

# Sisteme de reglare automata

- **Obiective**

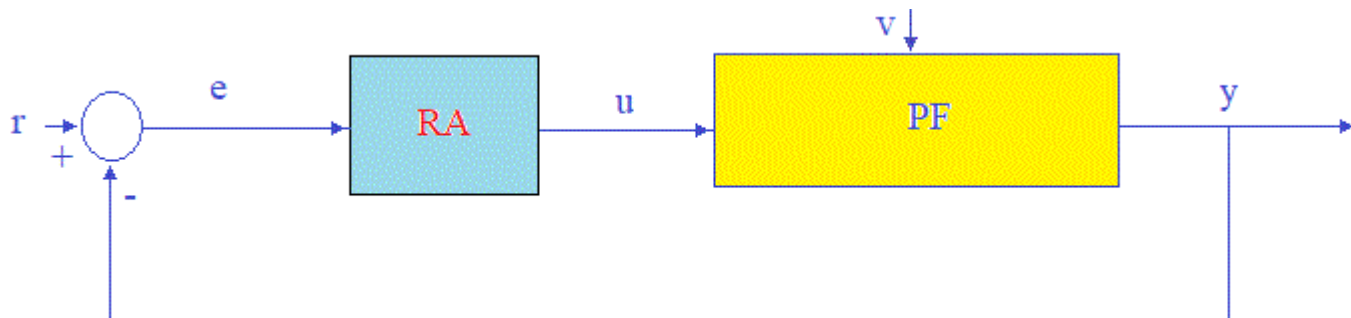
- Prezentarea unui sistem de reglare automata
- Prezentarea unui sistem SCADA care contine sisteme de reglare automata
- Prezentarea unui sistem SCADA pentru controlul debitului
- Prezentarea unui sistem SCADA pentru controlul temperaturii

- **Organizarea sarcinilor de lucru**

- Parcurgeti cele patru capitole ale cursului.
- In cadrul fiecarui capitol urmariti exemplele ilustrative si incercati sa le realizati in medul de dezvoltare "Citect".
- Fixati principalele idei ale cursului, prezentate în rezumat.
- Completati testul de autoevaluare.
- Timpul de lucru pentru parcurgerea testului de autoevaluare este de 15 minute.

## 1. Prezentarea unui sistem de reglare automata

SRA - Sistemele de Reglare Automata sunt sisteme cu bucla de reactie (loop control) care functioneaza pe baza analizei in permanenta a valorii de iesire (marimii reglate), preluata prin intermediul reactiei negative. Diferenta dintre valoarea de iesire si valoarea de referinta, numita "eroare", este folosita de SRA pentru eliminarea acesteia sau mentinerea ei in anumite intervale prestabilite.



Un SRA se compune din urmatoarele elemente:

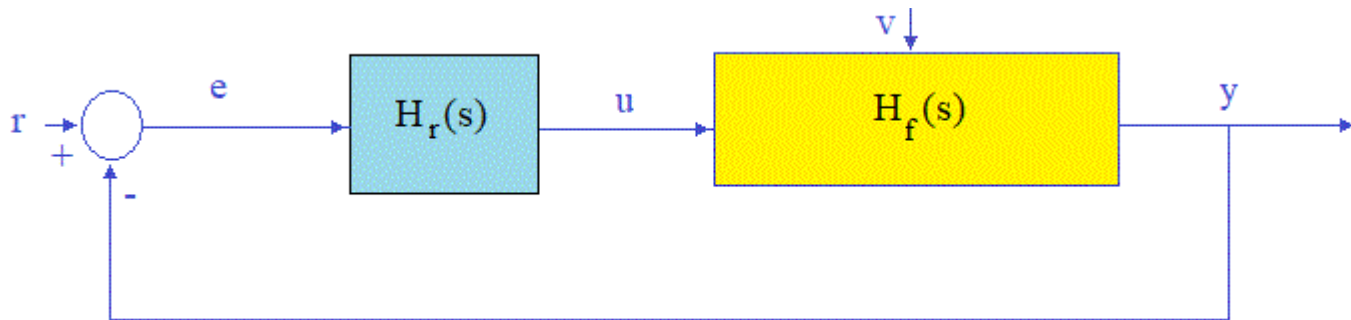
- RA - Regulator automat;
- PF - Partea fixata(Procesul condus);

Marimile definite:

- r - Referinta;
- e - Eroarea;
- u - Comanda;

- $v$  - Perturbatia;
- $y$  - Variabila de proces(Marimea reglata, iesirea);

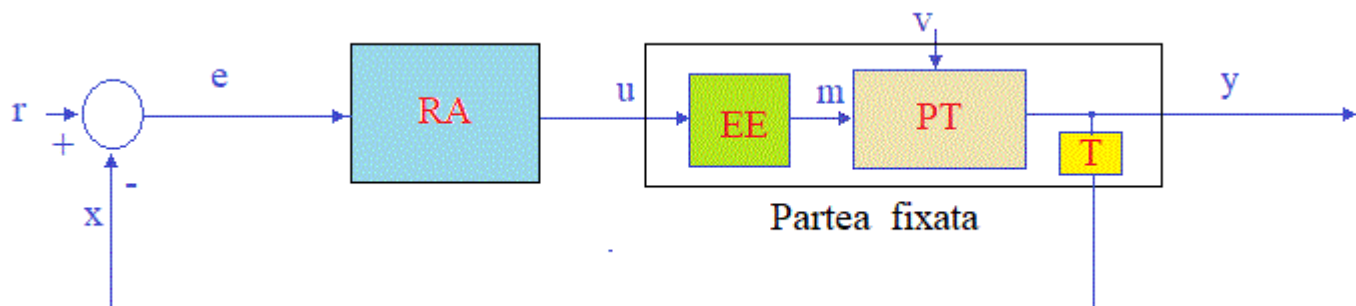
Daca definim  $H_r(s)$  functia de transfer a regulatorului si  $H_f(s)$  functia de transfer a partii fixate, Schema unui SRA devine:



Partea fixata(PF) contine procesul tehnologic(PT) asupra caruia actioneaza regulatorul automat(RA).

Procesul tehnologic(PT) este comandat de catre regulatorul automat(RA) prin intermediul elementelor de executie(EE). Citirea marimii reglate( $y$  - variabila de proces) se face prin intermediul traductorilor(T) care ofera marimea  $x$  proportionla cu marimea  $y$  adica  $x=k*y$ .

Putem deci reprezenta un SRA sub forma:



Unde:

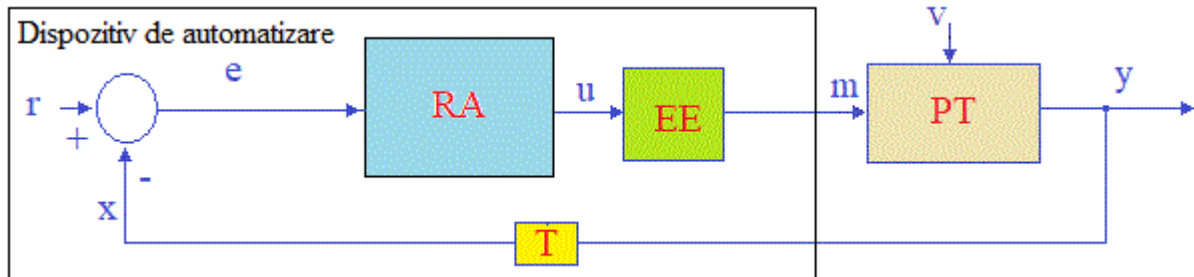
- RA - Regulator automat;
- PT - Procesul tehnologic;
- EE - Element de Executie;
- T - Traductor;

Marimile definite:

- $r$  - Referinta;
- $e$  - Eroarea;
- $u$  - Comanda;
- $m$  - Executie;
- $v$  - Perturbatia;
- $y$  - Variabila de proces(Marimea reglata, iesirea);

- x - Reactia;

Mult mai fireasca ar fi reprezentarea in care procesul tehnologic(PT) ar fi reprezentat separat, la care se adauga in mod firesc dispozitivul de automatizare. Aceasta reprezentare ar corespunde dezvoltarii firesti a sistemelor tehnologice in care initial se realizeaza un proces tehnologic dupa care se automatizeaza.



## 2. Sisteme de reglare automata in SCADA TIA-Portl

Un SRA de tip PID mentine valoarea iesirii y in jurul valorii de referinta r prin intermediul comenzii u. Comanda u este generata de SRA prin intermediul unei functii de transfer caracterizata de trei constante:  $k_p, k_i, k_d$  si  $T_i$ .

- $k_p$  este constanta - Proporzionala (cu valori intre 0-4)
- $k_i$  este constanta - Integrativa (cu valori intre 0-5)
- $k_d$  este constanta - Derivativa (cu valori intre 0.2-2)
- $T_e$  este timpul de esantionare (250 ms adica timpul de Refresh al HMI)

In imaginea de jos este reprezentata comanda u (cu rosu) si iesirea y(cu verde).

| Sistem de Reglare Automata PID                      |         |
|-----------------------------------------------------|---------|
| Your browser does not support the HTML5 canvas tag. |         |
| r:                                                  | $k_p$ : |
| ki:                                                 | $k_d$ : |

Pentru implementarea regulatorului PID se va folosi metoda "Velocity". Conform acesti metode, comanda u din momentul k depinde de referinta r, de iesirea y si de  $e_v$  ( $e_v$  din momentul k-1). Intervalul de timp dintre momentul k si momentul k-1 este  $T_e$  (Timpul de esantionare).

Comanda u se obtine prin insumarea elementului proportional, integrativ (inte) si derivativ(deriv) astfel:

```
e = r - y;
inte = inte + e * Te;
deriv = (e - e_v) / Te;
```

```
u = kp * e + ki * inte + kd * deriv;  
e_v = e;
```

In care:

- y este iesirea din pasul curent
- r este valoarea referinta
- e este eroarea calculata (r-y) din pasul curent
- e\_v este valoarea calculata pentru e in pasul anterior
- u este valoarea calculata pentru comanda din pasul curent
- kp este constanta - Proportionala (cu valori intre 0-4)
- ki este constanta - Integrativa (cu valori intre 0-5)
- kd este constanta - Derivativa (cu valori intre 0.2-2)
- Te este timpul de esantionare (250 ms adica timpul de Refresh al HMI)

Variabila de proces adica iesirea y reprezentand totodata marimea reglata ar trebui sa provina din procesul tehnologic(PT) prin plasarea unui traductor. Folosind de exemplu functia de transfer a unui sistem de ordinul 5 si anume:  $H_f(s)=1/(5s+1)$ . Dupa discretizare obtinem relatia:

$$y = (u \cdot T_e + 5 \cdot y_v) / (5 + T_e);$$

In care:

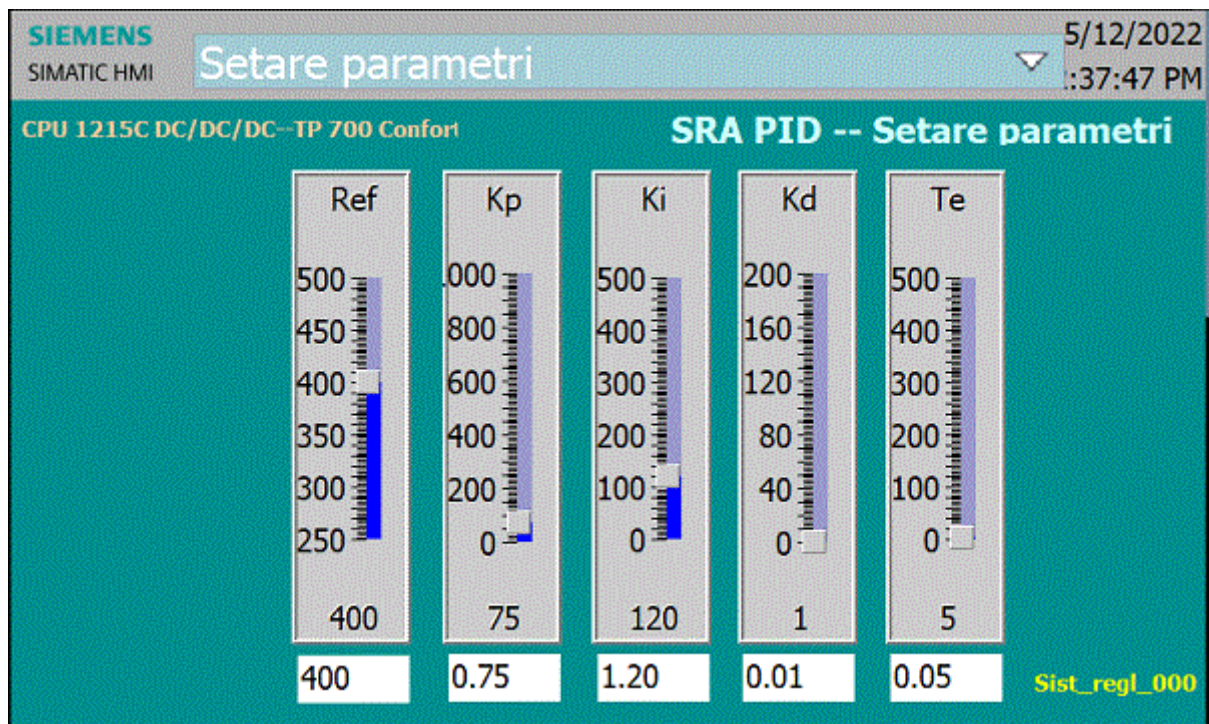
- y este valoarea calculata pentru iesire din pasul curent
- y\_v este valoarea calculata pentru y in pasul anterior
- u este valoarea calculata pentru comanda din pasul curent
- Te este timpul de esantionare (250 ms adica timpul de Refresh al HMI)

Dupa cum se observa ea depinde de de y\_v(y din pasul anterior), comanda u din pasul curent si de timpul de esantionare Te.

## Sistem de reglare automata PID realizat in SCADA TIA-Portal

Pe baza ecuatiilor descrise mai sus, vom realiza proiectul SCADA: Sist regl 000 proiect ce contine un SRA (sistem de reglare automata) de tip PID (proportional-integrativ-derivativ) discret.

In cadrul acestui proiect, vom crea pentru inceput, Screen-ul **Sra\_pid\_init** pentru initializarea parametrilor.



Pentru a crea acest Screen, avem nevoie de tag-urile: Kp, Ki, Kd, Te de tip "real" si r (Ref: set point) de tip "int".

Obiectele de tip "Slider" utilizate pentru setarea parametrilor, nu admit decat atribuirea de tag-uri de tip "int".

Vom defini tag-urile: Kp\_int, Ki\_int, Kd\_int, Te\_int de tip "int" dupa care vom calcula valoarea tag-urilor: Kp, Ki, Kd, Te prin impartirea acestora la 100 daca dorim doua zecimale de exemplu.

Pe evenimentul "Change al fiecarui obiect "Slider" vom atribui functia "sra\_init\_p()"

```

Sub init p()
    If r=0 Then
        SmartTags("Kp_int")=75
        SmartTags("Ki_int")=120
        SmartTags("Kd_int")=1
        SmartTags("Te_int")=5
        SmartTags("r")=400
    End If
    SmartTags("y")=0
    SmartTags("y_v")=0
    SmartTags("inte")=0
    SmartTags("deriv")=0
    SmartTags("e")=0
    SmartTags("e_v")=0
    SmartTags("u")=0
    SmartTags("Kp")=SmartTags("Kp_int")/100
    SmartTags("Ki")=SmartTags("Ki_int")/100
    SmartTags("Kd")=SmartTags("Kd_int")/100
    SmartTags("Te")=SmartTags("Te_int")/100
    SmartTags("r_scal")=SmartTags("r")/10
End Sub

```



## Implementarea algoritmului de regle automate in HMI

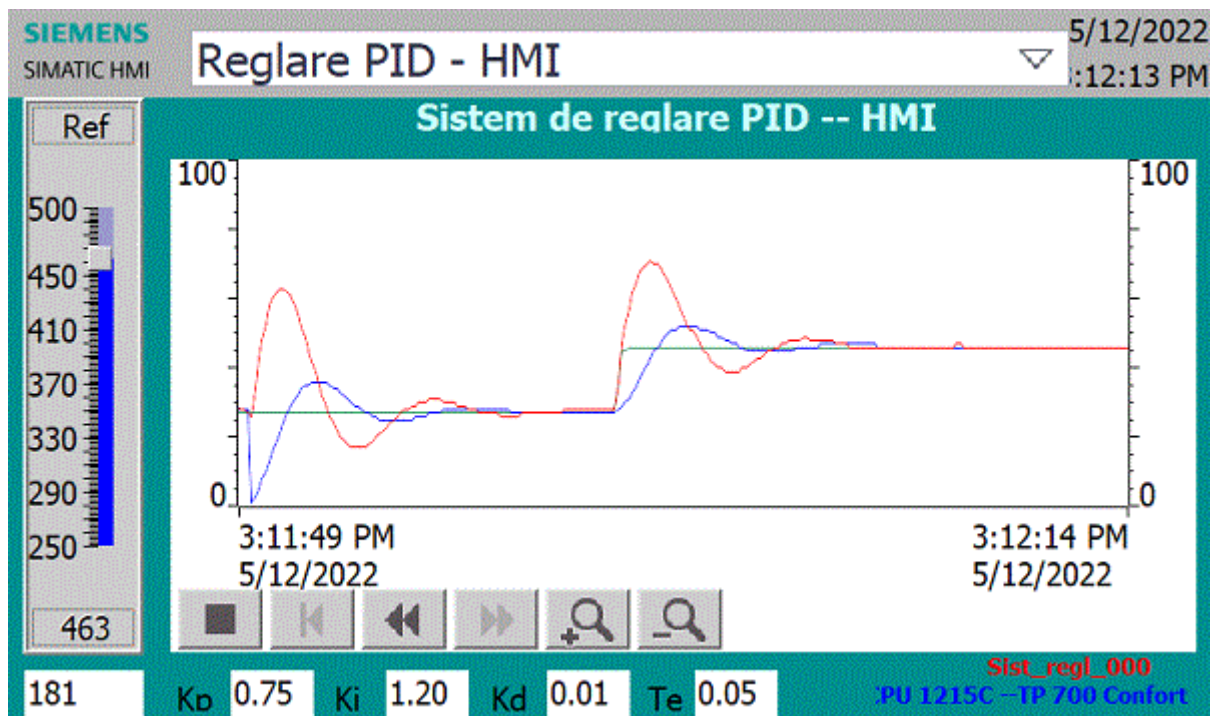
Pentru inceput vom implementa algoritmul pentru SRA in HMI.

Vom crea in continuare, in cadrul acestui proiect, Screen-ul **Sra\_pid\_hmi** unde vom afisa grafic evolutia in timp a marimilor u,y,r, folosind obiectul "Trend View". Obiectul "Trend View" admite adaugarea de trend-uri dar tag-urile corespunzatoare trebuie sa fie scalate in domeniul 0-100. Vom introduce pe langa Tagurile u,y,r tag-urile u\_scal, y\_scal, r\_scal scalate in domeniul 0-100

| HMI tags |                  |                   |           |               |
|----------|------------------|-------------------|-----------|---------------|
|          | Name ▲           | Tag table         | Data type | Connection    |
| [-]      | Contor           | Default tag table | UInt      | HMI_Connect   |
| [-]      | deriv            | Default tag table | Real      | HMI_Connect   |
| [-]      | e                | Default tag table | Real      | HMI_Connect   |
| [-]      | e_v              | Default tag table | Real      | HMI_Connect   |
| [-]      | inte             | Default tag table | Real      | HMI_Connect   |
| [-]      | Kd               | Default tag table | Real      | HMI_Connect   |
| [-]      | Kd_int           | Default tag table | Int       | <Internal tag |
| [-]      | Ki               | Default tag table | Real      | HMI_Connect   |
| [-]      | Ki_int           | Default tag table | Int       | <Internal tag |
| [-]      | Kp               | Default tag table | Real      | HMI_Connect   |
| [-]      | Kp_int           | Default tag table | Int       | <Internal tag |
| [-]      | mod              | Default tag table | UInt      | HMI_Connect   |
| [-]      | r                | Default tag table | Int       | HMI_Connect   |
| [-]      | r_scal           | Default tag table | Int       | HMI_Connect   |
| [-]      | Tag_ScreenNumber | Default tag table | UInt      | <Internal tag |
| [-]      | Te               | Default tag table | Real      | HMI_Connect   |
| [-]      | Te_int           | Default tag table | Int       | <Internal tag |
| [-]      | u                | Default tag table | Real      | HMI_Connect   |
| [-]      | u_scal           | Default tag table | Int       | HMI_Connect   |
| [-]      | y                | Default tag table | Real      | HMI_Connect   |
| [-]      | y_scal           | Default tag table | Int       | HMI_Connect   |
| [-]      | y_v              | Default tag table | Real      | HMI_Connect   |

Algoritmul pentru implementarea SRA presupune reglarea in mai multe iteratii.

Pe evenimentul "Load" al screen-ului "Sra\_pid\_hmi" se vol lansa functiile "init\_p()" si "pid\_hmi()".



Functia "pid\_hmi()" implementeaza algoritmul PID.

```

Sub pid_hmi()
  Dim inte, deriv, tm
  tm=Timer()
  Do While SmartTags("Tag_ScreenNumber")=3
    If Timer > tm+0.01 Then
      e = r - y
      inte = inte + e * Te
      deriv = (e - e_v) / Te
      u = Kp * e + Ki * inte + Kd * deriv
      e_v = e
      y = (u*Te + 5 * y_v) / (5 + Te)
      y_v=y
      u_scal=u/10
      y_scal=y/10
      r_scal=r/10
      tm=Timer()
    End If
  Loop
End Sub

```

## Implementarea algoritmului de regle automata in PLC

Implementarea algoritmul de regle automata in HMI, nu este recomandat deoarece functionarea procesului reglat depinde de HMI care este mult mai lent. Pe de alta parte HMI-ul trebuie sa fie tot timpul pornit, functionarea procesului depinzand de HMI. Solutia mai eficienta consta in implementarea algoritmului de regle automata in PLC.



Vom crea in continuare, in cadrul acestui proiect, Screen-ul **Sra\_pid\_plc** si vom implementa functia de reglare in PLC. Avem nevoie de urmatoarele tag-uri PLC:

| PLC tags |        |                   |           |         |                          |                                     |
|----------|--------|-------------------|-----------|---------|--------------------------|-------------------------------------|
|          | Name   | Tag table         | Data type | Address | Retain                   | Access...                           |
| 1        | Contor | Default tag table | UInt      | %MW0    | <input type="checkbox"/> | <input checked="" type="checkbox"/> |
| 2        | e      | Default tag table | Real      | %MD4    | <input type="checkbox"/> | <input checked="" type="checkbox"/> |
| 3        | e_v    | Default tag table | Real      | %MD8    | <input type="checkbox"/> | <input checked="" type="checkbox"/> |
| 4        | Kd     | Default tag table | Real      | %MD12   | <input type="checkbox"/> | <input checked="" type="checkbox"/> |
| 5        | Ki     | Default tag table | Real      | %MD16   | <input type="checkbox"/> | <input checked="" type="checkbox"/> |
| 6        | Kp     | Default tag table | Real      | %MD20   | <input type="checkbox"/> | <input checked="" type="checkbox"/> |
| 7        | Te     | Default tag table | Real      | %MD24   | <input type="checkbox"/> | <input checked="" type="checkbox"/> |
| 8        | u      | Default tag table | Real      | %MD28   | <input type="checkbox"/> | <input checked="" type="checkbox"/> |
| 9        | y      | Default tag table | Real      | %MD32   | <input type="checkbox"/> | <input checked="" type="checkbox"/> |
| 10       | y_v    | Default tag table | Real      | %MD36   | <input type="checkbox"/> | <input checked="" type="checkbox"/> |
| 11       | r      | Default tag table | Int       | %MW40   | <input type="checkbox"/> | <input checked="" type="checkbox"/> |
| 12       | r_scal | Default tag table | Int       | %MW42   | <input type="checkbox"/> | <input checked="" type="checkbox"/> |
| 13       | u_scal | Default tag table | Int       | %MW44   | <input type="checkbox"/> | <input checked="" type="checkbox"/> |
| 14       | y_scal | Default tag table | Int       | %MW46   | <input type="checkbox"/> | <input checked="" type="checkbox"/> |
| 15       | inte   | Default tag table | Real      | %MD48   | <input type="checkbox"/> | <input checked="" type="checkbox"/> |
| 16       | deriv  | Default tag table | Real      | %MD52   | <input type="checkbox"/> | <input checked="" type="checkbox"/> |
| 17       | mod    | Default tag table | UInt      | %MW56   | <input type="checkbox"/> | <input checked="" type="checkbox"/> |

Vom defini in Program Block -- Cyclic Interrupt --urmatoarea rutina:

```

"Contor" := "Contor" + 1;
IF "Contor" > 999 THEN
    "Contor" := 0;
END_IF;
IF "mod" = 2 THEN
    "e" := INT_TO_REAL("r") - "y";
    "inte" := "inte" + "e" * "Te";
    "deriv" := ("e" - "e_v") / "Te";
    "u" := "Kp" * "e" + "Ki" * "inte" + "Kd" * "deriv";
    "e_v" := "e";
    "y" := ("u" * "Te" + 5 * "y_v") / (5 + "Te");
    "y_v" := "y";
    "u_scal" := REAL_TO_INT("u" / 10);
    "y_scal" := REAL_TO_INT("y" / 10);
    "r_scal" := REAL_TO_INT("r" / 10);
END_IF;

```

Toate variabilele utilizate in rutina de sus trebuiesc initializate, vom defini in Program Block -- Startup -- urmatoarea rutina:

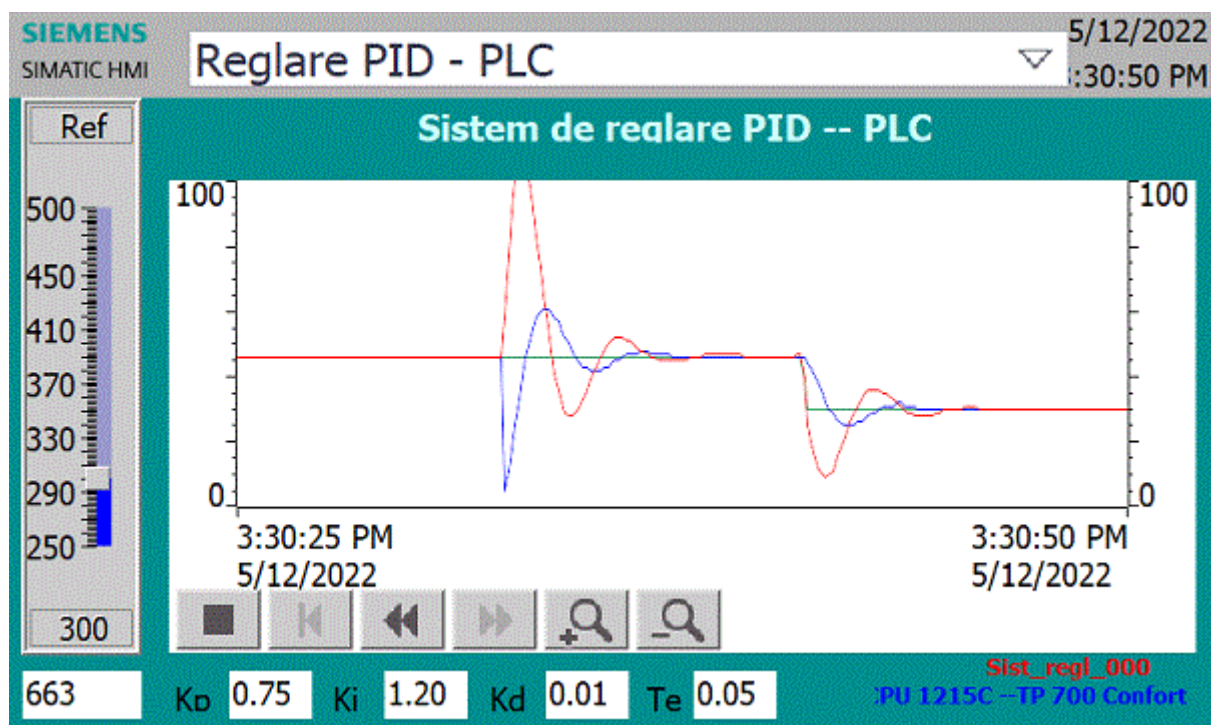


```

"mod" := 0;
"Contor" := 0;
"Kp" := 0.75;
"Ki" := 1.2;
"Kd" := 0.01;
"r" := 400;
"r_scal" := 0;
"e" := 0;
"e_v" := 0;
"y" := 0;
"y_v" := 0;
"r" := 400;
"y_scal" := 0;
"inte" := 0;
"deriv" := 0;
"u" := 0;
"u_scal" := 0;

```

Putem crea acum creen-ul **Sra\_pid\_plc**.

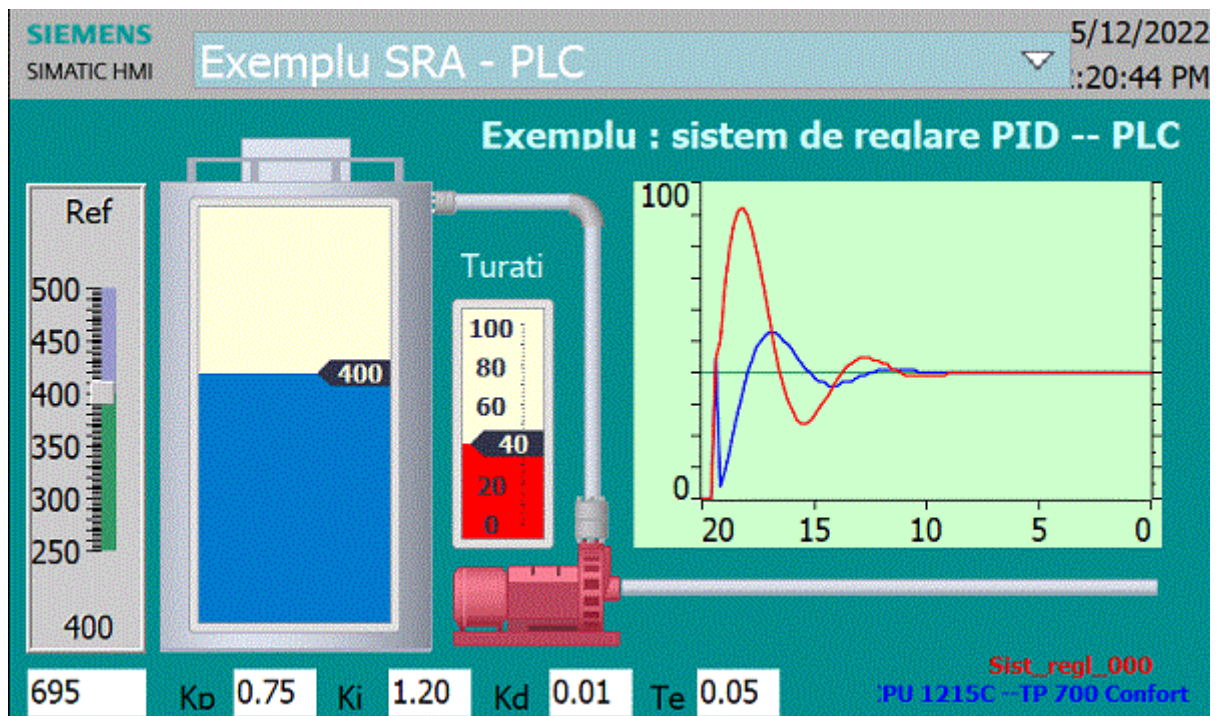


Pe evenimentul "Load" al screen-ului "Sra\_pid\_hmi" se va lansa functia "init\_p()" si vom seta Tag-ul Mod=2, lansandu-se astfel functia de reglare din PLC.

### Sistem de reglare automata a nivelului unui lichid

Urmatorul screen **Sra\_pid\_nivel** foloseste functia de reglare din PLC pentru a regla nivelul de lichid dintr-un tank.

Pe evenimentul "Load" al screen-ului "Sra\_pid\_nivel" se va lansa functia "init\_p()" si va seta Tag-ul Mod=2



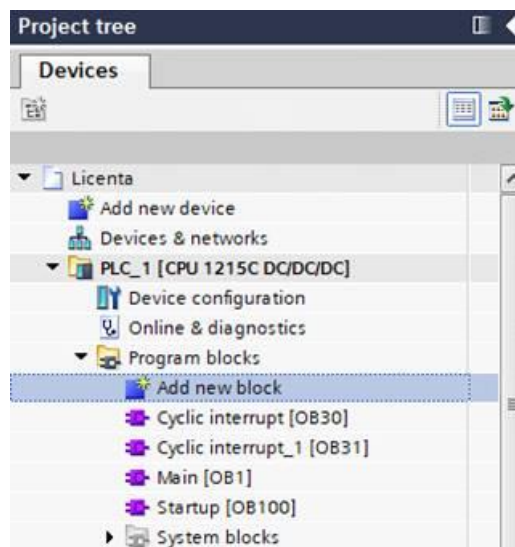
### 3. Utilizarea unui bloc PID Compact

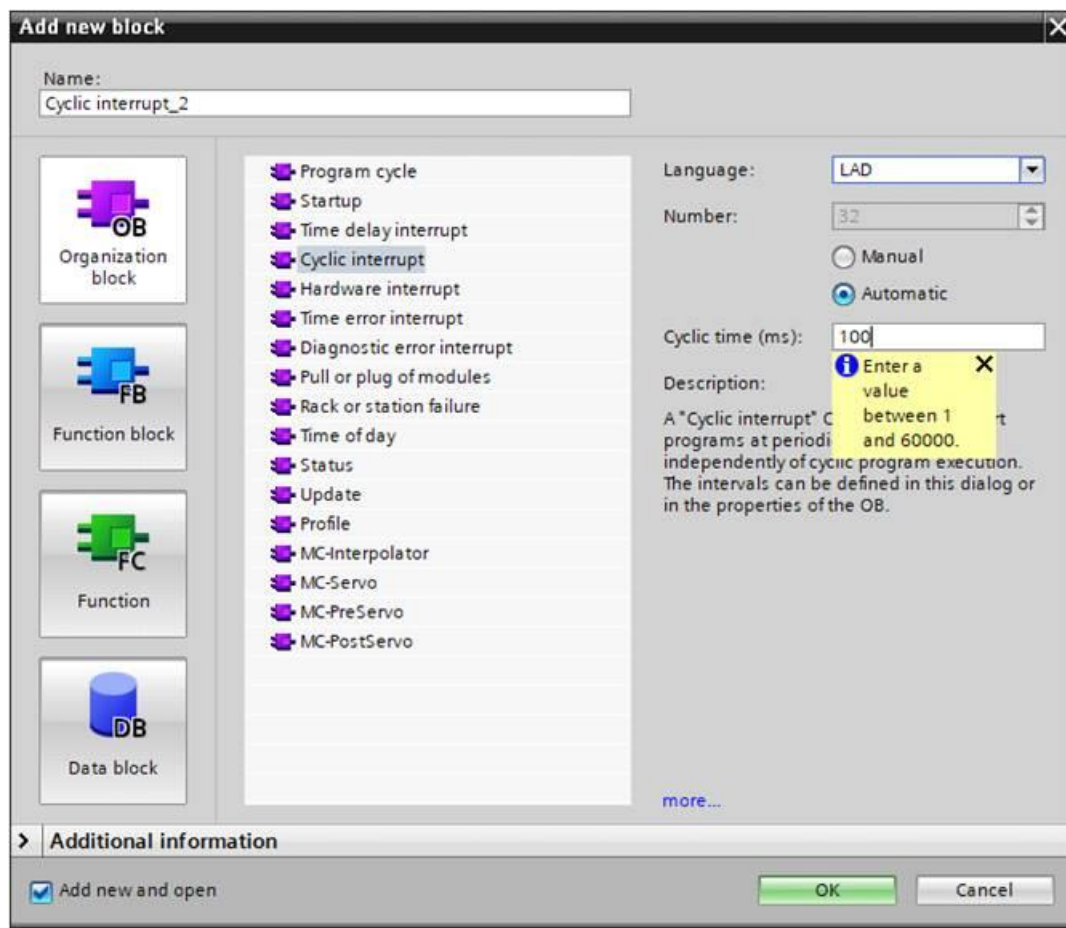
Capitol realizat cu contributia stud: Iulia Coros

Vom folosi in continuare blocul PID Compact pentru a realiza proiectul: [Sist\\_regl\\_001](#) proiect ce contine un SRA de tip PID Compact.

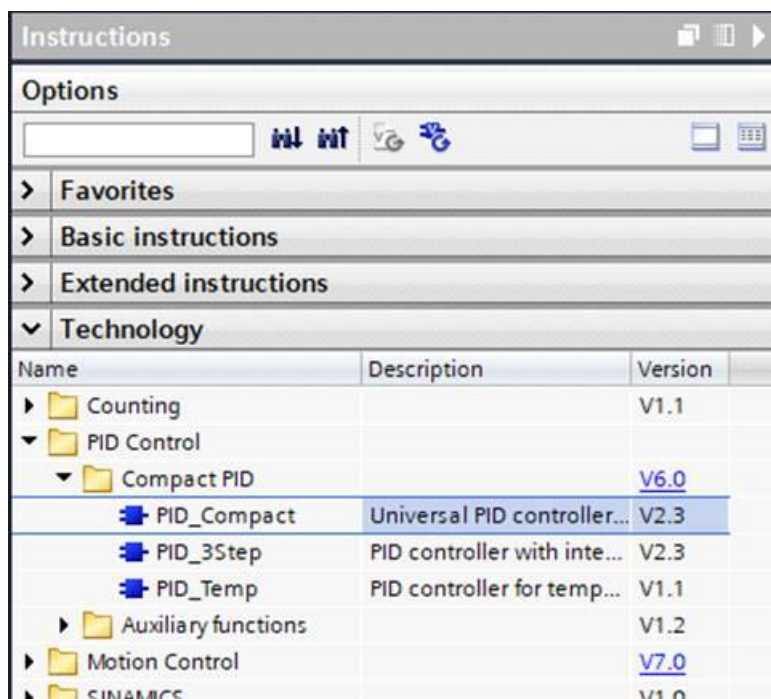
#### Configurare bloc PID Compact

**Pasul 1** – Adăugăm în proiectul nostru un nou bloc de tipul Cyclic interrupt în limbaj Leader și setăm Cyclic time pe 100 de milisecunde.





**Pasul 2** – În acest bloc creat adăugăm blocul PID Compact pe care îl găsim la categoria Technology.

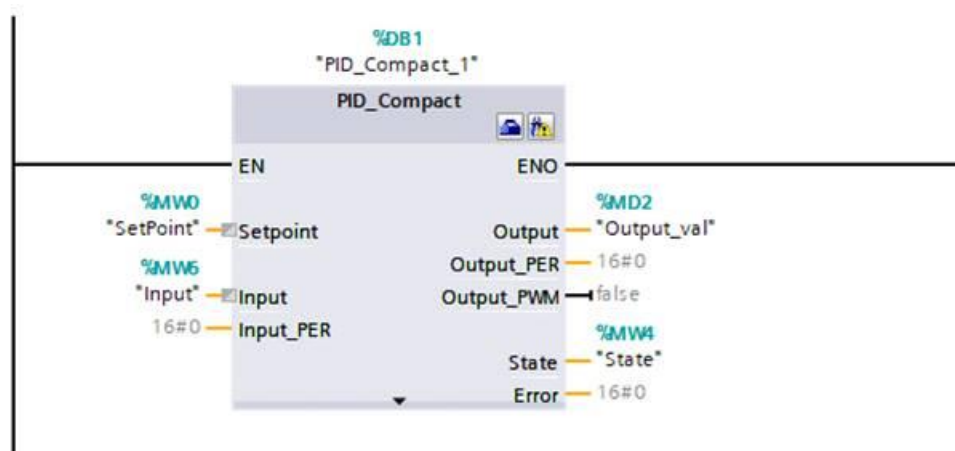


**Pasul 3** – Ne creăm în tabelul de tag-uri următoarele variabile.

| Default tag table |            |           |         |                          |                                     |                                     |                                     |  |
|-------------------|------------|-----------|---------|--------------------------|-------------------------------------|-------------------------------------|-------------------------------------|--|
|                   | Name       | Data type | Address | Retain                   | Acces...                            | Writa...                            | Visibl...                           |  |
| 1                 | SetPoint   | Int       | %MW0    | <input type="checkbox"/> | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> |  |
| 2                 | Output_val | Real      | %MD2    | <input type="checkbox"/> | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> |  |
| 3                 | State      | Int       | %MW4    | <input type="checkbox"/> | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> |  |
| 4                 | Input      | Int       | %MW6    | <input type="checkbox"/> | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> |  |

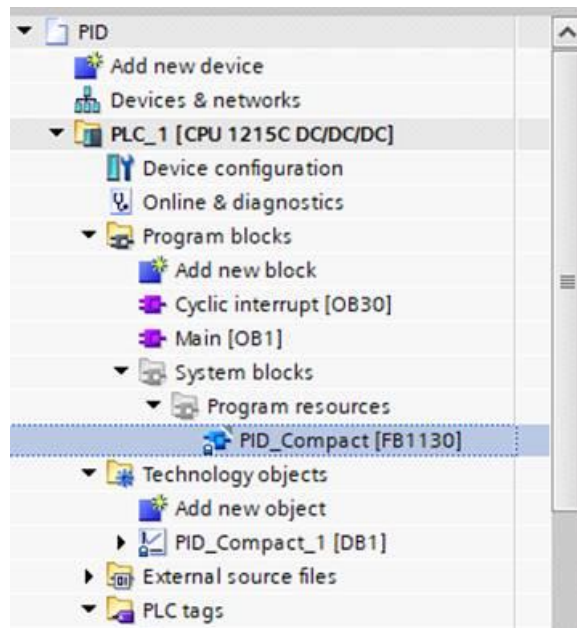
Atenție! Variabila pe care dorim să o atribuim ieșirii (Output) trebuie să fie de tip REAL.

Atribuim aceste tag-uri blocului PID în felul următor:

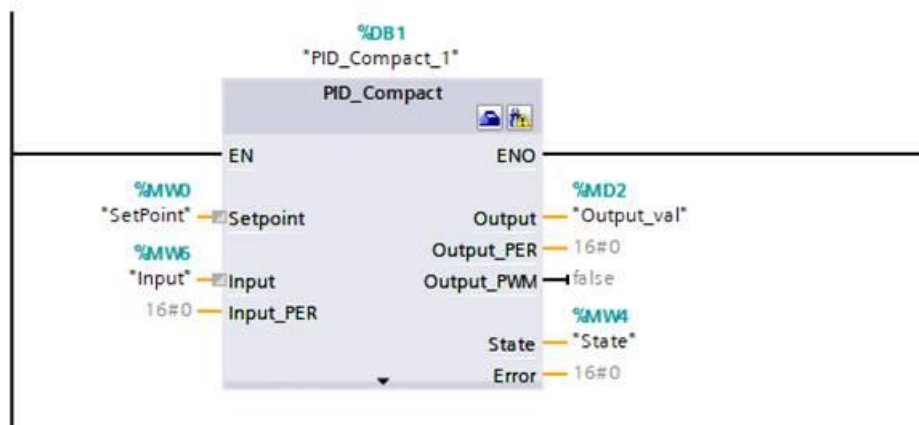


Putem vedea că regulatorul a fost adăugat în folderul System blocks și în Technology Objects.

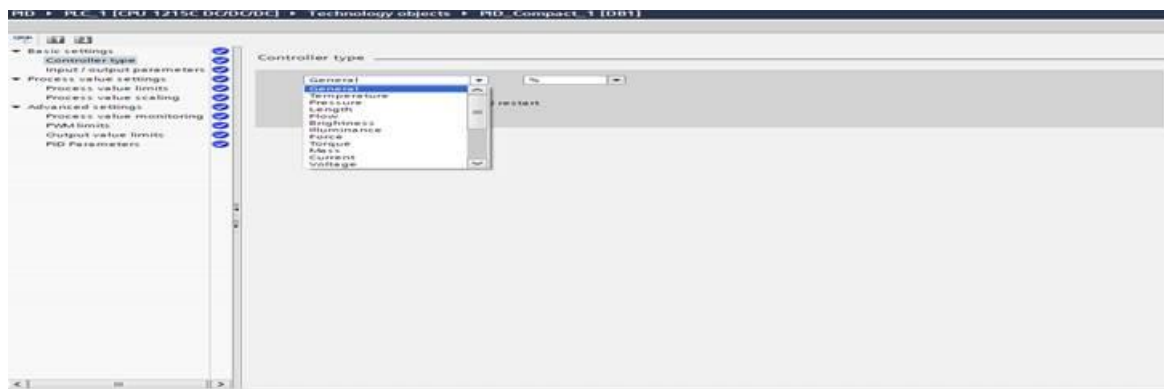




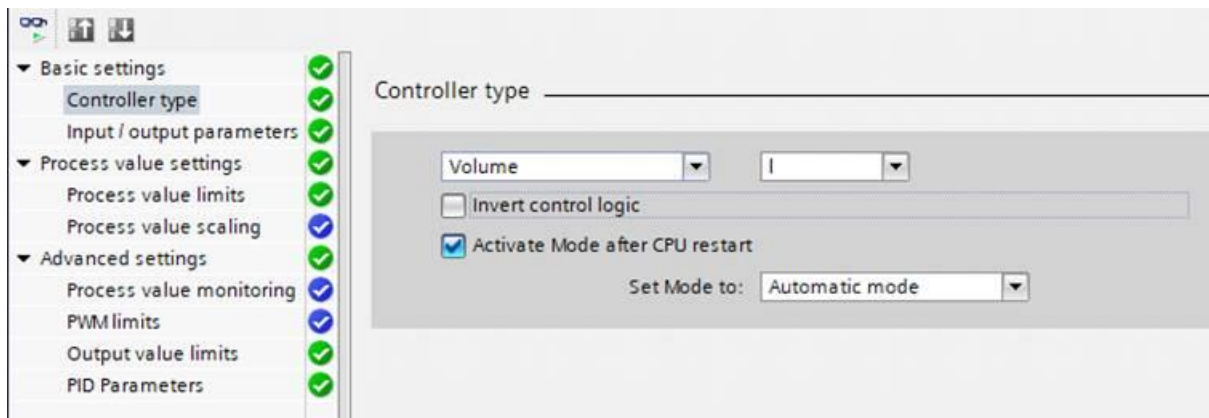
**Pasul 4** – Pentru meniul de configurare a regulatorului apăsăm pe această iconiță:



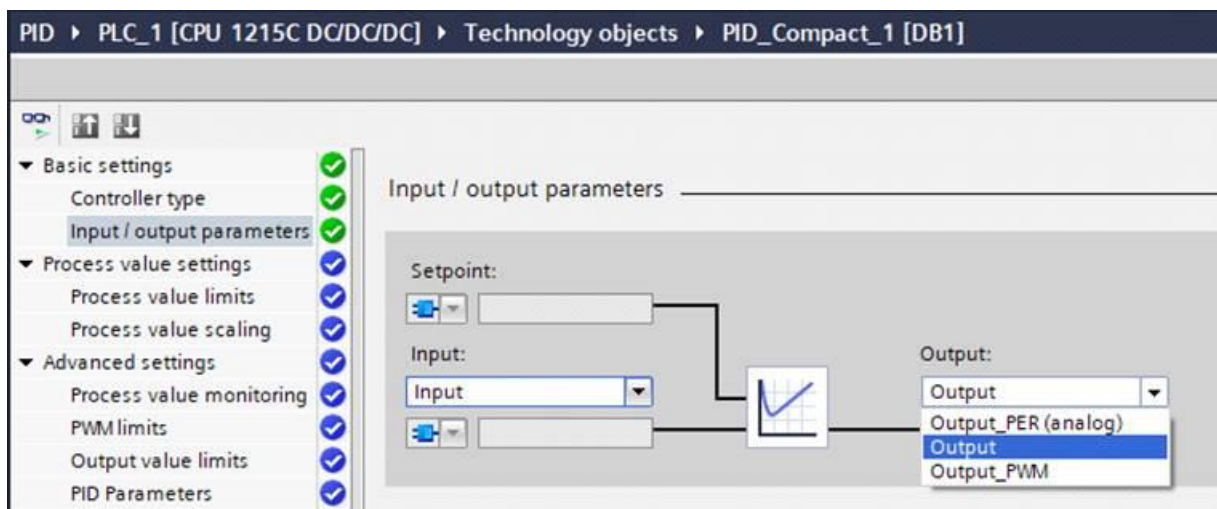
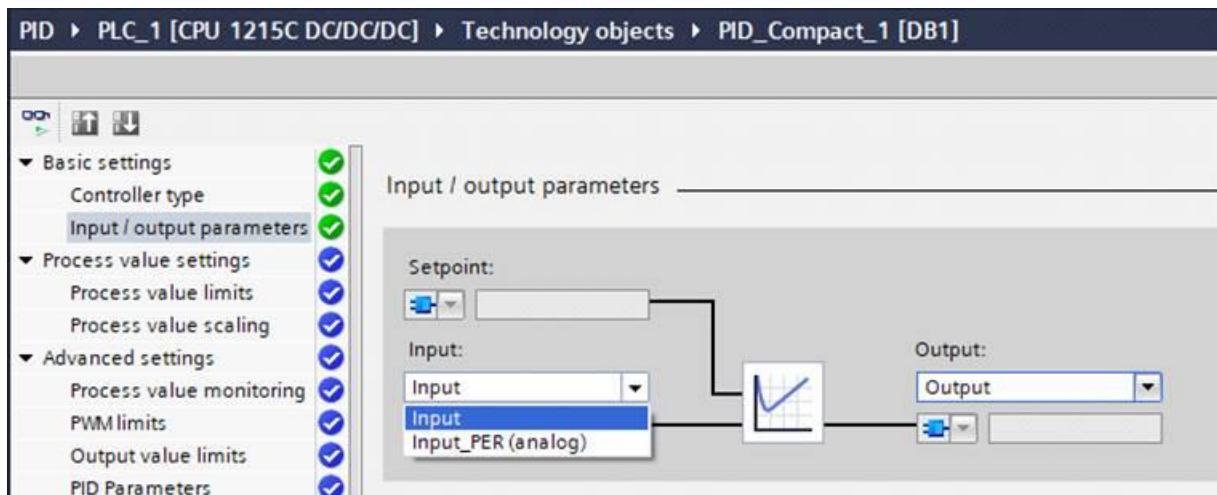
Din aceste setări de configurare putem selecta tipul de regulator în funcție de parametrul pe care dorim să îl reglăm.



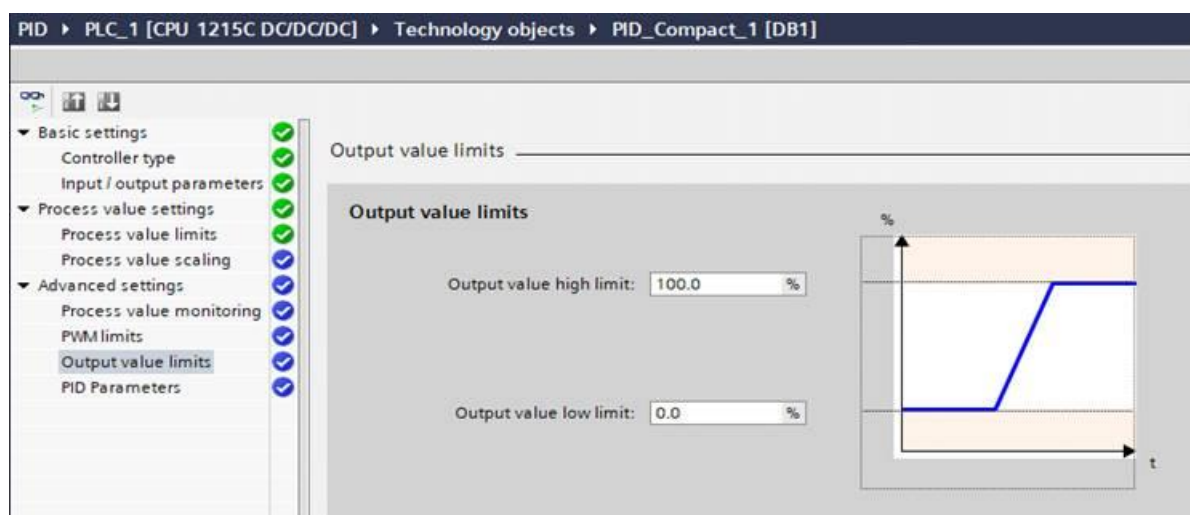
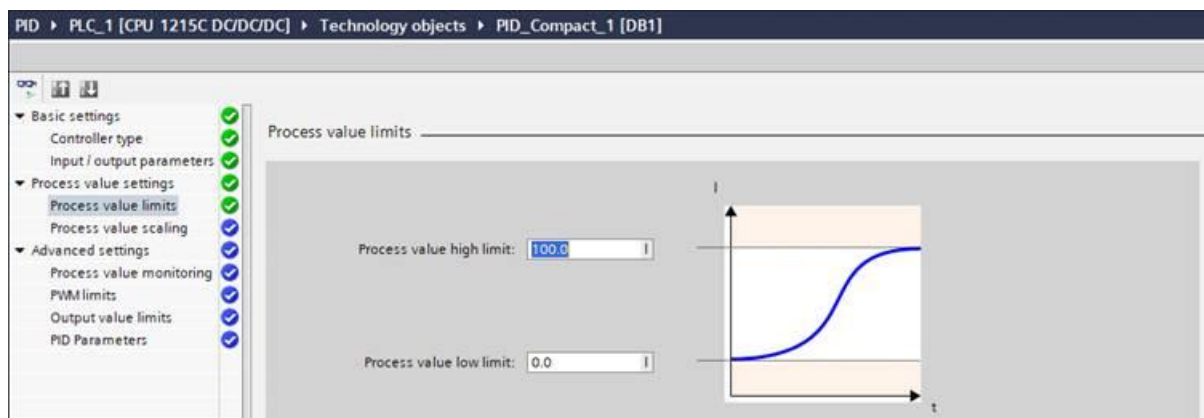
Foarte important! Regulatorul funcționează în modul automat doar după ce pornim simularea și dăm un restart (Stop CPU – Start CPU) unității centrale de procesare de pe butoanele:



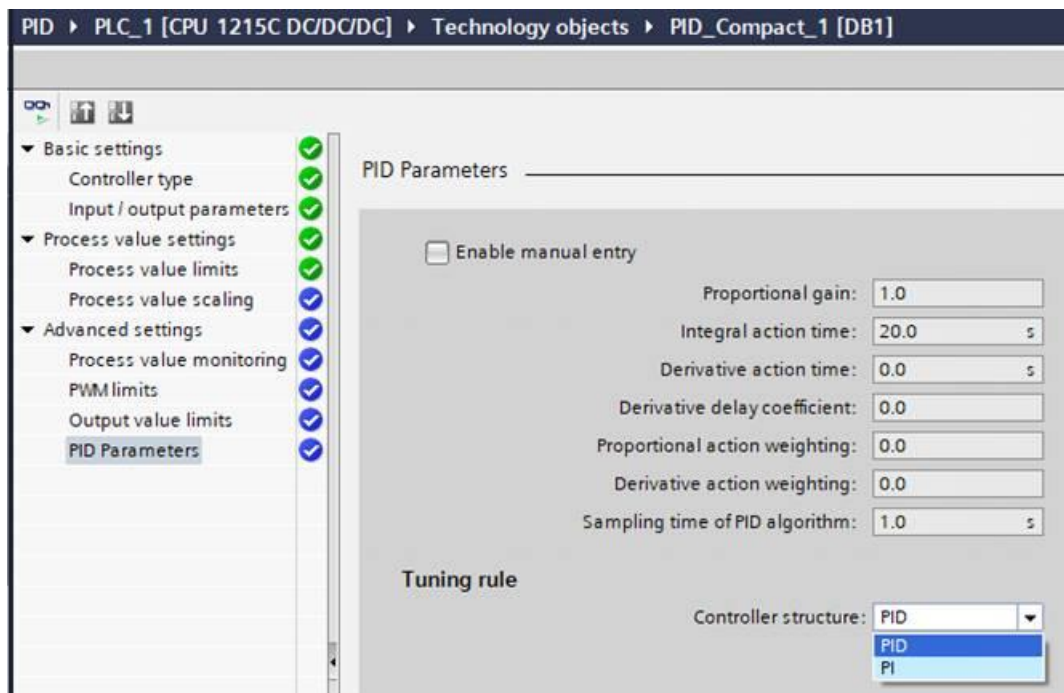
Alegem tipul intrării și ieșirii pe care dorim să le folosim.



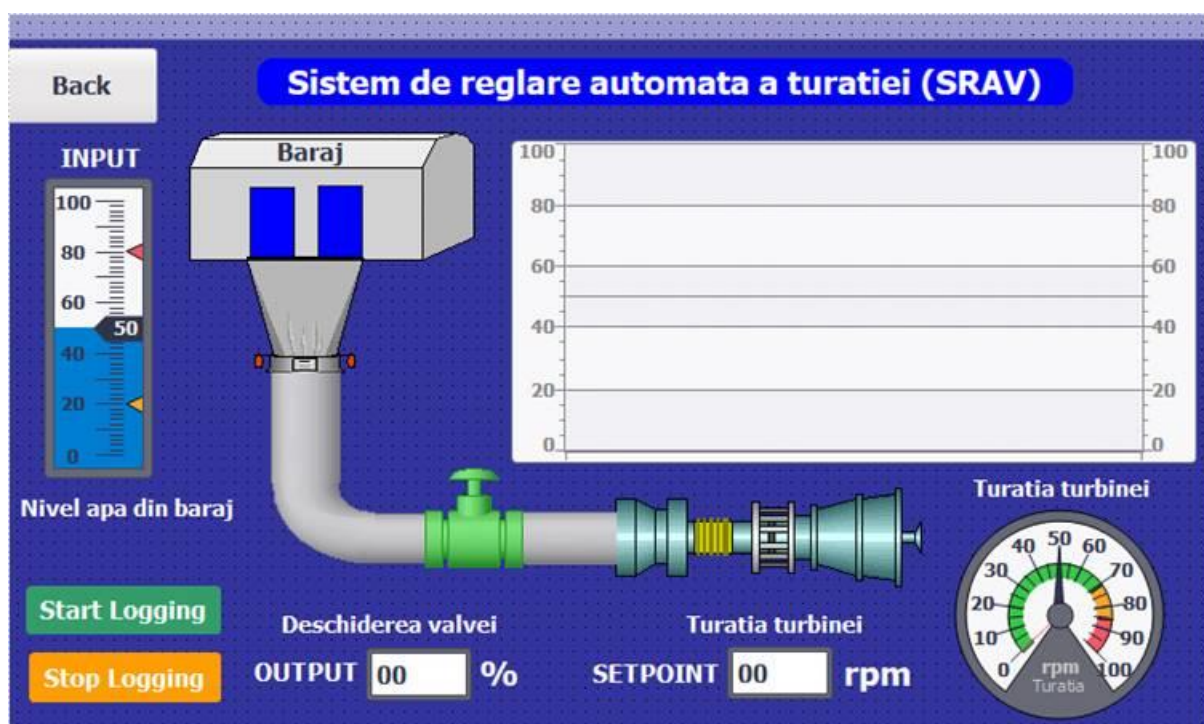
Setăm limitele superioare și inferioare ale semnalelor de intrare și ieșire.



Tot din acest meniu putem seta manual parametri regulatorului sau schimba tipul acestuia (PID sau PI).

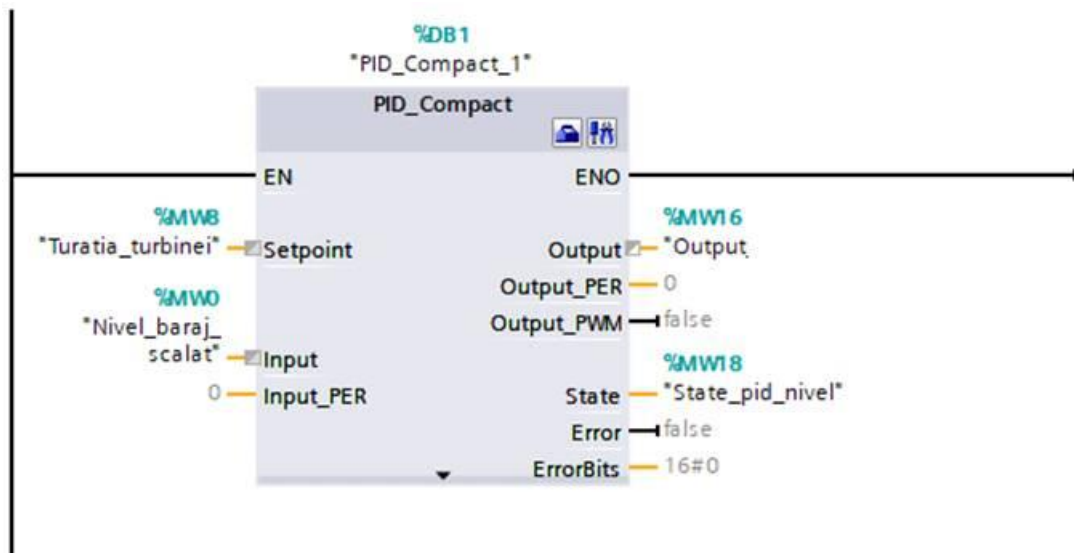


### Exemplu de utilizare – Sistem de reglare automată a turației unei turbine (SRAV)



Am adăugat blocul PID Compact și i-am setat ca și intrare nivelul apei din baraj, referința este turația turbinei pe care dorim să o menținem mereu la o valoare constantă, și la ieșire controlăm deschiderea valvei de admisie.





Așadar, noi vrem să menținem o turație constantă a turbinei și pentru acest lucru trebuie să reglăm deschiderea valvei de admisie în funcție de nivelul apei din baraj. Cu cât nivelul apei din baraj este mai mare, cu atât presiunea este mai mare și asta înseamnă ca forța cu care apa împinge lamelele turbinei este mai mare. Deci modelând deschiderea valvei, reglăm forța cu care apa împinge lamelele turbinei și aceasta se va învârti cu o turație constantă.

***Dacă creștem nivelul apei din baraj, deschiderea valvei trebuie să scadă.***

***Dacă scădem nivelul apei din baraj, deschiderea valvei trebuie să crească.***

