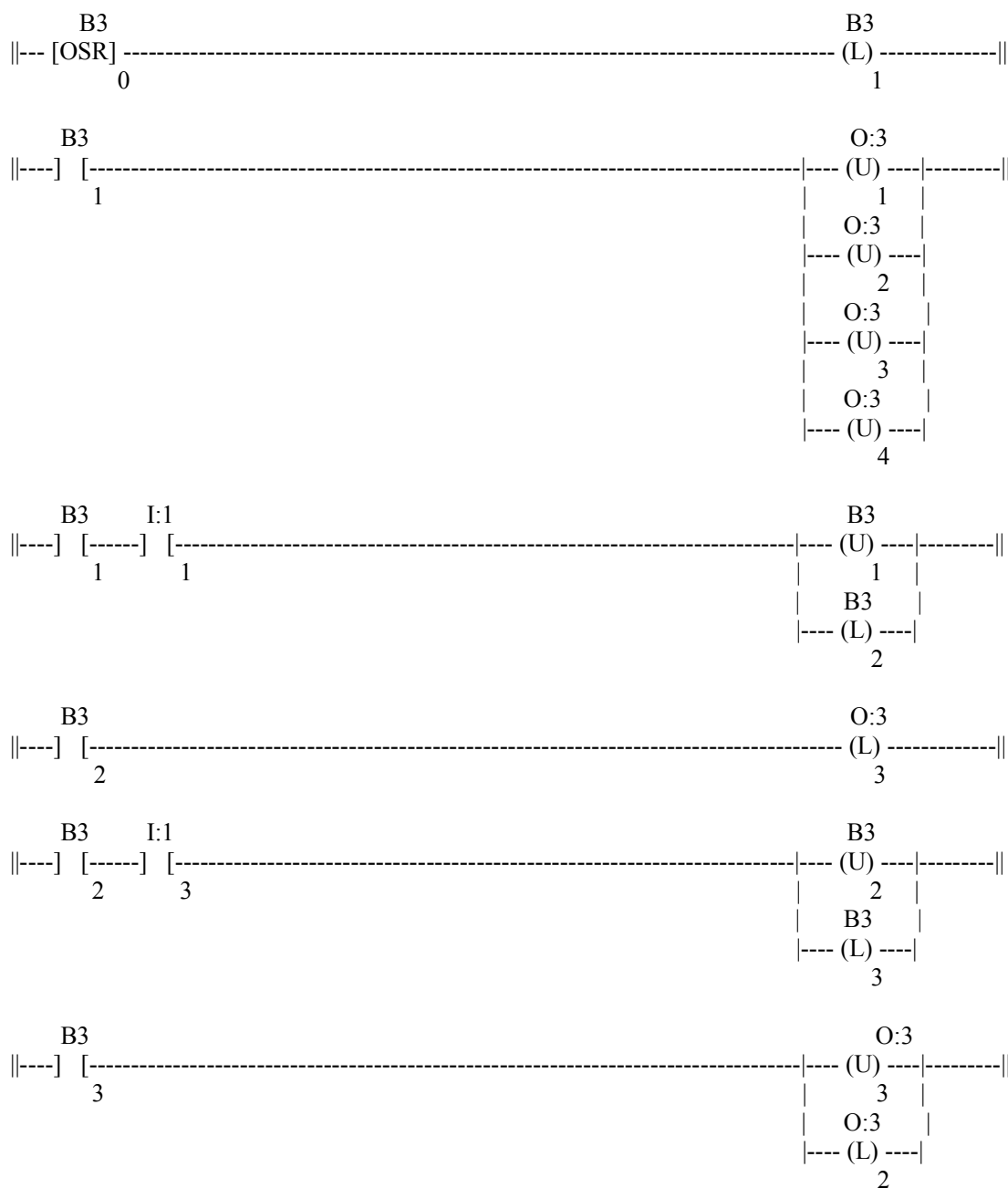
Intrare fizică	Adresă internă	Ieșire fizică	Adresă internă
Pornire	I:1/1	M1S	O:3/1
I1	I:1/2	M1D	O:3/2
I2	I:1/3	M2S	O:3/3
I3	I:1/4	M2J	O:3/4
I4	I:1/5		

Asocierea etapelor cu biți din fișierul de bit B3 și alegerea fișierului de timer este prezentată în tabelul 5.2:

Tabelul 5.2

Etapa	Adresa bit	Temporizare	Fișier de timer
1	B3/1	Temporizare 1	T4:0
2	B3/2		
3	B3/3		
4	B3/4		
5	B3/5		
6	B3/6		
7	B3/7		
8	B3/8		

Diagrama Ladder este prezenta în continuare:



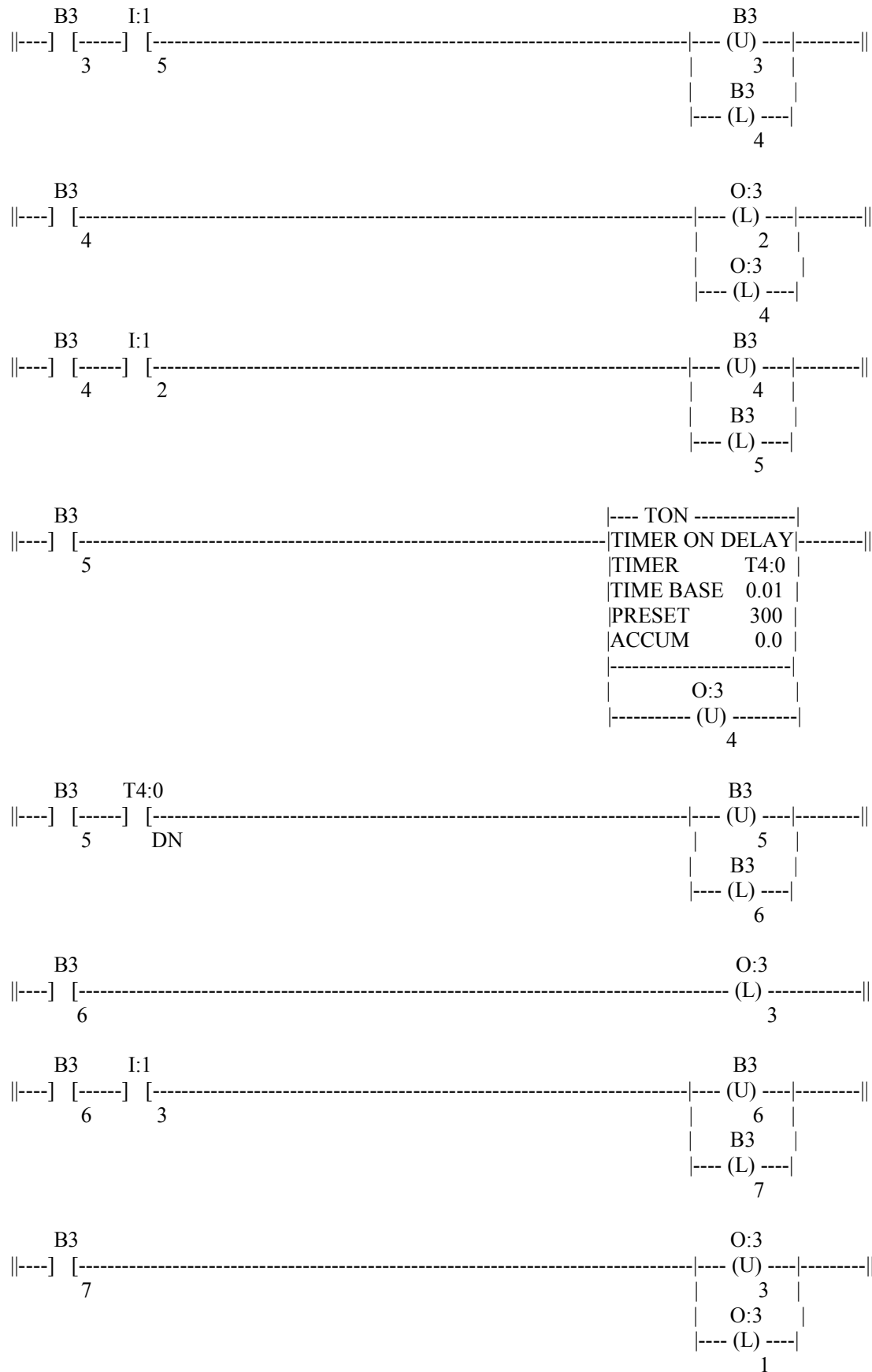




Fig.5.4 Diagrama Ladder

Problema 2: Comanda mișcării oscilatorii a unui mobil

1. Descrierea procesului:

Un mobil alunecă pe un șurub mișcat de un motor acționat de 2 contactoare (Cd – dreapta și Cs – stânga). Mobilul trebuie să realizeze o mișcare oscilatorie continuă din momentul în care se primește comanda (impuls) de la butonul M. Un impuls de la butonul P trebuie să oprească motorul, dar nu imediat, ci la finalul mișcării începute. Un impuls de la butonul E produce o retragere imediată a mobilului în poziția de origine, iar sistemul se mai poate pune în mișcare doar apăsând butonul R.

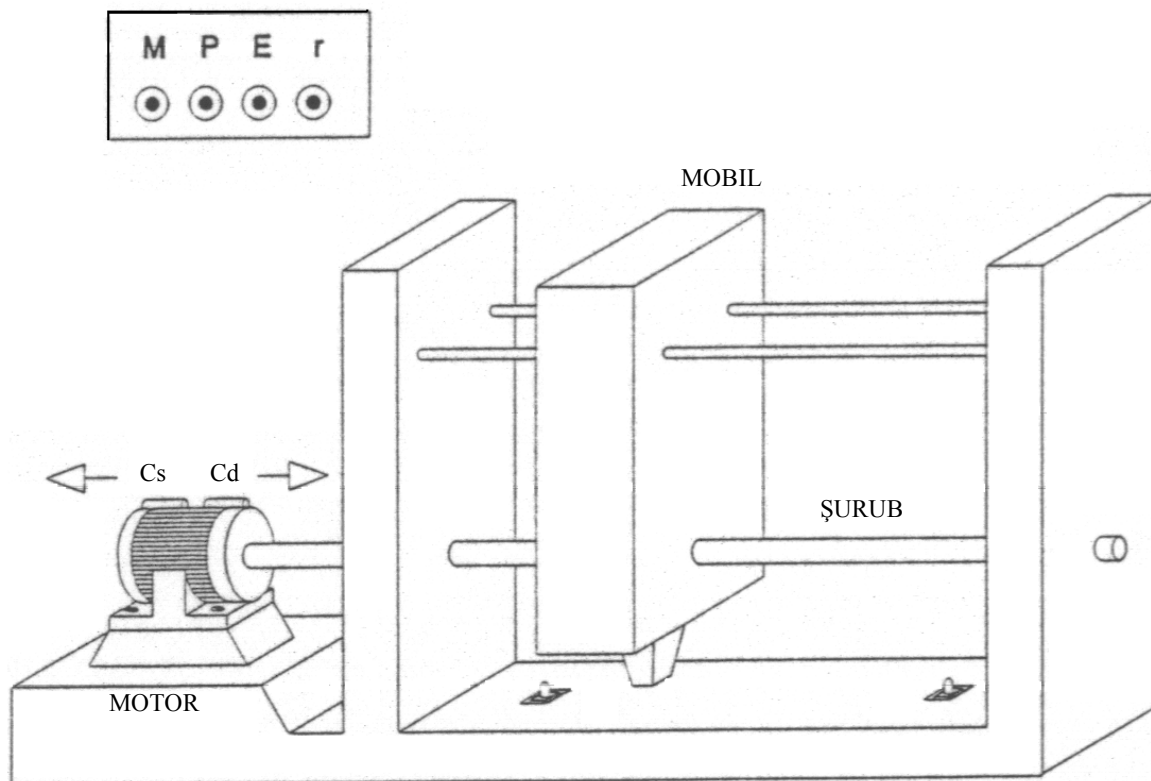


Fig. 5.5 Mișcarea oscilatorie a unui mobil

Element de execuție:

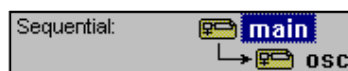
- 1 motor cu 2 sensuri de rotație

Elemente de măsură:

- 2 limitatoare de cursă

2. Soluția de automatizare

Pentru controlul acestei aplicații se alege un automat programabil de tip PEP Smart PLC pentru care se dezvoltă un proiect Isagraf. Proiectul conține 2 programe, un program principal și un program copil, numit „Osc”. Programul „Osc” este pornit și oprit de către programul principal și este responsabil cu realizarea mișcării oscilatorii și sesizarea apăsării butoanelor, conform specificațiilor aplicației.



Dicționarul de variabile globale:

Variabile de intrare digitale:

- M : buton de pornire mișcare oscilatorie
- P : buton de oprire motor
- E : buton de retragere în poziția de origine
- RST : buton de repornire
- L0 : limitator stânga
- L1 : limitator dreapta

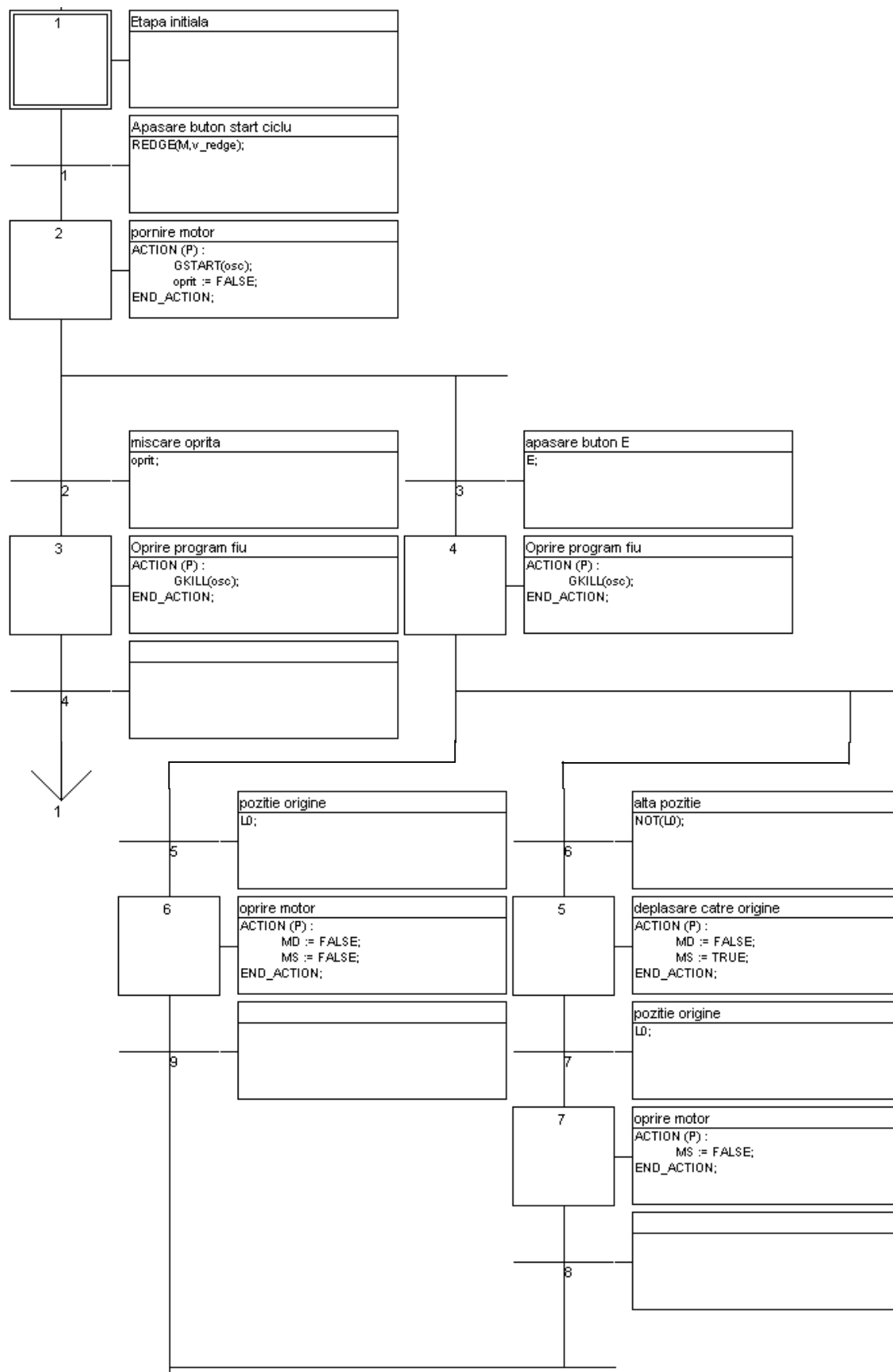
Variabile de ieșire digitale:

- MS : comandă motor stânga
- MD : comandă motor dreapta

Variabile interne de tip Boolean:

- oprit : are valoarea TRUE când mișcarea oscilatorie este oprită
- v_redge : necesar funcției REDGE

Programul „Main” este prezentat în figura 5.6.



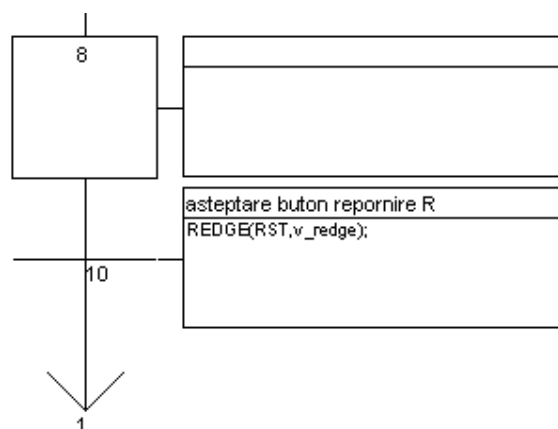
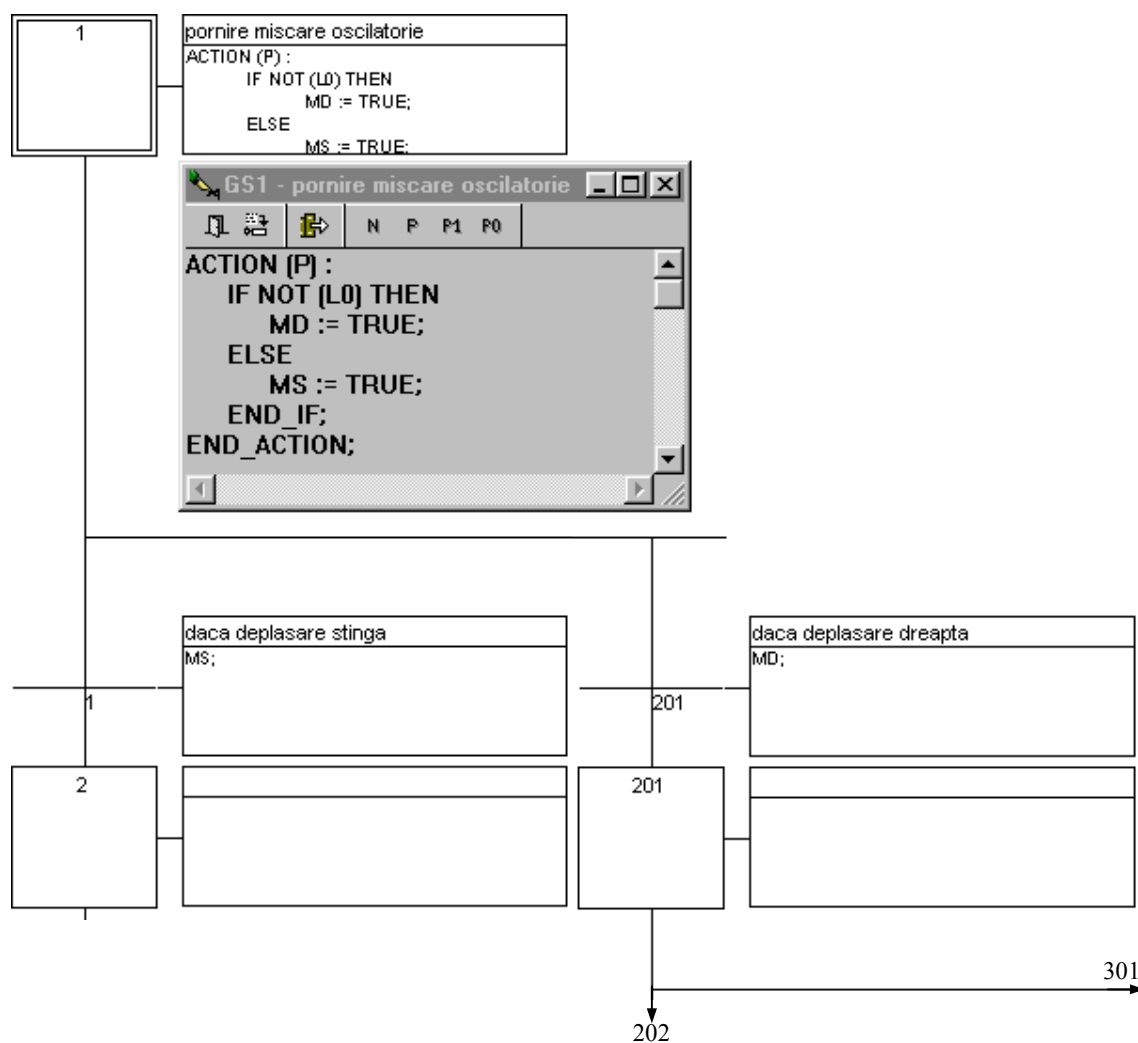


Fig.5.6 Programul „Main”

Programul fiu „Osc” este prezentat în figura 5.7.



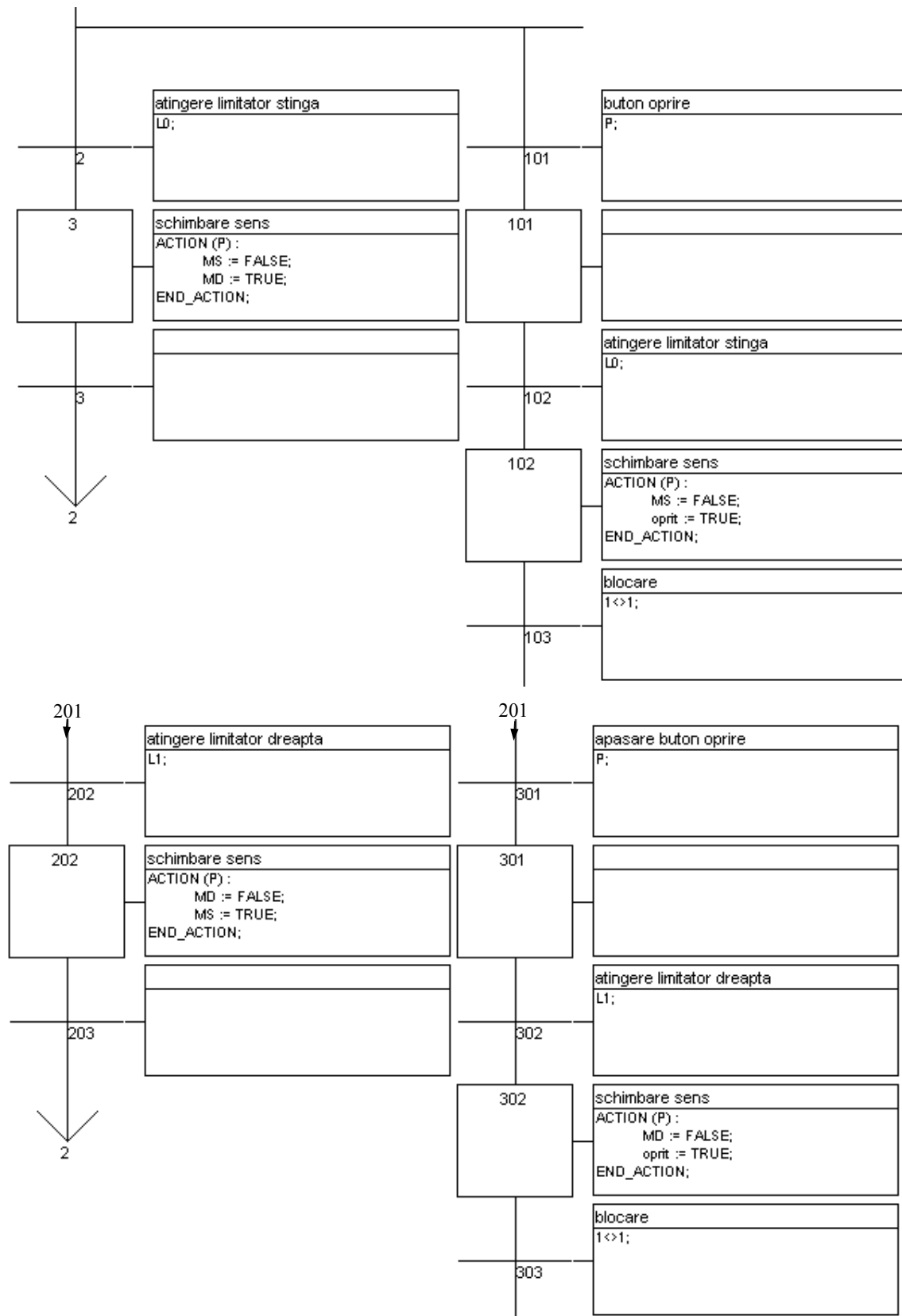


Fig. 5.7 Programul copil „Osc”

Comentarii :

- Funcția REDGE este folosită pentru detectarea impulsurilor produse prin apăsarea butoanelor
- Comunicația între programe este realizată prin intermediul variabilei „oprit”
- Programul principal oprește execuția programului copil „Osc” atunci când variabila „oprit” are valoare „TRUE” sau mișcarea este întreruptă de apăsarea butonului E

Propunere:

Să se modifice programul în condițiile în care la apăsarea butonului de oprire, ciclul să se încheie totdeauna când mobilul ajunge prima dată în partea dreaptă.

Problema 3: Detectia și expulzarea automată a sticlelor fără dop

1. Descrierea procesului:

Una din fazele de producție într-o linie de îmbuteliere constă în așezarea unui dop, ca urmare a încheierii secvenței de umplere. Sticlele se deplasează pe banda 1, separate de aceeași distanță și cu viteză constantă. Scopul aplicației este detectarea și extragerea sticlelor care ies din faza de închidere fără dopul corespunzător; pe lângă aceasta, dacă într-o perioadă determinată de timp (în acest caz 7 sticle), sunt rejectate mai mult de 3 sticle consecutive, trebuie activată o alarmă. Repornirea ciclului se face prin apăsarea butonului Pc. Pentru detecția sticlei defecte se conjugă acțiunile unui senzor inductiv, care detectează prezența dopului și un echipament foto electric care semnalează prezența unei sticle.

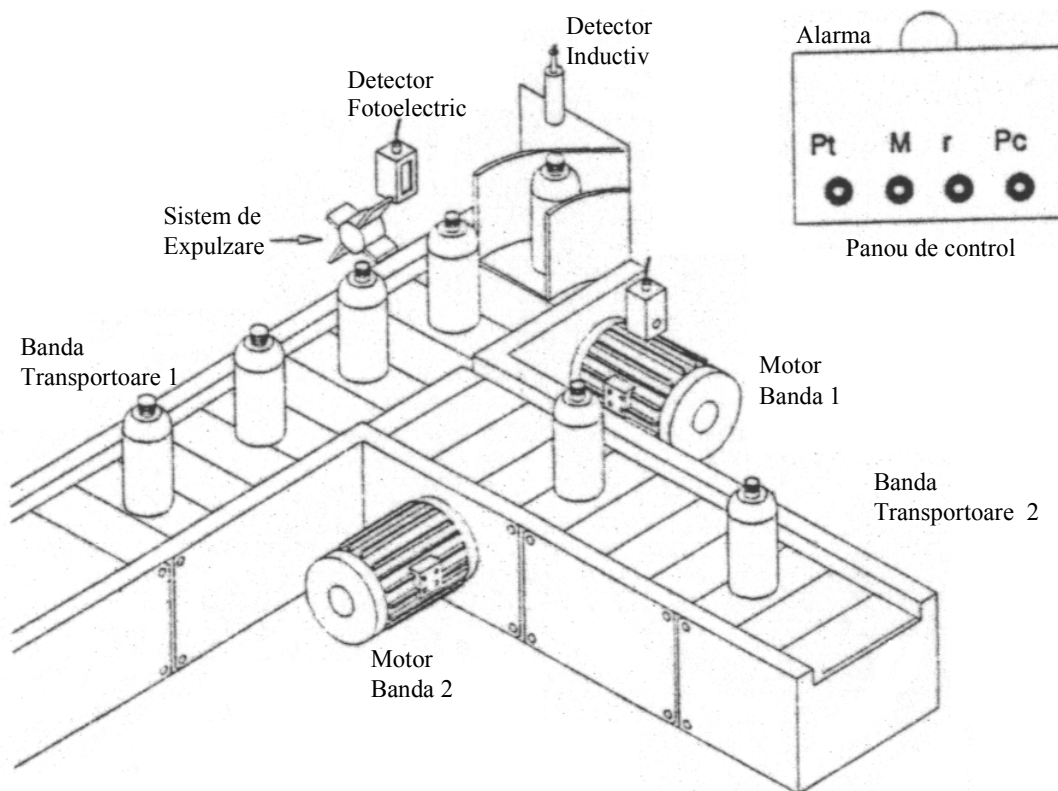


Fig. 5.8 Detectia și expulzarea sticlelor fără dop

Procesul este inițiat prin apăsarea butonului M, care produce pornirea benzii transportoare 1. Atunci când se detectează o sticlă fără dop, este oprită banda 1 și este pornită banda transportoare 2 (dacă era oprită). În momentul în care sticla fără dop se găsește în zona de expulzare, este activat mecanismul de expulzare. Banda 1 va fi repornită în momentul în care sticla fără dop nu se mai găsește în zona de expulzare (practic semnalul transmis de detectorul fotoelectric are valoarea logică fals). Banda 2 va fi oprită după 5 secunde de la începerea expulzării ultimei sticle.

Elemente de execuție:

- 2 motoare care acționează 2 benzi transportoare
- 1 dispozitiv de expulzare a sticlelor fără dop

Elemente de măsură:

- 1 detector inductiv pentru dopuri
- 1 detector fotoelectric pentru sticle

2. Soluția de automatizare:

Pentru controlul acestei aplicații se alege un automat programabil de tip PEP Smart pentru care se dezvoltă un proiect Isagraf, cu un singur program principal, dar cu 2 secțiuni programate și anume secțiunea secvențială și secțiunea de sfârșit, „end section” ca în figura 5.9.



Fig. 5.9 Secțiunile proiectului

Dicționarul de variabile globale:

❑ *Variabile de intrare booleene:*

- M : buton de pornire
- R : buton de rearmare
- I : detector inductiv
- F : detector fotoelectric
- Pc : buton de punere la zero și oprire alarmă

❑ *Variabile de ieșire booleene:*

- Banda_1 : comandă pornire / oprire bandă 1
- Banda_2 : comandă pornire / oprire bandă 2
- Alarma : comandă alarmă
- Expulzare : comandă dispozitiv de evacuare a sticlelor fără dop

Variabile interne de tip Integer:

- Nr_sticle : contorizează numărul de sticle fără dop expulzate consecutiv

Variabile interne de tip timer:

- Timer : folosit la contorizarea celor 5 secunde de activare a conveiorului 2

Programul principal „main” este prezentat în figura 5.10

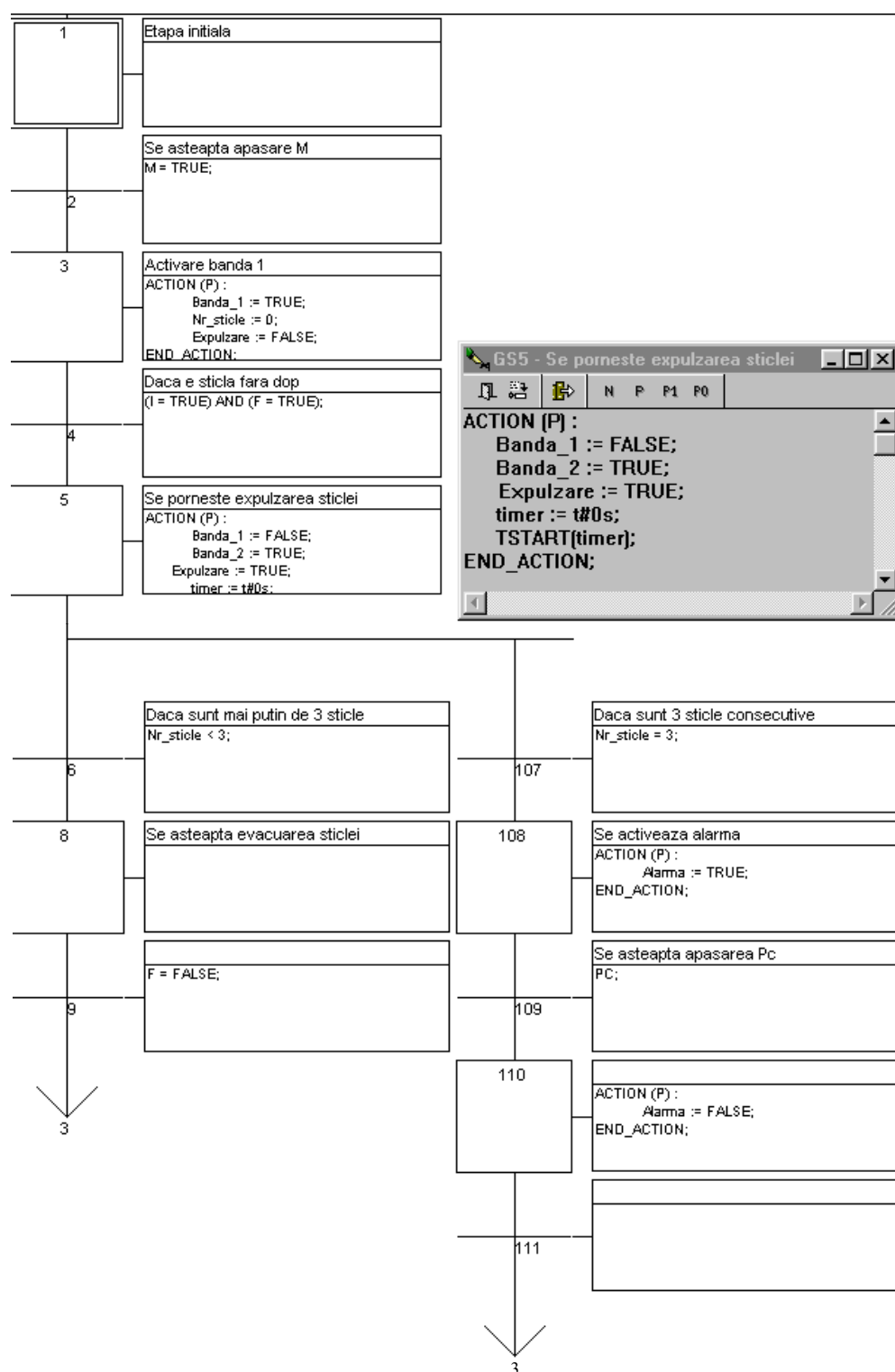


Fig. 5.10 Programul principal, secțiunea secvențială

Programul „timing” din secțiunea de end este prezentat în figura 5.11

```
IF (timer >= t#5s) THEN
  Banda_2 := FALSE;
  TSTOP (timer);
  timer := t#0s;
ELSE
  ENDIF;
```

Fig. 5.11 Programul principal, secțiunea de „end”

Observații:

- secțiunea de end este necesară pentru ca banda 2 să poate fi oprită în orice moment dacă perioada ei de activare a expirat
- deoarece secțiunea de end se execută la fiecare ciclu automat, testarea timer_ului se va face la fiecare ciclu automat.
- contorizarea timpului este făcută cu ajutorul funcțiilor TSTART și TSTOP
- dacă o sticlă trebuie să fie expulzată în timp ce o altă sticlă se găsește pe conveiorul 2, timer_ul este resetat și este reactivată incrementarea timer_ului

Propuneri:

- Să se construiască o diagramă Ladder pentru un automat de tip Allen Bradley, care să controleze acest proces

Problema 4: Stație automată de spălat autovehicule

1. Descrierea procesului:

Scopul proiectării acestui sistem de control îl reprezintă automatizarea unei stații de spălat autovehicule. Vehiculele vor trebui să treacă succesiv prin 4 posturi de lucru, înmuiere, spălare cu detergent, clătire și uscare. Procesul este inițiat de apăsarea unui buton de pornire, care determină activarea benzii transportoare iar vehiculele vor trece succesiv prin cele 4 posturi. Bariera, în condiții normale, trebuie să stea ridicată și semaforul dezactivat. Când în stație sunt detectate 4 vehicule, câte unul în fiecare post, bariera trebuie coborâtă și semaforul activat, indicând faptul că nu se mai poate trece. Atât bariera cât și semaforul rămân în această stare până când cele 4 vehicule au părăsit stația, moment în care bariera trebuie ridicată și semaforul dezactivat.

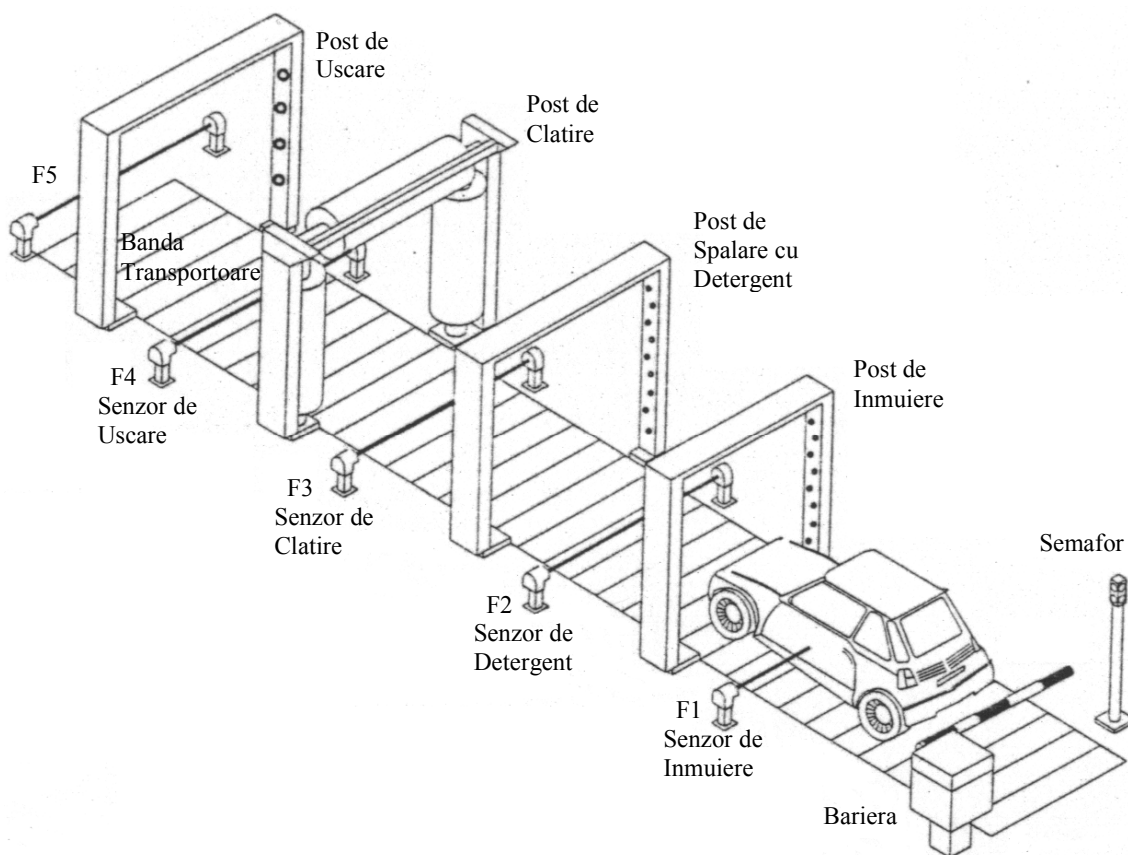


Fig. 5.12 Stație automată de spălat autovehicule

În momentul în care se activează celula fotoelectrică 1, se va activa postul 1. Când se activează fotocelula 2 iar fotocelula 1 nu mai este activată, se dezactivează postul 1. În momentul activării fotocelulei 2 se activează postul 2. Acesta va fi dezactivat când fotocelula 2 nu este activă dar fotocelula 3 este activă. Analog pentru postul 3. Postul 4 se dezactivează când fotocelula 4 se dezactivează iar fotocelula 5 se activează..

Elemente de execuție:

- motorul benzii transportoare
- motorul barierei cu 2 senzuri de rotație
- 1 semafor
- 4 posturi de lucru

Elemente de măsură:

- 5 celule fotoelectrice
- 2 limitatoare de cursă ale barierei

2. Soluția de automatizare

Pentru controlul acestei aplicații se alege un automat programabil de tip PEP Smart pentru care se dezvoltă un proiect Isagraf, proiect ce consta din 6 programe SFC ce rulează în paralel. Structura proiectului Isagraf este prezentată în figura 5.13.

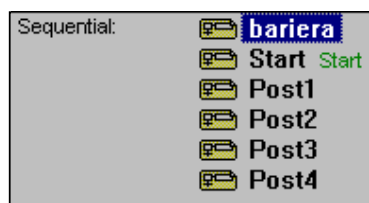


Fig. 5.13 Programele componente ale proiectului Isagraf

Dicționarul de variabile globale:

□ Variabile de intrare booleene:

- start : contact de pornire
- Limita_SUS : limitator de cursă sus pentru barieră
- Limita_JOS : limitator de cursă jos pentru barieră
- Foto_1 : fotocelula postului 1
- Foto_2 : fotocelula postului 2
- Foto_3 : fotocelula postului 3
- Foto_4 : fotocelula postului 4
- Foto_5 : fotocelula postului 5

□ Variabile de ieșire booleene:

- Banda : comandă pornire / oprire bandă
- Bariera_sus : comandă ridicare barieră
- Bariera_jos : comandă coborâre barieră
- Semafor : comandă activare / dezactivare semafor
- Înmuire : comandă activare dezactivare post înmuire
- Detergent : comandă activare dezactivare post detergent
- Clătire : comandă activare dezactivare post clătire
- Uscare : comandă activare dezactivare post uscare

Programul „bariera” este prezentat în figura 5.14

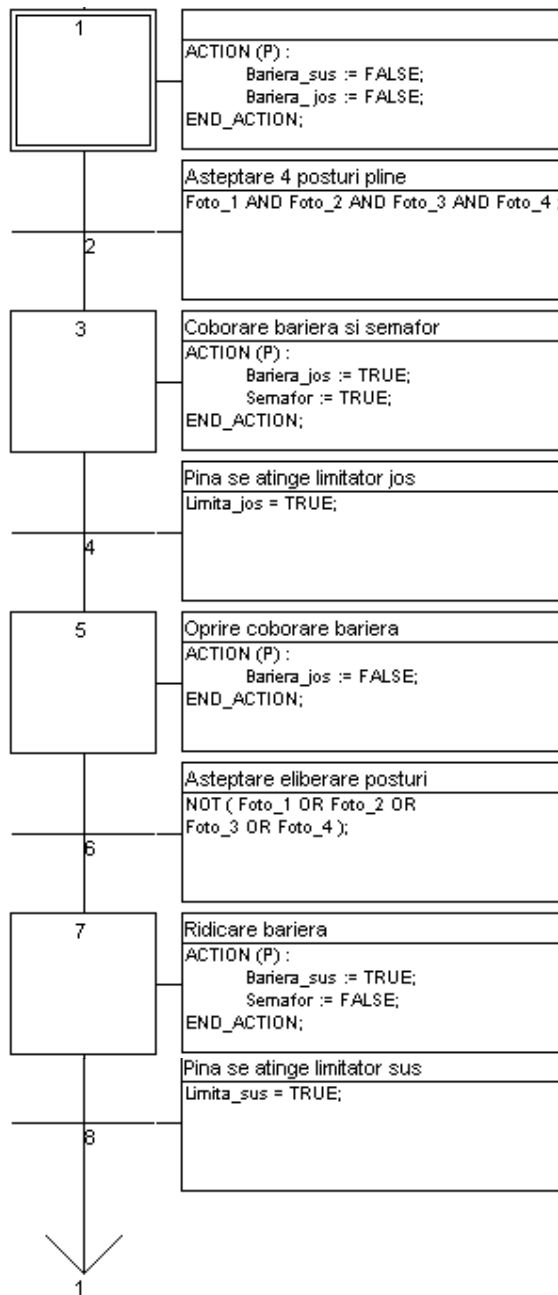


Fig. 5.14 Programul „Bariera”

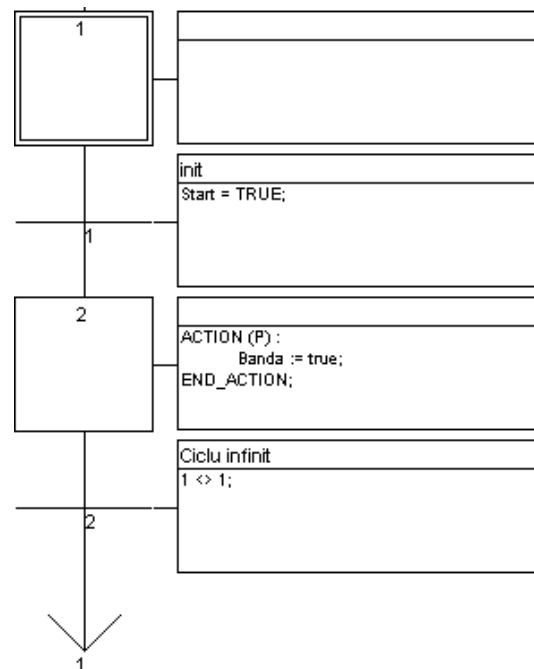


Fig.5.15 Programul „Start”

Programul „Post1” este prezentat în figura 5.16.

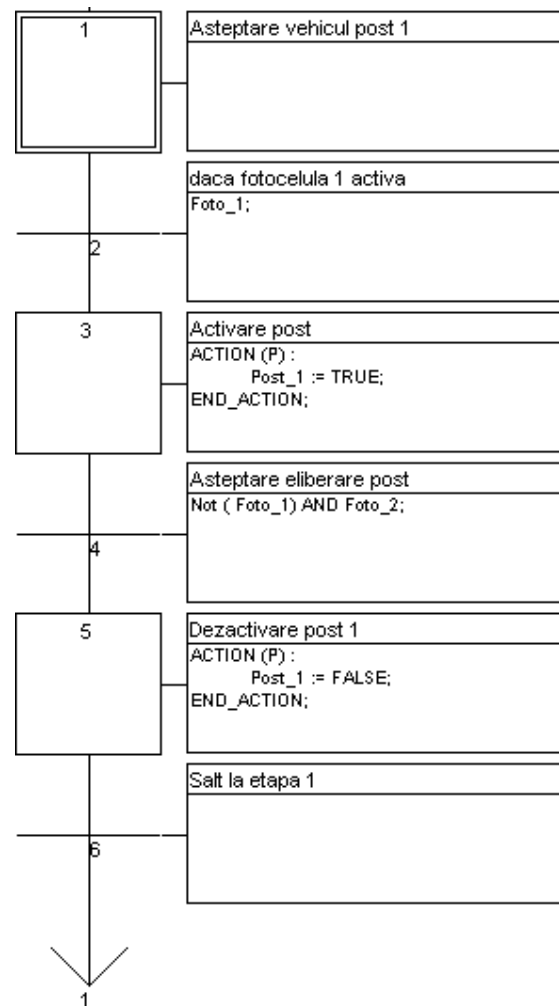


Fig.5.16 Programul „Post1”

Programul „Post2” este prezentat în figura 5.17.

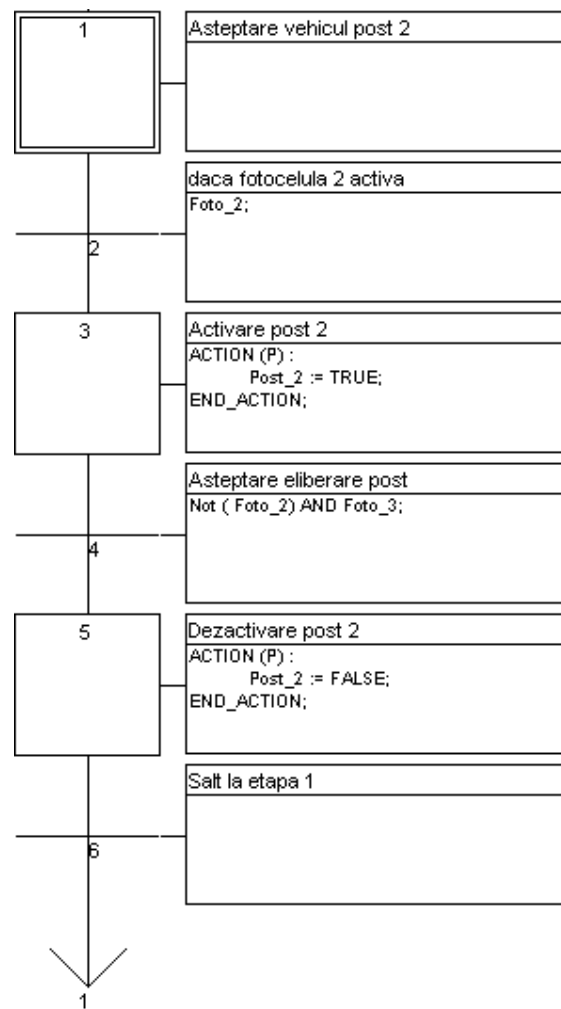


Fig.5.17 Programul „Post2”

Programul „Post3” este prezentat în figura 5.18.

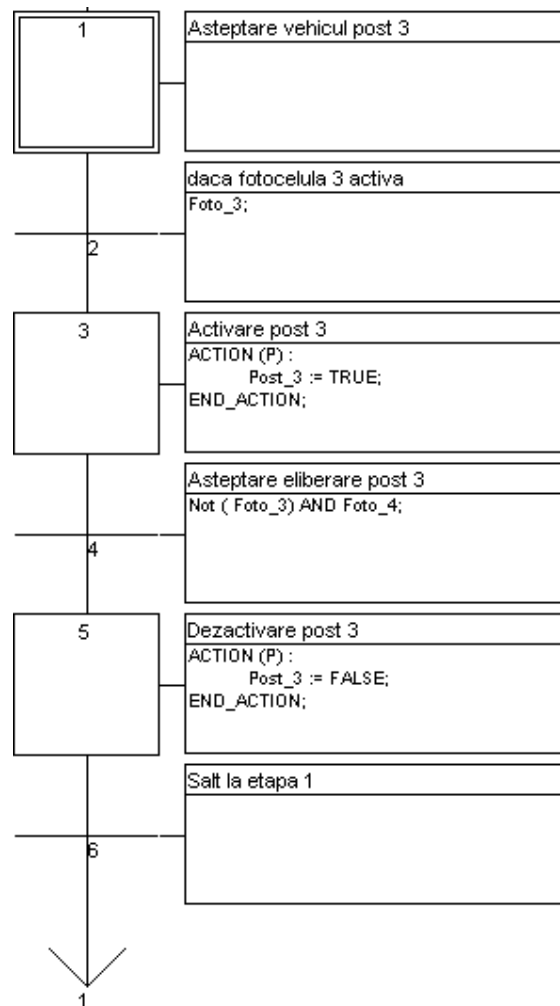


Fig.5.18 Programul „Post3”

Programul „Post4” este prezentat în figura 5.19

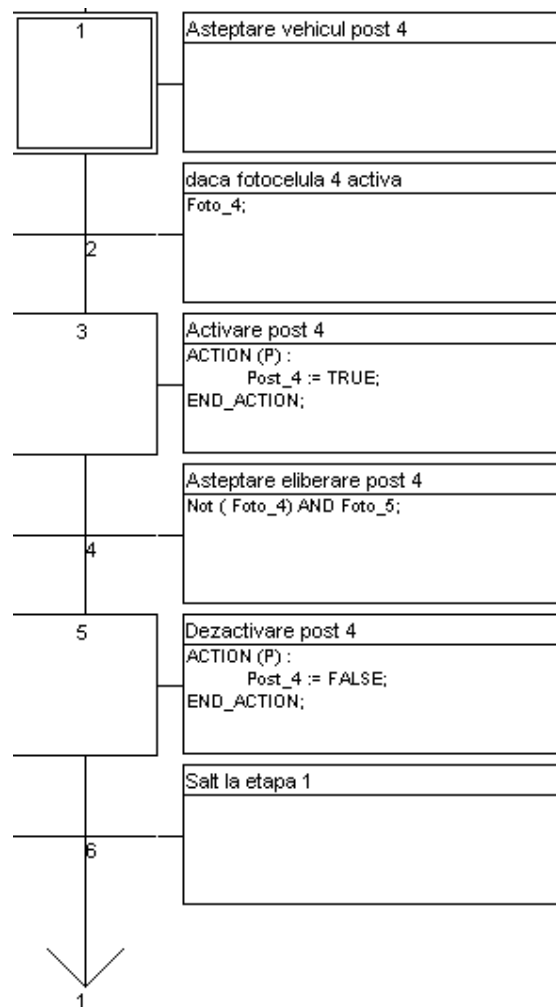


Fig.5.19 Programul „Post4”

Comentarii:

- Toate programele dezvoltate sunt independente și rulează în paralel, fapt ce ușurează foarte mult înțelegerea programului

Propunere:

- Să se modifice programul în cazul în care bariera se va ridica atunci când se eliberează primul post de lucru
- Să se modifice proiectul astfel încât să se dezvolte un singur program care să automatizeze acest proces

Problema 5: Elevator clasificator de pachete

1. Descrierea procesului:

Pe o bandă transportoare vin 2 tipuri de pachete (mic și mare). Tipul pachetului este determinat de un cântar, ulterior pachetele fiind transportate în direcții diferite în funcție de tipul pachetului. Procesul pornește cu transportul unui pachet către cântar; aici pachetul este cântărit fiind astfel identificat în funcție de greutatea citită. În continuare pachetul este transportat pe banda 1 până la planul elevator. Cilindrul C ridică pachetele. Apoi pachetele sunt transportate diferit; pachetele mici sunt plasate pe banda 2 de cilindrul A, iar pachetele mari sunt așezate pe banda 3 de cilindrul B. Cilindrul elevator C se retrage doar când cilindrii A și B au atins poziția finală.

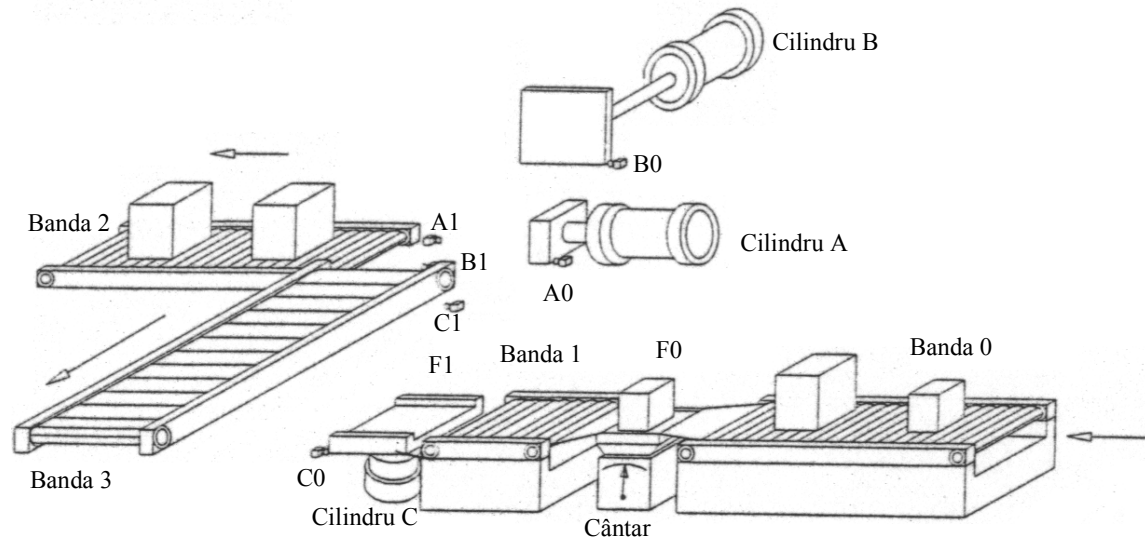


Fig. 5.20 Elevator clasificator pentru pachete

Elemente de execuție:

- 3 cilindri cu dublu efect (A, B, C)
- o basculă însărcinată cu clasificarea pachetelor
- 4 benzi transportoare

Elemente de măsură:

- 6 limitatoare de cursă
- 2 detectoare de poziție

2. Soluția de automatizare

Pentru controlul acestei aplicații s-a ales un automat programabil de tip PEP Smart pentru care s-a dezvoltat un proiect Isagraf ce cuprinde 5 programe secvențiale ce rulează în paralel și un program în secțiunea de „Begin”, program ce se execută la începutul fiecărui ciclu automat. Structura proiectului Isagraf este prezentată în figura 5.21

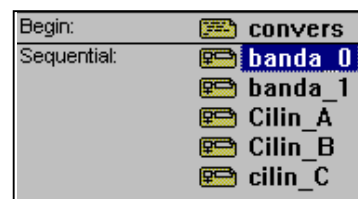


Fig. 5.21 Structura proiectului Isagraf

*Dicționarul de variabile globale:**Variabile de intrare booleene:*

- foto_0: fotocelula 0
- foto_1 : fotocelula 1
- A0 : Limită retragere cilindru A
- A1 : Limită avans cilindru A
- B0 : Limită retragere cilindru B
- B1 : Limită avans cilindru B
- C0 : Limită coborâre cilindru C
- C1 : Limită ridicare cilindru C

Variabile de ieșire booleene:

- Banda_0 : activare / dezactivare banda 0
- Banda_1 : activare / dezactivare banda 1
- Banda_2 : activare / dezactivare banda 2
- Banda_3 : activare / dezactivare banda 3
- A_avansat : avans cilindru A
- A_retras : retragere cilindru A
- B_avansat : avans cilindru B
- B_retras : retragere cilindru B
- C_ridicare : ridicare cilindru C
- A_retragere : retragere cilindru A

Variabile interne booleene:

- eroare : cod de eroare la cântărirea pachetelor

Variabile globale analogice

- cantar : variabilă internă, reprezintă valoarea reală a greutății de pe cântar (integer)
- traductor_cantar : variabilă de intrare, valoarea primită de la traductorul cântarului (integer, între 0 – 4096)
- pachet_actual : variabilă internă, în care se memorează tipul pachetului actual ce urmează a fi transportat
- pachet_viitor : variabilă internă, în care se memorează tipul pachetului de pe cântar (următorul ce va fi transportat)
- lim_inf : constantă, greutatea minimă a pachetului mic
- lim_sup_mic : greutatea maximă a pachetului mic
- mic : constantă cu valoarea 1
- mare : constantă cu valoarea 2

Programul “Banda_0” este prezentat în figura 5.22.

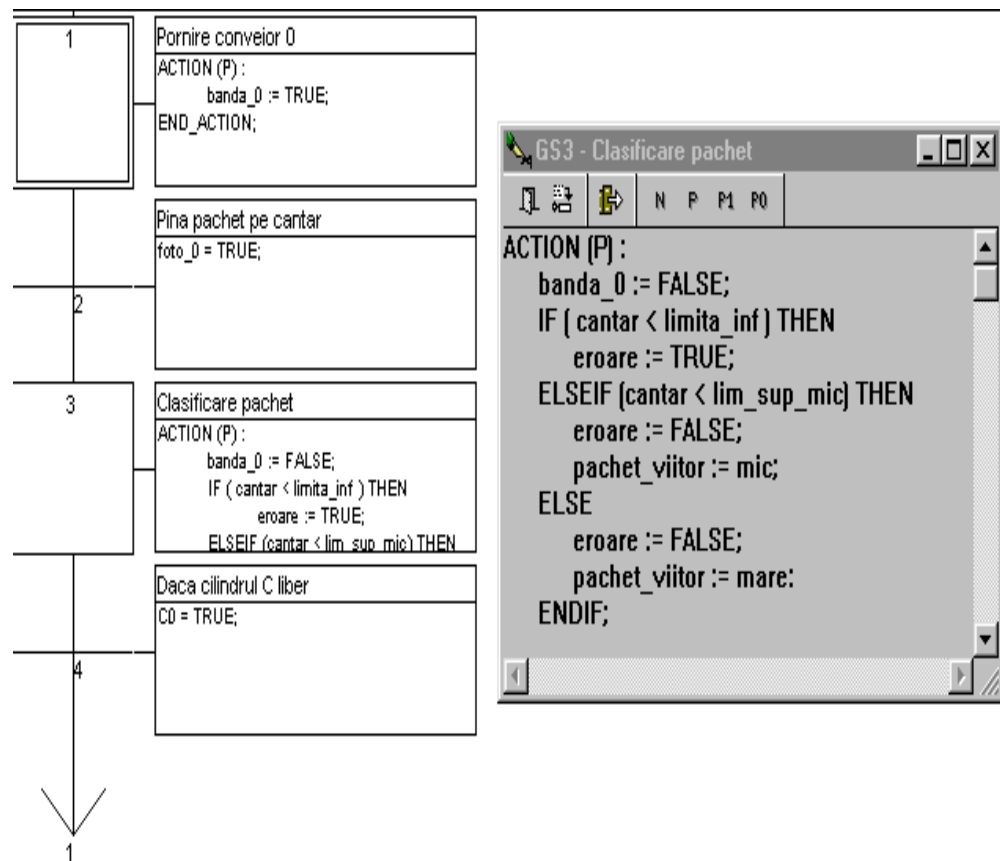


Fig.5.22 Programul Banda_0

Programul “Banda_1” este prezentat în figura 5.23.

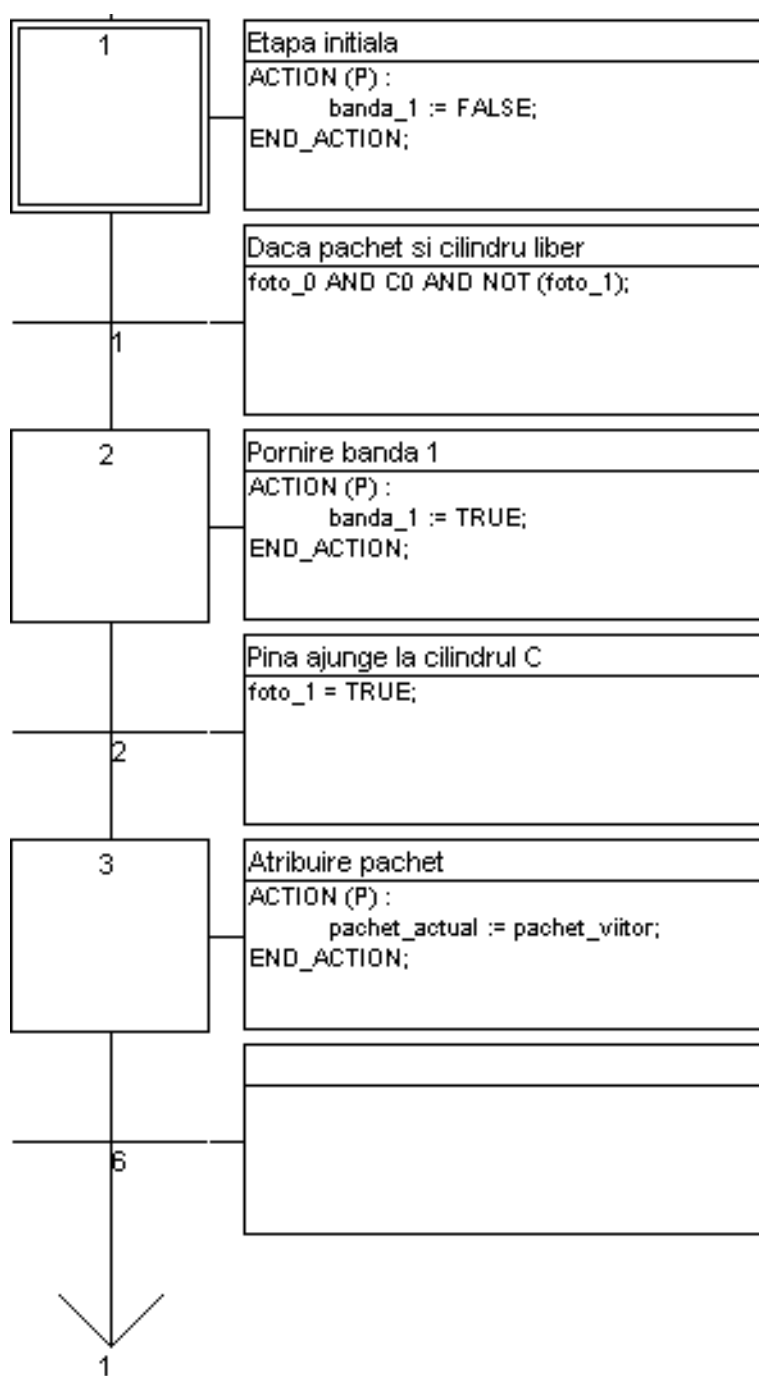


Fig.5.23 Programul „Banda_1”

Programul “Cilin_A” este prezentat în figura 5.24.

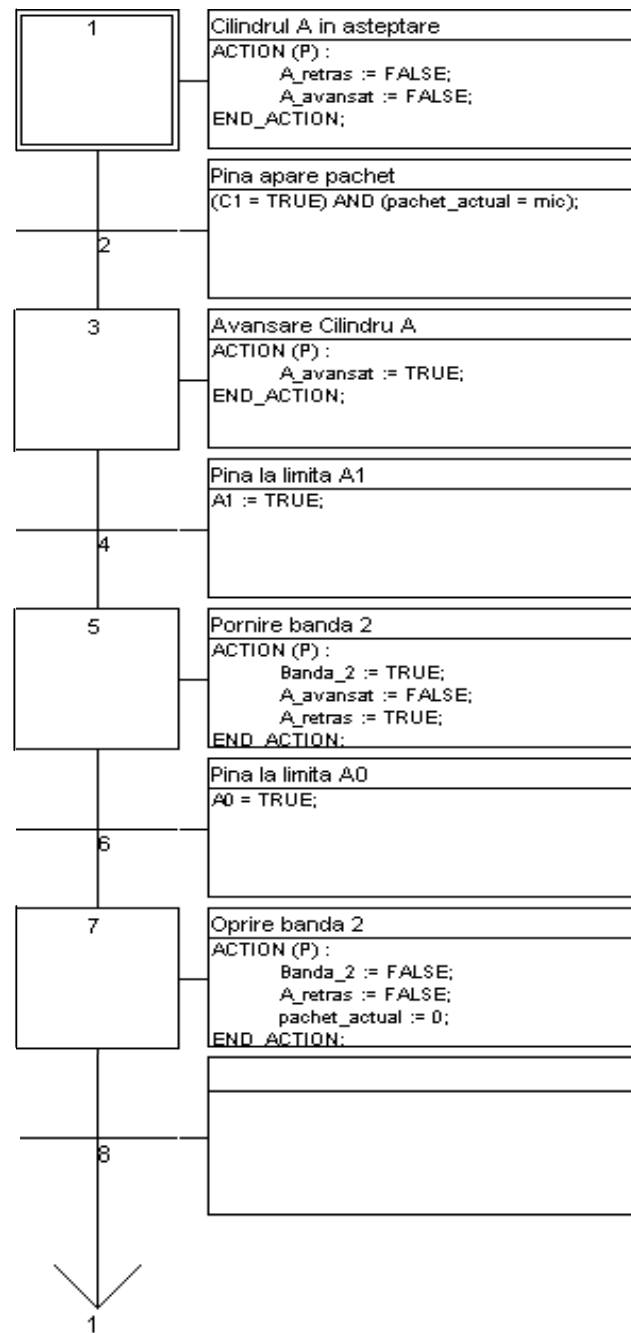


Fig.5.24 Programul Cilin_A

Programul “Cilin_B” este prezentat în figura 5.25

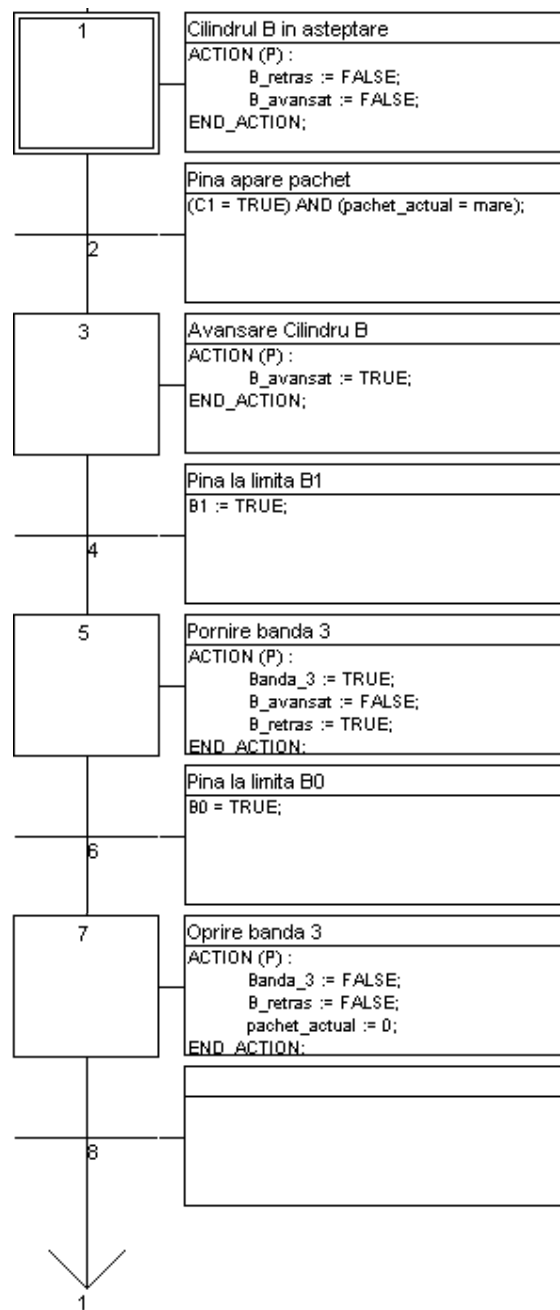


Fig.5.25 Programul Cilin_B

Programul “Cilin_C” este prezentat în figura 5.26

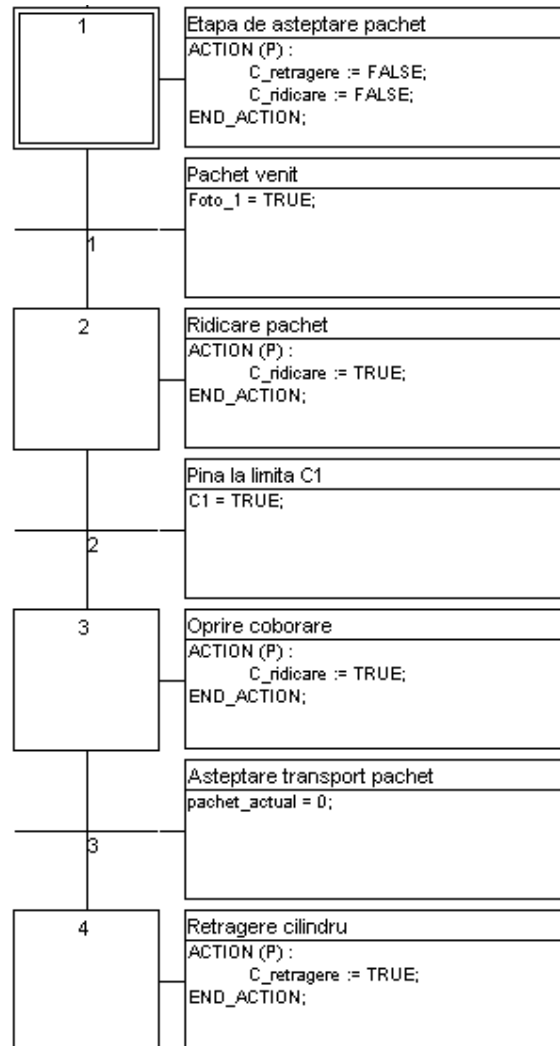


Fig.5.26 Programul “Cilin_C”

Programul „convers” din secțiunea Begin

Programul “convers” realizează conversia din unități CAN (Convertor Analog Numeric) în valori exprimate în unități de măsură ingineresti. Intrarea analogică a modului de intrare lucrează pe 12 biți și măsoară un curent de 0 - 20 mA , dar traductorul de la cântar generează un curent de 4 - 20mA. În această situație trebuie făcută o translație de scală. Se observă că la valoarea minimă a domeniului de măsură traductorul generează 4 mA, corespunzătoare valorii 819 citită de automat, valoare pentru care automatul trebuie să indice valoarea minimă a mărimii măsurate. Astfel formula de conversie este:

$$\text{Val_ing} = (\text{Val_cit} - 819) * (\text{Ds} - \text{Di}) / (4095 - 819)$$

Unde :

- Val_ing – valoarea în unități ingineresti
- Val_cit – valoarea citită în unități CAN
- Di – domeniul inferior de măsură
- Ds – domeniul superior de măsură

În cazul nostru Ds=100, Di=0, astfel încât instrucțiunea ce se execută în programul „convers” este

cantar := INT(((REAL(traductor_cantar)- 819)* 100.0) / (3276) ;

Propunere:

- Să se modifice programul de conversie în cazul în care traductorul de temperatură are ca domeniu -15 +150 grade și generează un curent în gama 2-10 mA

Problema 6: Controlul temperaturii unui lichid

1. Descrierea procesului:

Problema constă în menținerea temperaturii unui lichid între 2 valori determinate (60 și 65 grade), în timp ce nivelul în cele 2 rezervoare păstrează o capacitate determinată. Dacă temperatura se găsește între limitele fixate, valva 1 se va deschide iar valva 2 se va deschide pînă când rezervorul 2 ajunge la capacitatea fixată; în acel moment, valva 2 se va închide și va rămâne așa pînă când lichidul din rezervorul 2 se va găsi sub limita fixată. Când temperatura atinge marginile de temperatură fixată, valvele de intrare și de ieșire se vor închide (indiferent dacă rezervorul 2 și-a recuperat nivelul) și vor rămâne închise pînă când temperatura va fi reglată. Totdeauna va fi prioritară variația temperaturii față de variația nivelului de lichid.

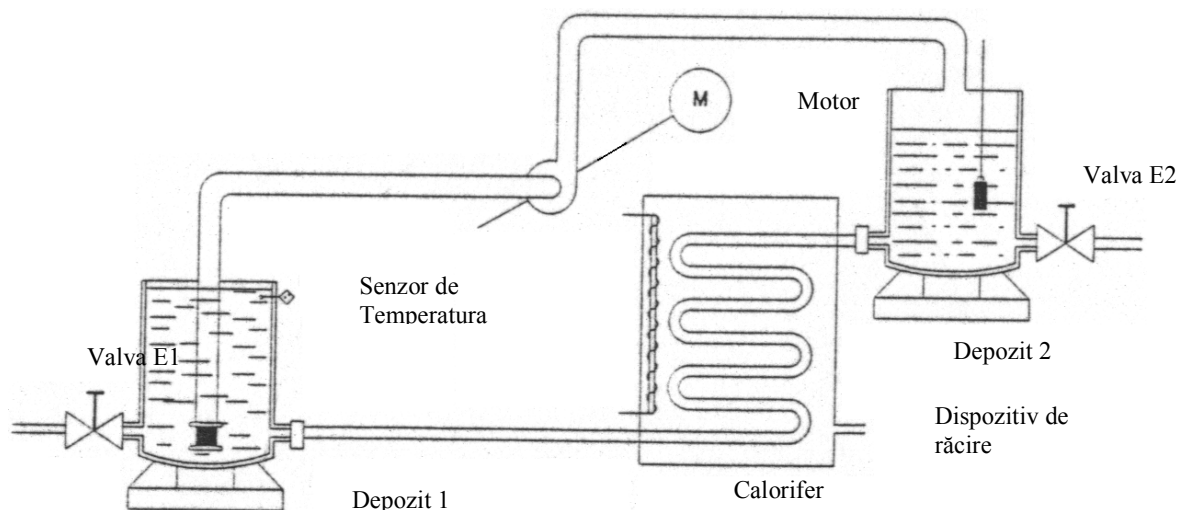


Fig. 5.27 Controlul temperaturii unui lichid

Elemente de execuție:

- 1 pompă cu motorul său
- 1 motor al echipamentului de pompat aerul
- 2 electrovalve
- 1 rezistență

Elemente de măsură:

- 2 senzori de nivel
- 1 traductor de temperatură

2. Soluția de automatizare

Pentru controlul acestei aplicații s-a ales un automat programabil de tip PEP Smart pentru care s-a dezvoltat un proiect Isagraf ce cuprinde un program principal „Main” și programul secțiunii de Begin, numit „Convers”.

Dicționarul de variabile globale:

Variabile de intrare booleene:

- senzor_plin : are valoarea TRUE când depozitul 2 este plin
- senzor_gol : are valoarea TRUE când depozitul 2 este gol

Variabile de ieșire booleene:

- valva_1 : comanda electrovalvei 1, atunci când are valoarea TRUE electrovalva 1 se închide
- valva_2 : comanda electrovalvei 2, atunci când are valoarea TRUE electrovalva 2 se închide
- rezistentă : comanda cuplarea / decuplarea rezistentei
- racire : comandă dispozitivul de pompat aer
- pompa : comandă motorul pompei

Variabile analogice :

- temp : variabilă internă de tip Real, reprezintă temperatura apei în grade Celsius
- temp_citita : variabilă de intrare de tip Real ce reprezintă valoarea analogică (între 0 și 4095) primită de la traductorul de temperatură

Programul “Convers” al secțiunii de Begin:

Programul “convers” realizează conversia din unități CAN (Convertor Analog Numeric) în valori exprimate în unități de măsură ingineresti. Intrarea analogică a modulului de intrare lucrează pe 12 biți și măsoară un curent de 0 - 20 mA , dar traductorul de la cântar generează un curent de 4 - 20mA. În această situație trebuie făcută o translație de scală. Se observă că la valoarea minimă a domeniului de măsură traductorul generează 4 mA, corespunzătoare valorii 819 citită de automat, valoare pentru care automatul trebuie să indice valoarea minimă a mărimii măsurate. Astfel formula de conversie este:

$$\text{Val_ing} = (\text{Val_cit} - 819) * (\text{Ds} - \text{Di}) / (4095 - 819)$$

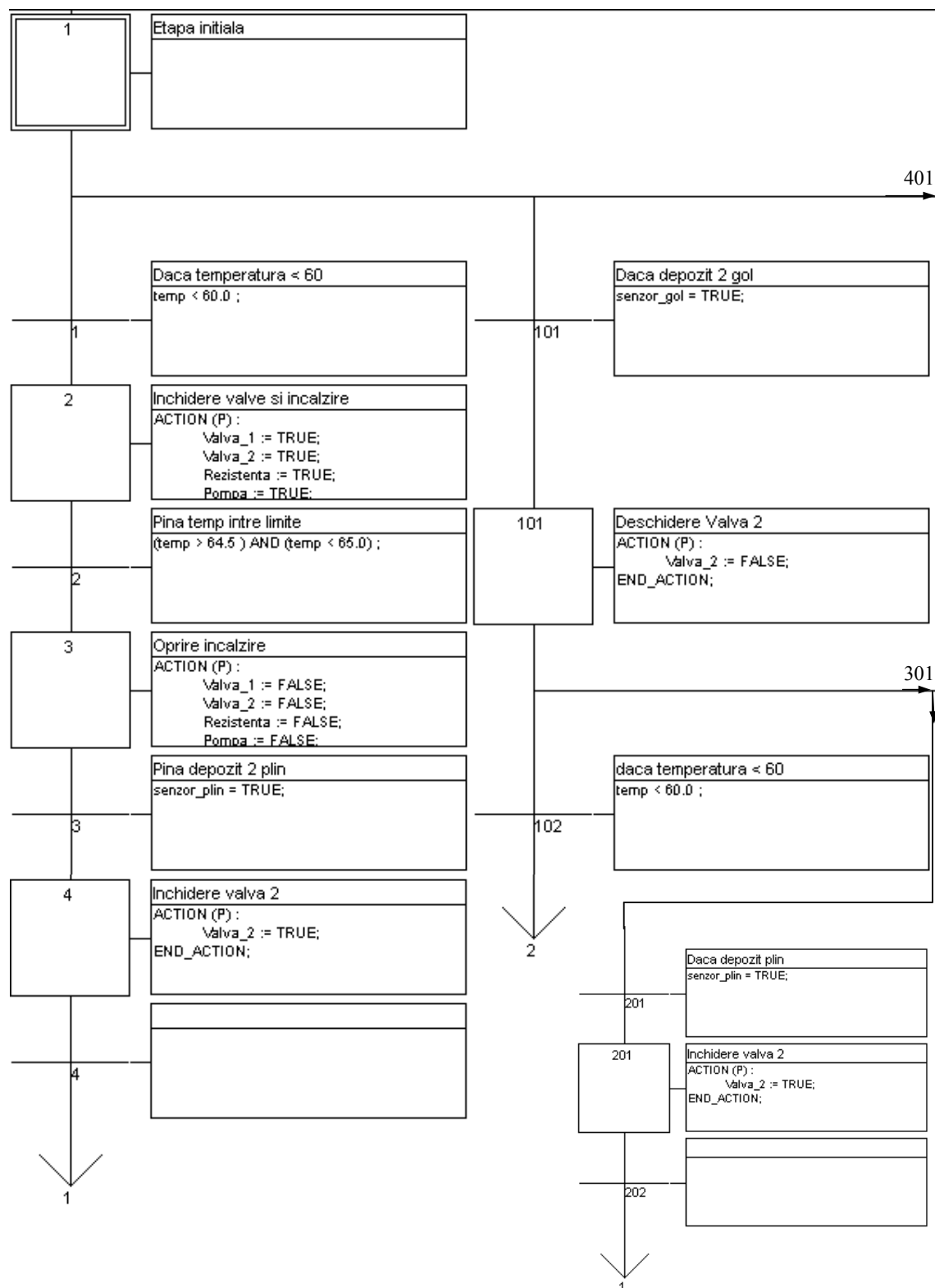
Unde :

- Val_ing – valoarea în unități ingineresti
- Val_cit – valoarea citită în unități CAN
- Di – domeniul inferior de măsură
- Ds – domeniul superior de măsură

În cazul nostru Ds=100, Di=0, astfel încât instrucțiunea ce se execută în programul „convers” este

temp := INT(((REAL(temp_citita)- 819)* 100.0) / (3276) ;

Programul „Main” este prezentat în figura 5.28



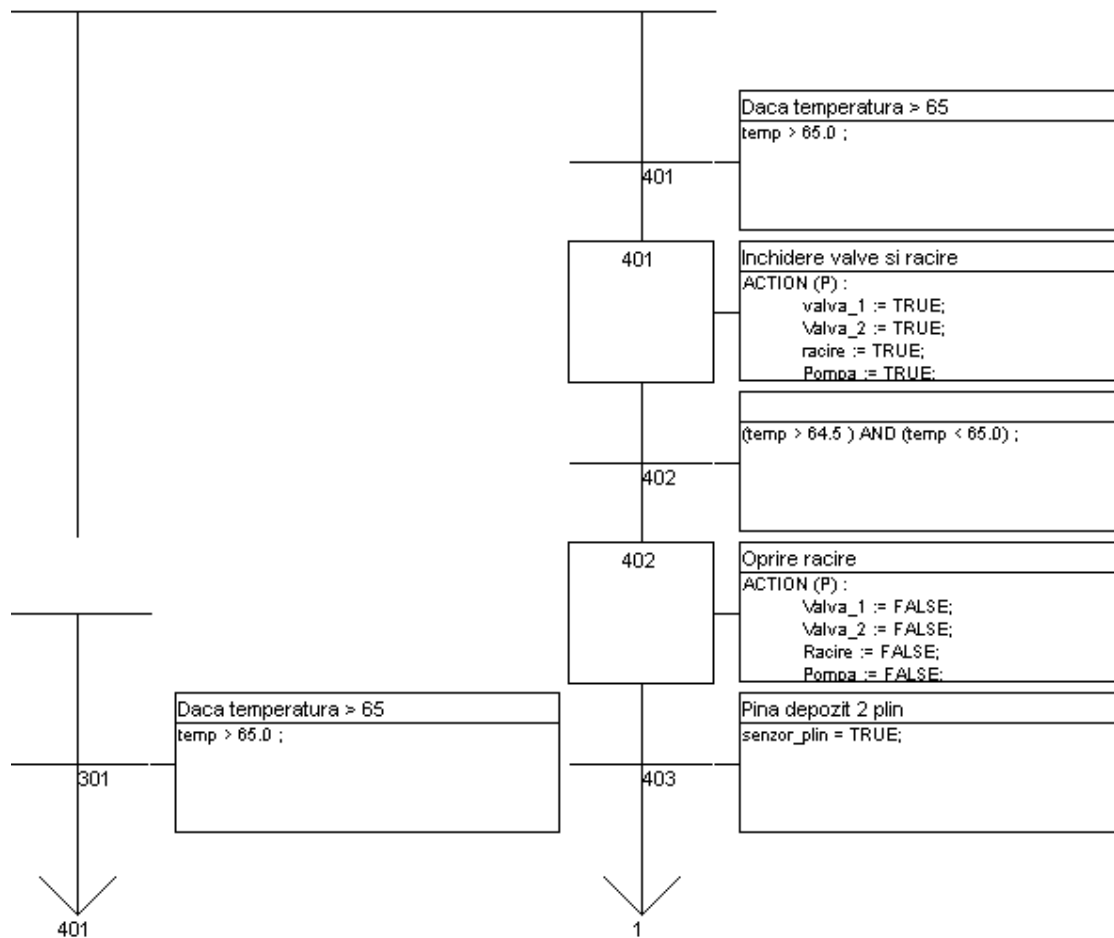


Fig.5.28 Programul „Main”

Comentarii:

- Programul folosește variabile analogice deoarece evenimentele ce produc modificări în sistem depind de o mărime cu variație continuă (temperatura)

Problema 7 : Dozare și malaxare automată

1. Descrierea procesului:

Un malaxor pivotant primește produsele A și B cântărite de bascula C și brichete solubile aduse una câte una pe o bandă transportoare. Automatizarea permite realizarea unei amestecări a celor 3 produse. Ciclul de realizat este următorul: la acționarea butonului de alimentare se pornește cântărirea și alimentarea produselor în următorul mod:

- cântărirea produsului A prin deschiderea valvei Va, până la referința A
- cântărirea produsului B prin deschiderea valvei Vb, până la referința B
- apoi, golirea basculei în malaxor prin deschiderea valvei Vc până la referința zero
- simultan cu precedentele operații are loc malaxorului alimentarea cu 2 brichete solubile

Ciclul se termină cu rotația malaxorului un anumit timp t și apoi pivotarea lui, menținându-se rotația în timpul golirii.

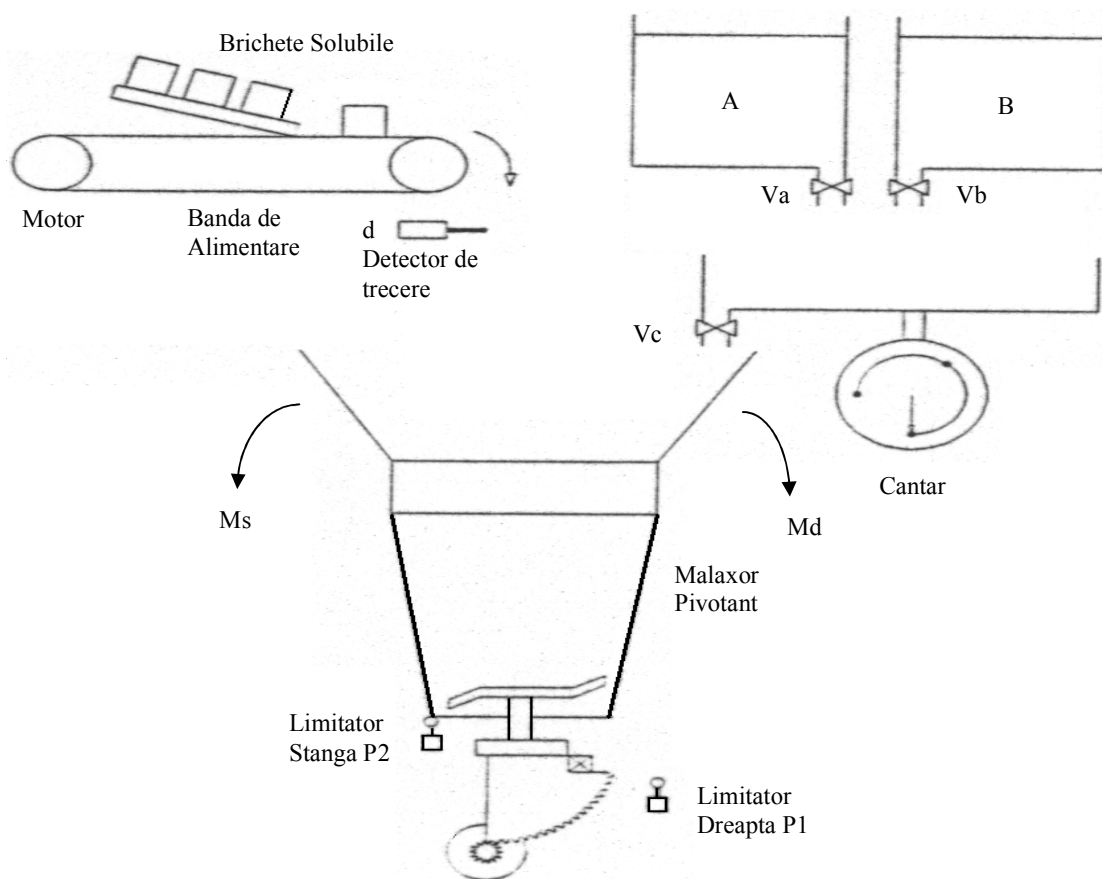


Fig. 5.29 Proces de dozare și malaxare

Elemente de execuție:

- 3 electrovalve (Va, Vb, Vc)
- Motorul benzii transportoare cu un singur sens de rotație
- Motorul de rotație al malaxorului
- Motorul de pivotare al malaxorului, cu două sensuri de rotație

Elemente de măsură:

- 3 senzori de greutate, pentru referințele A, B și zero

- 2 limitatoare de cursă
- 1 detector de trecere

2. Soluții de automatizare

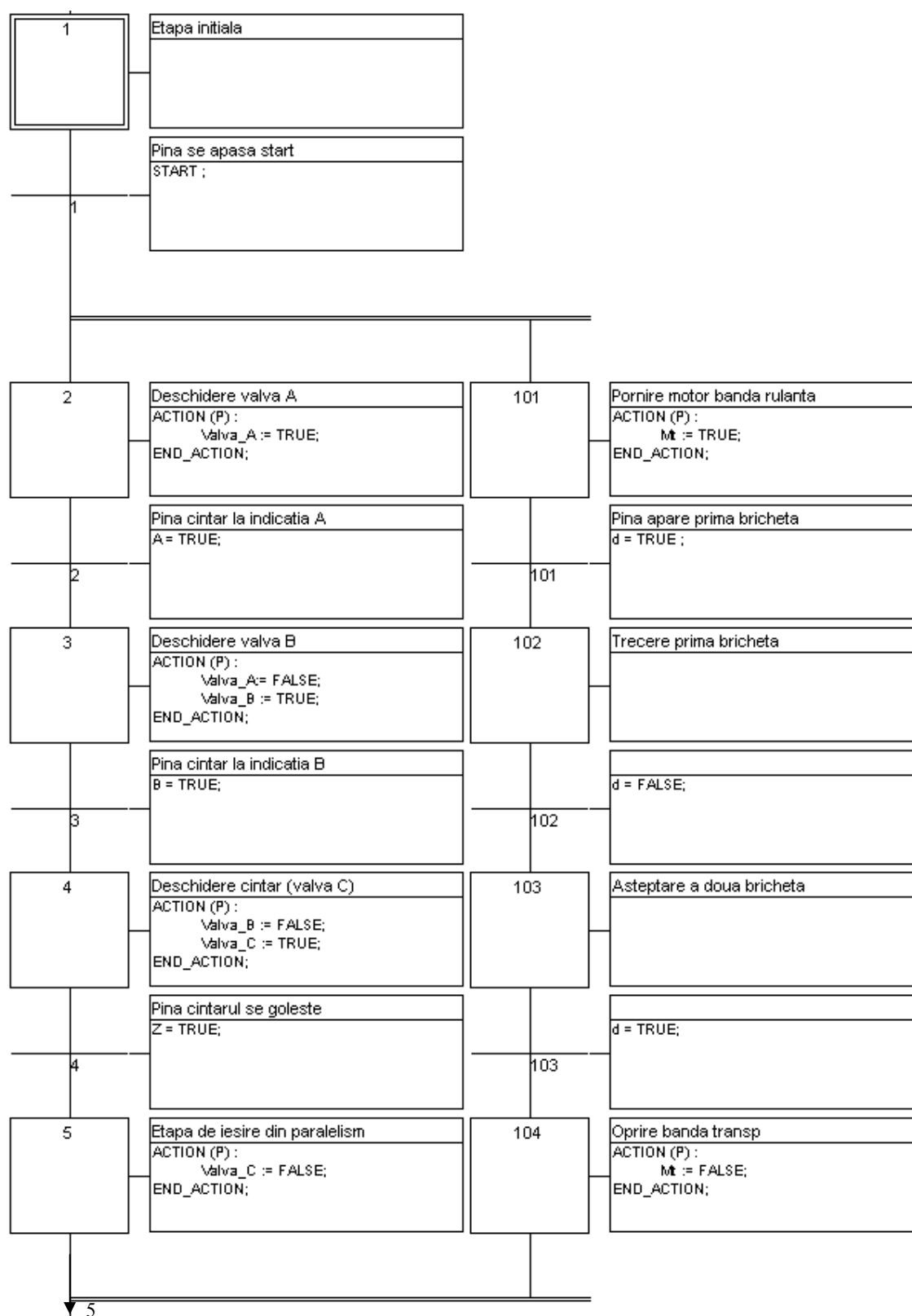
2.1 Varianta 1 : implementarea în mediul Isagraf

Această variantă presupune utilizarea unui automat programabil de tip PEP Smart, pentru care s-a dezvoltat un proiect Isagraf ce cuprinde 1 program principal.

Dicționarul de variabile globale:

- ❑ *Variabile de intrare booleene:*
 - Start : buton de pornire
 - d : detector de trecere brichete solubile
 - A : detectorul greutății produsului A
 - B : detectorul greutății produselor A + B
 - Lim_stanga : limitator stânga malaxor
 - Lim_dreapta : limitator dreapta malaxor
- ❑ *Variabile de ieșire booleene:*
 - Valva_A : comandă deschiderea / închiderea valvei A
 - Valva_B : comandă închiderea / închiderea valvei B
 - Valva_C : comandă deschiderea / închiderea valvei C
 - Mt : comandă motorul benzii transportoare
 - Rotire : comandă rotația malaxorului
 - Piv_dreapta : comandă pivotarea către dreapta a malaxorului
 - Piv_stanga : comandă pivotarea către stânga a malaxorului
- ❑ *Variabile globale de tip timer:*
 - timer : temporizare folosită la rotația malaxorului

Programul principal „Main” este prezentat în figura 5.30:



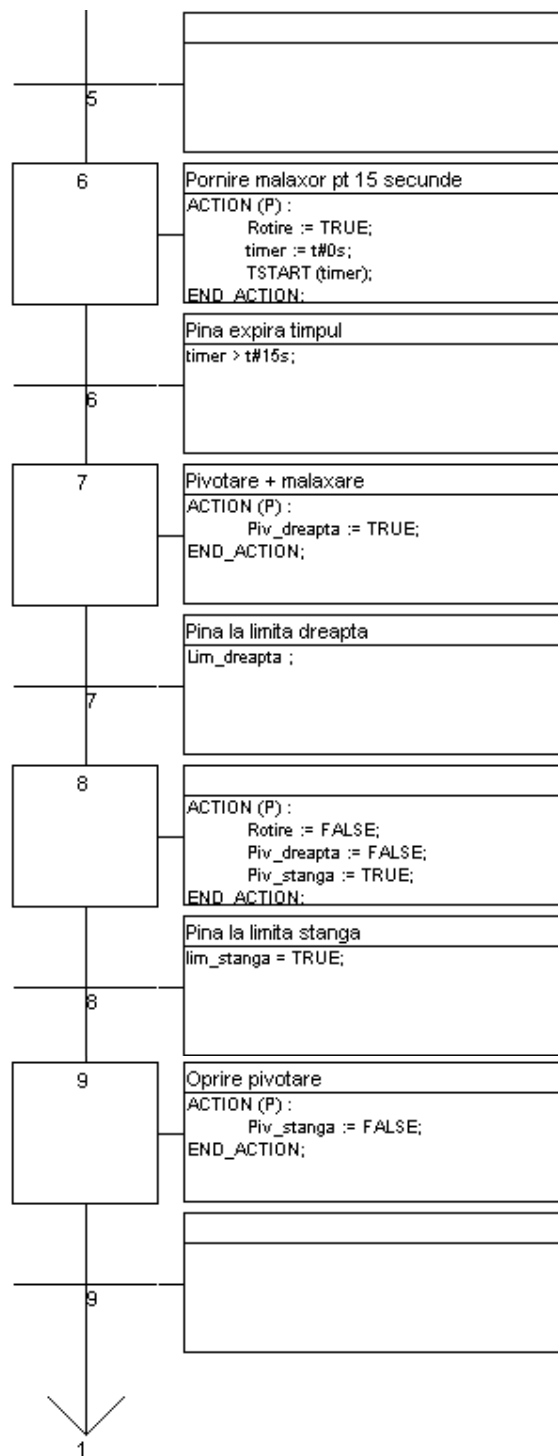


Fig. 5.30 Programul principal

2.2 Varianta 2 : implementarea în limbajul Ladder Diagram

Această variantă presupune folosirea unui automat programabil Allen Bradley, tip SLC500, pentru care se dezvoltă o diagramă de tip Grafcet și o diagramă de tip Ladder.

Diagrama Grafcet este prezentată în figura 5.31:

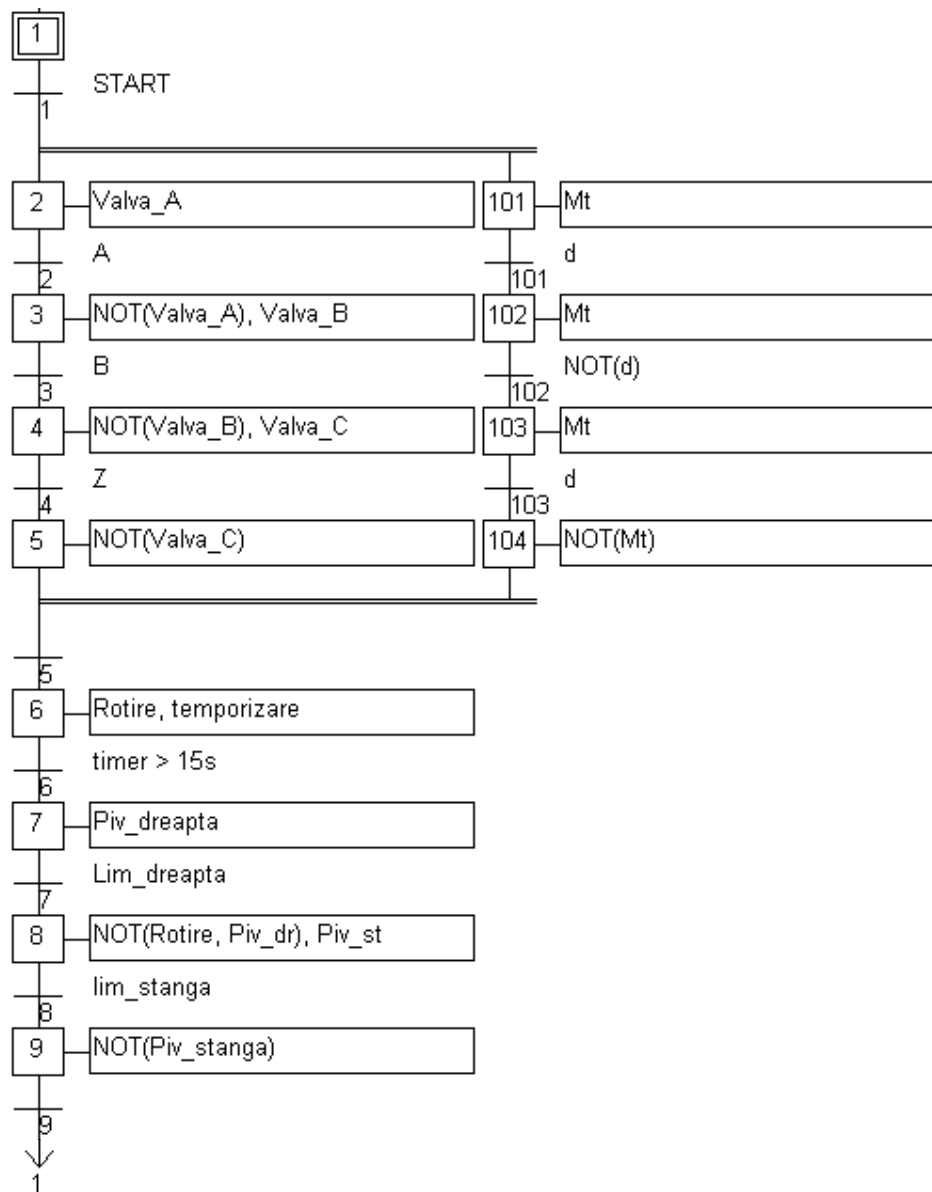


Fig. 5.31 Diagrama Grafcet

Asocierea intrărilor și ieșirilor fizice cu biți din regiștrii de intrare / ieșire este prezentată în tabelul 5.3:

Tabelul 5.3

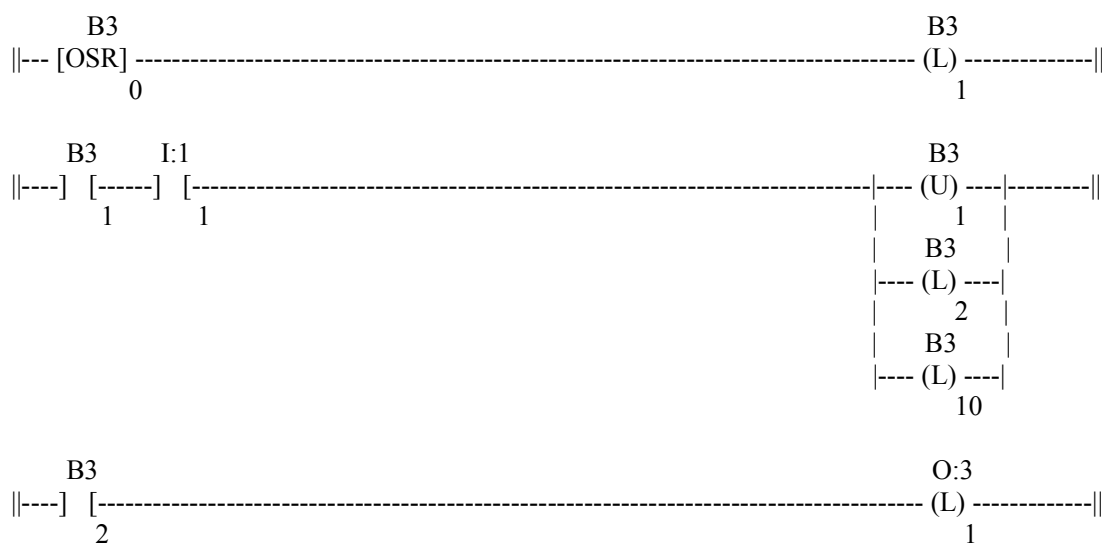
Intrare fizică	Adresă internă	Ieșire fizică	Adresă internă
START	I:1/1	Valva_A	O:3/1
A	I:1/2	Valva_B	O:3/2
B	I:1/3	Valva_C	O:3/3
Z	I:1/4	Mt	O:3/4
lim_dreapta	I:1/5	Rotire	O:3/5
d	I:1/6	Piv_dreapta	O:3/6
lim_stanga	I:1/7	Piv_stanga	O:3/7

Asocierea etapelor cu biți din fișierul de bit B3 și alegerea fișierului de timer este prezentată în tabelul 5.4:

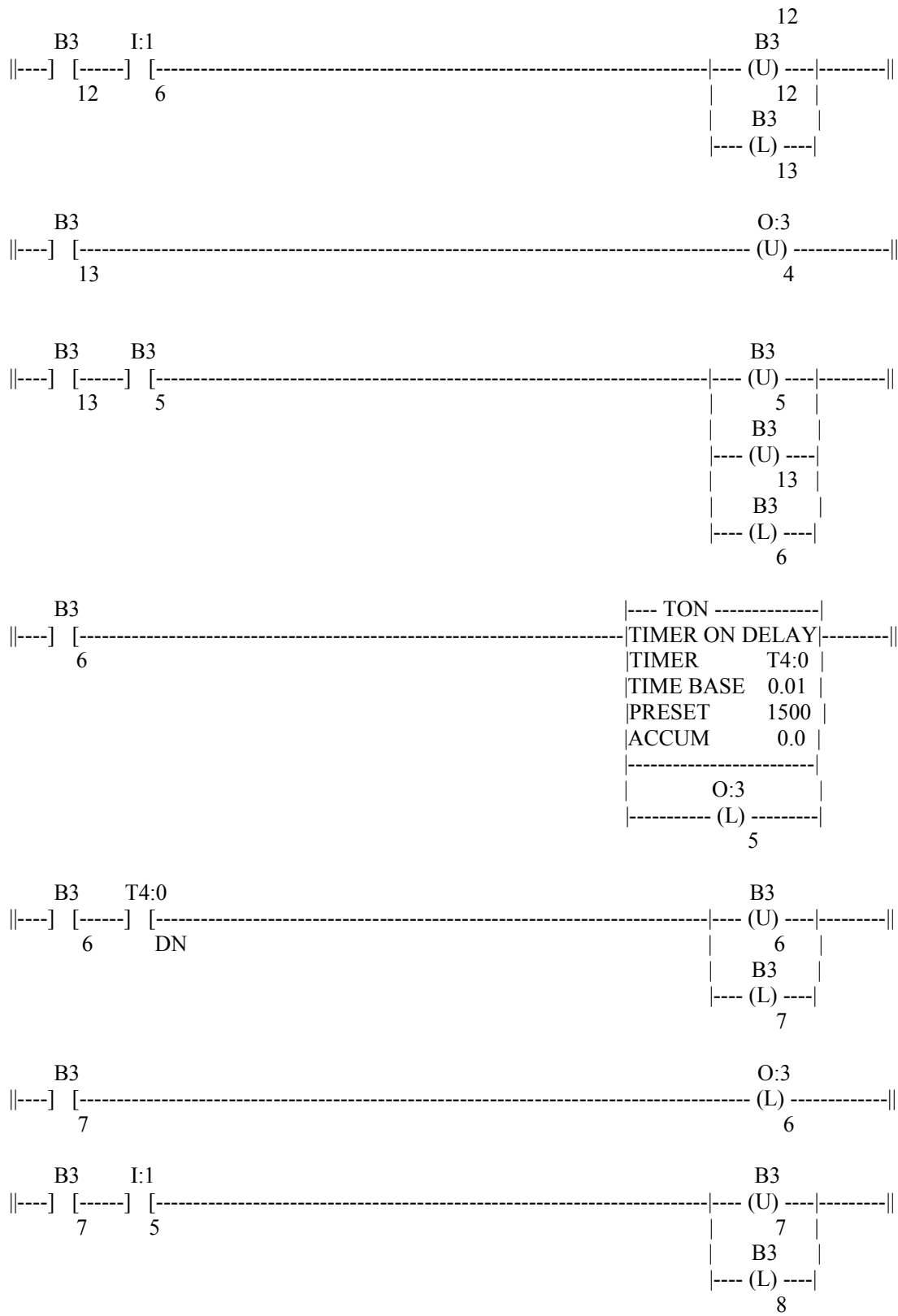
Tabelul 5.4

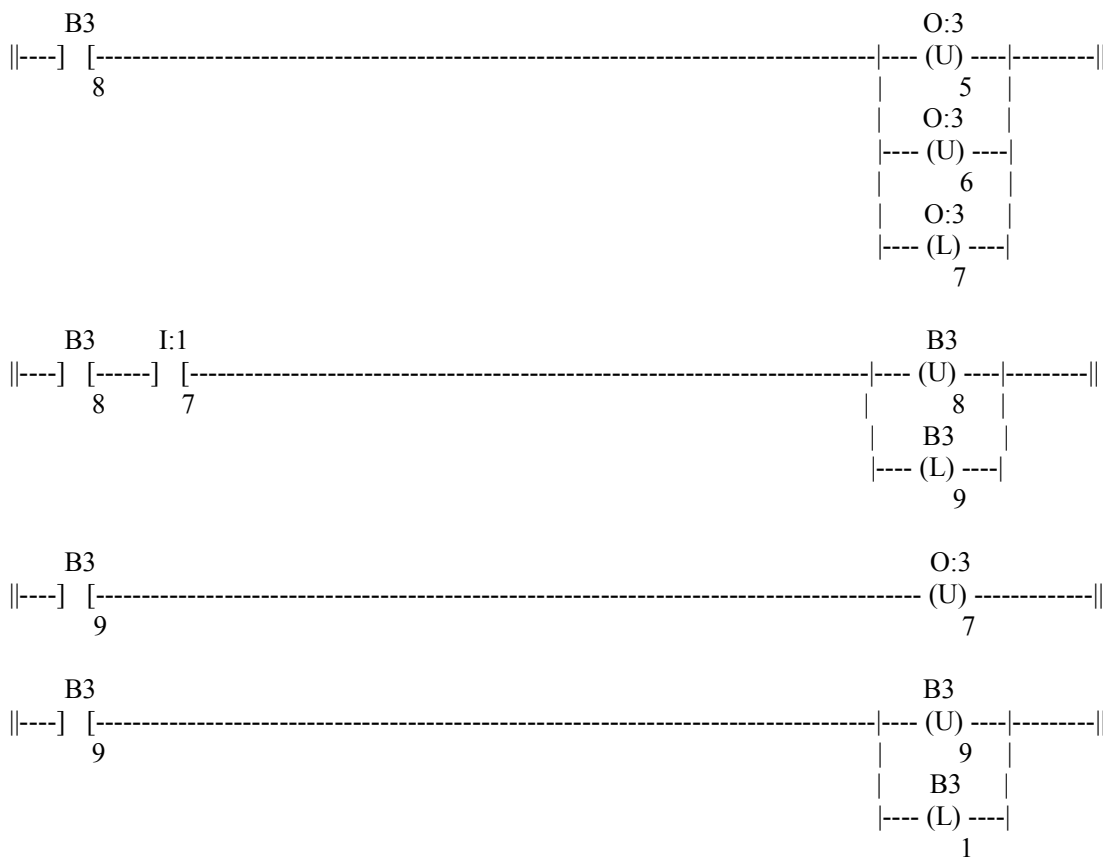
Etapă	Adresa bit	Temporizare	Fișier de timer
1	B3/1	Temporizare 1	T4:0
2	B3/2		
3	B3/3		
4	B3/4		
5	B3/5		
6	B3/6		
7	B3/7		
8	B3/8		
9	B3/9		
101	B3/10		
102	B3/11		
103	B3/12		
104	B3/13		

Diagrama Ladder este prezentată în continuare:









Comentarii:

- În diagrama SFC se pot observa cele 2 secvențe paralele, aducerea lichidelor și aducerea brichetelor în malaxor. Ieșirea din paralelism se face atunci când ambele secvențe s-au încheiat
- Temporizarea a fost realizată cu instrucțiunile TSTART și TSTOP

Propunere:

- Să se modifice diagrama Ladder folosind pentru activarea ieșirilor instrucțiunea OTE (acolo unde este posibil)

2. Soluția de automatizare

Pentru controlul acestei aplicații s-a ales un automat programabil de tip PEP Smart pentru care s-a dezvoltat un proiect Isagraf ce cuprinde un program principal, numit „main”.

Dicționarul de variabile globale:

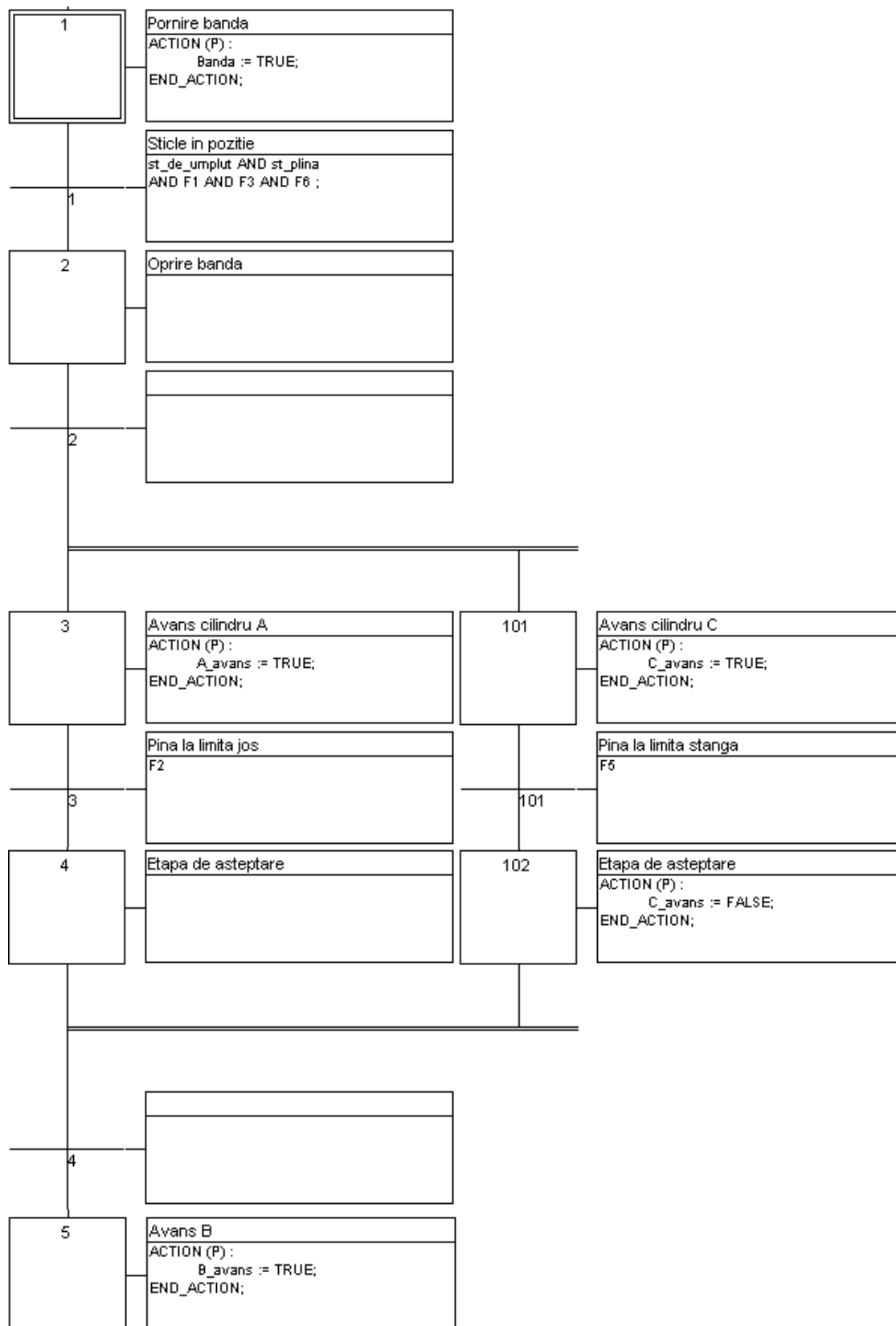
Variabile de intrare booleene:

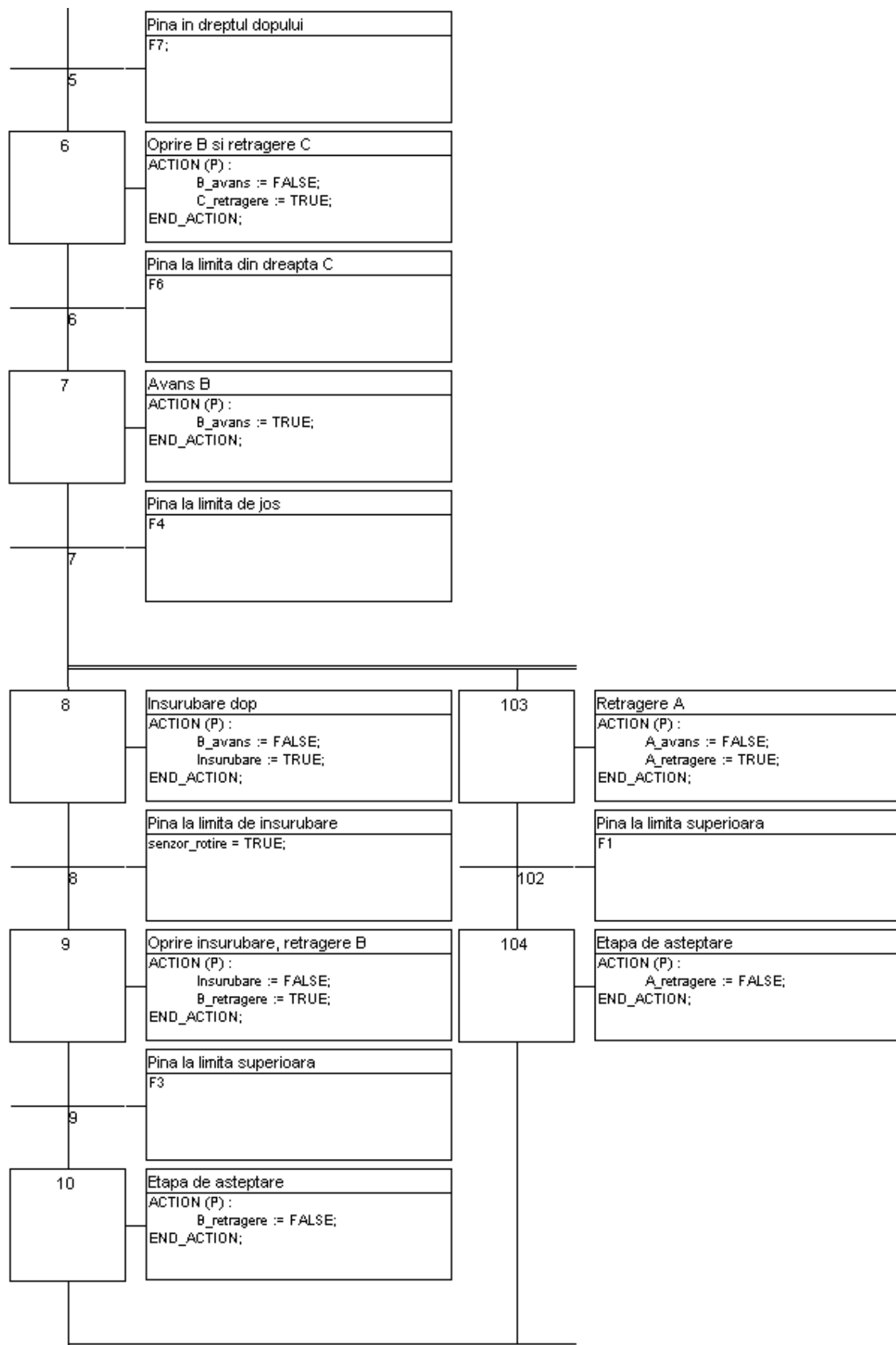
- F1 : Limitator sus cilindru A
- F2 : Limitator jos cilindru A
- F3 : Limitator sus cilindru B
- F4 : Limitator jos cilindru B
- F5 : Limitator stanga cilindru C
- F6 : Limitator dreapta cilindru C
- F7 : Detector pozitie de preluat dop
- senzor_rotire : Limitator rotire dispozitiv de înșurubare
- st_de_umplut : Senzor detecție sticlă de umplut
- st_plina : Senzor detecție sticlă plină

Variabile de ieșire booleene:

- Banda : comandă pornire / oprire bandă
- A_avans : comandă avans cilindru A
- A_retragere : comandă retragere cilindru A
- B_avans : comandă avans cilindru B
- B_retragere : comandă retragere cilindru B
- C_avans : comandă avans cilindru C
- C_retragere : comandă retragere cilindru C
- Inșurubare : comanda de înșurubare a dopului

Programul principal este prezentat în figura 5.33





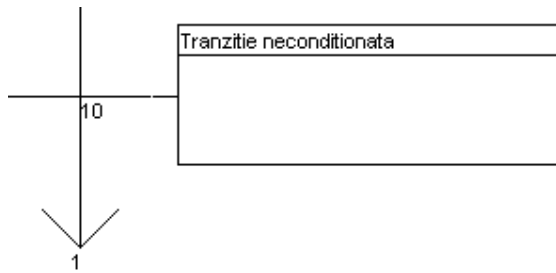


Fig. 5.33 Programul principal

Comentarii:

- În program au fost folosite 2 paralelisme, fiecare având câte 2 secvențe ce se execută simultan
- A doua secvență paralelă implementează condiția ca o dată cu umplerea unei sticle, alta să fie astupată

Propunere:

- Să se modifice programul în situația în care sticlele vin aleator pe banda transportoare

Elemente de execuție:

- 3 electrovalve
- 2 cilindri cu dublu efect
- 2 motoare ale benzilor transportoare

Elemente de măsură:

- 4 detectoare de poziție

2. Soluția de automatizare

Pentru controlul acestei aplicații s-a ales un automat programabil de tip PEP Smart pentru care s-a dezvoltat un proiect Isagraf ce cuprinde un program principal.

Dicționarul de variabile globale:

Variabile de intrare booleene:

- Poz_A: detectorul de poziție A
- Poz_B: detectorul de poziție B
- Poz_C: detectorul de poziție C
- Lim_4: limitatorul de avans al cilindrului E

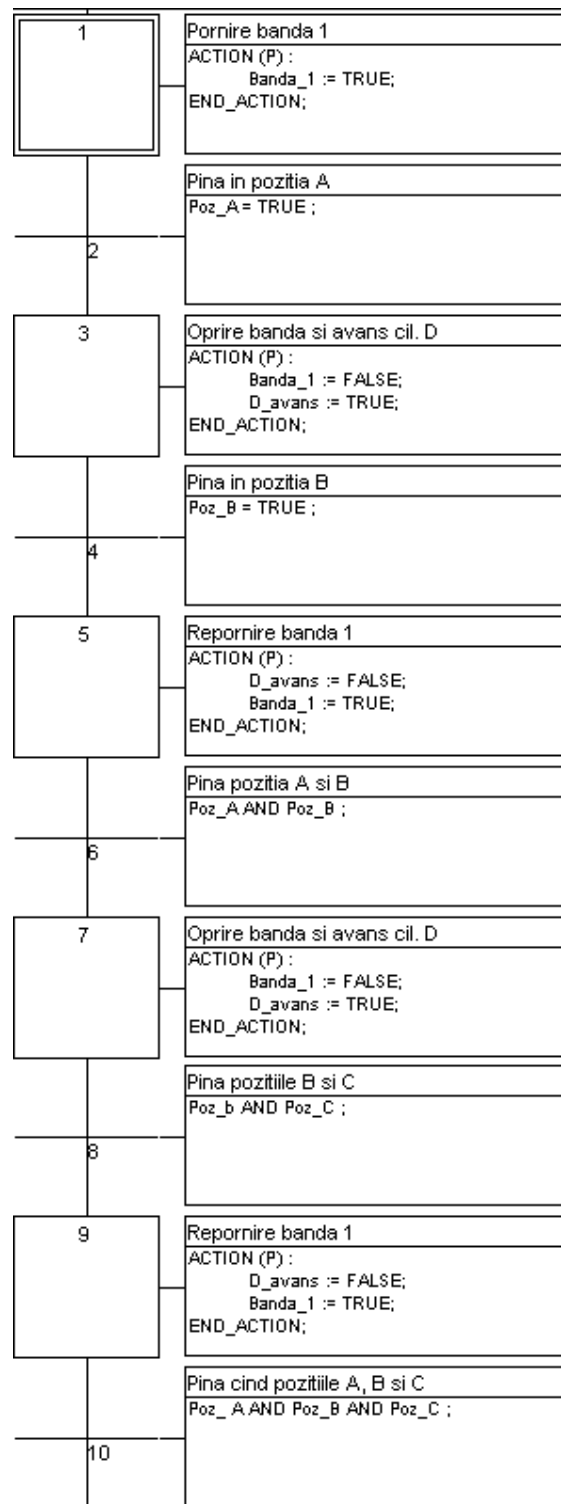
Variabile de ieșire booleene:

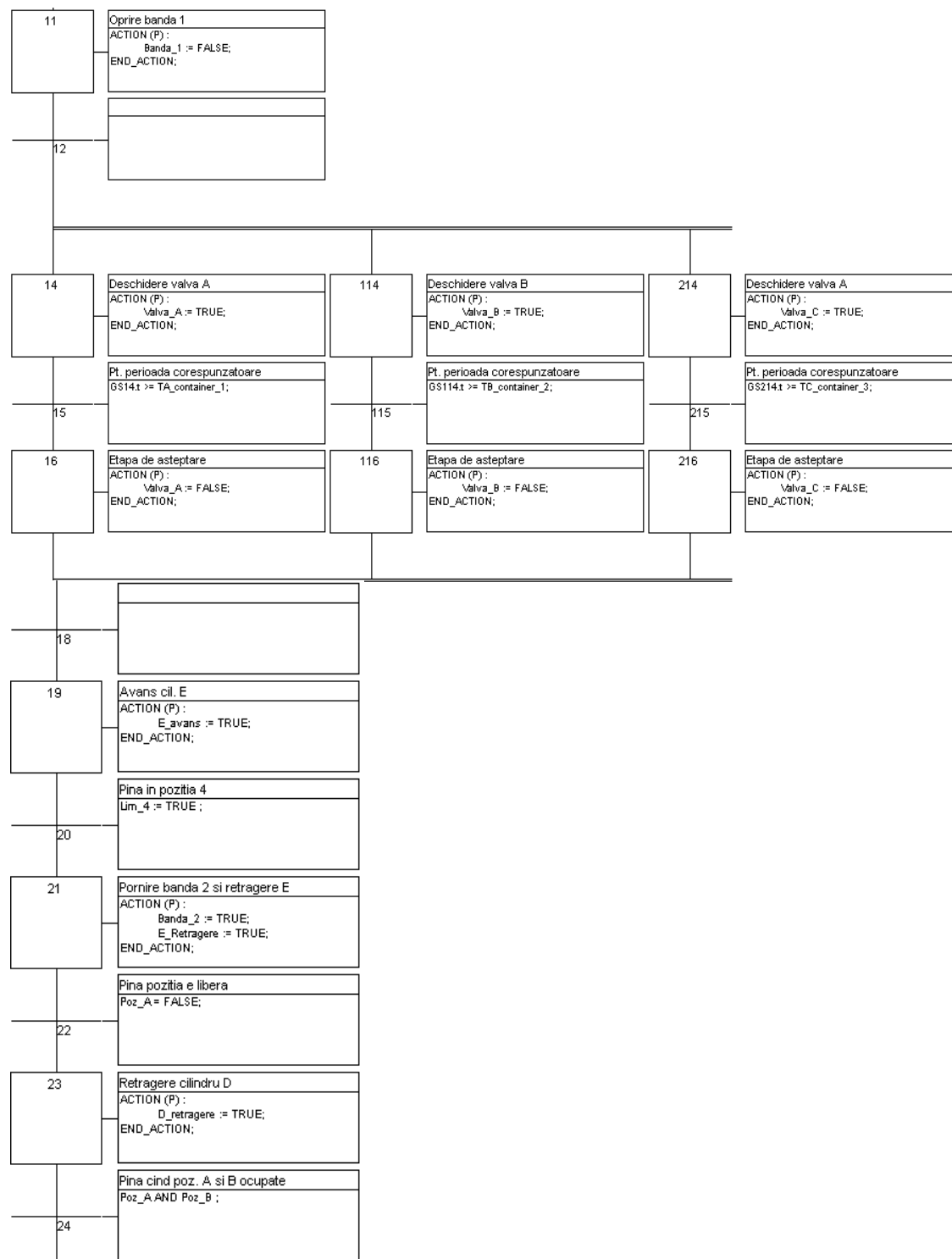
- Banda_1 : activare / dezactivare banda 1
- Banda_2 : activare / dezactivare banda 2
- D_avans : avans cilindru D
- D_retragere : retragere cilindru D
- E_avans : avans cilindru E
- E_retragere : retragere cilindru E
- Valva_A : comanda de deschidere-închidere valva A
- Valva_B : comanda de deschidere-închidere valva A
- Valva_C : comanda de deschidere-închidere valva A

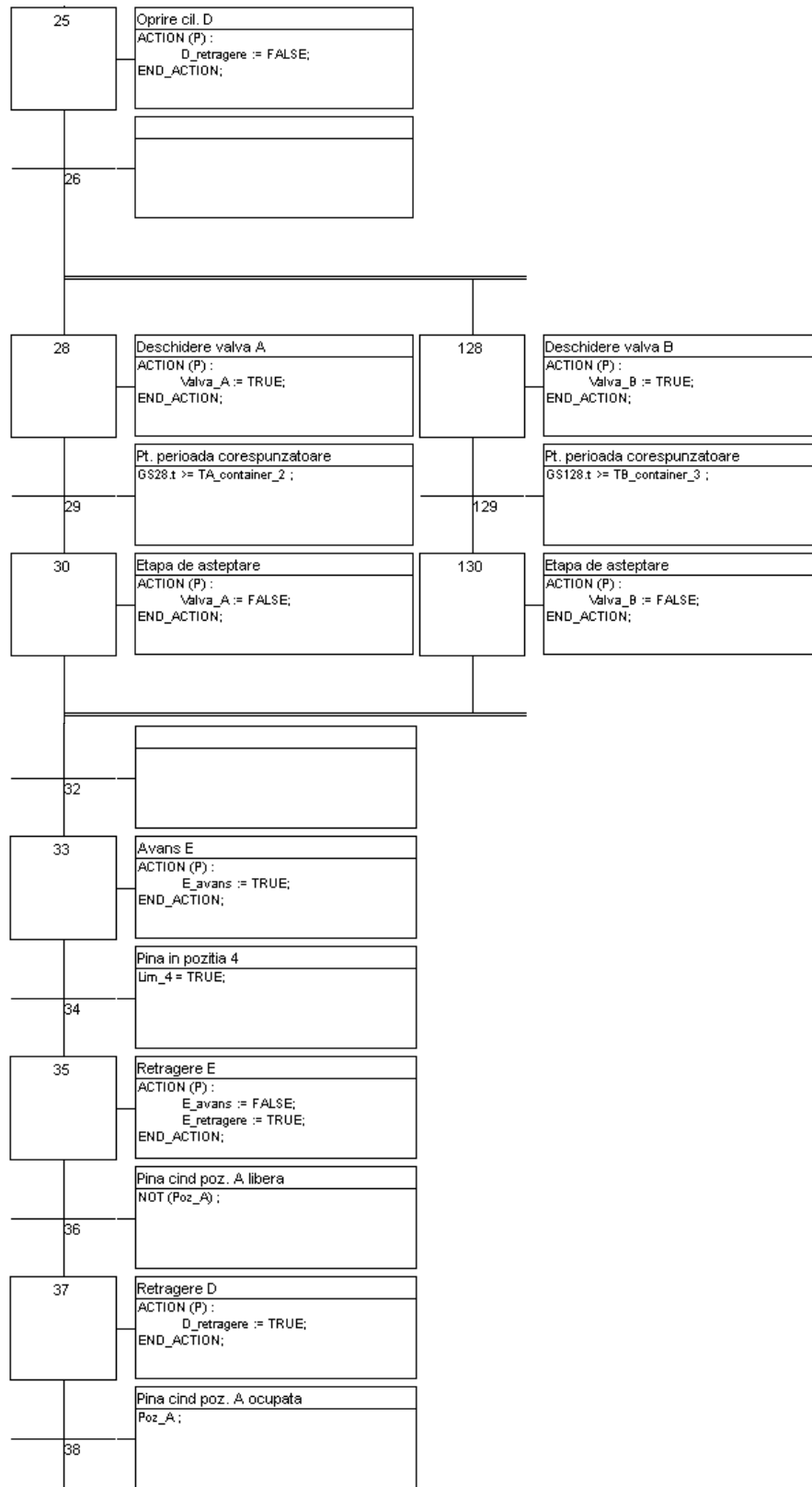
Constante interne de tip INTEGER:

- TA_container_1 : perioada de timp pentru umplerea containerului 1 cu lichid de tip A
- TB_container_2 : perioada de timp pentru umplerea containerului 2 cu lichid de tip B
- TC_container_3 : perioada de timp pentru umplerea containerului 3 cu lichid de tip C
- TA_container_2 : perioada de timp pentru umplerea containerului 2 cu lichid de tip A
- TB_container_3 : perioada de timp pentru umplerea containerului 3 cu lichid de tip B
- TA_container_3 : perioada de timp pentru umplerea containerului 3 cu lichid de tip A

Programul principal este prezentat în figura







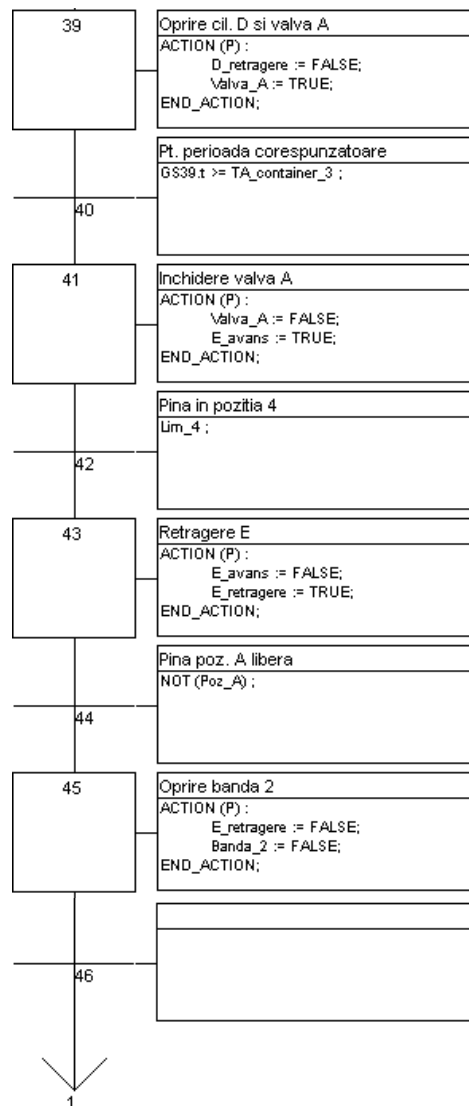


Fig. 5.35 Programul principal

Comentarii:

- În cadrul secvențelor de deschidere a valvelor, temporizările au fost implementate prin testarea permanentă a parametrului GSxxx.t asociat unei etape. Acest parametru indică timpul de când o etapă este activă. O altă variantă era folosirea unor variabile de tip Timer.
- Umplerea containerelor este făcută în paralel cu ajutorul elementului de paralelism al diagramelor SFC

Problema10: Proiectarea unui regulator PID pentru reglarea unei temperaturi

1. Descrierea procesului:

Problema constă în menținerea unei anumite temperaturi într-o instalație de granulare prin reglarea debitului de abur, dar, ca orice regulator, se poate acorda pentru o mare varietate de procese. Regulatorul PID trebuie să funcționeze în două regimuri de lucru, automat și manual, cu posibilitatea de a fi conectat în diferite scheme de reglare.

Temperatura este măsurată cu un traductor de temperatură având semnal de ieșire 4-20mA, măsura se filtrează iar valoarea reală se transmite regulatorului în procente. Referința se citește de la un panou operator care comunică cu automatul prin interfața serială 232.

Regulatorul face parte dintr-o schemă în care se intenționează menținerea constantă a temperaturii unui agent termic (în cazul acesta abur de joasă presiune 4-6 bar) prin comanda debitului de intrare al gazului metan.

Bucula de reglare în această situație este compusă din:

- termocuplu (pentru măsurarea temperaturii) care generează milivolți după o curbă care nu este liniară (ca alternativă la această soluție se poate utiliza orice alt traductor al cărui semnal este în curent unificat). Soluția cu automate programabile este destul de flexibilă și din faptul ca ele au module de intrare pentru toate tipurile de termocuple, eliminând astfel din buclă convertorul milivolți - curent unificat, dar alegerea între cele două metode se face strict economic.
- convertor 4-20 mA (pentru conversia în miliamperi a semnalului de la termocuplu (mV) și liniarizarea curbei valorilor)
- automat programabil SmartPLC cu module de intrări analogice, respectiv în termocuple și modul de ieșiri analogice care la rândul lor sunt niște convertoare analog-numerice și invers (pentru realizarea algoritmului de reglare)
- Electroventil comandat în curent de 4-20mA, reprezentând elementul de execuție (se mai poate folosi și un ventil pneumatic împreună cu un convertor electropneumatic dacă este necesar)

În fig. 4.12 este prezentată schematic bucla de reglare pentru varianta cu convertor, cealaltă fiind mai simplă (ieșirea termocuplului intrând direct în automat):

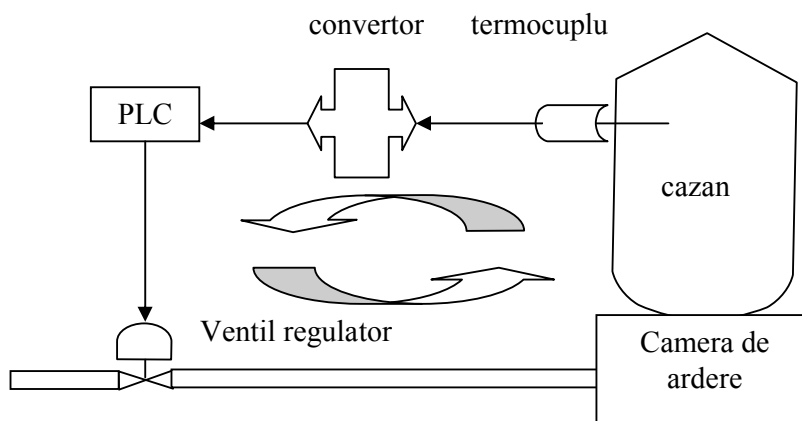


Fig. 5.36 Structura buclei de reglare folosind convertor tensiune - curent

Regulatorul se acordează prin:

- banda de proporționalitate [în procente]
- timp de integrare [în secunde]
- timp de derivare [în secunde]

Valorile de acordare se citesc și ele de la interfața operator odată cu referința.

În regim manual regulatorul transmite valoarea absolută a poziției elementului de execuție (comanda directă), venită tot de la consola operator, de unde se citește și regimul de lucru.

Trecerea din regim manual în regim automat și invers trebuie să se facă fără variații bruște de comandă, pentru a se evita intrarea sistemului în regimuri de lucru improprie. La comutarea regimurilor de lucru, pe cât posibil comanda va trebui să rămână aceeași.

Elementul de execuție primește la intrare valoarea absolută a poziției. Menținerea acestei valori este o problemă a elementului de execuție. Dacă e necesar se poate insera un regulator de poziție între regulatorul de temperatură și elementul de execuție.

2. Soluția de automatizare

Pentru implementarea acestei aplicații s-a ales un automat programabil de tip PEP Smart pentru care s-a dezvoltat un proiect Isagraf, a cărui structură ierarhică poate fi vizualizată în figura 5.37 și ale cărei secțiuni standard sunt descrise în continuare.

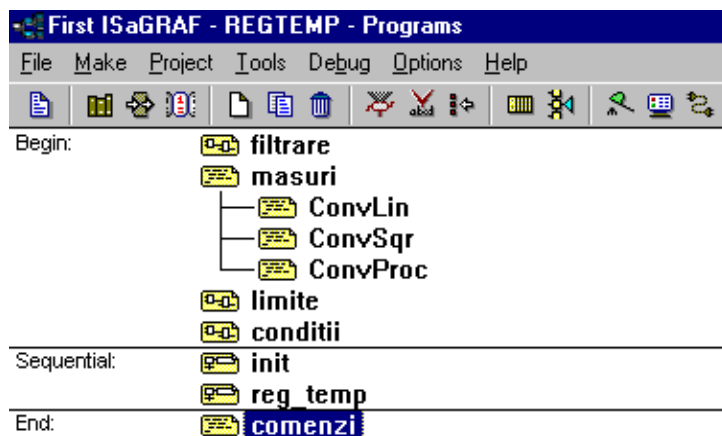


Fig. 5.37 Structura proiectului Isagraf pentru implementarea regulatorului

Inițializările de constante, măsurile, filtrările, adaptările, evaluarea existenței condițiilor de funcționare, interfața cu utilizatorul, sunt realizate în afara regulatorului de către alte taskuri care se ocupă doar de aceste lucruri. Comanda se adaptează și se transmite în afara regulatorului.

Numele de variabile s-au ales pentru varianta cu identificatori proprii. Pentru o versiune generică, impersonală, se lucrează cu array-uri (vectori de variabile), iar accesul se face prin indexul la variabila respectivă.

În cadrul secțiunii **BEGIN**, care se execută la începutul fiecărui ciclu de automat, au loc următoarele acțiuni:

- Măsurile, filtrări, testare de încadrare în limite, evaluare alarme, avarii, contorizări
- Adaptări, pregătire variabile de lucru
- Citirea variabilelor de control de la interfața operator

În cadrul secțiunii **SEQUENTIAL**, adică acolo unde este implementată logica de funcționare a programului, au loc, în principal, următoarele acțiuni:

- secvența de inițializare, în care sunt inițializate anumite constante și este pornit timer-ul pentru perioada de eșantionare.
- algoritmul de reglare, a cărui descriere detaliată este prezentată în subcapitolul următor

Nu întâmplător secțiunea secvențială are o singură stare pentru fiecare regim de lucru. Această necesitate a apărut datorită faptului că automatul (care poate lucra multitasking) nu trece la o altă activitate decât după terminarea unui ciclu mașină. Într-un ciclu mașină se execută secțiunea de BEGIN, cea de END și un anumit număr de stări din secțiunea **SEQUENTIAL**, cât permite durata unui ciclu mașină. Dacă pentru execuția acțiunilor asociate unei stări, procesorului i-ar lua un timp mai îndelungat decât perioada unui ciclu predefinit, ciclul va fi prelungit până când toate acțiunile asociate stării sunt executate.

Cu această metodă ciclul de reglare devine constant și cât mai mic posibil. Durata ciclului poate fi fixată și altfel (software), dar oricum, perioada va avea o valoare mai mare, pierzându-se astfel din timpul de răspuns al regulatorului. Mai mult, în cazul în care ar exista mai multe etape, apare posibilitatea ca comanda ce se dă elementului de execuție să nu fie consecința ultimei citiri, ceea ce face să se piardă chiar noțiunea de cauzalitate.

Secțiunea **END**, care se execută la sfârșitul fiecărui ciclu automat, după secțiunea secvențială, realizează în principal următoarele acțiuni:

- Adaptări de valori
- Actualizare ieșiri: concret, în cazul algoritmului de reglare ce respectă condițiile de mai sus, ieșirea este comanda către elementul de execuție calculată conform algoritmului la ultimul ciclu mașină și ca urmare a ultimei valori a intrării.
- Transmiterea către interfața utilizator a valorilor calculate, pentru vizualizare

Diagrama logică de funcționare a regulatorului (Sequential Function Chart), în care pot fi identificate etapele în care se poate găsi automatul și tranzițiile pe care le poate efectua, sunt ilustrate în figura 5.38.

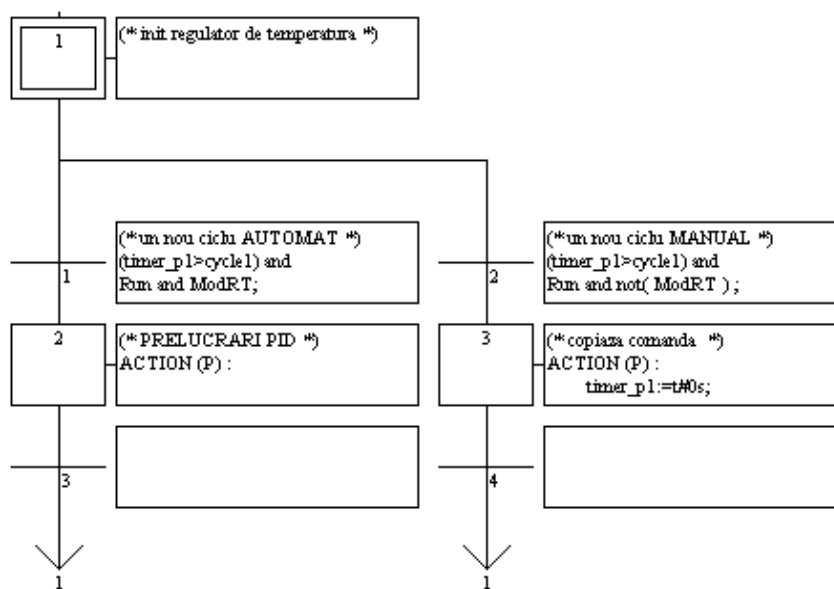


Fig. 5.38 Diagrama logică de funcționare a regulatorului

Se observă că diagrama conține 3 etape și pot fi efectuate 4 tranziții între aceste etape.

Etape:

- Etapa 1: inițializare regulator
- Etapa 2: se execută un pas de reglare în regim automat
- Etapa 3: se execută un pas de reglare în regim manual

Tranziții:

- Tranziția 1- S1 -> S2: se execută dacă sunt condiții de trecere în starea de calcul PID (regimul automat)
- Tranziția 2 - S1 -> S3: se execută dacă sunt condiții de trecere în starea de comandă manuală (regimul manual)
- Tranziția 3 - S2 -> S1: salt necondiționat în etapa inițială
- Tranziția 4- S3 -> S1: salt necondiționat în etapa inițială

Variabilele declarate în cadrul proiectului sunt prezentate în tabelul 5.5:

Tabel 5.5 Variabilele folosite în cadrul proiectului

Variabilă	Tip	Descriere
Run	Logic	Există condiții generale de funcționare pentru regulatorul PID, evaluate în exterior
	1	OK, adică sunt măsuri bune, elemente de execuție funcționează, resurse energetice ok, etc
	0	not OK, nu se face reglare cu regulatorul PID
ModRT	Logic	Regimul de lucru al regulatorului, evaluat în exterior
	1	regim AUTOMAT (PID)
	0	regim MANUAL (comandă directă)
timer_p1	Timer	Timer de ciclu de eșantionare
cycle1	Timer	Perioada de eșantionare, constantă [ms]
banda1	Real	Banda de proporționalitate, inversul amplificării, în %
Td1	Real	Constantă de timp pentru efectul de derivare, valoare de acord
Tint1	Real	Constantă de timp pentru efectul de integrare, valoare de acord
kp1	Real	Amplificarea, valoare calculată, dacă banda este valoarea de acord
deriv1	Real	Componenta derivativă, variabilă de calcul
w	real	Calcul parțial pentru incrementul de comandă, variabilă de lucru
T_pv	real	Valoarea curentă a temperaturii reglate, în %
T_pv1	real	Valoarea curentă, la pasul anterior, a temperaturii reglate, în %
T_pv2	real	Valoarea curentă, la pasul anterior, a temperaturii reglate, în %
T_sp	real	Valoarea impusă pentru temperatură, în %
err1	real	Eroarea ($t_{sp} - t_{pv}$) la pasul anterior
Comanda ventil	real	Comanda curentă, poziția absolută a elementului de execuție, în %
xo1	real	Comandă manuală, poziția absolută a elementului de execuție, în %

În **etapa 1** (Init) se actualizează (dacă e nevoie) valorile variabilelor de lucru (dacă regulatorul funcționează în alte regimuri de lucru, spre exemplu în cascadă, sau în regim de selecție).

Acțiunile asociate **etapei 2** (Regim automat) sunt descrise în tabelul 5.6:

Tabel 5.6 Acțiuni asociate etapei 2

inițializează ciclul	timer_p1:=t#0s;
calcul amplificare	kp1:=100.0/banda1;
calcul componentă derivativă	deriv1:=Td1*1000.0*(T_pv-2.0*T_pv1+T_pv2)/real(cycle1);
salvare valori pv anterioare	T_pv2:=T_pv1; T_pv1:=T_pv;
calcul noua comandă (pt timpi, se lucrează în [ms]=[s]*1000)	w:=(T_sp-T_pv)+(T_sp-T_pv)*real(cycle1)/(Tint1*1000.0)-err1-deriv1; Comanda ventil:=Comanda ventil+Kp1*w;
limitări la 0% și 100%	if(Comanda ventil>100.0) then Comanda ventil:=100.0; end_if; if(Comanda ventil< 0.0) then Comanda ventil:= 0.0; end_if;
salvare eroare	err1:=T_sp-T_pv;
pregătire trecere în regim manual	xo1:=Comanda ventil;

În etapa 3 (Regim manual) este inițializat ciclul și pregătește trecerea în regim automat:

Tranziția 1 (intrare în regulator automat s1 -> s2) are loc dacă s-a epuizat timpul pentru perioada de eșantionare și sunt condiții de reglare în regim automat:

timer_p1>cycle1 și Run și ModRT

Tranziția 2 (intrare în regulator manual s1 -> s3) are loc dacă s-a epuizat timpul pentru perioada de eșantionare și sunt condiții de reglare în regim manual:

timer_p1>cycle1 și Run și notModRT

E important să se efectueze toate calculele necesare într-o singură stare (un sigur ciclu IsaGRAF) și, pentru uniformitate, e de dorit ca pe oricare ramură (automat sau manual), durata unui ciclu de reglare să fie aceeași, cât mai scurtă, adică două cicluri IsaGRAF (init+automat sau init +manual), pentru ca modul de funcționare a ISaGRAF-ului (la un ciclu mașină execută secțiunea de BEGIN , din secțiunea SEQUENTIAL câte stări îi permite ciclul mașina și apoi secțiunea END) să nu influențeze (mărească) perioada de eșantionare.

Un ciclu de eșantionare, pentru valorile cu care se lucrează la reglare, se calculează ținând cont că un ciclu activ înseamnă stare – calcul - comanda + stare init (2 isa) iar un ciclu pasiv înseamnă stare init (1 isa) unde se așteaptă epuizarea timpului ales pentru ciclul de reglare. Dacă ciclul isa se fixează, atunci se poate alege ca perioada de eșantionare pentru reglare un multiplu de ciclu isa mai mare sau egal cu 2 și se introduc în secvență etape lipsite de acțiuni care asigură implicit realizarea perioadei. Dacă sunt necesare regulatoare suplimentare (de exemplu de poziție) va trebuie asigurată o sincronizare .

Oricum, secvența în două stări pe ramură este cea mai convenabilă în cazul în care se adoptă această soluție. Soluția corectă de realizare a unui regulator este aceea în care funcția de reglare este descrisă ca o funcție C ce se va apela chiar în secțiunea de BEGIN și va fi executată cu siguranță câte un pas incremental la fiecare ciclu mașină.

Diagrama SFC a programului principal de reglare este prezentată în figura 5.39.

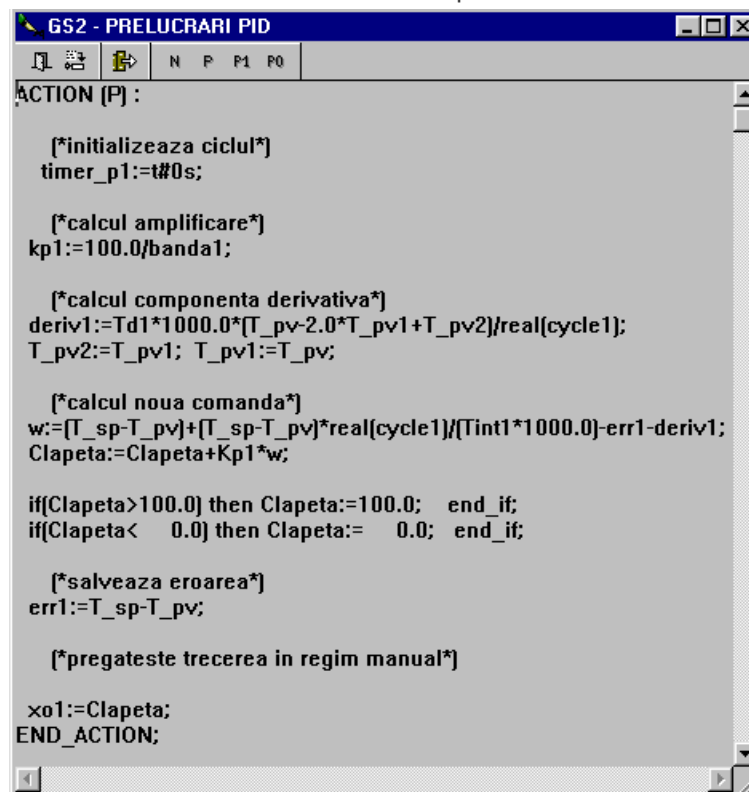
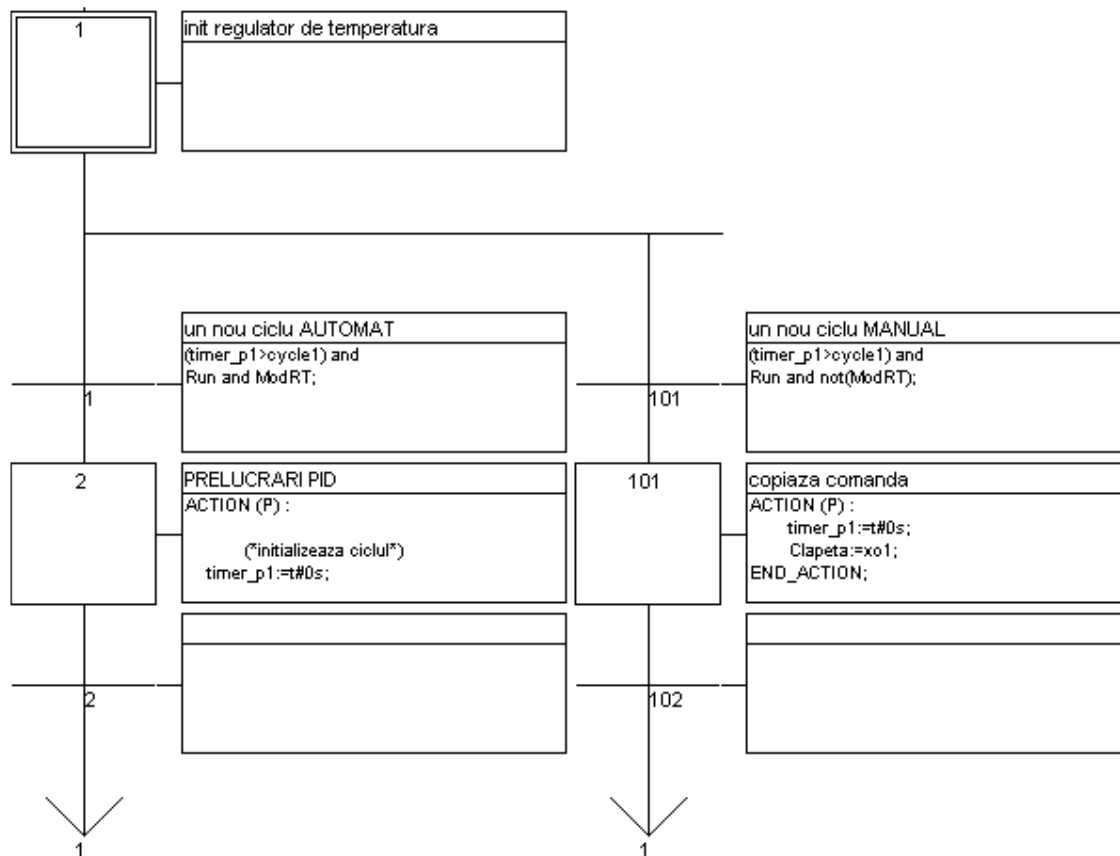


Fig. 5.39 Diagrama SFC a programului „Reg_temp”