

UNIVERSITATEA “PETRU MAIOR” TÎRGU-MUREȘ
FACULTATEA DE INGINERIE
SPECIALIZAREA : Sisteme automate de conducere a proceselor industriale



Proiect SCADA - Reglare automata de nivel cu regulator PI incremental

Îndrumător științific:
Sef lucr. Dr. ing. Turc Traian

Masterand :
Cordoș Petru Marian

TÎRGU-MUREȘ

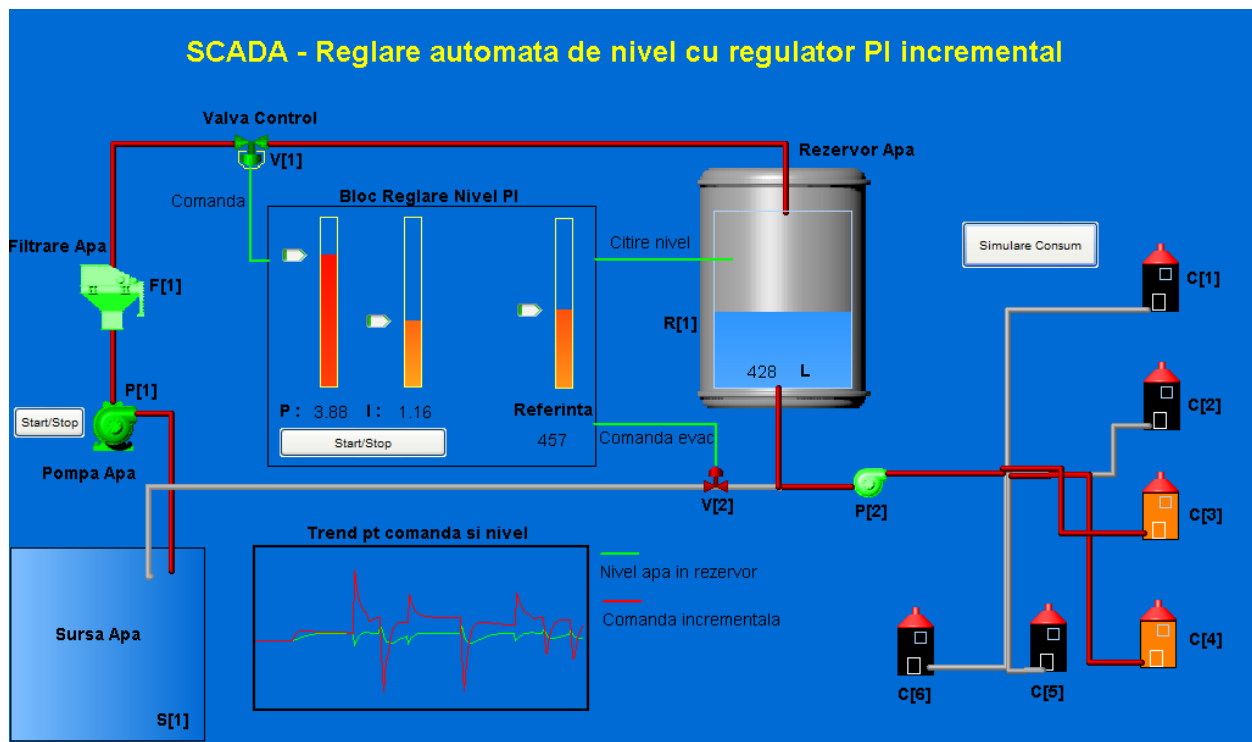
2018

1. Descrierea Proiectului

Lucrarea de fata prezinta si descrie implementarea in mediul SCADA a unui sistem de reglare automat pentru reglarea apei dintr-un rezervor. Regulatorul calculate este de tip proportional-integrativ discret, folosind metoda "Velocity PI" sau "incremental".

S-a considerat ca rezervorul are o functie de transfer clasica fixa de forma : $Hf = \frac{1}{5s+1}$.

Primul pas pentru implementarea regulatorului a fost discretizarea acestei functii de transfer si calculul formei generale a regulatorului. Implementarea regulatorului nu este de tip fix, parametrii P,I pot fi reglati din slidere, precum si referinta sistemului. Pagina SCADA are o rata de reimprospatare de 0.25 secunde, asa ca si timpul de esantionare a fost luat dupa aceeasi valoare.



Proiectul presupune si o parte de simulare in timp real a reglarii, considerand si eliminand fluctuatiile provenite de la consumatori.

La pornirea programului, se apeleaza functia de initializare() pentru a aduce toate valorile variabilelor de tip TAG la valoarea 0. Butoanele pentru pornirea/oprirea reglarii sau simularii se activeaza doar dupa ce exista apa in sistem (pipes). Odata ce in valva V[1] exista apa, butonul de reglare este setat pe enable. Similar, si pentru inceperea simularii consumului, butonul este activat doar daca exista apa in rezervor. In cazul in care nu se consuma apa, sistemul intra in bucla inchisa, functionand dupa modul clasic cu evacuare prin cadere libera activand valva V[2].

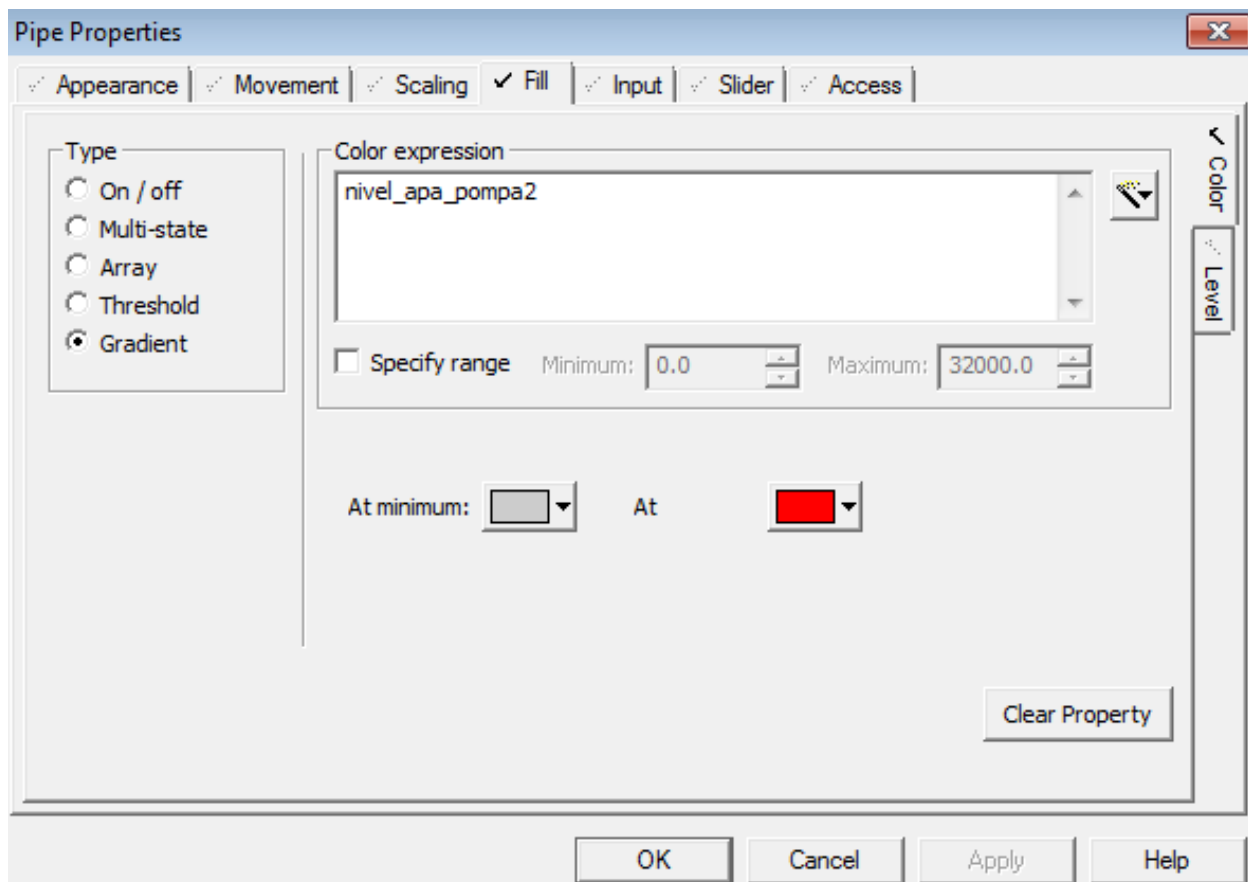
În blocul de reglare se pot seta parametrii regulatorului și referința de urmărire dorită în rezervor.

Simularea de consum s-a implementat activând aleatoriu un număr de consumatori dintr-un tablou al acestora. În momentul când un consumator este activ și consumă apă din rezervor, se schimbă și culoarea acestuia în HMI. Similar, rețeaua de țevi pentru transportul apei își schimbă culoarea în funcție de stare, s-a folosit o animație de tip “gradient”.

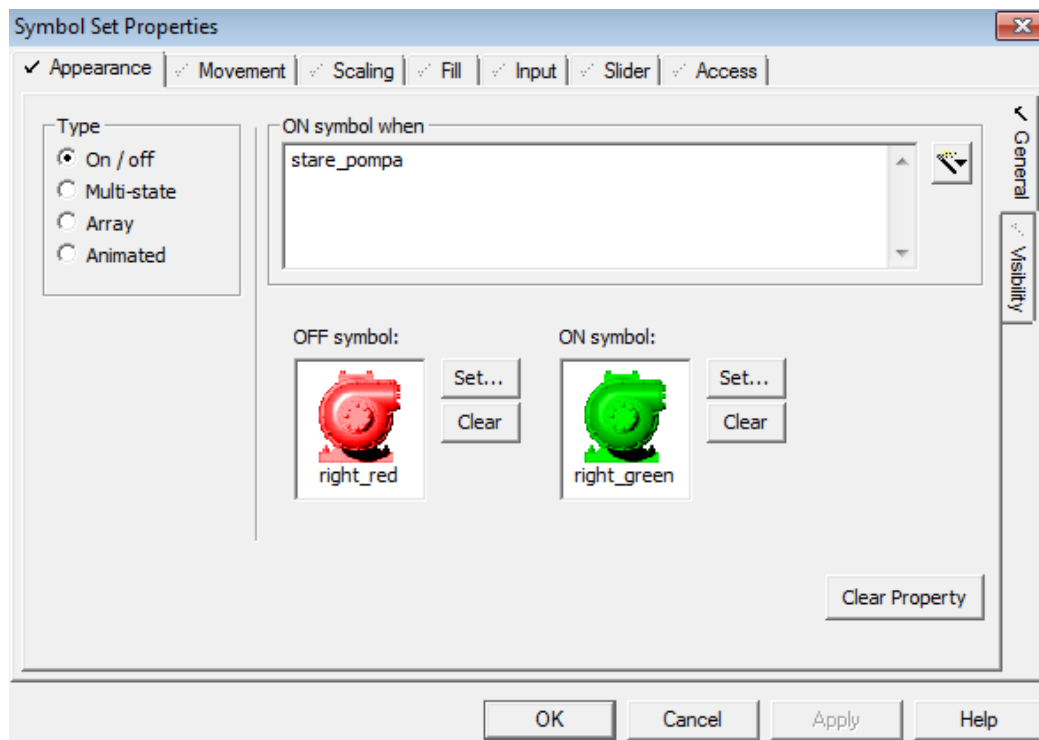
Nu în ultimul rând, s-a implementat și un widget de tip “trend” care arată în timp real evoluția nivelului de apă în rezervor și efortul regulatorului de a menține nivelul, afișând comanda.

2. Implementare

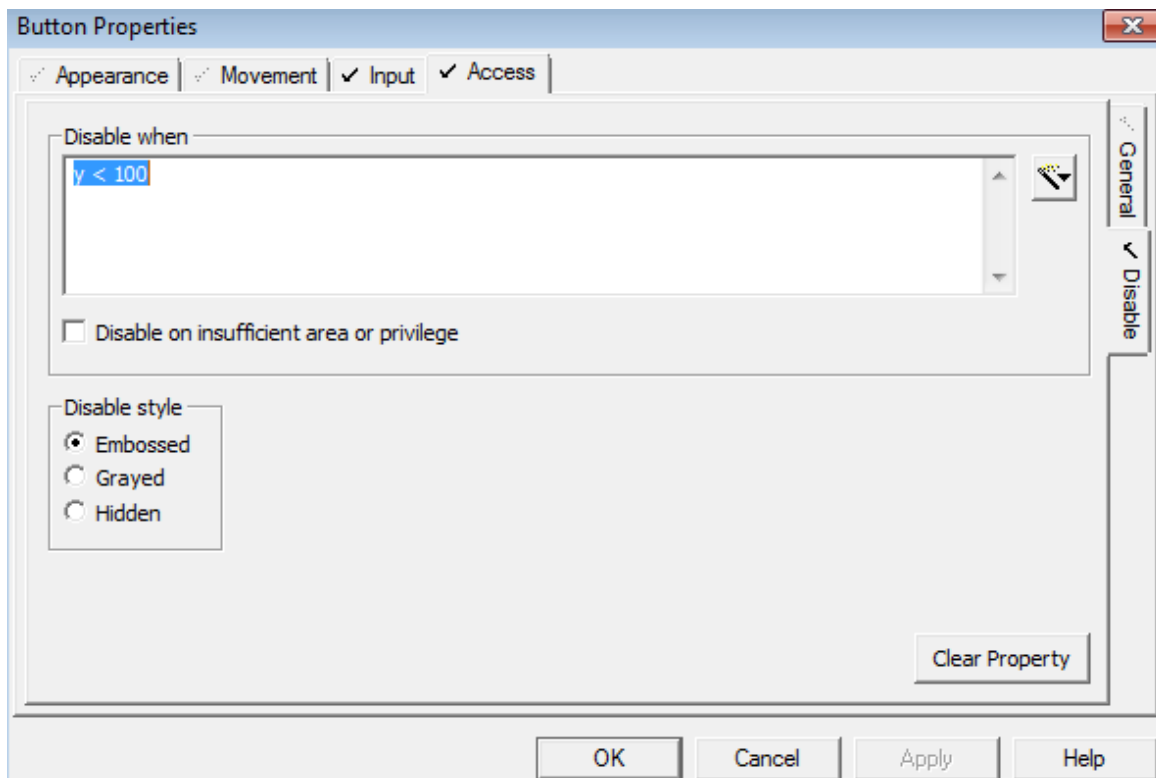
S-au folosit o serie de tag-uri locale de tip digital pentru logica decizională și înt pentru logica de animație. În paralel, s-a folosit și un tablou de valori digitale pentru simularea consumatorilor de apă. Toate țevile simulate în sistem sunt animate, folosind metoda gradient după cum urmează :



Pompele, valve, blocul de filtrare sunt implementate folosind symbol set de tip on/off.

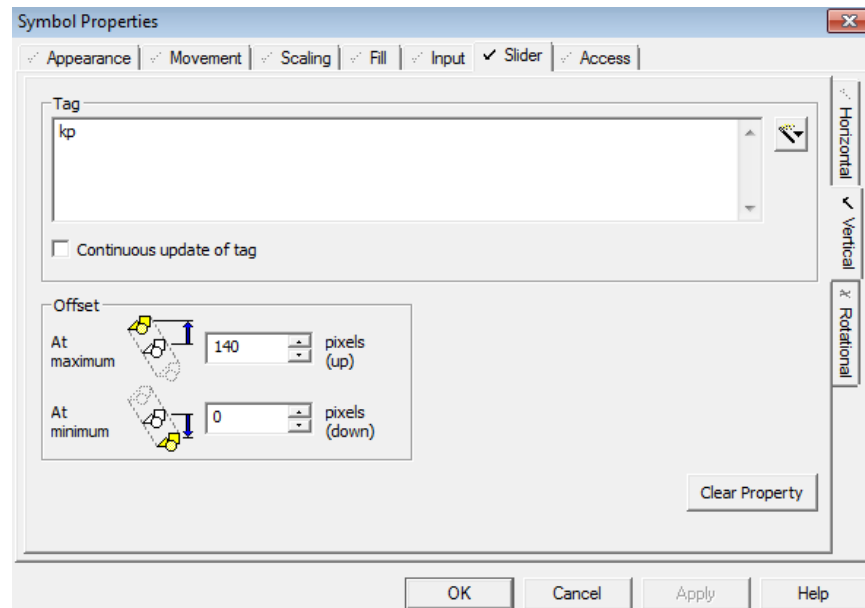


Activarea si dezactivarea butoanelor s-a implementat utilizand tabul Access al obiectelor :

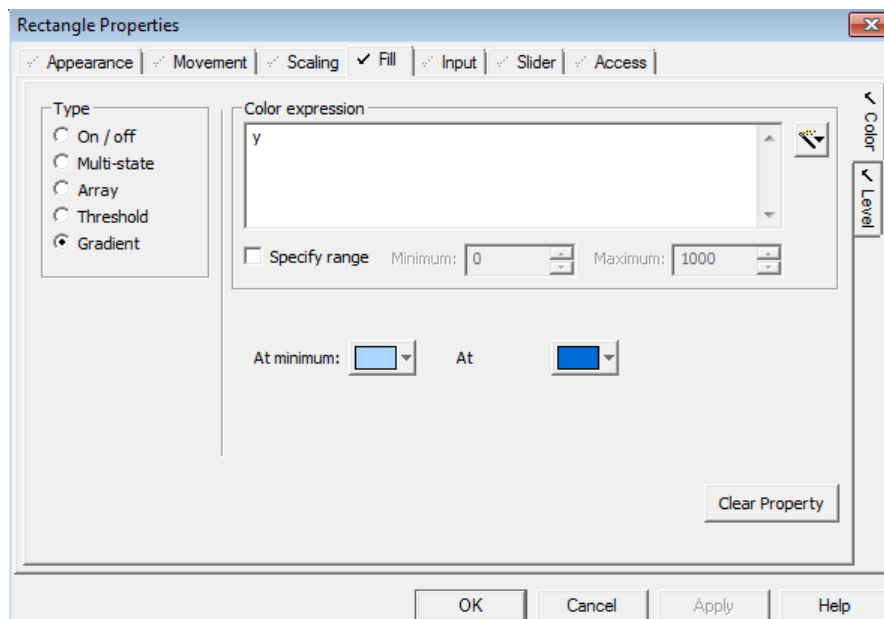


Implementarea blocului regulator se face folosind tehnica sliderelor, care numara pixelii si seteaza astfel proportional cu diferenta acestora valoarea in memorie. Atasat s-au folosit rectangles animate care isi schimba culoarea in functie de valoarea aleasa. Acest mod de implementare a fost folosit pentru alegerea P, kp si referintei. Blocul regulator citeste in permanenta nivelul de apa din rezervor si genereaza comenzi pentru valvele V[1] si V[2]. Calculul comenzii se face folosind formula :

$$u = u_1 + e \cdot (k_p + (T_i \cdot (T_e / 2))) + e_1 \cdot ((T_i \cdot (T_e / 2)) - k_p);$$

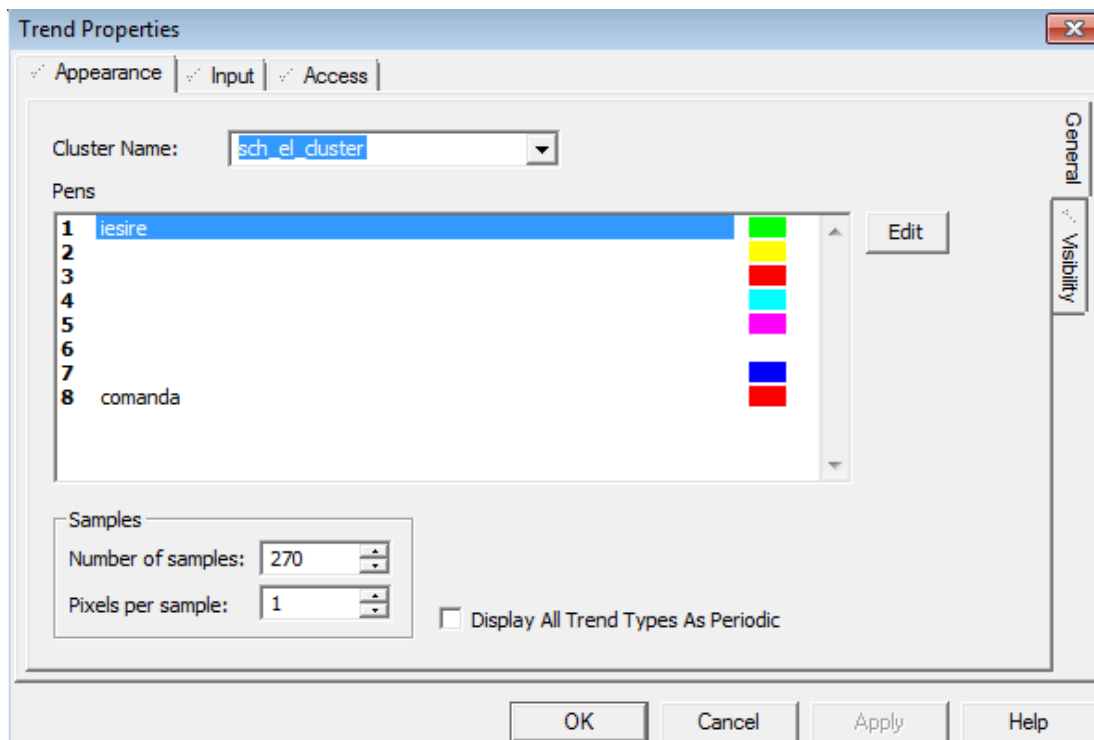


Rezervorul de apa este generat folosind un symbol set peste care s-a desenat un rectangle animat cu gradient in functie de valoarea de iesire y :



S-au generat in total un numar de 5 obiecte de tip "function" (codul atasat mai jos) care calculeaza in mod repetat comanda regulatorului in functie de referinta si iesire, consumul de apa din rezervor si logica programului general.

Obiectul de tip trend functioneaza doar daca se folosesc TAGuri trend carora le sunt atasate TAGuri generale (nu si taguri locale). S-au creat doua taguri de tip trend care proceseaza valorile pentru afisare :



3. Codul Sursa

```
FUNCTION fct_pompa()
```

```
    y = 0;
```

```
    nivel_apa_pompa = 0;
```

```
    nivel_apa_pompa2 = 0;
```

```
    stare_filtrare = 0;
```

```
    Toggle(stare_pompa);
```

```
    IF stare_pompa THEN
```

```
        nivel_apa_pompa = nivel_apa_pompa + 25;
```

```
        Sleep(0.5);
```

```

    nivel_apa_pompa = nivel_apa_pompa + 25;
    Sleep(0.5);
    nivel_apa_pompa2 = nivel_apa_pompa2 + 25;
    Toggle(stare_filtrare);
    nivel_apa_pompa = nivel_apa_pompa + 25;
    Sleep(0.5);
    nivel_apa_pompa2 = nivel_apa_pompa2 + 25;
    nivel_apa_pompa = nivel_apa_pompa + 25;
    Sleep(0.5);
    nivel_apa_pompa2 = nivel_apa_pompa2 + 25;
    Sleep(0.5);
    nivel_apa_pompa2 = nivel_apa_pompa2 + 25;
    Sleep(0.5);
ELSE
    nivel_apa_pompa = 0;
    nivel_apa_pompa2 = 0;
END

END

FUNCTION reglare()
    IF nivel_pompa3 = 100 THEN
        e = r - y;
        //u = u1 + e*(kp + ((kp*Te)/Ti)) - e1*kp;
        u = u1 + e*(kp + (Ti*(Te/2))) + e1*((Ti*(Te / 2)) - kp);
        y = (u*Te + 5 * y1) / (5 + Te);

        e1 = e;
        y1 = y;
        u1 = u;
    END
END
END

```

FUNCTION regulator()

Toggle(stare_regulator);

IF stare_regulator **THEN**

 nivel_pompa3 = nivel_pompa3 + 25;

Sleep(0.5);

 nivel_pompa3 = nivel_pompa3 + 25;

Sleep(0.5);

 nivel_pompa3 = nivel_pompa3 + 25;

Sleep(0.5);

 nivel_pompa3 = nivel_pompa3 + 25;

Sleep(0.5);

ELSE

 nivel_pompa3 = nivel_pompa3 - 25;

 y = y - 100;

Sleep(0.5);

 nivel_pompa3 = nivel_pompa3 - 25;

 y = y - 100;

Sleep(0.5);

 nivel_pompa3 = nivel_pompa3 - 25;

 y = y - 100;

Sleep(0.5);

 nivel_pompa3 = nivel_pompa3 - 25;

 y = 0;

Sleep(0.5);

END

END

FUNCTION evac()

IF y>100 **THEN**

 evac_temp = 1;

ELSE


```

        evac_temp = 0;
    END
    IF consumatori = 0 AND y > 100 THEN
        evacuaire = 0;
    ELSE
        evacuaire = 1;
    END
END

FUNCTION consum()
schimba_stare = schimba_stare + 1;
    IF y > 100 AND consumatori = 1 AND schimba_stare > 30 THEN
        c[1] = Rand(2);
        c[2] = Rand(2);
        c[3] = Rand(2);
        c[4] = Rand(2);
        c[5] = Rand(2);
        c[6] = Rand(2);
        schimba_stare = 0;
    END
END

FUNCTION calcul_consum()
INT i;
FOR i=1 TO 6 DO
    IF c[i] = 1 THEN
        y = y - 25;
    END
END
END
END

```

```
FUNCTION initializare()  
    IF flag_init = 0 THEN  
        y1=0.0;  
        y=0.0;  
        u=0.0;  
        u1=0.0;  
        e=0.0;  
        e1=0.0;  
        Te=0.25;  
        flag_init = 1;  
    END  
END
```