



## Prefață

# **Prefață**

Această lucrare este destinată defășurării lucrărilor de laborator la disciplina Sisteme de conducere a proceselor continue prevăzută în programa de învățământ a anului VI a Facultății de Automatică și Calculatoare, specializarea Ingineria Sistemelor Automate, din Universitatea Tehnică din Cluj-Napoca.



# Cuprins

1.	Simularea sistemelor automate .....	1
1.	Scopul lucrării .....	1
2.	Considerații teoretice .....	1
2.1.	Prezentarea mediului Matlab SIMULINK .....	1
2.2.	Posibilități de construcție și analiză a modelelor.....	3
3.	Desfășurarea lucrării .....	23
4.	Conținutul referatului.....	29
2.	SIMATIC Step 7 .....	30
1.	Scopul lucrării .....	30
2.	Considerații teoretice .....	30
2.1.	Automate programabile SIMATIC S7-300 .....	30
2.2.	Mediul de programare Step 7 .....	43
2.3.	Crearea unei aplicații Step 7 .....	44
2.4.	Instrumente în STEP 7 .....	46
2.5.	Crearea și editarea unui proiect .....	49
3.	Desfășurarea lucrării .....	67
3.1.	Releu cu automenținere .....	67
3.2.	Citirea și afișarea unei mărimi analogice.....	72
3.3.	Compararea a două tensiuni.....	74
3.4.	Regulator P .....	75
4.	Conținutul referatului .....	75
3.	Reglarea nivelului .....	76
1.	Scopul lucrării .....	76
2.	Considerații teoretice .....	76
2.1.	Descriere FESTO Compact Workstation.....	76
2.2.	Componentele stației FESTO .....	78
2.3.	Configurarea sistemului.....	84

---

## Cuprins

---

2.4. Conexiuni .....	84
3. Desfășurarea lucrării .....	85
3.1. Configurarea PLC .....	86
3.2. Control continuu FB41 „CONT_C” .....	86
3.3. Programarea PLC .....	90
3.4. Identificarea experimentală a procesului .....	92
3.5. Determinarea parametrilor modelului procesului și acordarea regulatorului .....	95
4. Conținutul referatului .....	99
4. Reglarea temperaturii unui letcon .....	100
1. Scopul lucrării .....	100
2. Considerații teoretice .....	100
2.1. Sisteme de reglare a temperaturii.....	100
2.2. Descrierea elementelor componente ale părții fixate.....	102
2.3. Descrierea standului.....	108
3. Desfășurarea lucrării .....	108
3.1. Configurarea PLC .....	108
3.2. Control continuu al temperaturii FB58 „TCONT_CP” .....	109
3.3. Programarea PLC .....	113
3.4. Identificarea experimentală a procesului și implementarea regulatorului acordat .....	114
3.5. Crearea unui proiect WinCC Flexible .....	114
3.6. Achiziția datelor .....	120
4. Conținutul referatului .....	121
5. Reglarea temperaturii într-un tub .....	122
1. Scopul lucrării .....	122
2. Considerații teoretice .....	122
2.1. Descrierea standului didactic .....	122
2.2. Element de execuție .....	125
2.3. Ventilator .....	128

---

## Cuprins

2.4. Senzorul LM35 .....	128
3. Desfășurarea lucrării .....	130
3.1. Configurarea PLC.....	130
3.2. Control continuu al temperaturii FB58 „TCONT_CP” .....	130
3.3. Programarea PLC .....	131
3.4. Identificarea experimentală a procesului .....	132
4. Conținutul referatului .....	133
6. Reglarea turației motorului de curent continuu.....	134
1. Scopul lucrării .....	134
2. Considerații teoretice .....	134
2.1. Motorul de curent continuu .....	134
2.2. Modelarea procesului.....	135
2.3. Descrierea standului.....	137
2.4. Conexiuni.....	139
3. Desfășurarea lucrării .....	139
4. Conținutul referatului .....	141
7. Anexa 1. Descriere dispozitive stand FESTO .....	142
8. Anexa 2. Stand cu PLC S7-300 .....	148
9. Anexa 3. Achiziție de date .....	153
10. Bibliografie .....	156



# 1. Simularea sistemelor automate



## 1. Scopul lucrării

Se prezintă etapele semnificative pentru simularea sistemelor dinamice în SIMULINK, posibilitățile de manipulare a blocurilor și de construcție a modelelor, modul de utilizare a instrumentelor de analiză oferite de SIMULINK. Sunt expuse detalii privind construcția și simularea unor modele simple de sisteme de reglare automată, sunt interpretate răspunsuri indiciale în raport cu comportarea regulatorului și în raport cu parametrii și structura părții fixate.



## 2. Considerații teoretice

### 2.1. Prezentarea mediului Matlab SIMULINK

SIMULINK este un program pentru simularea sistemelor dinamice. Ca și extensie a programului MATLAB, SIMULINK adaugă multe trăsături specifice sistemelor dinamice păstrând toate funcționalitățile generale ale MATLAB-ului.

După instalarea programului SIMULINK directorul MATLAB va conține următoarele subdirectoare adăugate, de regulă, în subdirectorul TOOLBOX\SIMULINK:

*SIMULINK* - grupul de instrumente sub forma fișierelor cu extensia *M*;

*BLOKS* - grupul de instrumente bloc, blocuri folosind funcții *S*, exemple de mascare, exemple și modele de funcții *S*;

*SIMDEMOS* - demonstrații sub forma unor fișiere cu extensia *M*.

SIMULINK are două etape succesive de utilizare: *definirea sau construcția modelului și analiza acestuia*. O sesiune tipică de lucru începe prin defininirea unui model sau încărcarea unuia definit anterior, după care se poate trece la analiza acestuia. În practică acești doi pași sunt adesea execuți iterativ pe măsură ce proiectantul definește și modifică modelul pentru a obține comportamentul dorit.

Pentru a ușura definirea modelului, SIMULINK adaugă o nouă clasă de ferestre denumite *scheme bloc*. În aceste ferestre modelele sunt create și modificate utilizând aproape în totalitate mouse-ul. Performanțele obținute în utilizarea SIMULINK-ului depind în parte de familiarizarea cu metodele de manipulare a componentelor din aceste ferestre.

După ce se definește un model, acesta poate fi analizat fie prin folosirea opțiunilor oferite de meniurile SIMULINK-ului, fie prin introducerea de comenzi în fereastra MATLAB-ului. Instrumentele oferite de program cuprind câteva variante de algoritmi de simulare, *linmod*, un instrument pentru extragerea modelului liniar al sistemelor, și *trim* un instrument pentru găsirea punctelor de echilibru.

O simulare poate fi urmărită în timp ce se desfășoară, la sfârșitul acesteia rezultatele finale fiind disponibile în spațiul de lucru al MATLAB-ului.

Pentru simularea sistemelor dinamice în SIMULINK este necesară parcurgerea unor etape prezentate pe scurt în continuare:

1. Se introduce comanda *simulink* la prompterul MATLAB pentru a deschide biblioteca bloc principală;
2. Se selectează *New* din meniul *File* pentru a deschide o fereastră goală, în care se poate construi un model de sistem. Noua fereastră este denumită *Untitled* și va putea fi redenumită când se salvează ca fișier;
3. Se deschid una sau mai multe biblioteci și cu ajutorul mouse-ului se trag blocurile necesare în fereastra activă;
4. După ce blocurile au fost poziționate se vor trage linii pentru a le conecta poziționând cursorul mouse-ului pe portul unui bloc și ținând apăsat butonul din stânga al acestuia;
5. Fiecare bloc poate fi deschis prin dublu clic pentru a modifica parametrii interni. Ca parametri pot fi introduse orice expresii permise în MATLAB;
6. Se salvează modelul sistemului selectând opțiunea *Save* din meniul *File*;
7. Simularea poate fi declanșată selectând opțiunea *Start* din meniul *Simulation*. În timpul simulării opțiunea *Start* se transformă în *Stop*. Dacă se selectează *Stop* meniul va afișa din nou *Start* ;

8. Parametrii simulării pot fi ajustați selectând opțiunea *Parameters* din meniul *Simulation*, ceea ce va avea ca rezultat afișarea unei căsuțe de dialog numită *Simulation Parameters* care permite modificarea lor;
9. Se poate monitoriza comportarea sistemului modelat folosind un bloc *Scope* (osciloscop), un bloc *Grafic* (reprezentare grafică) sau un bloc *To Workspace* (spre spațiul de lucru) pentru a trimite date în spațiul de lucru MATLAB, unde se pot aplica ulterior funcții specifice acestuia asupra rezultatelor simulării.

Fereastrye SIMULINK au fost dimensionate în aşa fel încât să corespundă celor mai des întâlnite rezoluții. Totuși, dacă monitorul are o rezoluție excepțional de mare sau de mică, dimensiunea ferestrelor s-ar putea să fie nepotrivită.

Când se deschide o bibliotecă bloc sau se rulează un program demonstrativ pentru prima dată, dacă dimensiunea implicită nu este corectă, se poate redimensiona fereastra trăgând cu mouse-ul de margini. Pot fi apoi salvate noile dimensiuni selectând opțiunea *Save* din meniul *File*. Data viitoare când se deschide biblioteca bloc sau se rulează programul demonstrativ, fereastra va avea noua dimensiune.

## 2.2.Pozibilități de construcție și analiză a modelelor

Pentru a stăpâni SIMULINK este necesar să se cunoască posibilitățile de manipulare a blocurilor și de construcție a noi modele. Este necesară de asemenea o familiarizare cu tipurile de blocuri disponibile. În sfârșit, trebuie învățat cum se utilizează instrumentele de analiză oferite de SIMULINK . Construcția și simularea unui model simplu, descrise în următorul capitol, poate constitui o introducere în aceste concepte.

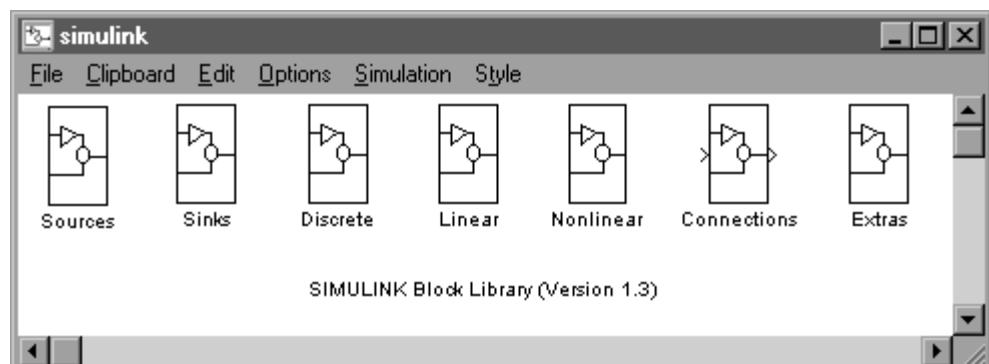
## Lucrarea 1

### *Copierea blocurilor în fereastra de lucru*

SIMULINK folosește reprezentarea sistemelor prin diagrame bloc. Definirea unui sistem este asemănătoare desenării unei diagrame bloc. Nu se desenează însă fiecare bloc individual, ci acestea se pot copia din bibliotecile standard de blocuri furnizate de SIMULINK sau din bibliotecile construite de utilizator. O bibliotecă nu este altceva decât un model care include anumite blocuri și subsisteme.

Biblioteca standard de blocuri este organizată în câteva subsisteme, grupând blocurile în funcție de comportarea lor. Blocurile pot fi copiate în schema bloc (fereastra de lucru) din aceste subsisteme sau din orice alte biblioteci sau modele. Plasând blocurile care se folosesc cel mai des într-un astfel de subsistem și setând preferințele personale ca valori implicate, pot fi construite biblioteci personale de blocuri. Acestea vor fi de un real folos în construcția viitoarelor modele.

Biblioteca de blocuri a SIMULINK-ului se deschide tastând comanda: *simulink*. Această comandă va afișa o fereastră conținând iconurile subsistemelor de blocuri care alcătuiesc biblioteca standard (figura 1.1). Aceste subsisteme pot fi deschise (prin dublu clic) pentru a afișa blocurile pe care le conțin și care pot fi copiate în schema bloc.

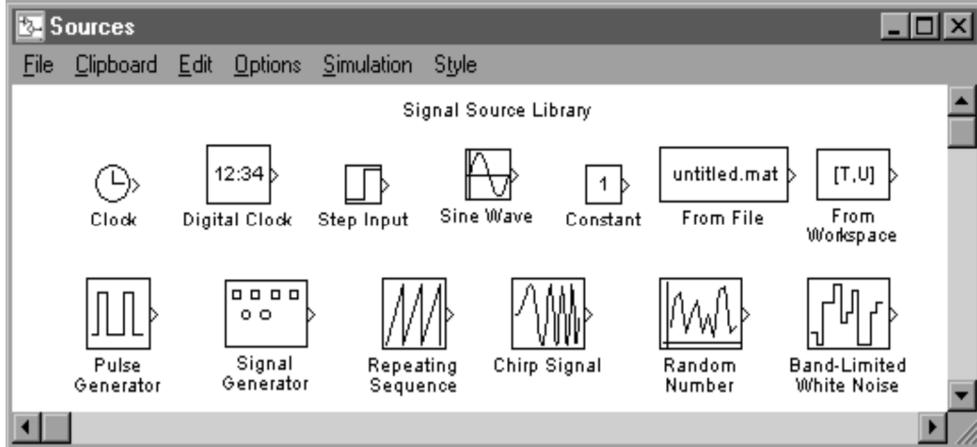


**Figura 1.1.** Biblioteca de blocuri

Biblioteca *Sources*, de exemplu, conține blocuri capabile să genereze semnale. Se poate deschide subsistemul *Sources* executând un dublu clic pe

## Lucrarea 1

iconul *Sources*. Acesta va afișa o nouă fereastră conținând toate blocurile care produc semnale de ieșire dar nu au intrări (figura 1.2).



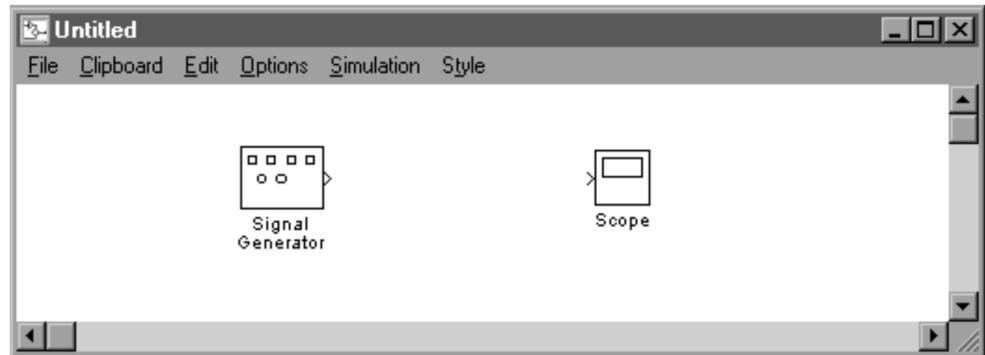
**Figura 1. 2.** Semnale de intrare

Selectând opțiunea *New* din meniul *File* se va deschide o fereastră goală în care se poate începe construcția modelului. Fereastra se va deschide într-o poziție prestabilită, dar poate fi mutată într-o poziție mai convenabilă, mărită sau micșorată.

Blocurile pot fi copiate dintr-o poziție în alta trăgându-le cu mouse-ul din poziția originală până în noua fereastră. Tragând spre exemplu blocul *Signal Generator* din fereastra *Sources* în nouă, încă nenumărată fereastră, se poate observa că este deplasată de fapt doar o copie a sa; există în continuare un bloc *Signal Generator* în fereastra *Sources*.

Pentru a observa evoluția semnalului în timpul simulării poate fi folosit blocul *Scope*. Se deschide biblioteca *Sinks* și se trage o copie a blocului *Scope* în fereastra de lucru, în care se vor găsi acum cele două blocuri (figura 1.3).

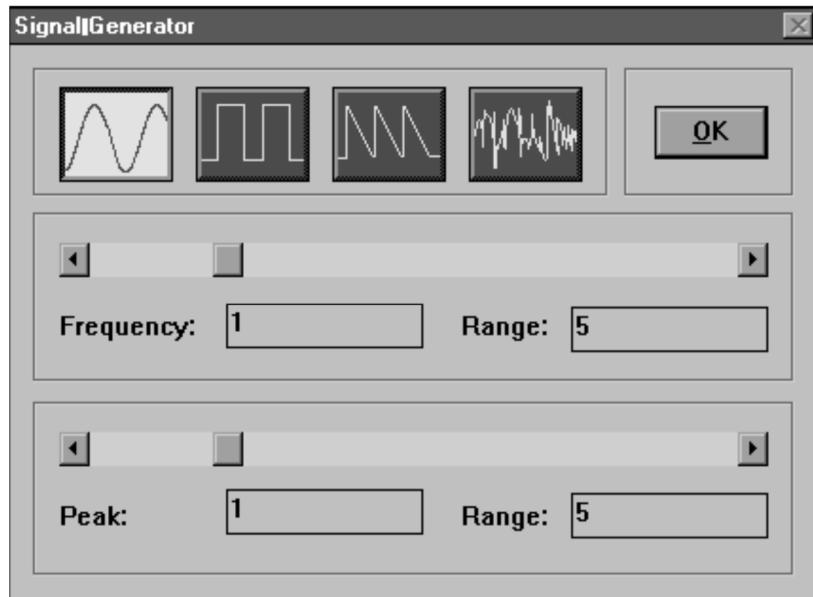
## Lucrarea 1



**Figura 1.3.** Fereastra de lucru

Blocurile copiate au parametrii identici cu cei ai blocurilor originale. Majoritatea blocurilor își expun parametrii în ferestre separate. Aceste ferestre permit controlarea comportamentului blocului prin modificarea valorii parametrilor săi.

Se poate deschide blocul *Signal Generator* executând un dublu clic pe iconul său. Această acțiune va avea ca rezultat afișarea unei ferestre de dialog conținând parametrii de control ai formei de undă, amplitudinii și frecvenței generate de bloc (figura 1.4).



**Figura 1.4.** Generator semnal

Pentru a fixa valoarea frecvenței semnalului generat se selectează căsuța text numită *Frequency* poziționând cursorul pe ea și se introduce valoarea dorită. Dacă se execută un dublu clic întreaga căsuță va fi evidențiată și ceea ce se tastează va înlocui valorile anterioare. În aceste cazuri, de introducere explicită a valorii frecvenței, variabila *Range* nu este luată în considerare. Valoarea frecvenței poate fi stabilită între 0 și valoarea introdusă pentru variabila *Range* prin poziționarea corespunzătoare a cursorului baretei situată deasupra căsuței *Frequency*. Amplitudinea semnalului generat se stabilește asemănător cu cea pentru frecvență, acționând în căsuța *Peak*. Pentru a închide această fereastră se execută un clic pe butonul *OK* sau se tastează *Enter*.

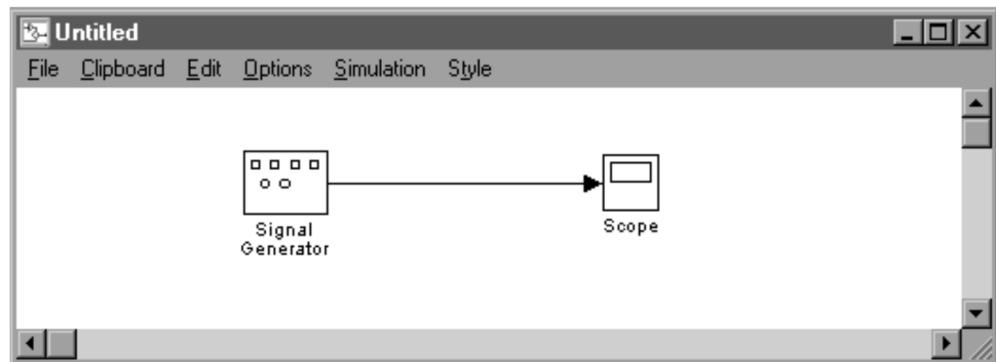
La deschiderea blocului *Scope* se obține o fereastră care conține o reprezentare grafică a unui osciloscop. Se poate alege această fereastră într-o poziție convenabilă. Folosind cursorul pot fi selectate rubricile corespunzătoare pentru a schimba limitele implicate ale axelor orizontală și verticală introducând noile valori.

### *Conecțarea blocurilor*

Semnul ( $>$ ) orientat spre exteriorul iconului blocului *Signal Generator* reprezintă portul său de ieșire, iar semnul ( $<$ ) orientat spre interiorul iconului blocului *Scope* reprezintă portul său de intrare. Pentru a conecta aceste două blocuri se apăsa butonul din stânga al mouse-ului când cursorul se află deasupra unui port și se trage o linie de conectare între cele două porturi, după care se eliberează butonul.

Când blocurile sunt conectate corect semnele  $>$ ,  $<$  dispar fiind înlocuite de o săgeată care indică sensul deplasării datelor (figura 1.5).

Pe măsură ce se desenează linia ea poate sări în unghiuri care sunt multipli de 45° și capătul liniei nu va rămâne neapărat sub cursor. Linia poate fi desenată sub orice unghi înținând apăsat ambele butoane ale mouse-ului (stâng și drept în același timp) în timp ce se trage cursorul mișcând mouse-ul.



**Figura 1.5.** Fereastra de lucru

Capătul liniei sare de asemenea pentru a se conecta la orice port de tipul corect ( intrare sau ieșire ), dacă este tras cursorul destul de aproape de acesta sau deasupra iconului blocului. Dacă butoanele mouse-ului sunt eliberate înainte ca blocurile să fie conectate linia se va termina cu un semn indicând sensul deplasării datelor ( < sau > ). Pentru a adăuga un alt segment se execută un clic pe capătul liniei și se trage în orice direcție; linia va respecta regulile descrise mai sus în ceea ce privește unghiurile.

În felul acesta pot fi trasate linii de conectare ocolind alte blocuri. Nu există limite privind numărul sau lungimea liniilor. Pentru a realiza legătura între generatorul de semnal și osciloscop se procedează conform indicațiilor de mai sus. Legătura va determina blocul *Scope* să afișeze semnalul generat de blocul *Signal Generator*. În felul acesta se poate testa dacă setările parametrilor fiecărui bloc în parte au fost corecte.

În cazul în care se consideră că o linie este trasată incorect, linia respectivă se poate șterge sau edita. Se selectează linia executând un clic oriunde pe ea. Toate colțurile liniei vor fi evidențiate prin mici dreptunghiuri negre numite *mânere*. Odată linia selectată se apasă tasta *Delete* sau se selectează opțiunea *Cut* sau *Clear* din meniul *Edit*, pentru a șterge linia din model.

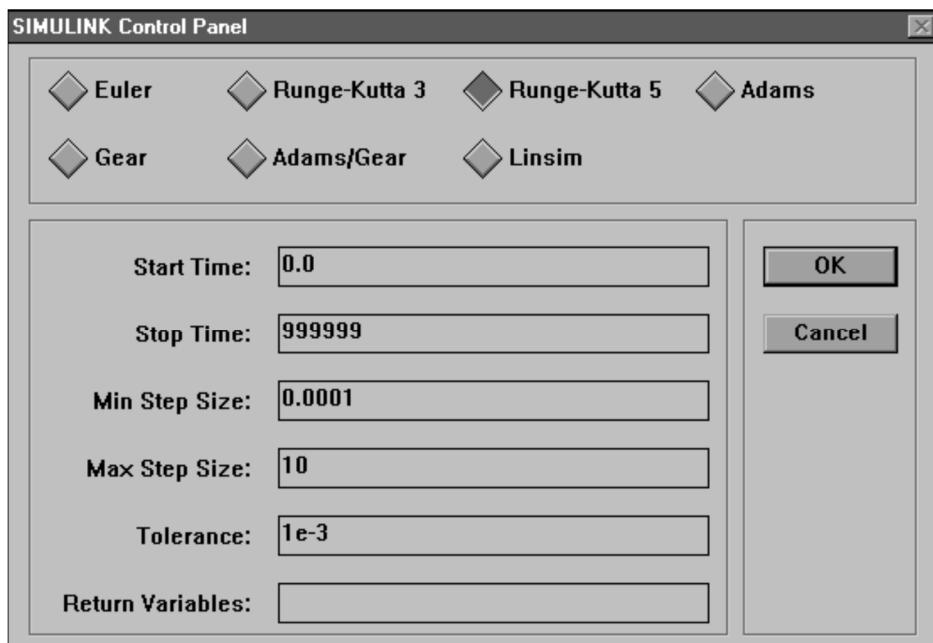
### Simularea sistemului

Când modelul este finalizat, se poate începe simularea. Pentru a alege metoda de integrare și parametrii care vor fi folosiți în timpul simulării, se selectează opțiunea *Parameters* din meniul *Simulation*.

Ca rezultat se va deschide o fereastră, numită *Control Panel*, care afișează toți parametrii simulării care pot fi modificați (figura 1.6).

În fereastră sunt afișați parametrii implicați. Aceștia pot fi schimbați acum pentru a putea îmbunătăți simularea. Selectând căsuța text numită *Maximum Step Size*, se poate schimba valoarea maximă a pasului de integrare, după care se execută un clic pe butonul *OK* pentru a închide fereastra de dialog. Simularea se decanșează selectând opțiunea *Start* din meniul *Simulation*.

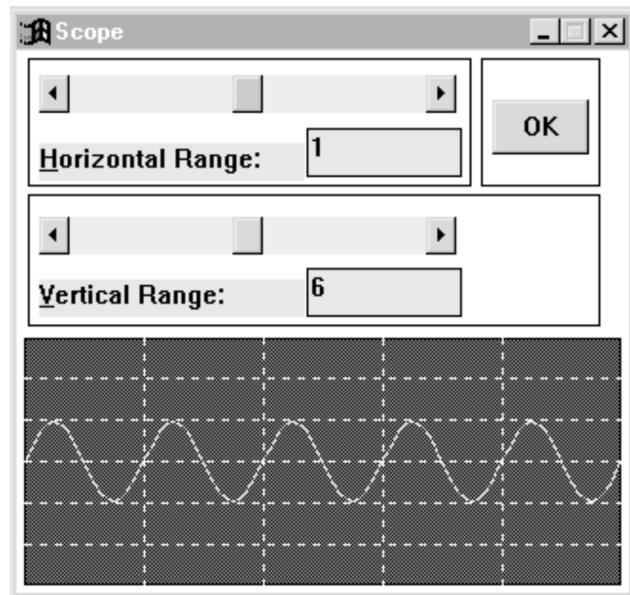
Dacă fereastra blocului *Scope* nu este deschisă, se va executa un dublu clic pe bloc; el se va deschide chiar dacă simularea a început deja. Blocul *Signal Generator* generează funcția selectată pentru fiecare pas al timpului iar blocul *Scope* va afișa graficul acestei funcții.



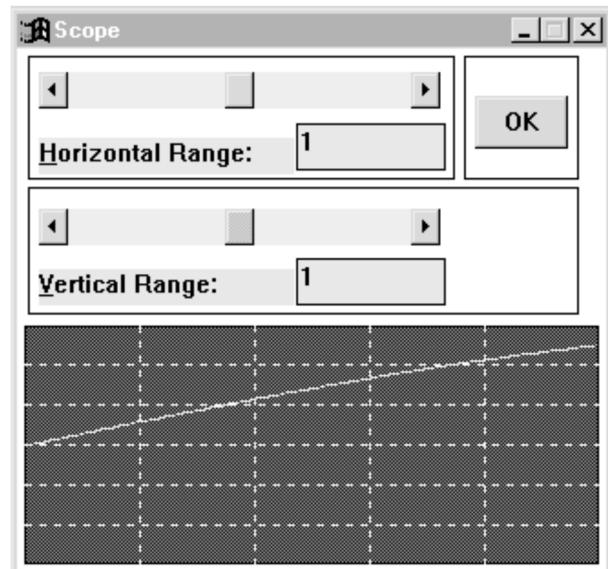
**Figura 1.6.** Control Panel

## Lucrarea 1

Cu valorile implicate ale parametrilor celor două blocuri, care nu sunt corelate, graficul afișat este cel din figura 1.7.



**Figura 1.7.** Scope



**Figura 1.8.** Scope

Pentru a obține graficul din figura 1.8., se va introduce pentru frecvență generatorului valoarea  $2 \cdot 3.14 \cdot 5 = 31.4$ , iar pentru amplitudine valoarea 2 (amplitudinea vârf la vârf a semnalului va fi 2V). Trebuie remarcat aici că valoarea introdusă pentru frecvență reprezintă de fapt valoarea pulsației ( $\omega = 2\pi f = 31.4$  corespunzătoare unei frecvențe  $f = 5$  Hz). Parametrii osciloscopului vor fi setați la valorile: *Horizontal Range* = 1 (capătul de scală pe axa timpului 1 sec.) și *Vertical Range* = 6 (1 V / diviziune).

Parametrii de control ai blocurilor *Scope* și *Signal Generator* sunt accesibili în orice moment al simulării. Se poate selecta o formă de undă diferită în blocul *Signal Generator* și se va observa imediat în fereastra *Scope*.

Schimbând și valorile altor parametrii ai blocurilor *Scope* și *Signal Generator* se poate observa modul în care se modifică graficul afișat. Simularea se va opri când valoarea parametrului *Stop Time* din fereastra *Control Panel* este atinsă. Simularea poate fi oprită oricând selectând opțiunea *Stop* din meniu *Simulation*.

### *Salvarea modelului*

În acest moment există în fereastra de lucru un model funcțional care poate fi salvat pe disc ca fișier MATLAB cu extensia *M*. Un fișier de acest tip conține toate comenzi necesare pentru a descrie modelul astfel încât acesta să poată fi simulaț sau modificat ulterior. Se alege opțiunea *Save* sau *Save As* din meniu *File* pentru a afișa o fereastră de dialog în care se poate introduce numele și directorul în care modelul curent va fi salvat.

Se introduce numele modelului, de exemplu *model1* și se acționează butonul *OK* pentru a salva modelul în directorul curent. Extensia *M* va fi adăugată implicit chiar dacă nu este introdusă de la tastatură. Se poate schimba directorul în care va fi salvat modelul introducând calea (directorul) în fața numelui fișierului sau executând un clic pe numele directorului dorit și apoi acționând butonul *OK*. Pentru a selecta un fișier existent, se execută un dublu clic pe numele respectivului fișier sau un clic pe numele fișierului și apoi un clic pe butonul *OK*. Vor fi vizibile și numele altor fișiere mutând bara de derulare în sus sau în jos.

După ce se salvează un model într-un fișier, numele ferestrei în care este afișat modelul va fi identic cu numele fișierului respectiv. Fereastra modelului se poate închide selectând opțiunea *Close* din meniul *File*.

Acum există deja un model salvat pe disc și poate fi deschis acum introducând la prompterul MATLAB comanda :

- *model1*

Modelul va fi afișat într-o fereastră *SIMULINK* în aceeași stare în care era când a fost salvat. De remarcat că în același fel am deschis biblioteca de blocuri standard la începutul sesiunii de lucru anterioare, prin execuția din *MATLAB* a comenzi:

- *simulink*

### *Manipularea blocurilor în fereastra de lucru*

Blocurile pot fi mutate oriunde în fereastră. Chiar dacă un bloc este mutat conexiunile acestuia se păstrează. Dacă sunt mutate mai multe blocuri odată își vor păstra poziția relativă, liniile de conectare mutându-se împreună cu ele. Blocurile *Scope*, *Signal Generator* și linia de conectare dintre ele se pot selecta apăsând butonul din stânga al mouse-ului când cursorul se află deasupra și în stânga obiectelor și tragându-l până într-o poziție aflată sub și în dreapta obiectelor.

Vor apărea mai multe mânere în colțurile iconurilor blocurilor selectate. Tragând unul dintre blocurile selectate se va observa că tot grupul se mișcă simultan cu obiectul selectat.

Pentru a face loc și altor blocuri se mută blocurile în partea de sus a ferestrei. Se aduce apoi o copie a blocului *Gain* din subsistemul *Linear* aparținând bibliotecii de blocuri standard.

Se deschide acest bloc executând un dublu clic pe iconul său. În fereastra de dialog afișată la deschiderea blocului se va schimba valoarea parametrului *Gain* la 2 și apoi se închide fereastra. Blocul *Gain* este un bloc care-și afișează valoarea parametrului intern ; valoarea introdusă ca parametru al blocului *Gain*, în cazul nostru 2, este afișată în interiorul iconului blocului.

Este necesară acum conectarea blocului *Signal Generator* la blocul *Gain*. Nici un port de ieșire nu este vizibil pe blocul *Signal Generator*, din moment ce acesta este deja conectat la un bloc, aşa că se va începe noua linie din

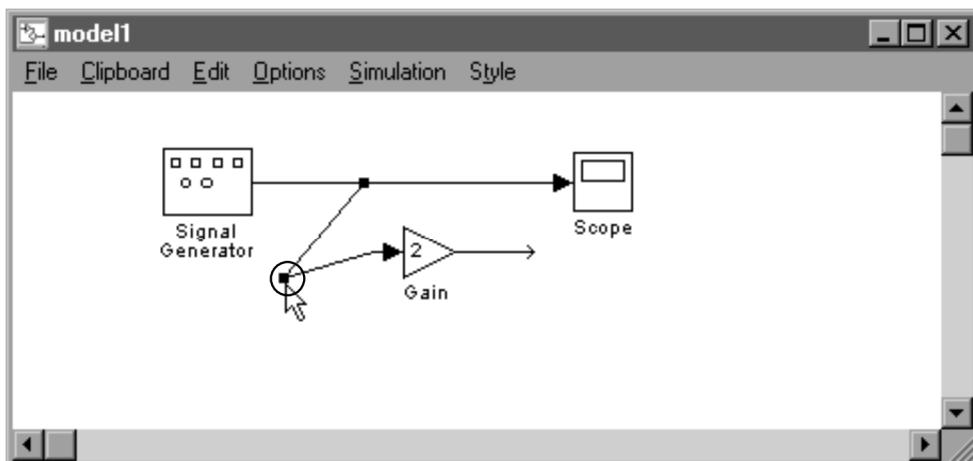
## Lucrarea 1

punctul în care este conectată linia deja existentă. Se trage cursorul spre blocul *Gain* conectând linia de acesta. Linia poate fi desenată direct spre portul de intrare al blocului *Gain* sau poate fi desenată din segmente la diferite unghiuri, de preferință la unghiuri drepte.

Pentru a muta punctul de întâlnire a două segmente de linie, se execută un clic pe punctul respectiv și se trage cursorul. Cele două segmente care se întâlnesc în acel punct sunt întinse pe măsură ce cursorul se mișcă ; când butonul mouse-ului este eliberat liniile vor rămâne în poziția la care se află în acel moment.

Se poate de asemenea muta un segment de linie executând un clic în mijlocul său și trăgându-l în noua poziție. Cele două segmente conectate la capetele segmentului mutat se vor întinde pentru a-și menține conexiunile. Panta și lungimea segmentului mutat rămân aceleași.

Pentru a introduce un cot suplimentar într-un segment de dreaptă se țin apăsat ambele butoane ale mouse-ului trăgând cursorul. Segmentul se va frângă, punctul de ruptură urmărind cursorul până când butoanele vor fi eliberate (figura 1.9).



**Figura 1.9.** Fereastra de lucru

### Vizualizarea rezultatelor

Odată legat de blocul *Signal Generator*, blocul *Gain* va primi în timpul simulării același semnal ca și blocul *Scope*.

Pentru a adăuga un alt bloc *Scope* în model, nu este necesar să fie copiat din biblioteca de blocuri. Se poate face o copie a blocului deja existent în sistem ținând apăsat butonul din dreapta și trăgând blocul într-o altă poziție aflată în aceeași fereastră. Noua copie a blocului *Scope* va avea aceleași valori ale parametrilor ca și originalul. SIMULINK-ul adaugă automat un număr numelui blocului (noul bloc se va numi *Scope 1*) atunci când acesta va fi copiat în aceeași fereastră. Acest lucru se întâmplă deoarece numele blocurilor trebuie să fie unice în interiorul unei ferestre. Se poate, desigur, redenumi un bloc selectând numele acestuia și introducând un nume nou.

Se conectează blocul *Gain* cu noul bloc *Scope*. Se deschid ambele blocuri *Scope* și se startează simularea. Blocul *Gain* amplifică semnalul primit de la blocul *Signal Generator*, în al doilea bloc *Scope* fiind afișată deci o sinusoidă cu amplitudine dublă.

În cazul în care trebuie îndepărtat un bloc sau o linie dintr-un model se selectează obiectul respectiv și apăsând tasta *Delete* sau selectând opțiunile *Cut* sau *Clear* din meniul *Edit*, obiectul este șters.

Se selectează al doilea bloc *Scope*; după aceea ținând apăsată tasta *Shift*, se selectează linia care îl conectează cu blocul *Gain* și linia care conectează blocul *Signal Generator* și primul bloc *Scope*. Se poate face aceasta executând câte un clic pe fiecare obiect folosind butonul din stânga al mouse-ului. Se îndepărtează obiectele din model selectând opțiunea *Cut* din meniul *Edit*. Se trage în fereastra modelului un bloc *Mux* din biblioteca de blocuri *Connections*.

Se deschide blocul *Mux* și se schimbă valoarea parametrului *Number of Inputs* la 2, după care se închide fereastra de dialog. Ca rezultat blocul *Mux* va avea acum două porturi de intrare. Se conectează ieșirile blocurilor *Signal Generator* și *Gain* la câte unul din aceste porturi de intrare și portul de ieșire al blocului *Mux* la portul de intrare al blocului *Scope*.

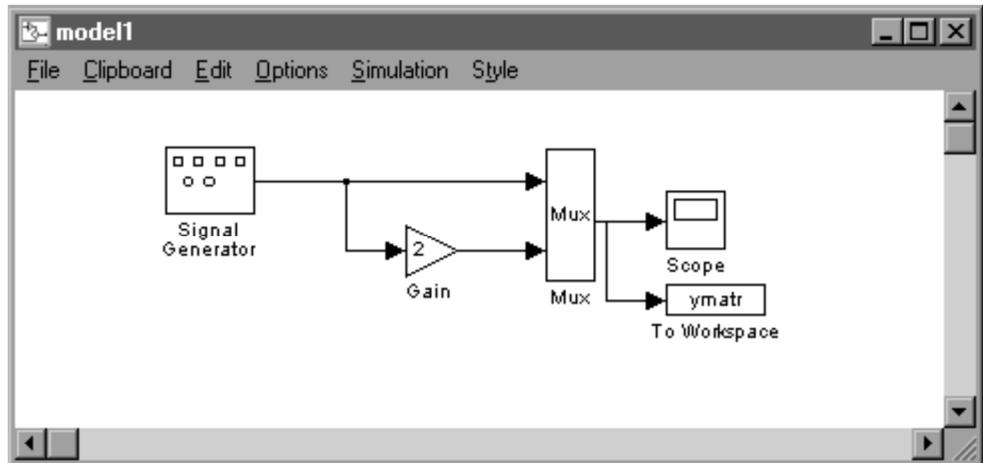
Blocul *Mux* își va multiplexa cele două intrări într-un singur vector de ieșire. Linia de conectare dintre blocurile *Scope* și *Mux* este o linie *multivariabilă* care transmite mai multe valori odată.

Se trage acum o copie a blocului *To Workspace* din biblioteca *Sinks* în fereastra modelului. Valoarea parametrului *Variable Name* este afişată în interiorul iconului blocului ; valoarea implicită este *yout*. Se poate schimba această valoare introducând un nou nume în căsuța text *Variable name* situată în fereastra de dialog a blocului. Blocul *To Workspace* ia un vector de orice lungime la intrare și îl transmite zonei de lucru MATLAB sub forma unei matrici. Matricea va avea câte o linie pentru fiecare pas al simulării, numărul maxim de linii fiind specificat prin valoarea unuia din parametrii blocului, și câte o coloană pentru fiecare element al vectorului de intrare.

Pentru a schimba numele variabilei MATLAB produse de simulare în *ymatr* se deschide blocul *To Workspace* și se introduce noul nume în căsuța text *Variable name* (valoarea implicită maximă a numărului de linii, 1000, este suficientă pentru acest exemplu). Se închide fereastra de dialog și se va vedea noul nume afișat în iconul blocului (figura 1.10).

O altă modalitate de a conecta ieșirea unui bloc la intrarea unuia sau mai multor altor blocuri este de a folosi o linie deja existentă. Se poziționează cursorul pe linia care leagă blocurile *Mux* și *Scope*, se apasă butonul din dreapta al mouse-ului trăgând cursorul spre blocul *To Workspace*. Se poate observa că SIMULINK-ul pune un mic cerculeț la intersecția celor două linii pentru a arăta că ieșirea blocului *Mux* va fi transmisă și prin această linie. Se conectează linia la blocul *To Workspace*.

Se startează simularea modelului. Se poate observa că blocul *Scope* va afișa două grafice suprapuse datorită faptului că blocul său de intrare are două elemente.



**Figura 1.10.** Fereastra de lucru

La sfârșitul simulării se salvează modelul selectând opțiunea *Save* din meniul *File*; versiunea actuală a modelului va fi salvată peste versiunea rămasă de la prima construcție. Pentru a salva modelul curent sub un alt nume se selectează opțiunea *Save As* din meniul *File* și se introduce noul nume.

Dacă se selectează fereastra de lucru MATLAB și se introduc comanda *whos*, se va afișa:

Name	Size	Elements	Bytes	Density	Complex
ymatr	1000 by 2	2000	16000	Full	No

Grand total is 2000 elements using 16000 bytes.

Numele variabilei *ymatr* poate apărea afișat împreună cu alte variabile MATLAB.

Acest exercițiu are rolul de a demonstra că ieșirea oricărui bloc poate fi analizată mai departe folosind întreaga putere a MATLAB-ului. Dacă se introduce comanda *plot(ymatr)* aceasta va avea ca efect afișarea datelor achiziționate în timpul simulării. Graficul va arăta ca cel afișat de blocul *Scope*. Rezultatele simulării pot fi salvate într-un fișier MATLAB folosind comanda *save*. Un vector de valori în timp poate fi obținut conectând un bloc *Clock* la blocul *To Workspace* sau introducând un nume de variabilă (ex. *t*) în câmpul *Return Variables* din fereastra *Simulation Parameters*. Acest vector este necesar deoarece, cu excepția situației în care pasul minim în timp este setat egal cu pasul maxim, simularea se desfășoară în pași inegali.

### *Posibilități de analiză a modelelor*

S-au folosit până acum uneltele de bază oferite de SIMULINK pentru construcția modelelor. Acest capitol prezintă instrumentele ce permit analiza modelelor SIMULINK. Instrumentele conțin funcții pentru simulare, liniarizare și determinarea punctului de echilibru (optimizare).

#### *Controlul simulării*

Există trei moduri diferite de a utiliza SIMULINK. Cel mai interactiv mod presupune controlul simulării cu ajutorul meniurilor și urmărirea rezultatelor folosind blocuri *Scope* sau *Graph*. Această metodă e ușor de folosit și de învățat și poate furniza rapid rezultate. Acest mod de utilizare este deosebit de eficient atunci când se crează și verifică un sistem pentru prima dată.

Al doilea mod presupune folosirea modelelor predefinite furnizate de SIMULINK și lansarea funcțiilor de analiză din linia de comandă. Această metodă nu este la fel de interactivă ca prima dar oferă o flexibilitate mai mare. Transmiterea rezultatelor în zona de lucru MATLAB permite analiza și vizualizarea ulterioară a datelor folosind funcții grafice specifice acestuia.

Cel mai complex și flexibil mod de utilizare necesită accesul direct al *funcțiilor S* ale modelului. Orice model SIMULINK este disponibil în MATLAB sub forma unei *funcții S* care conține comportamentul dinamic al sistemului. Fiecare *funcție S* are un nume identic cu cel al modelului corespunzător. O *funcție S* poate fi apelată în mai multe moduri pentru a furniza informații despre modelul pe care îl reprezintă. Se pot obține informații despre numărul de intrări, ieșiri și stări ale modelului (continue sau discrete). Se pot obține deasemenea ieșirile și derivatele acestora. Toate instrumentele de analiză furnizate de SIMULINK interacționează cu modelele prin intermediul *funcțiilor S*. Fișierele de tip *M*, *linmod* și *trim*, sunt exemple care folosesc aceste funcții. (Pentru mai multe informații despre funcțiile S se poate consulta *Gidul utilizator SIMULINK*.)

Cele trei moduri de utilizare nu se exclud reciproc și nu se pot delimita exact. Ele pot fi folosite succesiv sau simultan de-a lungul diferitelor etape de dezvoltare a modelului.

Simularea modelelor SIMULINK implică integrarea numerică a unor sisteme de ecuații diferențiale. SIMULINK furnizează o serie de algoritmi de integrare pentru simularea unor astfel de ecuații. Din păcate, datorită diversității comportamentelor dinamice ale unui sistem, nici una dintre aceste metode nu garantează simularea eficientă și exactă a oricărui tip de model. Alegerea potrivită a metodei de integrare și setarea cu atenție a parametrilor simulării, sunt condiții necesare pentru obținerea unor rezultate exakte.

O simulare poate fi pornită fie din linia de comandă, fie din meniu *Simulation*.

Dacă se pornește simularea selectând opțiunea *Start* din meniu *Simulation* este necesară setarea parametrilor simulării în fereastra de dialog *Control Panel* care apare selectând opțiunea *Parameters* din meniu *Simulation*.

Fereastra *Control Panel* are câmpuri în care se pot introduce numere sau orice expresii MATLAB corecte, de exemplu, variabilele: *tfinal*, *pasmin*, *pasmax* și *err* introduse în ordinea câmpurilor, și care sunt definite în spațiul de lucru MATLAB.

Variabilele  $[t, x, y]$ , introduse la rubrica *Return Variables*, sunt utilizate pentru transmiterea timpului (întotdeauna), stării (dacă există) și traекторiilor de ieșire (dacă sunt utilizate porturi de ieșire - biblioteca *Sinks*) în spațiul de lucru MATLAB. Timpii de start și stop sunt setați în variabilele *tstart* și respectiv *tfinal*. Parametrii de integrare *err*, *pasmin* și *pasmax*, controlează eroarea relativă locală, lățimea minimă și respectiv lățimea maximă a pasului de timp al simulării.

Pornirea unei simulări din meniu permite executarea anumitor operații interactiv în timpul simulării :

- Schimbarea parametrilor unui bloc, cu condiția ca aceasta să nu modifice numărul de stări, intrări sau ieșiri ale blocului respectiv;
- Schimbarea parametrilor simulării, cu excepția valorilor introduse la rubricile *Return Variables* și *Start Time*.
- Schimbarea algoritmului de integrare.
- Pentru blocurile discrete, schimbarea timpului de eșantionare.
- Simularea unui alt sistem în același timp.

- Executând un clic pe o linie de conectare se poate vedea ieșirea sa într-un bloc *Scope* flotant. Un bloc *Scope* neconectat se numește flotant. El va afișa ieșirea oricărei linii pe care se execută un clic în timpul simulării.

Adăugarea sau ștergerea liniilor sau blocurilor în timpul simulării produce oprirea acesteia. Selectarea opțiunii *Restart* din meniu *Simulation* va produce reluarea simulării și afișarea rezultatelor modificărilor.

Orice simulare pornită din meniu poate fi pornită și din linia de comandă. De exemplu, pentru a configura o simulare cu aceiași parametrii ca cei introdusi anterior în fereastra *Control Panel*, se poate folosi comanda:

$$[ t, x, y ] = \text{linsim} ( NmodelIN, [ tstart, tfinal ], x0, [ err, pasmin, pasmax ] );$$

unde *model1* este numele diagramei bloc a sistemului.

Comanda *linsim('system',tf)* integrează sistemul descris de modelul SIMULINK (sau o altă funcție de tip *S-function*) >*system*= de la 0 la TF (timpul final). Fără argumentele din membrul stâng rezultatele sunt afișate sub formă grafică. Dacă modelul *system* include porturi de ieșire, *linsim* va reprezenta grafic traectoriile ieșirii în raport cu timpul (altfel sunt reprezentate grafic traectoriile de stare). Utilizată cu argumentele din membrul stâng  $[ t, x, y ] = \text{linsim}('system',tf)$ , comanda *linsim* returnează, fără reprezentare grafică, vectorul *t* ce cuprinde succesiunea momentelor de integrare, traectoriile ieșirilor *y* și traectoriile de stare *x*.

O altă formă a comenzi  $[ t, x, y ] = \text{linsim}('system',tf,x0)$ , permite definirea condițiilor inițiale *x0*, altele decât cele definite în *system*.

$[ t, x, y ] = \text{linsim}('system',tf,x0, options)$  permite utilizare unor parametrii opționali definiți mai jos:

<i>options(1)</i> - eroarea relativă (default 1e-3).	
<i>options(2)</i> - pasul minim de integrare	(default tf/2000).
<i>options(3)</i> - pasul maxim de integrare	(default tf/50).
<i>options(4)</i> - neutilizat	(default 0).
<i>options(5)</i> - afișează parametri de alarmă	(default 0).

*options(6)* - validează reprezentarea (default - validare dacă nu sunt argumentele grafice a parametrilor din membrul stâng, alfel invalidare) (1 validare, 2 invalidare)

Condițiile inițiale, care nu pot fi setate în meniu, sunt definite în vectorul  $x0$ . Aceste condiții vor înlocui condițiile inițiale înscrise în blocuri cu excepția situației în care  $x0$  este o matrice goală.

Pornirea simulării din linia de comandă prezintă următoarele avantaje :

- Condițiile inițiale din blocuri pot fi suprascrise;
- Dacă nu se specifică nici un argument în partea stângă a expresiei, graficele ieșirilor vor fi automat afișate sau, când sistemul nu are ieșiri, se vor afișa traекторiile de stare;
- O simulare poate fi rulată dintr-un fișier de tip  $M$  ceea ce permite schimbarea iterativă a parametrilor blocurilor;
- Pentru modele mici, simularea rulează ceva mai repede.

Toți algoritmii de integrare au sintaxe de lansare identice să că metode diferite pot fi selectate schimbând doar numele funcției:

```
[ t, x, y ] = euler ( NsystemN , tfinal );  
[ t, x, y ] = rk23 ( NsystemN , tfinal );  
[ t, x, y ] = rk45 ( NsystemN , tfinal );  
[ t, x, y ] = adams ( NsystemN , tfinal );  
[ t, x, y ] = gear ( NsystemN , tfinal ).
```

Performanțele simulărilor, în termeni de viteză și precizie diferă în funcție de model și condiții inițiale. Sunt prezentate în continuare câteva recomandări privind modul de alegere a metodei de integrare:

*linsim* - metodă de integrare pentru modele dinamice liniare - se recomandă utilizarea ei în special pentru modele liniare. Aceste modele au de regulă în componența lor blocuri de tipul *Transfer Function*, *State-Space*, *Zero-Pole-Gain*, *Sum*, și *Gain*.

Metoda lucrează bine și dacă sistemul conține un număr mic de blocuri neliniare. Rezultatele sunt bune, comparate cu alte metode, dacă în componența modelului intră atât blocuri liniare cu dinamică rapidă cât și lentă (sisteme dificile).

*rk23, rk45* - metode Runge-Kutta de ordinul 3, respectiv de ordinul 5 - asigură rezultate mai bune decât celelalte metode în cazul sistemelor ce conțin neliniarități puternice și discontinuități. Asigură de asemenea bune performanțe în simularea sistemelor mixte, ce conțin atât blocuri continue cât și discrete. Aceste metode nu dau rezultate satisfăcătoare dacă în componența modelului intră atât blocuri cu dinamică rapidă cât și lentă. În acest caz se recomandă metodele *linsim* sau *gear*.

*gear* - se utilizează în simularea sistemelor neliniare fără discontinuități. Metoda este destinată simulării sistemelor ce conțin atât blocuri cu dinamică rapidă cât și lentă. Posibil să nu lucreze bine pentru sisteme cu singularități sau dacă sistemul este perturbat prin schimbări rapide ale intrărilor.

*adams* - se utilizează în simularea sistemelor neliniare fără discontinuități și fără constante de timp cu variații mari.

*euler* - se utilizează numai pentru verificarea rezultatelor obținute prin celelalte metode.

### *Vizualizarea traекторiilor ieșirilor*

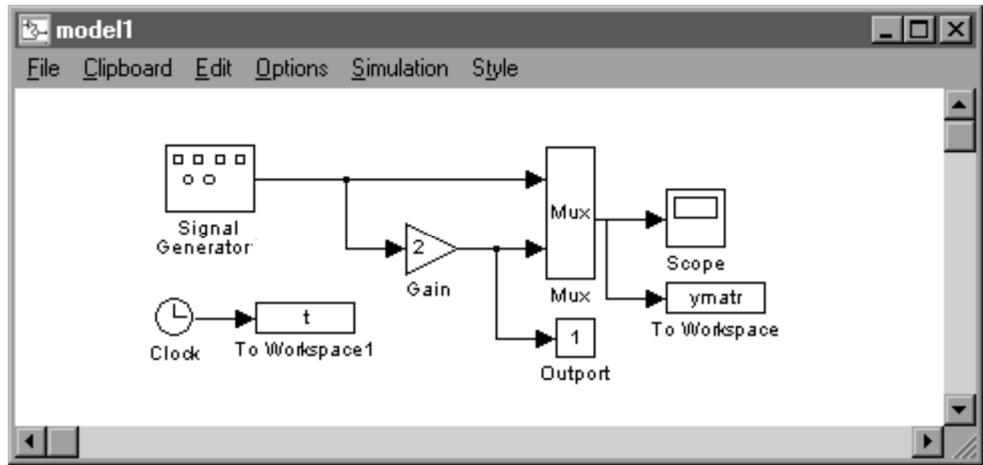
Traекторiile ieșirilor din SIMULINK pot fi afișate folosind una din metodele următoare:

- blocuri Scope sau Graph;
- variabilele returnate și comenzi de afișare MATLAB;
- blocuri To Workspace și comenzi de afișare MATLAB.

Dacă se folosesc blocuri *Scope* sau *Graph* simularea poate fi pornită din meniu sau din linia de comandă. Afișajul blocului *Scope* este foarte simplu; nu există nici o indicație de scală pe axe, există doar posibilitatea specificării capitelor de scală prin variabilele *Range*. Afișajul blocului *Graph* gradează axele în funcție de valorile introduse la setarea parametrilor.

Returnând variabilele în zona de lucru MATLAB se pot folosi comenzi grafice ale acestuia pentru afișarea unor grafice mai explicite.

Blocul etichetat *Outport 1* este un port de ieșire luat din lista conexiunilor blocurilor (biblioteca *Connections*) (figura 1.11).



**Figura 1.11.** Fereastra de lucru

Traекторiile ieșirilor se obțin în vectorul de ieșire  $y$  returnat de funcția de integrare ; de exemplu, dacă avem un sistem numit *model1* comanda:

$$[t, x, y] = \text{linsim} (N\text{model1}, 2);$$

va furniza istoria evoluției în timp a sistemului până la  $t_{final}=2$ , care poate fi afișată cu comanda:

$$\text{plot}(t, y)$$

Blocul *To Workspace* poate fi de asemenea folosit pentru a obține traectoriile ieșirilor. Variabilele sunt disponibile în spațiul de lucru MATLAB după ce simularea este completă. Vectorul timp se obține legând un bloc *Clock* de un bloc *To Workspace* (figura 1.11). Vectorul timp poate fi obținut și prin returnarea lui de către funcția de integrare:

$$t = \text{linsim} (N\text{model1}, 2);$$

Se poate observa că blocul *To Workspace* poate primi intrări multiple corespunzătoare traectoriilor stocate ca și vectori coloane ai variabilelor returnate.

### *Condiții inițiale*

Condițiile inițiale, aplicate sistemului în momentul de timp specificat de valoarea variabilei *tstart*, sunt în general setate în interiorul blocurilor. Condițiile inițiale din interiorul blocurilor pot fi modificate furnizând un vector suplimentar *x0*:

$$[t, x, y] = \text{linsim} (NmodelIN, tfinal, x0);$$

Când vectorul este vid ([ ]) sau nu este specificat, sunt folosite condițiile inițiale definite în blocuri; altfel sunt folosite condițiile definite în vectorul *x0*. (Condițiile inițiale nu pot fi modificate dacă modelul este simulat din meniu )

Condițiile inițiale posibil fixate în blocuri pot fi descoperite folosind comanda :

$$[sizes, x0] = model1;$$

unde *model1* este numele modelului.

Ordinea stărilor este disponibilă folosind un al treilea argument în partea stângă :

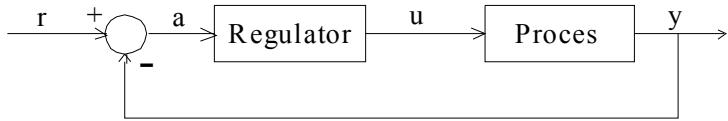
$$[sizes, x0, xstr] = model1;$$

unde *xstr* este o variabilă sir de caractere care conține în linia *i* numele blocului și calea asociată stării *I*.



## 3. Desfășurarea lucrării

Se propune în continuare realizarea modelului SIMULINK al unui sistem simplu de reglare automată, corespunzător schemei bloc din figura 1.12 și simularea răspunsului la un semnal de referință de tip treaptă. Se va considera că blocul notat *Proces* reprezintă modelul procesului, considerat aici liniar, model reprezentat prin funcția de transfer, și înglobează elementul de execuție, procesul propriu-zis, traductorul și adaptorul de măsură. Se va considera de asemenea un regulator continuu de tip PID.



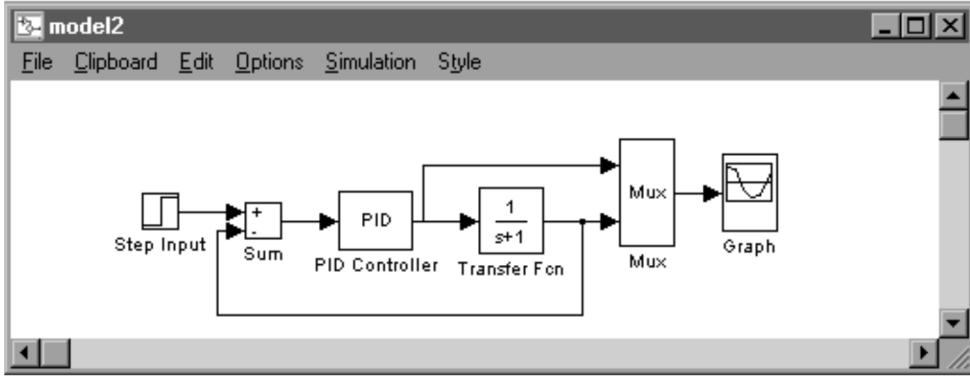
**Figura 1.12.** Schema bloc sistem de reglare

Pentru realizarea modelului simulink al acestui sistem se va selecta din nou opțiunea *New* din meniul *File* și se va deschide o fereastră nouă în care se poate începe construcția modelului. Se salvează această fereastră într-un fișier numit *model2*. În această fereastră se vor copia următoarele blocuri:

- generatorul de semnal treaptă, blocul *Step input* din biblioteca *Sources*;
- sumator de semnale, blocul *Sum* și modelul procesului, blocul *Transfer Fcn* din biblioteca *Linear*;
- regulator PID, blocul *PID Controller* din biblioteca *Extras*, grupul de modele intitulat *PID Controllers*;
- pentru vizualizarea evoluției ieșirii, blocul *Graph* din biblioteca *Sinks*.
- pentru a vizualiza și evoluția comenzi, blocul *Mux* din biblioteca de blocuri *Connections*.

Denumirile implicate ale blocurilor pot fi redenumite într-o manieră mai sugestivă. Un bloc se poate redenumi selectând numele acestuia și introducând un nume nou.

Conform celor prezentate la modul de conectare a blocurilor SIMULINK, se vor trasa liniile ce conectează blocurile copiate în fereastra de lucru (figura 1.13).



**Figura 1.13.** Fereastra de lucru

Pentru a fixa parametrii semnalului de referință se deschide blocul *Step input* și se introduce 0 pentru parametrul *Step time*, 0 pentru *Initial value* și 1 pentru *Final value*. Astfel, acest bloc va genera o treaptă unitară (cu salt de la 0 la 1) în momentul declanșării simulării.

Se deschide apoi blocul *Sum* și se fixează semnele cu care intră semnalele în acest bloc, respectiv în ordinea +– (referință cu + și reacția cu –).

Se va considera un proces simplu, caracterizat printr-o funcție de transfer de ordinul I, respectiv:

$$H_p(s) = \frac{k_p}{T_p s + 1}$$

Pentru a fixa parametrii procesului se deschide blocul *Transfer Fcn* și se introduce [kp] (vectorul corespunzător coeficienților numărătorului funcției de transfer) pentru *Numerator* și [Tp 1] (vectorul corespunzător coeficienților numitorului funcției de transfer) pentru *Denominator*.

Se deschide în continuare blocul *PID Controller*. Se observă că funcția de transfer implementată de acest bloc este de forma:

$$H_R(s) = P + \frac{I}{s} + D s$$

Pentru termenii proporțional, integral și derivativ, respectiv P, I, D se acceptă și introducerea unor expresii matematice.

Vor fi luate în considerare modificări de tip treaptă ale referinței.

Acordarea regulatorului se va face prin impunerea unui răspuns al sistemului în buclă închisă (metoda analitică de acordarea regulatorului),

respectiv prin impunerea timpului de stabilire  $T_s$  sau timpul de răspuns. Comportarea sistemului în buclă închisă va fi dată de:

$$H_0(s) = \frac{I}{T_0 s + I}$$

unde, constanta de timp corespunzătoare sistemului în buclă închisă,  $T_0$  poate fi calculată în funcție de timpul de răspuns:  $T_0 = T_s / 2.3$  dacă se consideră ca timp de răspuns intervalul de timp după care ieșirea atinge 90% din valoarea de regim staționar,  $T_0 = T_c / 3$  pentru un procent de 95% și respectiv  $T_0 = T_c / 3.9$  pentru un procent de 98%. Aceste relații sunt valabile numai pentru sisteme de ordinul I în care ieșirea evoluează exponențial la modificări de tip treaptă ale intrării.

Funcția de transfer a sistemului în buclă închisă dată de relația:

$$H_0(s) = \frac{H_p H_R}{I + H_p H_R}$$

de unde:

$$H_R(s) = \frac{H_0}{H_p(I - H_0)} = \frac{I}{H_p T_0 s} = \frac{T_p}{k_p T_0} + \frac{I}{k_p T_0} \frac{I}{s} = P + \frac{I}{s}$$

A rezultat deci un regulator cu structură PI ( $D=0$ ). În consecință, după deschiderea blocului *PID Controller* se introduce pentru parametrul P expresia  $T_p/(k_p \cdot T_0)$ , pentru parametrul I expresia  $1/(k_p \cdot T_0)$  și 0 pentru parametrul D.

Blocul *Graph* este folosit pentru vizualizarea evoluției ieșirii procesului. Se deschide și acest bloc și se introduce pentru parametrul *Time range* valoarea corespunzătoare capătului de scală pentru axa timpului (în cazul nostru 100), iar pentru parametrii *y-min* și *y-max* valoarea minimă și respectiv cea maximă pentru reprezentarea pe ordonată (valori rezonabile pot fi *y-min*=0 și *y-max*=2). Ultima căsuță de text se referă la culoarea reprezentării grafice respectiv *y-* (yellow) linie continuă de culoare galbenă pentru primul semnal, *g--* (green) linie intreruptă de culoare verde pentru al doilea semnal etc. Nu se modifică parametrii din această rubrică.

În acest moment modelul este finalizat și se poate începe simularea. Pentru a alege metoda de integrare și parametrii care vor fi folosiți în timpul simulării, se selectează opțiunea *Parameters* din meniul *Simulation*. Ca

rezultat se va deschide o fereastra *Control Panel*, afișând valorile implicate ale parametrilor simulării dintre care se va modifica doar parametrul *Stop Time* pentru care se va introduce valoarea 100. Pasul de integrare se alege automat astfel încât eroarea să fie mai mică decât valoarea corespunzătoare a parametrului *err*. Se salvează din nou modelul.

Se decanșează simularea selectând opțiunea *Start* din meniul *Simulation*. Programul afișează însă un mesaj de eroare deformată prezentată în figura 1.14.

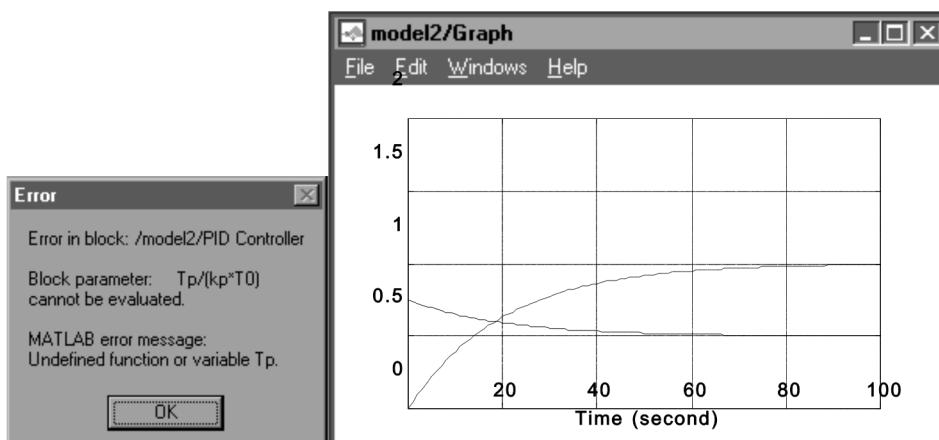


Figura 1.14. Mesaj eroare

Figura 1.15. Rezultate simulare

Pentru a rezolva această problemă se trece în fereastra MATLAB și se inițializează următorii parametri:

- $kp = 2$ ,  $Tp = 30$ , -funcția de transfer a procesului;
- $T0 = 20$ , performanțele impuse sistemului.

Dacă se declanșează din nou simularea, rezultatele vor fi afișate sub formă graficului prezentat în figura 1.15, unde cu linie continuă este reprezentată evoluția comenzi și cu linie întretreruptă cea a ieșirii. Se poate observa că timpul de răspuns al sistemului este de cca 60 secunde. În continuare se pot modifica atât parametrii modelului procesului ( $kp$  și  $Tp$ ), cât și performanțele impuse ( $T0$ ), vizualizându-se și interpretându-se rezultatele de fiecare dată.

SIMULINK are posibilitatea de a simula și sisteme discrete (sisteme cu eșantionare). Modelele pot fi de tip *multirate* aceasta însemnând că pot

conține blocuri care lucrează cu perioade de eșantionare diferite. Modelele pot fi construite numai din blocuri discrete sau pot fi mixte, conținând combinații de blocuri discrete și continue. Pentru simularea modelelor mixte metodele de integrare Runge-Kutta cu pas variabil sunt superioare celorlalte metode în ceea ce privește viteza și precizia.

Majoritatea blocurilor discrete sunt incluse în biblioteca de blocuri *Discrete*. Fiecare dintre aceste blocuri are incorporat un element de eșantionare pe intrări și un extrapolator de ordinul zero pe ieșiri. Intrările și ieșirile blocurilor discrete sunt actualizate numai la momente de timp ce corespund multiplilor perioadei de eșantionare. Perioada de eșantionare a unui bloc discret se poate seta în căsuța corespunzătoare (*Sample Time*) din fereastra de dialog afișată la selecția blocului.

Ca o aplicație imediată a utilizării blocurilor discrete se va înlocui regulatorul continuu PID din modelul precedent printr-un regulator discret. Una dintre posibilități constă în înlocuirea blocului *PID Controller* cu blocul *Discrete Transfer Fcn* din biblioteca de blocuri *Discrete*. Pentru aceasta se salvează modelul într-un nou fișier sub numele *model3*, se sterge blocul *PID Controller* și se copiază blocul *Discrete Transfer Fcn*. Funcția de transfer în s a regulatorului PI utilizat în exemplul precedent este de forma:

$$H_R(s) = \frac{P s + I}{s}$$

Se trece acum în fereastra MATLAB și se discretizează funcția de transfer continuă a regulatorului, cu o perioadă de eșantionare de 1 sec. (mică în raport cu constantele de timp). Se poate realiza aceasta cu comanda:

$$[numz,denz] = c2dm([P\ I], [1\ 0], 1).$$

Se revine acum în fereastra *model3*, se selectează blocul *Discrete Transfer Fcn* introduce pentru parametrul *Numerator* expresia *numz*, pentru *Denominator* expresia *denz* iar pentru *Sample time* valoarea 1. Rezultatele simulării vor fi identice cu cele obținute în cazul regulatorului continuu. Se pot face mai multe simulări pentru valori diferite ale perioadei de eșantionare. De fiecare dată însă, este necesară discretizarea regulatorului continuu pentru valoarea respectivă a perioadei de eșantionare, utilizând funcția *c2dm*.

Un model mai eficient se poate realiza utilizând pentru regulatorul discret blocul *S-Domain to Z-Domain* din biblioteca *Extras*, grupul de blocuri

*Controllers.* Acest bloc are ca parametri de intrare numărătorul și numitorul unei funcții de transfer în s și perioada de eșantionare. El discretizează automat funcția de transfer continuă corespunzător perioadei de eșantionare introdusă și simulează modelul discret. Se va încerca aceasta ca temă.

#### **4. Conținutul referatului**

- prezentarea unui rezumat al etapelor semnificative pentru simularea sistemelor dinamice în SIMULINK;
- prezentarea unui rezumat al conținutului bibliotecilor standard;
- prezentarea sumară a unei metode de acordare a regulatorului, alta decât cea prezentată în lucrare, simularea comportării sistemului obținut și interpretarea performanțelor de reglare pentru diverse valori ale parametrilor părții fixate.

## **2. SIMATIC Step 7**



### **1. Scopul lucrării**

Obiectivul lucrării de laborator este familiarizarea automatele programabile din seria Siemens SIMATIC S7-300 și cu mediul de programare Simatic Step 7 și realizarea câtorva aplicații utilizând standul de laborator cu PLC S7-300. Pentru testarea aplicațiilor se va folosi atât simultorul mediului de programare cât și un PLC Siemens Simatic S7-300.



### **2. Considerații teoretice**

#### **2.1. Automate programabile SIMATIC S7-300**

O Unitate Logică de Control Programabilă (PLC) este un computer digital folosit pentru automatizarea unor procese electro-mecanice, cum ar fi controlul mașinilor din cadrul unei linii de producție al unei fabrici, chiar controlul mașinilor și instalațiilor din parcuri de distracție. PLC-urile sunt utilizate în multe ramuri ale industriei pentru a realiza funcții de control și pentru eficientizarea procesului tehnologic (industria alimentară, industria construcțiilor de mașini etc.).

Spre deosebire de calculatoarele personale, PLC-urile sunt computere specializate proiectate pentru a fi utilizate pentru procese cu numeroase intrari și iesiri și construite în astfel încât să reziste variațiilor de temperatură, vibratiilor și impactului, imune la zgomote electrice. Programele create sunt stocate în memoria non-volatile. Un PLC este un exemplu de sistem în timp real deoarece iesirile sunt un răspuns la condițiile de la intrare și sunt strâns legate de timp.

Un PLC, este un mic computer cu un microprocesor folosit pentru automatizarea proceselor. Programul unui PLC poate adesea controla secvențe complexe și de cele mai multe ori este scris de către un inginer. Programul este apoi salvat în memoria EEPROM.

## Lucrarea 2

PLC-urile au fost inventate ca o alternativa mai putin costisitoare la vechile sisteme care foloseau zeci sau sute de relee și timere. Adesea un PLC poate fi programat să înlocuiască sute de relee. Automatele programabile au fost inițial folosite de industria constructoare de mașini.

Funcționalitatea unui PLC s-a dezvoltat de-a lungul anilor pentru a include controlul mișcării, control de proces, Sisteme de Control Distribuit și rețele complexe.

La primele PLC-uri funcțiile decizionale erau implementate cu ajutorul unor simple diagrame ladder (*Ladder Diagram*) inspirate de schemele electrice de conexiuni. Astfel electricienilor le era ușor să depaneze problemele de circuit având diagramele schematizate cu logica lader.

În prezent, linia ce delimită un computer programabil de un PLC este tot mai subțire. PLC-urile s-au dovedit a fi mai robuste, în timp ce computerele au încă deficiențe. Folosind standardul IEC 61131-3 acum este posibilă programarea PLC folosind limbaje de programare structurată și operații logice elementare. La unele PLC este disponibilă programarea grafică denumită (*Sequential Function Charts*) bazată pe *Grafset*.

Deosebirea principală față de alte calculatoare este aceea că PLC-urile au fost create pentru a rezista în condiții severe (praf, umezeală, căldură, frig etc.) și au proprietatea că pot fi adăugate module de intrare/ieșire necesare pentru legătura dintre PLC și senzori sau elemente de execuție. Pot fi utilizate pentru interpretarea semnalelor analogice (temperatură, presiune, umiditate, etc.), intrări digitale de la relee sau poziții din cadrul unor sisteme complexe de poziționare, unele chiar pot fi utilizate pentru procesare de imagini (în cadrul unor procese ce necesită inspecția vizuală-analiza unor cipuri, inspector pentru recipiente etc.). Din punct de vedere al elementelor de execuție prin intermediul PLC-urilor pot fi comandate motoare electrice, cilindri pneumatici și hidraulici, relee magnetice și bobine sau ieșiri analogice. Pot fi construite ca un sistem compact sau având posibilitatea de a se adăuga module de întare/ieșire.

SIMATIC S7-300 este un automat programabil cu foarte multe aplicații de succes în diferite sectoare industriale. S7-300 e folosit pentru a soluționa probleme în special în domeniul ingineriei asigurând o platformă universală

## Lucrarea 2

in automatică ceea ce înseamnă o soluție optimă pentru aplicații în instalațiile centralizate și distribuite.



**Figura 2.1.** Familia de automate programabile Simatic S7

Familia S7 cuprinde automate programabile (PLC) cu procesoare puternice, cu funcții integrate, comod de utilizat.

Simatic S7-200 este un micro-PLC cu înaltimea de numai 8 cm, în formatul clasic cu cleme pe marginile de sus și de jos. Unitatea centrală cu intrări/ieșiri integrate poate fi utilizată singură sau se pot adăuga module de extensie pentru suplimentarea numărului de semnale. S7-200 este recomandabil pentru un număr de semnale de ordinul zecilor.

Domeniile de utilizare sunt: prese de ambalat, agregate pentru ciment și betoane, mecanisme de absorbtie, echipamente pentru lubrifierea roților, struguri, controlul sistemelor de acces (uși, porți), lifturi hidraulice, echipamente de transmisie, industrie alimentară, laboratoare, monitorizare, instalații electrice.

Simatic S7-300 este un automat programabil mediu, format din module cu înaltimea de 12,5 cm, alăturate pe orizontală. Este recomandabil

## Lucrarea 2

pentru un număr de semnale de ordinul miilor, reprezentând o soluție convenabilă pentru o mare varietate de aplicații ce nu necesită performanțe deosebite. Acesta este prezentat mai pe larg în continuare.

Domeniile de aplicații pentru SIMATIC S7-300 sunt: aplicații generale, echipamente cu destinații speciale, controlul proceselor, industria textilă, mașini de împachetat, ingineria intalațiilor, industria electrică și multe altele.

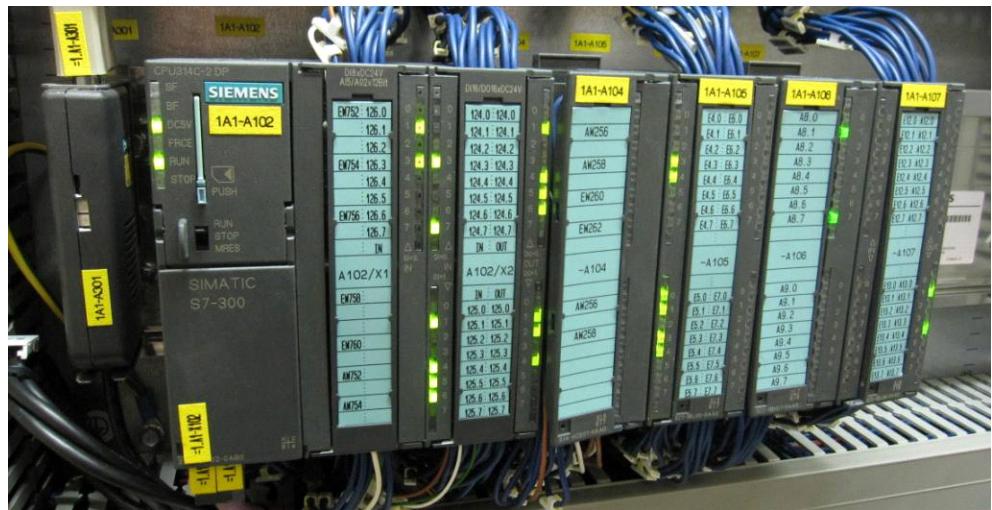
Capul de serie *Simatic S7-400* are cele mai puternice caracteristici, cele mai variate opțiuni și cele mai multe posibilități. Modulele se montează într-un sertar de 48 cm lățime. Este recomandabil pentru un număr de semnale de ordinul miilor.

Numărul de semnale menționat orientativ mai sus se referă la un singur automat. În mod normal se folosesc mai multe automate distribuite în proces, conectate printr-o rețea "fieldbus" la posturi operator (calculatoare sau panouri).

Domeniile de aplicații pentru *SIMATIC S7-400* sunt: aplicații generale, inginerie mecanică, construcție motoare, echipamente de depozitare, construcție piese de mașini, inginerie de proces, echipamente de control și instrumentație, industria textilă, mașini de împachetat, echipamente speciale.

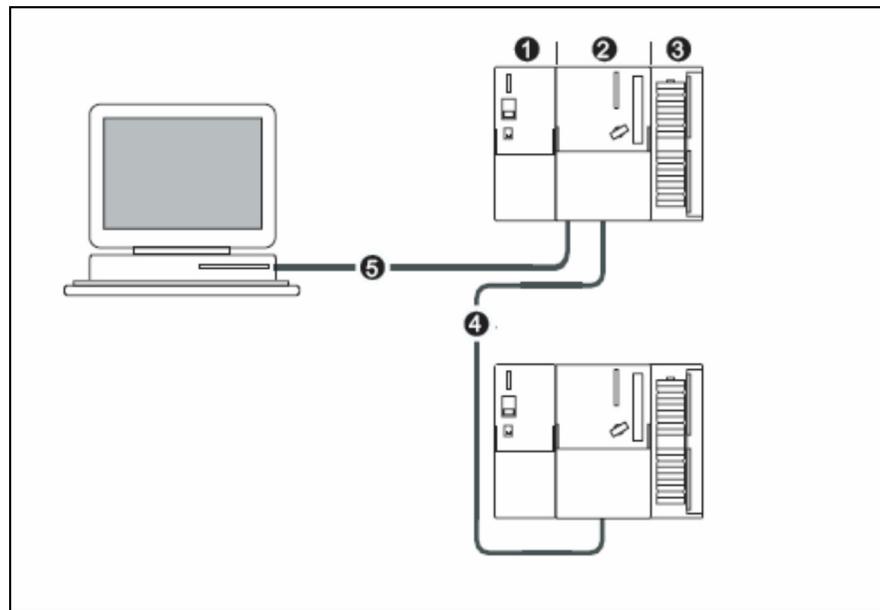
Revenind la automatele programabile S7-300, se poate observa că se dispune de un mare număr de componente compacte, modulare, cu dimensiuni mici, care pot fi puse împreună în orice combinație. Aceasta înseamnă că sistemul poate fi proiectat pentru a se potrivi exact unei anumite aplicații. Pe lângă module, mai este nevoie de o șină DIN pe care sunt fixate modulele, asigurând o construcție robustă. În fiecare modul este integrat câte un conector de magistrală. Pot fi puse până la 8 module împreună pe o linie (controler central/unitate de expansiune) și până la 32 module pe un sistem, acestea fiind alăturate pe orizontală. Poate fi conectat la rețea prin: Interfata Multipunct(MPI), PROFIBUS sau Ethernet Industrial.

## Lucrarea 2



**Figura 2.2.** Structura unui sistem cu S7-300

În următoarea diagramă este prezentată o posibilă configurație.



**Figura 2.3.** Structura unui sistem cu S7-300

## Lucrarea 2

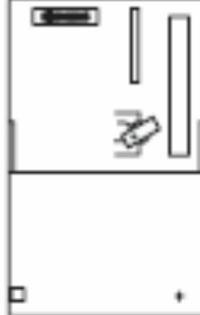
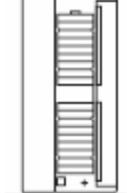
În diagramă sunt folosite următoarele module:

- (1) – sursa de alimentare;
- (2) – unitate centrală de prelucrare;
- (3) – modul de semnal;
- (4) – cablu magistrală;
- (5) – cablu pentru conectarea unui dispozitiv de programare.

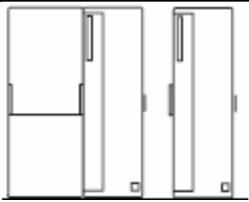
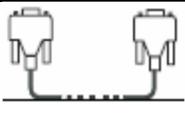
În tabelul 2.1 vor fi prezentate principalele module ale unui SIMATIC S7-300 și funcțiile acestora:

Componente	Funcție	Reprezentare
Șină de montare (suport) Accesorii: Element cu armătură de contact	Reprezintă suportul mecanic pentru SIMATIC S7-300	
Sursă de alimentare	Convertește tensiunea de la rețea (120 - 230 VAC) la 24 VDC, tensiunea de funcționare și este folosită pentru a furniza energie CPU și modulelor S7-300.	

## Lucrarea 2

<p>CPU (Unitatea Centrală de Prelucrare)</p> <p>Accesorii: conectori I/O (pentru CPU cu interfețe periferice integrate)</p>	<p>Execută programul utilizatorului, asigură interfațarea cu semnale analogice și numerice (pentru CPU cu interfețe periferice integrate) și comunică cu alte puncte ale rețelei prin interfața MPI sau Profibus. Poate fi master sau slave.</p>	
<p>Module de Semnal (MS) – module digitale de I/O, module analogice de I/O.</p> <p>Accesorii: conectori.</p>	<p>Asigură interfațarea cu semnale analogice și numerice. Răspund unor nivele diferite de procesare a semnalului pentru S7-300</p>	
<p>Module de funcții (FM)</p> <p>Accesorii: conectori.</p>	<p>Îndeplinesc obligațiile de procesare a semnalului privind timpul critic și memoria, cum ar fi poziția sau reglarea</p>	
<p>Procesor de comunicare (CP).</p> <p>Accesorii: Cablu de conectare</p>	<p>Acesta asigură comunicarea pentru CPU, de exemplu CP 314-2DP conectat la Magistrala DP</p>	

## Lucrarea 2

Modul Interfață (IM) Accesorii: cablu de conectare.	Modul de expansiune - bucleaza magistrala de la un rack (sina) la urmatorul	
Cablu magistrală cu conector magistrală	Conectează nodurile unei MPI sau magistrale ale unei rețele secundare	
Cablu dispozitiv de programare (PG)	Conectează un dispozitiv de programare (PG/PC) la un CPU	
Repetor RS485	Folosit pentru a amplifica semnalele într-o MPI (Multi Point Interface) sau magistrală (PROFIBUS) ale unei rețele secundare pentru cuplarea segmentelor acestora.	
Dispozitiv de programare (PG) sau PC cu pachetul soft STEP 7	Este nevoie de un dispozitiv de programare pentru a configura, seta parametrii sistemului, pentru a realiza programul și pentru a testa S7-300	

**Tabelul 2.1.** SIMATIC S7-300 și funcțiile acestora

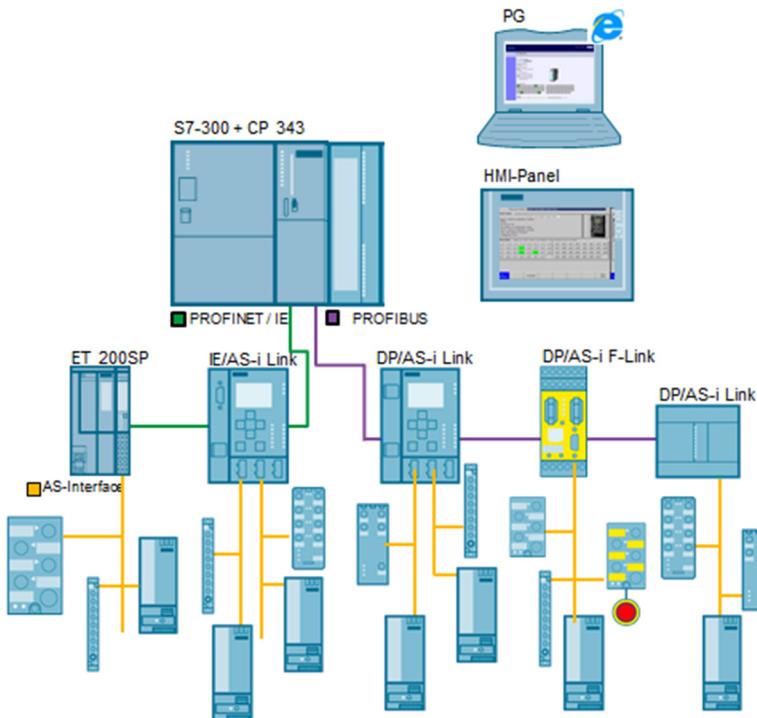
### *2.1.1. Caracteristici ale unității centrale de prelucrare (CPU)*

*Interfețe de comunicație integrate:*

**Interfața multi-punct (MPI):** MPI este soluția posibilă pentru comunicațiile cu sisteme PG/PC și alte sisteme automate SIMATIC S7. Pot fi conectate maximum 125 noduri MPI cu 187.5 kbiți/sec. Comunicatia poate asigura atât transferul datelor prelucrate între diferite controlere, ceea ce înseamnă că, de exemplu, un CPU poate accesa intrările/ieșirile altui controler, cât și control și monitorizarea funcționării automatului programabil de către un sistem PG/PC.

**Magistrala Profibus-DP:** Protocolul Profibus DP este folosit pentru comunicații de viteză mare cu dispozitive I/O distribuite. Există o gamă largă de dispozitive Profibus, de la simple module de intrare sau ieșire, traductoare sau elemente de execuție, la regulatoare de turăție, convertizoare, automate programabile etc. Simatic S7-300 poate fi conectat la magistrala ProfiBus-DP pentru a realiza structuri distribuite relativ mari. Profibus DP (“Distributed Periperal”-periferice distribuite) este un protocol de comunicație la distanță definit prin standardul european EN 50170. Dispozitivele cu protocoale ce respectă acest standard sunt compatibile chiar dacă producătorii lor sunt diferiți. DP precizează faptul că intrările și ieșirile pot fi la distanță (remote I/O), nu neapărat incluse în dispozitivul respectiv. Modulele distribuite de I/O sunt configurate folosind STEP7 în același fel ca și modulele centralizate de I/O. S7-300 poate fi folosit de asemenea ca master sau slave.

**Funcții partajate (folosite în comun):** funcțiile de monitorizare și funcțiile dispozitivului de programare sunt posibile în ambele interfețe (MPI, DP), de exemplu: programarea folosind dispozitiv de programare/PC pentru distanțe relativ mari. Un dispozitiv de programare poate de asemenea opera asupra mai multor CPU sau un număr de dispozitive de programare pot accesa un singur CPU. Există de asemenea o funcție de traseu, care permite unui dispozitiv de programare conectat la orice punct în rețea să acceseze nodurile acestei rețele. Cu ajutorul acestor module de comunicație pot fi implementate multe funcții complexe.



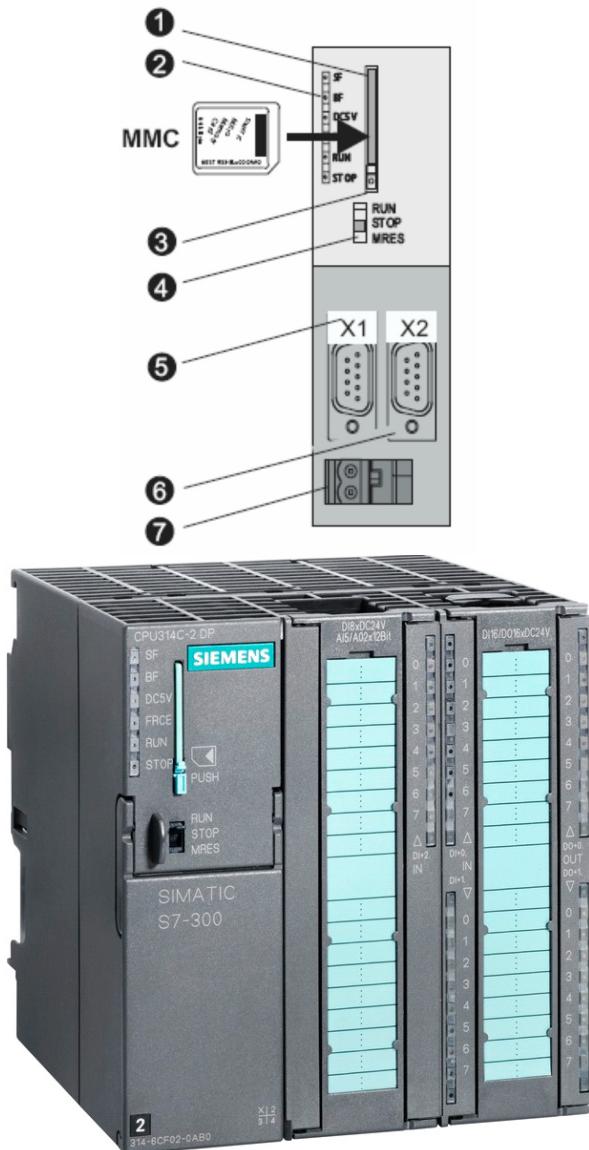
**Figura 2.4.** Opțiuni de conectare prin interfețele integrate ale CPU SIMATIC S7-300

**Periferice integrate și funcții tehnologice:** Depinzând de tipul lor, CPU au încorporate intrări și ieșiri digitale și analogice și intrări speciale pentru numărare și măsurare cu generatoare de impuls sau encodere. Unele CPU au de asemenea funcții integrate pentru control PID sau controlul poziției cu ieșiri speciale.

Intrucât laboratorul de echipamente de automatizare este dotat cu standuri având la bază automate programabile SIMATIC S7 314-2DP, În continuare se vor face referiri la acest tip de automat. CPU 314C-2 DP este un CPU compact potrivit pentru aplicații de marime medie. Intrările/ieșirile digitale și analogice integrate permit o conectarea directă între automatul programabil și proces iar interfata master slave PROFIBUS DP permite conectarea unor unități separate de intrare/ieșire prevăzute cu acest tip de comunicație. CPU 314C-2 DP poate fi folosit atât ca o unitate locală pentru pre-procesare rapidă, cât și ca un control la nivel înalt cu magistrala de sistem subordonată.

## Lucrarea 2

Pe paoul frontal al unității centrale se găsesc o serie de LED-uri pentru semnalizarea stării automatului programabil, butonul pentru stabilirea modului de lucru și slotul pentru cartela de memorie iar în partea de jos, sub capacul ce acoperă aproximativ jumătate din panoul frontal se găsesc conectorii de comunicație.



**Figura 2.5. SIMATIC S7-300**

## Lucrarea 2

### 1. Cartela de memorie (Memory Card)

Pe unitatea centrală există un slot rezervat pentru o cartela de memorie. Cartela de memorie salvează conținutul programului în ipoteza unei întreruperi de tensiune, fără a fi nevoie de o baterie de back-up.

### 2. Leduri pentru afișarea stării de funcționare și erori (Status and error displays).

SF	Rosu	Eroare hardware sau software
BF	Rosu	Eroare de comunicare (doar la CPU care au interfata DP)
DC5V	Verde	Alimentarea 5V pentru CPU si magistrala S7-300 sunt ok.
FRCE	Galben	Comanda fortată activă
RUN	Verde	CPU în RUN ; LED-ul pulsează în STARTUP cu frecvența de 2Hz ; în PAUSE cu 0.5Hz
STOP	Galben	CPU în STOP resp. în PAUSE sau STARTUP ; LED ul clipește rar (0,5Hz) pentru o cerere de resetare de memorie, iar în timpul resetării clipește mai des (2 Hz).

**Tabelul 2.2.** Semnificație LED-uri CPU

### 3. Dispozitiv pentru ejectarea cardului de memorie

4. Selector pentru modul de lucru (Mode Selector), este un comutator cu două pozitii, STOP și RUN și cu funcție de push buton pentru MRES din poziția STOP

MRES - Funcția de resetare a memoriei (Module Reset)

STOP - Modul Stop, programul nu este executat.

RUN - Execuția programului.

### 5. Interfata cu capacitate multipunct (MPI)

Interfata Multipoint integrată poate să stabilească până la 12 conexiuni simultane la S7-300/400 sau la PG, PC, OP. Printre aceste conexiuni, una este întotdeauna rezervată PG și o altă pentru OP.

### 6. Interfata PROFIBUS DP:

Interfata master slave PROFIBUS DP face posibilă crearea sistemelor automatizate descentralizate de viteze foarte mari și ușor de folosit. Din

## Lucrarea 2

punctul de vedere al utilizatorului intrarile/iesirile distribuite sunt tratate ca si intrari/iesiri centrale (aceeasi configuratie, adresare si programare).

### *2.1.2. Specificatii tehnice ale unitatii centrale de prelucrare CPU 314C-2DP*

<b>Memoria de lucru (apliatii)</b>	96 Kbytes/32 K instrucțiuni
<b>Backup (folosind MMC)</b>	toate modulele
<b>Numar de module</b>	128 FCs, 128 FBs, 127 DBs
<b>Prelucrarea programului</b>	ciclu liber (OB 1), controlate în timp real (OB 10), alarmă de întârziere (OB 20), regim de autodeclanșare în timp (OB 35), regim de autodeclanșare la întreruperi (OB 40), restart (OB 100, 102), eroare asincronă (OB 80..82,85,87), eșecul/restabilirea stației (OB 86), eroare sincronă (OB 121, 122);
<b>Timp de operare</b> - operații pe bit - operații pe cuvânt (word)	0.1 µs 0.2 µs
<b>Memorii/timere/contoare pe bit</b> - memorii pe bit - timere / contoare S7 - timere IEC /contoare IEC	256 bytes 256/256 Da
<b>Numar de module pe sistem</b>	31
<b>Numar de CC/EU</b>	1/3
<b>Domenii de adrese</b> - Domeniu adrese I/O	1024/1024

<ul style="list-style-type: none"> <li>- Procesare imagine I/O</li> <li>- Canale digitale (integrate)</li> <li>- Canale analogice (integrate)</li> </ul>	128/128 bytes Maxim 1016 Maxim 253												
<b>Funcții integrate</b>	<ul style="list-style-type: none"> <li>- Contoare</li> <li>- Ieșiri impuls</li> <li>- Măsurarea frecvenței</li> <li>- Poziționare buclă deschisă</li> <li>- Control integrat FB „Closed-loop”</li> </ul>												
<b>Intrări/Ieșiri integrate</b>	<table border="0"> <tr> <td>4</td> <td>encodere incrementale</td> </tr> <tr> <td>24V/60Khz</td> <td></td> </tr> <tr> <td>4</td> <td>canale cu modulare în timp, maxim 2.5 KHz</td> </tr> <tr> <td>4</td> <td>canale maxim 60 KHz</td> </tr> <tr> <td>SFB</td> <td>pentru poziționare unei axe via 2 DO, AO</td> </tr> <tr> <td colspan="2">Controler PID</td> </tr> </table>	4	encodere incrementale	24V/60Khz		4	canale cu modulare în timp, maxim 2.5 KHz	4	canale maxim 60 KHz	SFB	pentru poziționare unei axe via 2 DO, AO	Controler PID	
4	encodere incrementale												
24V/60Khz													
4	canale cu modulare în timp, maxim 2.5 KHz												
4	canale maxim 60 KHz												
SFB	pentru poziționare unei axe via 2 DO, AO												
Controler PID													
<b>Dimensiuni (în mm)</b>	125 x 125 x 130												

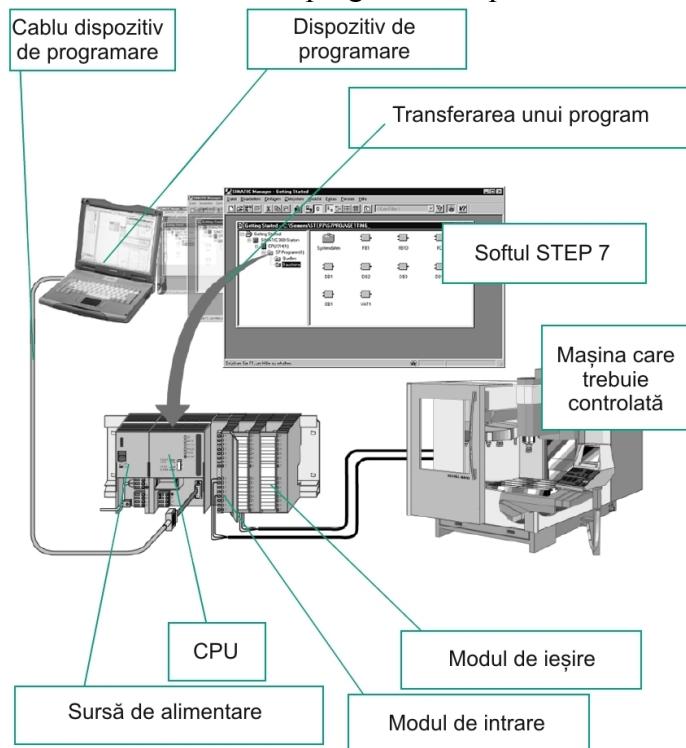
**Tabelul 2.3.** Specificații tehnice CPU 314C-2DP

## 2.2. Mediul de programare Step 7

Folosind softul STEP 7, se poate crea un program S7 în cadrul unui proiect. Controller-ul programabil S7 este format dintr-o sursă de alimentare, un CPU și module de intrare și ieșire (module I/O).

Controller-ul Logic Programabil (PLC) monitorizează și controlează un echipament (instalație) cu ajutorul programului S7.

Modulele I/O sunt accesate în programul S7 prin intermediul adreselor.



**Figura 2.6.** Viziune de ansamblu asupra unui proiect de automatizare

### 2.3.Crearea unei aplicații Step 7

Înainte de a crea un proiect, este bine de știut că implementarea unui program în STEP 7, pentru o anumită aplicație, poate fi realizată în mai multe moduri (figura 2.7).

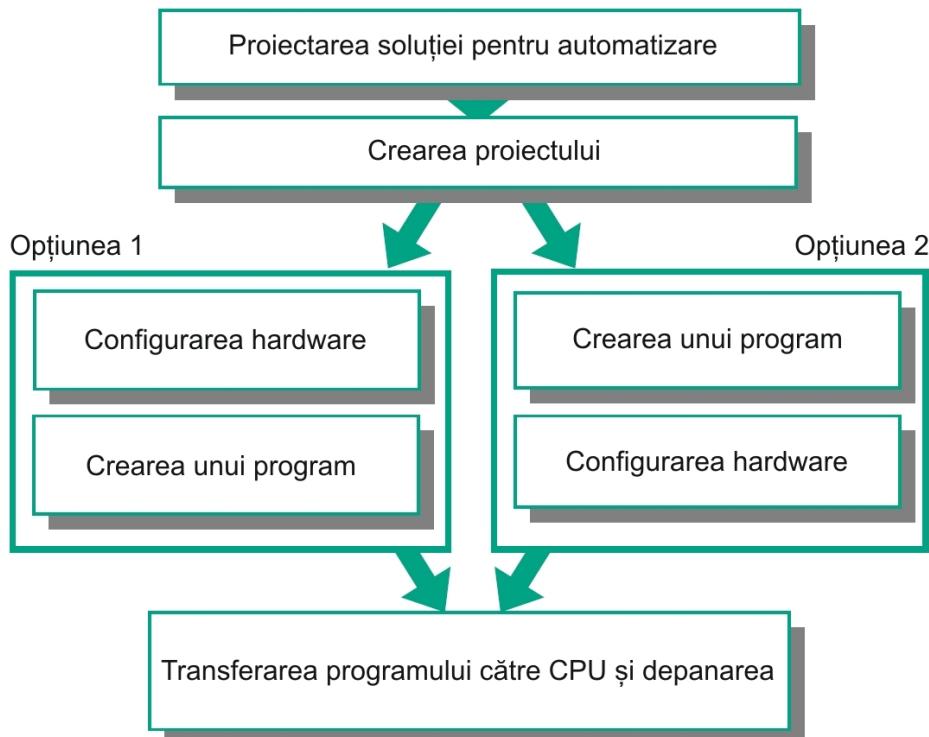
Dacă aplicația este una mai complexă, cu mai multe intrări și ieșiri, se recomandă realizarea configurării hardware la început. Avantajul este că STEP 7 afișează adresele posibile la Hardware Configuration Editor.

Dacă se alege a doua opțiune utilizatorul va trebui să aleagă fiecare adresă în funcție de componentele selectate și nu se poate apela la ajutor din partea mediului STEP 7.

La configurația hardware se pot defini adrese și se pot schimba parametrii și proprietățile modulelor.

## Lucrarea 2

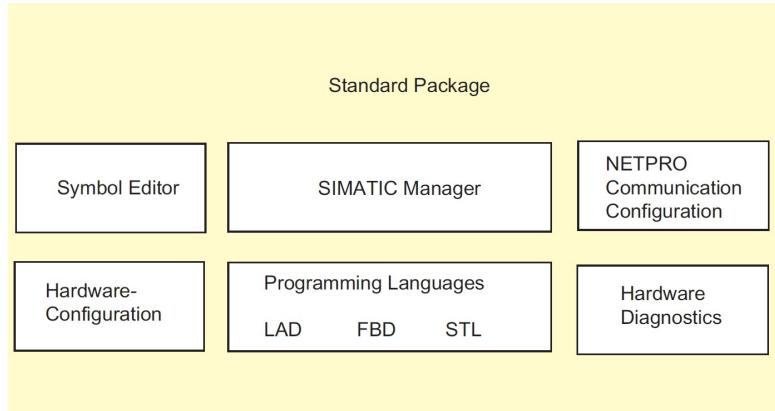
După acești pași se transferă programul către CPU și se testează funcționarea.



**Figura 2.7.** Crearea unei aplicații STEP 7

### **2.4.Instrumente în STEP 7**

Pachetul software STEP 7 înglobează o serie de aplicații (figura 2.8).

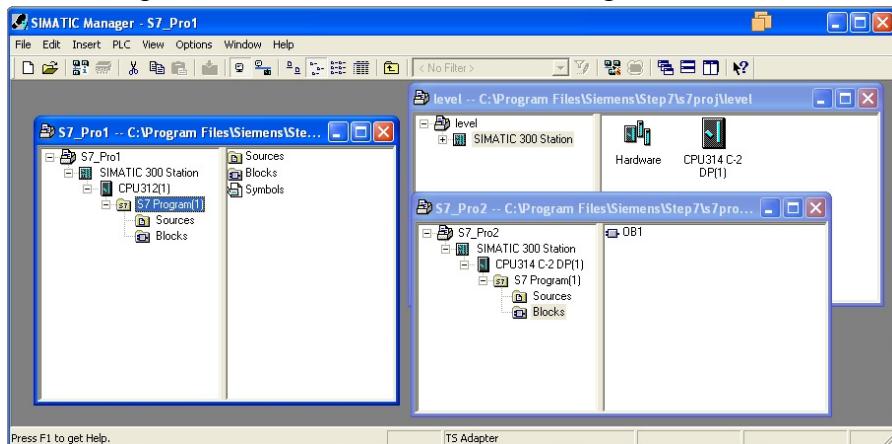


**Figura 2.8.** Instrumente STEP 7

Nu este neceasă deschiderea acestor aplicații separat. Ele sunt pornite automat când se selectează o funcție corespunzătoare sau se deschide un obiect.

#### **2.4.1. SIMATIC Manager**

SIMATIC Manager gestionează toate datele care aparțin unui proiect de automatizare, indiferent de sistemul de control programabil (S7/M7/C7) pentru care sunt concepute. Instrumentele necesare pentru editarea datelor selectate sunt pornite automat de SIMATIC Manager.



**Figura 2.9.** SIMATIC Manager

#### 2.4.2. *Symbol Editor*

Cu Symbol Editor se gestionează toate simbolurile aplicației. Sunt disponibile următoarele funcții:

- Crearea numelor și componentelor simbolice pentru semnalele procesului (intrări/ieșiri, bit memorie și blocuri);
- Funcții de sortare;
- Import/export la/de la alte programe Windows.

Tabela de simboluri creată cu acest instrument este disponibilă pentru poate celelalte instrumente ale pachetului SIMATIC. Orice modificare a proprietătilor unui simbol este aşadar recunoscută automat de toate celelalte instrumente.

#### 2.4.3. *Hardware Diagnostics*

Acste funcții oferă o imagine de ansamblu asupra stării controller-ului programabil. Se poate localiza un defect și se pot obține informații detaliate despre el.

- Afisează informații generale despre modul (de exemplu: numărul de ordine, versiune, numele), precum și starea modulului (de exemplu defecte);
- Afisează defectele modulului (de exemplu canal defect) pentru I/O central și slave DP;
- Afisează mesajele de la buffer-ul de diagnosticare.

Pentru CPU afișază următoarele informații suplimentare:

- Cauzele defecțiunilor la rularea unui program utilizator;
- Afisează durata ciclului (al celui mai lung, mai scurt, și al ultimului ciclu);
- Posibilități de comunicare pe MPI și de încărcare;
- Date de performanță (număr de intrări/ieșiri posibile, bit memorie, numărătoare, timere, și blocuri).

#### 2.4.4. Limbaje de programare

Este permisă programarea în:

- Ladder Logic (LAD);
- Function Block Diagram (FBD);
- Structured Control Language (SCL);
- Statement List (STL);
- Sequential Function Chart (SFC) – GRAPH.

- ✓ **Ladder Logic** este un limbaj grafic cu sintaxa similară cu o schema cu relee.
- ✓ **Function Block Diagram** este un limbaj grafic și folosește blocuri logice din algebra booleană. Funcțiile complexe (de exemplu funcțiile matematice) pot fi reprezentate direct în conjuncție cu blocurile logice.
- ✓ **Structured Control Language** este un limbaj textual de nivel înalt definit în standardul IEC 61131-3 și se utilizează la programarea algoritmilor complexi de control, a funcțiilor aritmetice și pentru sarcini de procesare de date.
- ✓ **Statement List** este un limbaj textual. Dacă un program este scris în STL, instrucțiunile individuale corespund pașilor cu care procesorul execută programul. Este conceput pentru crearea de rutine la nivel hardware și de programe utilizator cu optimizare folosirii memoriei PLC. STL cuprinde construcții de limbaj de nivel înalt, de exemplu, acces la date structurate și la parametrii blocurilor.
- ✓ **Sequential Function Chart** este utilizat pentru a descrie procedurile cu secvențe alternante sau paralele.

#### 2.4.5. Hardware Configuration

Acest instrument se utilizează pentru configurarea și atribuirea de parametrii părții hardware a unui proiect.

Sunt disponibile următoarele funcții:

- Configurarea controller-ului programabil;
- Configurarea modulelor I/O;

- Atribuirea de parametrii modulelor de funcții și procesoarelor de comunicație.

#### *2.4.6. NetPro (Network Configuration)*

Permite transferul de date time-driven și event-driven.

### **2.5.Crearea și editarea unui proiect**

Proiectele sunt folosite pentru a stoca datele și programele care sunt create când sunt puse împreună într-o soluție de automatizare.

Datele colecționate într-un proiect includ:

- Datele de configurare din structura hardware și parametrii pentru module
  - Datele de configurare pentru comunicarea în rețea
  - Programele pentru modulele programabile
- ✓ Sarcina principală la crearea unui proiect este pregătirea acestor date pentru programare
  - ✓ Datele sunt stocate într-un proiect sub formă de obiecte
  - ✓ Obiectele dintr-un proiect sunt aranjate într-o structură arbore (ierarhie de proiect)

#### *2.5.1. Organizarea generală a programului*

Modul de organizare a unui program STEP 7 are anumite particularități față de alte limbaje de programare. Un factor foarte important în aceasta organizare îl au OB-urile (Organization Block). De fapt aceste blocuri sunt singurele care se execută în rularea programului dintr-un CPU, restul funcțiilor fiind doar apelate din interiorul acestor blocuri.

Singurul bloc de organizare (OB) care va apărea obligatoriu în orice proiect dezvoltat în STEP 7 este OB1, un bloc care se execută într-o buclă infinită atât timp cât CPU-ul se află în modul RUN. Toate celelalte OB-uri reprezintă cazuri speciale în evolutia programului.

Pentru o mai bună organizare a programului sunt folosite anumite funcții, FC-uri și FB-uri, diferența dintre aceste 2 tipuri de funcții fiind aceea că FB vor avea întotdeauna un bloc de date asociat pe când FC-urile pot să apeleze

## Lucrarea 2

în mod indirect anumite blocuri de date dar nu vor avea un anumit bloc de date asociat.

Blocurile de date, DB-urile, sunt blocuri în care memoria CPU-ului poate fi împărtită în anumite segmente distincte, cu adrese distincte, în funcție de tipul de date care se dorește a fi memorat în acea locație.

Blocurile de date pot fi împărtite în două mari categorii:

- DB-uri generale în care sunt păstrate valori globale ale programului
- DB-uri instanțiate. Aceste DB-uri corespund fiecare unei anumite instantieri a unei funcții FB

De subliniat este faptul că pentru alte familii de PLC-uri produse de firma Siemens există programe dedicate.

Exemple:

- S100 → Step 5;
- S200 → Micro Win32;
- Logo-uri → Logo Soft Comfort vx.0;

### *2.5.2. Crearea unui proiect Step 7*

STEP 7 este pachetul software standard pentru configurarea și programarea controllerelor programabile SIMATIC. Este un mediu de programare complex alcătuit din mai multe module.

Fereastra principală este *SIMATIC Manager*, care devine activă când STEP 7 este pornit. În setarea standard se pornește STEP 7 Wizard, care oferă suport pentru crearea proiectelor STEP 7.

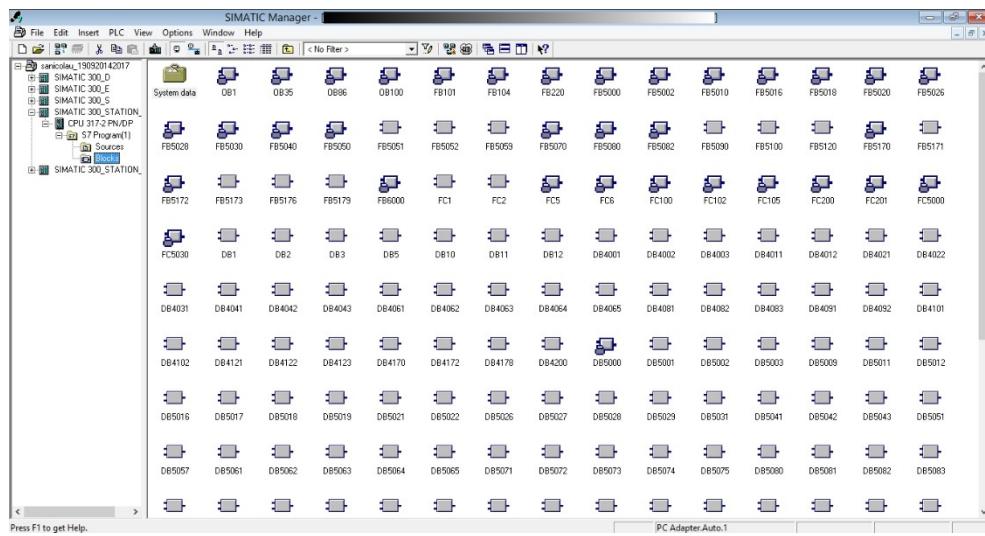


Cum se lansează mediul Simatic Manager?

Pentru lansarea mediului Simatic Manager parcurgeți secvența din Windows: *Start-> Programs(All Programs) -> Simatic -> Simatic Manager*

În figura este prezentată fereastra principală a mediului de dezvoltare *Simatic manager*.

## Lucrarea 2



**Figura 2.10.** Fereastra principală a mediului *Simatic Manager*

Un proiect este un set de obiecte asociate cu o aplicație.

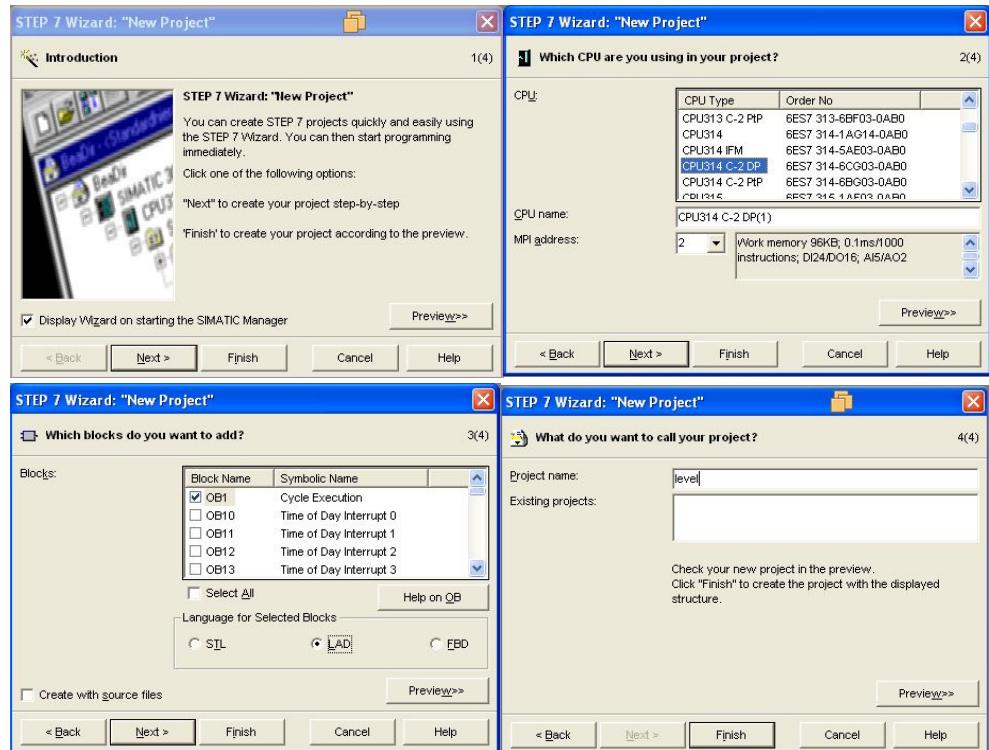
Pașii necesari creării unui proiect sunt prezentati în continuare.

- Dacă wizard-ul nu se pornește automat atunci selectați *New Project Wizard...* din meniul *File* (figura 2.11 – 1)
- Selectați tipul de procesor pentru care doriți să creați proiectul (figura 2.11 – 2). Se selectează modelul corespunzător de CPU în câmpul *CPU Type*, pentru exemplul prezentat se va alege *CPU314C-2DP*, apoi se atribuie numele CPU-ului în câmpul *CPU name*. *MPI address* rămâne neschimbată. Se activează butonul *Next* pentru a confirma setările și a ajunge la următoarea fereastră de dialog.
- Sectați blocurile care vor fi adăugate proiectului și limbajul de programare (figura 2.11 – 3). Se selectează blocul de organizare *OB1*, dacă acesta nu este deja selectat. *OB1* reprezintă cel mai înalt nivel de programare și organizează toate celelalte blocuri din programul STEP 7. Apoi se selectează unul dintre următoarele limbaje de programare: *Ladder Logic (LAD)*, *Statement List (STL)* sau *Function Block Diagram (FBD)*. Se alege *Ladder Logic* apoi

## Lucrarea 2

se apasă butonul Next. Limbajul de programare poate să fie schimbat și ulterior.

- Dați un nume proiectului. La Project name se atribuie proiectului numele dorit, apoi se apasă pe butonul Finish care va genera noul proiect. Proiectul va fi salvat la adresa ... \Program Files\Siemens\Step7\S7Proj (figura 2.11 – 4)



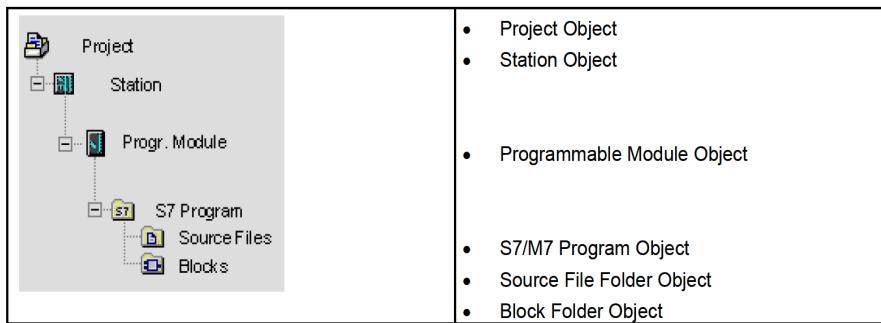
**Figura 2.11.** Pașii pentru crearea unui proiect în *Simatic Manager*

### 2.5.3. Posibilități de navigare prin structura proiectului

Proiectul nou creat este afișat împreună cu stația S7 selectată și CPU. Partea superioară a ierarhiei proiectului este structurată astfel:

1. Primul nivel: Proiectul
2. Nivelul doi: Subnoduri, stații, sau programe S7/M7
3. Nivelul trei: Depinde de obiectele de la nivelul 2

## Lucrarea 2



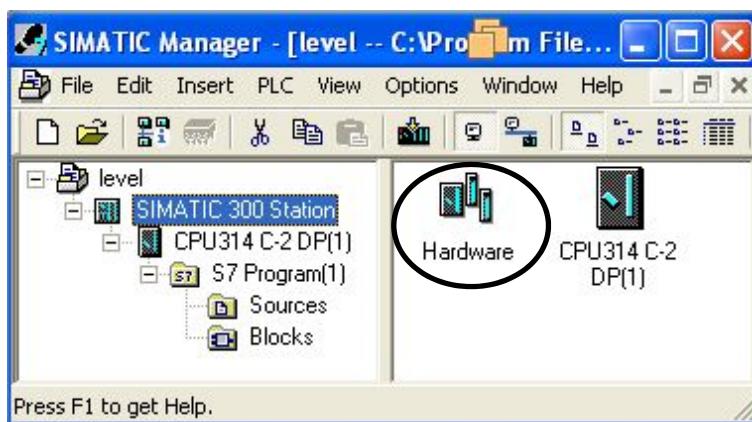
**Figura 2.12.** Structura proiectului

Dacă activăm folderul *S7 Program*, în fereastra din stânga vor apărea 3 foldere: *Sources*, *Blocks* și *Symbols*. Componenta *Symbols* va fi folosită pentru implementarea tabelei de simboluri, *Sources* este folosită pentru a stoca fișierele sursă ale programelor. Folderul *Blocks* conține momentan fișierul *System Data* și fișierul OB1 deja creat iar mai târziu va conține toate celelalte block-uri.



Cum se realizează configurarea hardware a automatului programabil?

Pentru orice proiect dezvoltat în STEP 7 stabilirea configurației hard este primul pas care trebuie făcut. Accesați modulul *Hardware Configuration* din *Simatic Manager* de la secțiunea *Simatic 300 Station* (figura 2.13).



**Figura 2.13.** Accesare Hardware Configuration

Din librăria de componenete, pe primul slot al configurației se adaugă sursa de alimentare. Apoi se verifică dacă numărul de ordine și versiunea procesorului corespund cu cele înscrise pe PLC. Dacă acestea diferă se adaugă un alt procesor din librerie, care va corespunde cu cel aflat pe stand.

În figura 2.14 se poate observa realizarea configurației hardware. În partea din dreapta a ecranului se găsesc librăriile cu componentele necesare configurației. În partea din stânga vor fi afișate două tabele cu componente hardware selectate din librării. CPU-ul care a fost selectat la crearea proiectului este afișat în ferestrele din stânga, în cazul nostru acesta este CPU 314C-2DP. Fiind un CPU care are conține și intrări/ieșiri analogice/numerice, vor apărea și acestea în cele două tabele.

Pentru a realiza configurarea, primul lucru de care avem nevoie este o ţină, „rack”, pe care vor fi atașate toate componente. Aceasta, ca de altfel toate celelalte componente, depind de stația SIMATIC folosită. Toate aceste componente diferă între familiile de automate (SIMATIC 300, SIMATIC 400). În cazul nostru, avem de a face cu un PLC din gama SIMATIC 300, și deci toate componentele se vor selecta din librăria corespunzătoare. Așa cum apare specificat în ferestrele din partea stângă a ecranului vom folosi o singură ţină care are alocat numărul 0. În slotul 2 este fixat CPU-ul cu modulele de intrare/ieșire. Numărul slotului apare scris și pe panoul frontal al CPU, o cifră în partea stângă jos, deasupra codului CPU-ului.

Este posibil ca slotul 1 să nu fie folosit, în cazul în care se dispune de o sursă de alimentare externă. Pentru PLC-urile care dispun de sursă de alimentare proprie, pe primul slot se introduce un modul de sursă de alimentare. Astfel vom căuta prin librăriile aflate în partea dreapta până găsim sursa de alimentare folosită (în cazul nostru PS307 2A) pe care o luăm cu drag and drop și o punem în slotul 1 (sau activăm slotul 1 și apoi dublu click pe sursa aleasă). Verificăm apoi dacă, codul înscris pe panoul frontal al sursei, în partea stângă jos, corespunde cu cel din tabelul de jos din coloana *Order number*. Va trebui să alegem sursa pentru care corespunde acest cod.

## Lucrarea 2

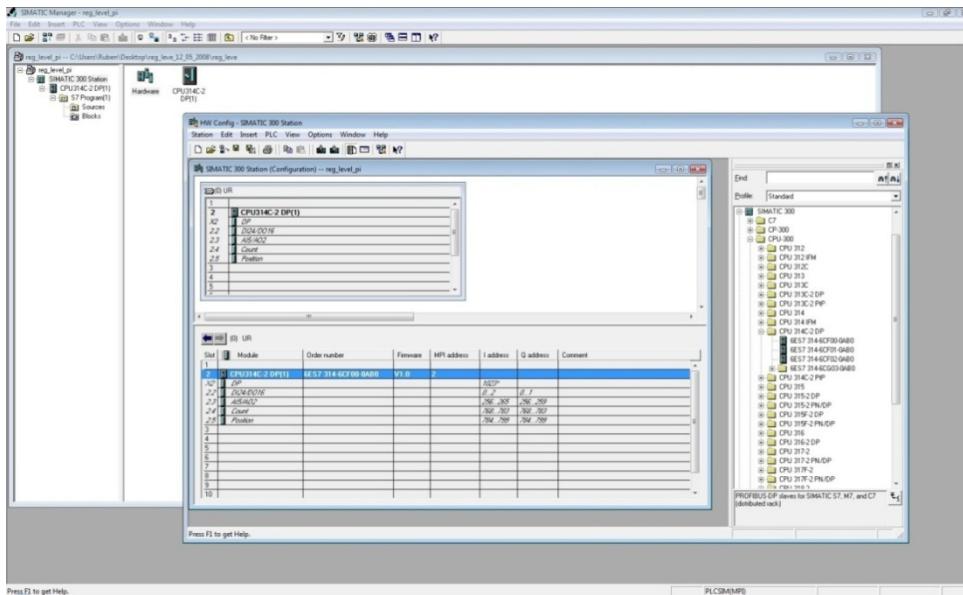


Figura 2.14. Realizarea configurației hardware

Urmează apoi să verificăm corespondența între codul înscris pe CPU (pe panoul frontal în partea stângă jos) și cel din coloana *Order number* tabelul de jos. Trebuie verificată deasemenea și versiunea, înscrisă în coloana *Firmware*, ce se va compara cu versiunea CPU-ului, care se poate citi ridicând capacul de pe panoul frontal ce acoperă conectorii de comunicație (în cazul nostru v2.6). Dacă seria și/sau versiunea CPU nu corespund cu cele înscrise pe panoul frontal al CPU atunci se va șterge CPU-ul adăugat implicit și se va înlocui cu cel corespunzător din libraria de componente, pe care îl adaugăm fie cu drag and drop și îl punem în slotul 1, fie activăm slotul 1 și apoi se dă dublu click în librerie, pe CPU-ul aleas.



Cum se configurează intrările și ieșirile PLC-ului?

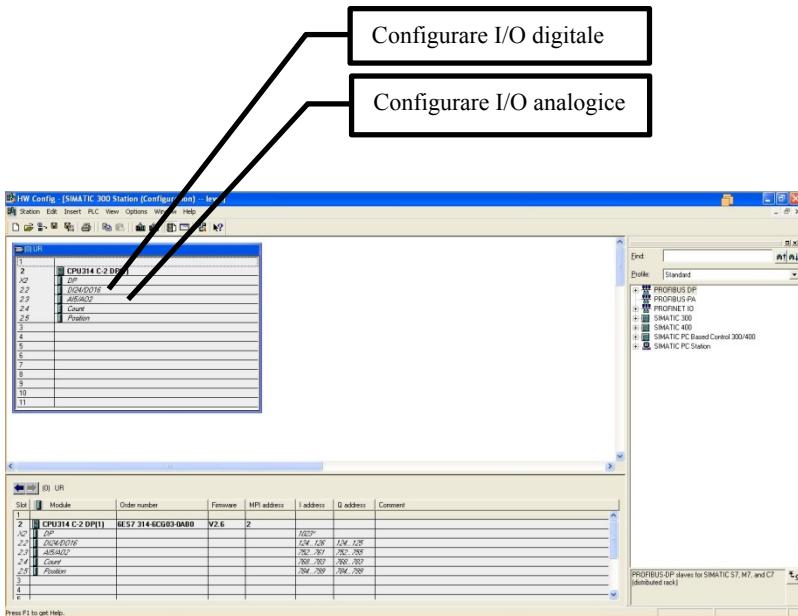
După alegerea CPU, se vor configura modulele de intrare/ieșire corespunzătoare. În cazul nostru este vorba despre un modul cu 24 de intrări digitale, 16 ieșiri digitale (DI24/DO16), 5 intrări analogice și respectiv 2 ieșiri analogice (AI5/AO2). Accesați secțiunea 2.2 (DI24/DO16) pentru

## Lucrarea 2

configurarea intrărilor și ieșirilor digitale și secțiunea 2.3 (AI5/AO2) pentru configurarea intrărilor și ieșirilor analogice (figura 2.15).

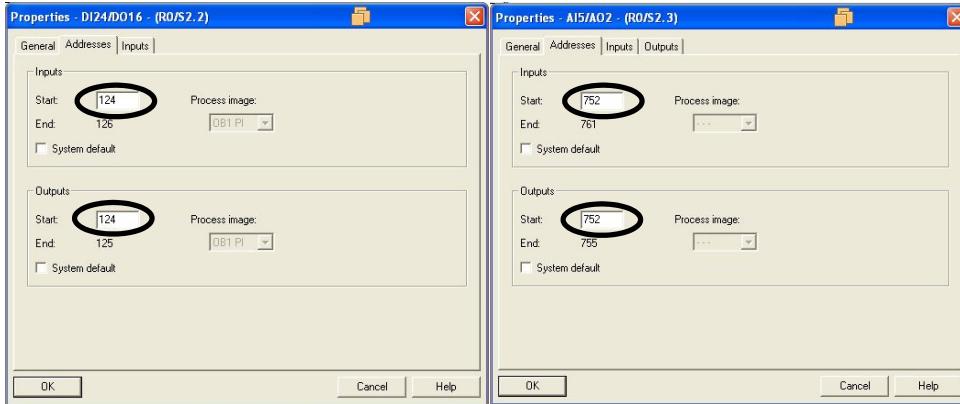
Adresele modulelor analogice și digitale pot fi configurate după dorință, dar respectând condiția ca ele să nu se suprapună. Modificarea adreselor de I/O (figura 2.16) se poate face debifând *System default* și scriind la *Start* adresa corespunzătoare primei intrări. Aici există posibilitatea de a comuta între o intrare/ieșire pe tensiune sau curent, dar nu este cazul aplicației curente.

- ✓ Intrările digitale vor începe de la 0 și se vor termina la 2: I0.0...I0.7, I1.0...I1.7, I2.0...I2.7
- ✓ Ieșirile digitale vor începe de la 0 și se vor termina la 1: Q0.0...Q0.7, Q1.0...Q1.7
- ✓ Intrările analogice vor începe de la 256 și se vor termina la 265: PIW 256...PIW 265
- ✓ Ieșirile analogice vor începe de la 256 și se vor termina la 259: PQW 256...PQW 259



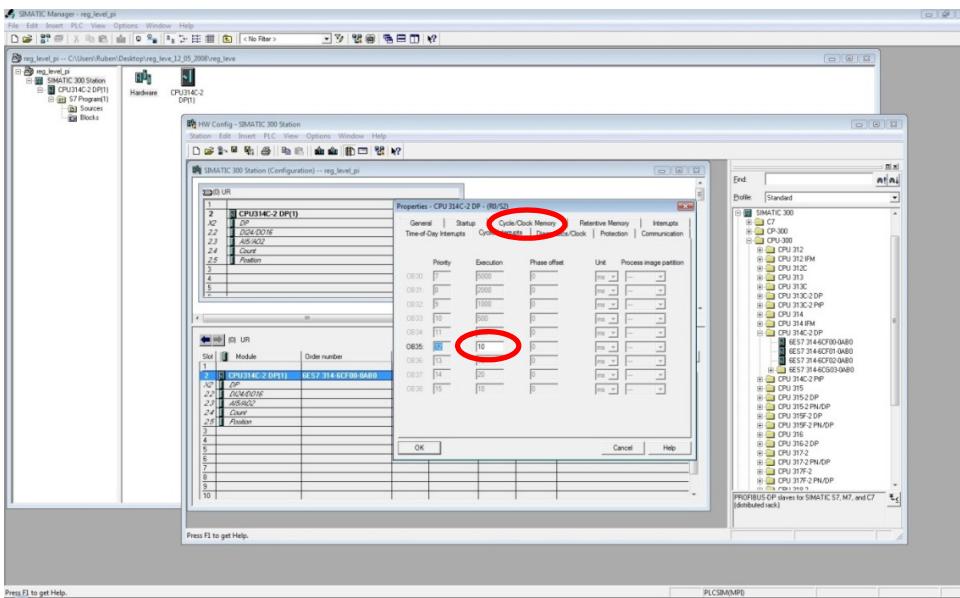
**Figura 2.15.** Accesare configurare adrese I/O

## Lucrarea 2



**Figura 2.16.** Modificare adrese intrare și ieșire

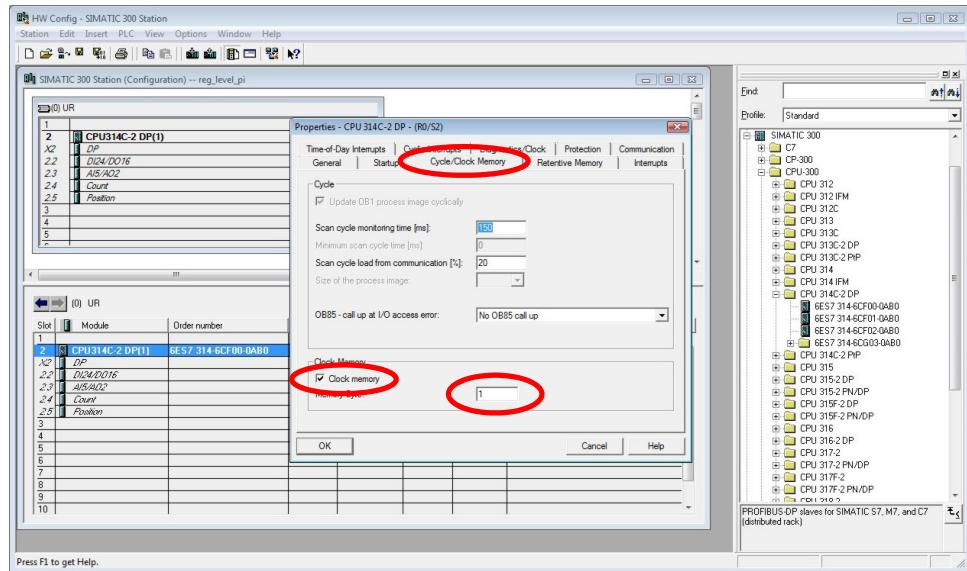
Se poate modifica și timpul de întârziere pentru întârzierea ciclică OB35, singura care este disponibilă pentru SIMATIC 300 CPU 314C-2 DP. Implicit acest timp este de 100 ms. Pentru modificare se va dă dublu click pe CPU și se va alege tab-ul Cyclic Interrupts. Câmpul *Execution* va fi setat la valoarea dorită în milisecunde.



**Figura 2.17.** Modificarea timpului pentru întârzierea ciclică OB35

## Lucrarea 2

Este permisă setarea unui octet de memorie pentru ceas. Pentru acesta se va da dublu click pe CPU și se va alege tab-ul *Cycle/Clock Memory*. Se va activa obținerea *Clock memory* și se va scrie în câmpul *Memory Byte* valoarea 1.



**Figura 2.18.** Setarea octetului de memorie pentru ceas

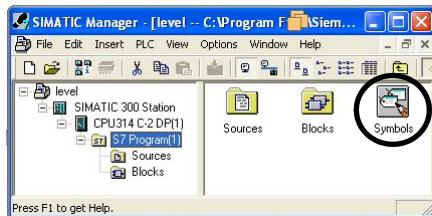
Dacă mai sunt și alte module disponibile (de exemplu module de comunicație) vor fi și acestea adăugate în configurația hardware.

Astfel, datele sunt pregătite pentru a fi transferate către CPU folosind comanda *Save and Compile* din meniul *Station*. STEP 7 va genera posibile soluții pentru orice erori ce ar putea să apară. Configurarea facută se poate verifica pentru erori folosind comanda *Consistency Check* din meniul *Station*. După ce aplicația *HW Config* a fost închisă simbolul *System Data* va apărea în folderul *Blocks*.



## Cum se editează simbolurile?

Tabelul de simboluri se accesează din *S7 Program -> Symbols*.



**Figura 2.19.** Accesare editor simboluri

Aici se asociază nume simbolic și tip de date adreselor absolute care vor fi utilizate ulterior în program (figura 2.20).

Status	Symbol /	Address	Data type	Comment
1	CONT_C	FB 41	FB 41	Continuous Control
2	CYC_INTERRUPTS	OB 35	OB 35	Cyclic Interrupt S
3	Cycle Execution	OB 1	OB 1	
4	Input	PWM 256	WORD	System Input Signal from level sensor
5	Output	PWM 256	WORD	System Output Signal (process command)
6				

**Figura 2.20.** Symbol Editor

## Adresarea absolută și simbolică

Într-un program STEP 7 se lucrează cu adrese, cum ar fi semnale I/O, bit de memorie, numărătoare, timere, blocuri de date, și funcții bloc.

## Lucrarea 2

Sunt permise două moduri de adresare:

✓ *Adresarea absolută*

- ⊕ O adresă absolută este alcătuită dintr-un identificator și o locație de memorie (de exemplu: Q 4.0, I 1.1, M 2.0, FB80).

✓ *Adresarea simbolică*

- ⊕ Programul este mai ușor de înțeles și se simplifică rezolvarea problemelor de depanare dacă se atribuie nume simbolice adreselor absolute. Astfel, o adresă din program poate fi accesată prin intermediul unui simbol.

STEP 7 poate interpreta numele simbolice în adresele absolute cerute în mod automat.

Se poate, de exemplu, atribui numele simbolic MOTOR\_ON adresei Q 1.0 și atunci se folosește MOTOR\_ON ca o adresă în program. Folosind adresarea simbolică este mai ușor de recunoscut în ce măsură elementele din program se potrivesc cu componentele proiectului.

### *Adrese și tipuri de date permise în Symbolic Table*

IEC	Descriere	Tip de date	Domeniu de adrese
I	Input bit	BOOL	0.0 la 65535.7
IB	Input byte	BYTE, CHAR	0 la 65535
IW	Input word	WORD, INT, S5TIME, DATE	0 la 65534
ID	Input double word	DWORD, DINT, REAL, TOD, TIME	0 la 65532
Q	Output bit	BOOL	0.0 la 65535.7
QB	Output byte	BYTE, CHAR	0 la 65535
QQW	Output word	WORD, INT, S5TIME, DATE	0 la 65534
QD	Output double word	DWORD, DINT, REAL, TOD, TIME	0 la 65532
M	Memory bit	BOOL	0.0 la 65535.7
MB	Memory byte	BYTE, CHAR	0 la 65535

MW	Memory word	WORD, INT, S5TIME, DATE	0 la 65534
MD	Memory double word	DWORD, DINT, REAL, TOD, TIME	0 la 65532
PIB	Peripheral input byte	BYTE, CHAR	0 la 65535
PQB	Peripheral output byte	BYTE, CHAR	0 la 65535
PIW	Peripheral input word	WORD, INT, S5TIME, DATE	0 la 65534
PQW	Peripheral output word	WORD, INT, S5TIME, DATE	0 la 65534
PID	Peripheral input double word	DWORD, DINT, REAL, TOD, TIME	0 la 65532
PQD	Peripheral output double word	DWORD, DINT, REAL, TOD, TIME	0 to 65532
T	Timer	TIMER	0 to 65535
C	Counter	COUNTER	0 to 65535
FB	Function block	FB	0 to 65535
OB	Organization block	OB	1 to 65535
DB	Data block	DB, FB, SFB, UDT	1 to 65535
FC	Function	FC	0 to 65535
SFB	System function block	SFB	0 to 65535
SFC	System function	SFC	0 to 65535
VAT	Variable table		0 to 65535
UDT	User-defined data type	UDT	0 to 65535

**Tabelul 2.4.** Tipuri de date acceptate în Step 7



Cum se crează o funcție?

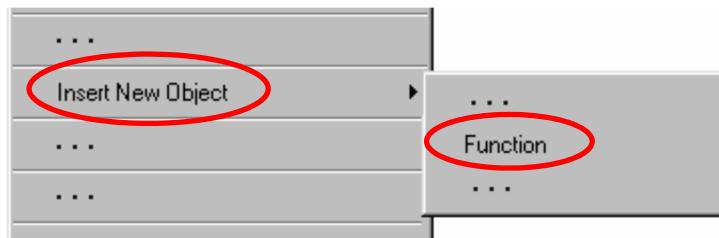
STEP 7 pune la dispoziție mai multe structuri de programare:

- ✚ Organization Block (OB)
  - ✚ Function (FC)
  - ✚ Function Block (FB)
  - ✚ Data Block (DB)
- ✓ În contrast cu FB-uri în FC-uri nu pot fi declarate variabile statice

## Lucrarea 2

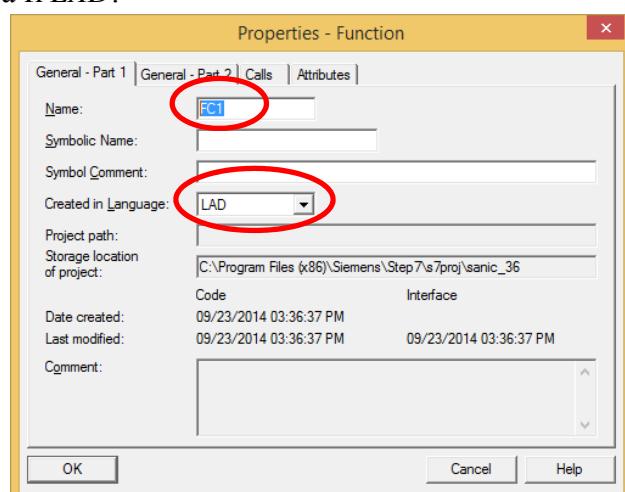
- ✓ Variabilele statice declarate în funcțiile bloc (FB) se păstrează când blocul este închis
- ✓ FB și FC sunt apelete în cadrul blocurilor de organizare (OB)

Inserarea unei funcții se poate realiza fie prin folosirea meniului contextual (figura 2.21), fie din meniul *Insert* al mediului de programare *Simatic Manager*. Mergeți la meniul *Insert* din *Simatic Manager* și selectați *S7 Block* și de aici *Function*.



**Figura 2.21.** Inserarea unei funcții din meniul pop-up

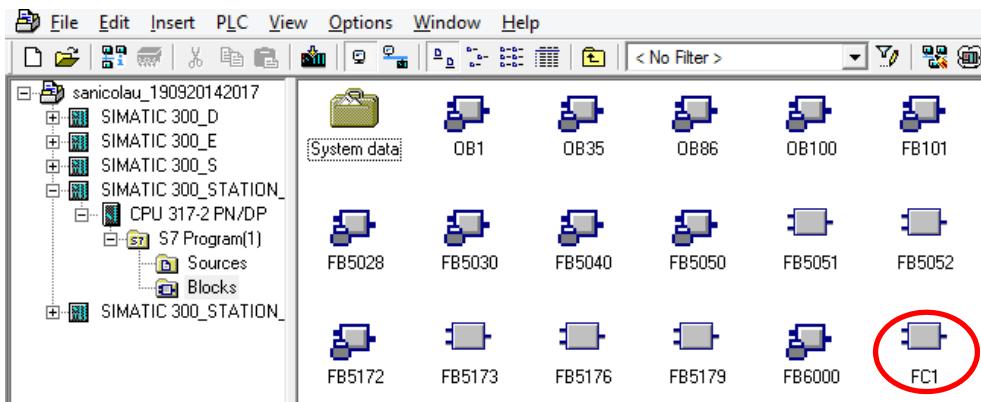
Ulterior inserării funcției sau la prima accesare a acestora se va deschide caseta de dialog *Properties – Function* unde pot fi modificate proprietățile funcției. Se stabilește numele funcției și se selectează limbajul de programare. Confirmarea cu *OK* va genera efectiv funcția dorită. Astfel, în secțiunea *Blocks* va fi adăugată o funcție cu numele specificat (ex. *FC1*). Limbajul de programare va fi *LAD*.



**Figura 2.22.** Proprietăți funcție FC1

## Lucrarea 2

Funcția FC1 este adăugată în directorul *Bolcks*. Se poate deschide cu dublu click.



**Figura 2.23.** Funcția FC1

Funcțiile în *Step 7* sunt blocuri logice „fără memorie”. Variabilele temporare care aparțin funcției sunt salvate în stiva locală de date. Aceste date se pierd atunci când funcția a fost executată. Pentru a salva permanent datele, funcțiile pot folosi blocuri de date partajate.



Care este blocul pentru control PID?

### Control continuu cu FB41 “CONT\_C”

FB41 este o funcție de control continuu implementată de STEP 7. Această funcție este utilizată de PLC-urile SIMATIC S7 pentru a controla procesele tehnice cu intrări continue și ieșiri variabile. În timpul atribuirii parametrilor se pot activa sau dezactiva subfuncții ale regulatorului PID pentru o mai bună adaptare a acestuia la proces.

Pentru configurarea parametrilor regulatorului se poate folosi blocul DB41. Acesta permite, pe lângă setarea parametrilor regulatorului, folosirea câtorva opțiuni de control a blocului regulator precum și monitorizarea grafică a parametrilor regulatorului.

Funcțiile regulatorului se bazează pe algoritmul de control PID al unui regulator cu semnal analogic. Efectele proporțional, integrator și derivativ

## Lucrarea 2

sunt conectate în paralel, iar acest lucru permite configurarea de regulatoare P, PI, PD, PID. Pot fi implementate și regulatoare pure I sau D. În figura 3.9 este prezentată schema bloc a regulatorului. În aceasta se pot observa toate mărimile de intrare și ieșire precum și funcțiile suplimentare pe care acest bloc le conține. În aplicația curentă funcția FB41 este apelată pentru implementarea unui regulator PI, cu observația că se trece comanda pe manual în cazul în care se dorește realizarea identificării sistemului. În acest scop s-au utilizat următoarele intrări: variabila de intrare (PV\_IN), valoarea presetată (SP\_INT), selectorul pentru control automat MAN\_ON (are valoare FALSE în cazul rulării pentru monitorizare). În regim automat, respectiv valoarea TRUE în cazul în care se urmărește realizarea identificării sistemului), intrarea MAN (pe care se aplică o treaptă de identificare), valoarea parametrului proporțional (GAIN), valoarea parametrului integrator (TI) și timpul dintre apelurile successive ale blocului (perioada de eșantionare).

Blocul FB41 implementează regulatorul a cărui structură este prezentată în figura 3.5. Se va urmări pe schema din figura 3.5 semnificația parametrilor de mai jos.

Principalii parametri utilizați, precum și intervalul de valori al acestora, sunt prezențați în cele ce urmează:

SP\_INT – referința procesului (în intervalul -100...100)

PV\_IN – măsura de la proces exprimată în procente (în intervalul -100...100)

PV\_PER – măsura de la proces exprimată în decimal (în intervalul -27648...27648)

GAIN – factorul de proporționalitate al regulatorului

TI – constanta de integrare a regulatorului (în secunde)

TD – constanta de derivare a regulatorului (în secunde)

LMN – comanda regulatorului (în domeniul 0...100)

LMN\_PER – comanda regulatorului scalată (în domeniul 0...27648)

MAN – comanda manuală (în domeniul 0...100)

PVPER\_ON – comutator măsură proces internă/periferică

MAN\_ON – comutator manual (bucla de control este întreruptă)/automat (bucla de control este activă)

## Lucrarea 2

CYCLE – timpul dintre apelurile succesive ale blocului regulatorului (egal cu 100 ms în mod standard)

Ieșirea procesului poate fi conectată pe 2 intrări, PV\_PER sau PV\_IN. Se utilizează PV\_PER dacă se dorește ca blocul FB 41 să facă sclarea intrării de la valoarea în decimal, disponibilă în urma conversiei analog-numerice, la valoarea în procente (necesară calculelor interne din blocul FB41) și respectiv PV\_IN dacă scalarea se va face separat în programul de aplicație.

Măsura din proces PV\_PER este scalată pentru a intra în domeniul de lucru al regulatorului.

$$CRP\_IN = PV\_PER \cdot 100/27648$$

$$PV\_PER \in [0 \dots 27648]$$

$$PV\_IN \in [0 \dots 100]$$

Uneori este necesară amplificarea și deplasarea semnalului primit din preces. Acest lucru se poate realiza în blocul FB41 prin setarea parametrilor PV\_fac și PV\_OFF, astfel:

$$\text{Ieșirea din PV_NORM} = (\text{ieșirea din CRP_IN}) * PV\_FAC + PV\_OFF$$

Implicit PV\_FAC are valoarea 1 și PV\_OFF are valoarea 0.

Comanda regulatorului LMN poate să fie scalată înainte de a se aplica elementului de execuție. Dacă este necesară, scalarea se face prin amplificarea semnalului și/sau deplasarea acestuia, după relația:

$$LMN = (\text{ieșirea din LMNLIMIT}) * LMN\_FAC + LMN\_OFF$$

Implicit LMN\_FAC are valoarea 1 și LMN\_OFF are valoarea 0.

$$LMN\_PER = LMN \cdot 27648/100$$

$$LMN\_PER \in [0 \dots 27648]$$

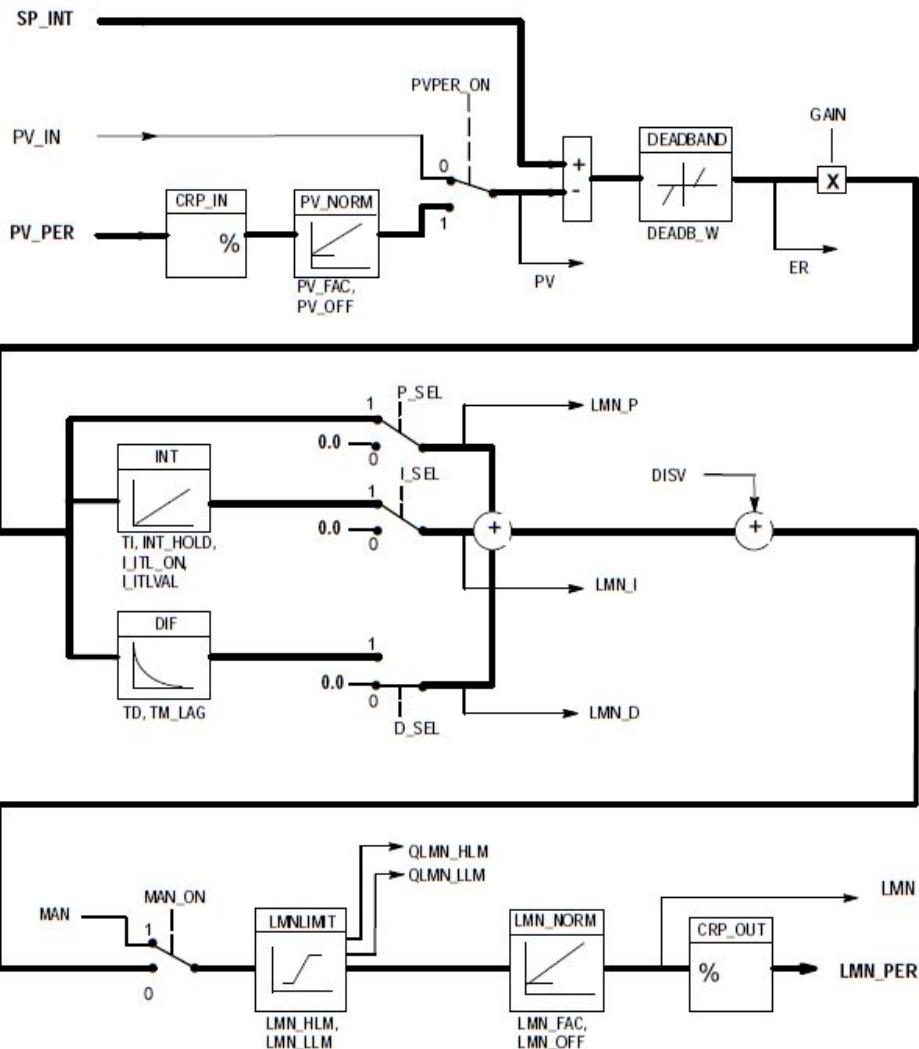
$$LMN \in [0 \dots 100]$$

Funcția de transfer a regulatorului este:

$$H_{PID}(s) = GAIN \left( 1 + \frac{1}{TI} \cdot \frac{1}{s} + TD \cdot s \cdot \frac{1}{TM\_LAG \cdot s + 1} \right)$$

**DB 41 – Bloc de date al regulatorului**

DB 41 este un bloc de date care este instantiat automat la apelarea functiei bloc FB41. În acest bloc se află toți parametrii regulatorului folosiți în funcția bloc FB41 și valorile corespunzătoare acestora.



**Figura 2.24.** Schema bloc FB41



Cum se adaugă proiectului blocul FB41?

Acet bloc se adaugă în blocul de organizare OB35 de la *Libraries -> Standard Library -> PID Control Blocks -> FB41 CONT\_C ICNT*.



Cum se asociază un bloc de date funcției bloc FB41?

Se va scrie adresa blocului de date în partea de sus a funcției bloc FB41. Asignare variabile de intrare și ieșire din proces se va face în blocul de organizare OB35.



Cum se configerează parametrii regulatorului?

Asignare parametriilor regulatorului se va face în blocul de date DB41,

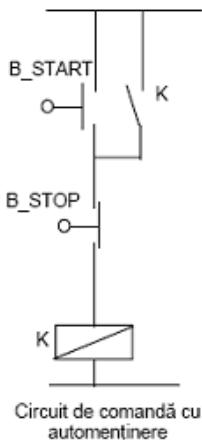


### 3. Desfășurarea lucrării

Standul utilizat pentru rezolvarea aplicațiilor este descris în detaliu în anexa 1.

#### 3.1. Releu cu automenținere

Pentru exemplificarea modului de lucru cu mediul Step 7 ne propunem realizarea unui program pentru comanda cu automenținere a unui releu. Programul are ca efect comanda unui releu prin apăsarea butoanelor B\_START (pentru anclansarea releului – închiderea contactelor N.O.) și B\_STOP (pentru dezanclansarea releului – întreruperea contactelor N.O.).



**Figura 2.25.** Circuit de comandă cu automenținere

Dezvoltarea programului pentru automatul Siemens 314C-2DP se va realiza în Simatic STEP 7 un mediu special dezvoltat de firma Siemens pentru programarea automatelor din familiile S300 și S400.

#### *Crearea tableei de simboluri*

Tabela de simboluri poate fi accesată de la nivelul *S7 Program* din *SIMATIC Manager* dând dublu click pe pictograma *Symbols*.

În tabela de simboluri se pot da nume și tipul de date adreselor absolute care vor putea fi folosite apoi în program. Aceste nume se pot folosi în toate părțile programului și se numesc variabile globale. Pentru aplicația propusă vom avea:

Symbol	Address	Data Type	Comment
Buton_1	I 1.0	BOOL	START
Buton_2	I 1.1	BOOL	STOP
Iesire	Q 0.0	BOOL	RELEU

**Tabelul 2.5.** Tabel simboluri

Pentru început, tabela de simboluri conține doar blocul de organizare predefinit OB1. Pentru exemplul nostru vom înlocui *Cycle Execution* cu *Main Program*. Dacă nu apare atunci se va adăuga astfel: la coloana *Symbols* se scrie *Main Program* și la coloana *Adresses* se scrie OB1. Poate fi adăugat și un comentariu. Apoi se va realiza tabela de simboluri de mai sus. În acest fel

## Lucrarea 2

---

se pot atribui denumiri simbolice tuturor adreselor absolute ale intrărilor și ieșirilor necesare în program. La final se salvează folosind opțiunea *Save* din meniu *File*.

### *Programarea blocului OB1 în Ladder Logic*

Vom programa acum circuitul START/STOP în Ladder Logic. În secțiunea *Blocks* activăm OB1. Va apărea o nouă fereastră. Dacă nu a fost selectat corect limbajul de programare, acesta se mai poate selecta de la View, în cazul nostru vom alege **LAD**.

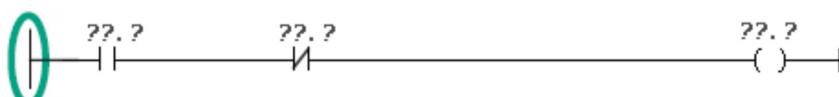
Dăm click în secțiunea de titlu a lui OB1 și scriem *Programul principal*.

Folosind limbajul de programare structurată se selectează cu mouse-ul icoanele de contact (normal închis sau normal deschis), releu sau ramificație și se inserează în spațiul de lucru unde se realizează diagrama structurată după logica impusă de aplicație. Selectăm treapta de program curentă pentru primul element.



Apăsăm butonul ce se află în toolbar și introducem un contact normal deschis , apoi un contact normal închis și un releu (coil – bobină) care se va plasa în partea dreaptă a diagramei.

Am realizat prima ramură din treapta de program. Pentru a realiza a doua ramură, în paralel cu prima, selectăm bara din stânga treptei de program și selectăm apoi începutul de ramificație . Introducem un contact normal deschis și sfârșitul ramificației .



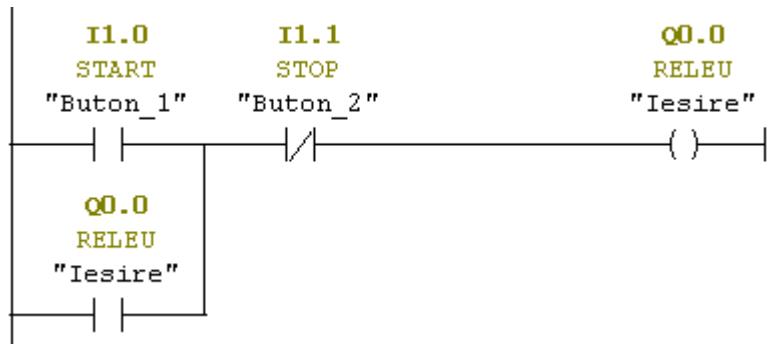
Din meniu *Options*, submeniu *Customize* și apoi *View* verificăm dacă *Symbolic Representation* și *Block/network comments* sunt activate.

## Lucrarea 2

Apăsăm semnul “??..?” corespunzător primului element al treptei și trecem numele simbolic “*Buton\_1*” pentru primul contact deschis. După introducerea primului caracter al numelui simbolului apare pe ecran o fereastră ajutătoare cu simbolurile definite în tabela de simboluri. Se va alege cu dublu clic stânga simbolul dorit. Se procedează la fel pentru toate elementele din treapta de program.

Dacă un simbol este scris cu roșu înseamnă că acesta nu există în tabela de simboluri sau există o eroare de sintaxă.

Se poate da un nume treptei (Network1:) și se poate insera un comentariu referitor la treapta respectivă.



### *Testarea programului*

#### *1. Offline*

Testarea offline a programului realizat se poate face utilizând simulatorul S7-PLCSIM. Prin deschiderea acestuia și realizarea unui download se permite încarcarea programului ca și cum am avea un automat programabil atașat, având astfel posibilitatea de a simula funcționarea programului.

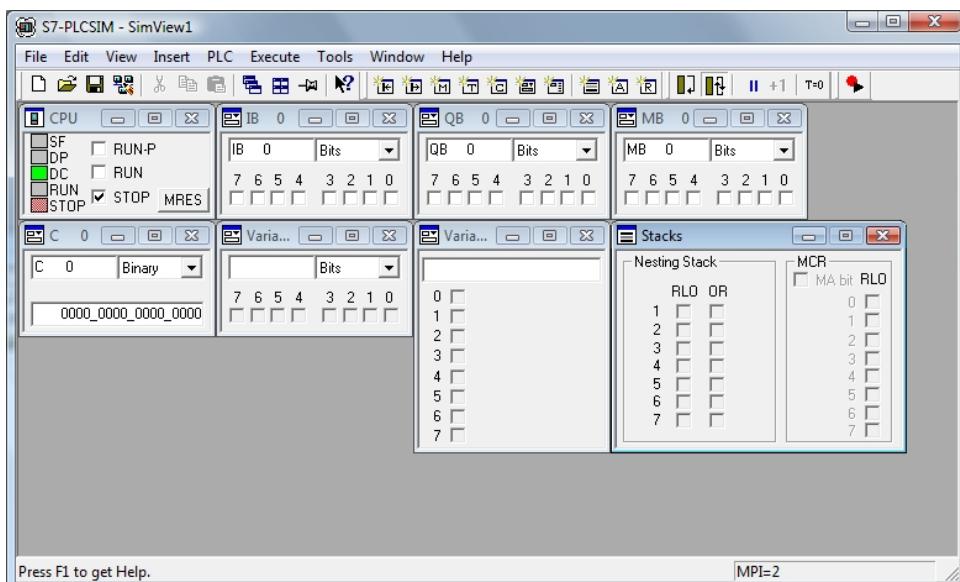
Simulatorul este disponibil în SIMATIC Manager, meniul *Options*, *Simulate Modules*. După activarea acestei opțiuni apare o fereastră cu S7-PLCSIM și fereastra de dialog *Open Project*. Se selectează *Select CPU access node* și *OK*. S7-PLCSIM va afișa o nouă fereastră de dialog *Select CPU access node* și denumirea proiectului deschis în STEP 7. Se va selecta *MPI* și apoi *OK*. Din SIMATIC Manager se activează folderul *Blocks* și descarcă

## Lucrarea 2

apoi programul activând *Download*  . În S7-PLCSIM se deschide fereastra de simulare (figura 16). Modul de lucru cu simulatorul este prezentat în fișierul „s7wsvhdb.pdf” sau „S7-PLCSIM - Testing Your S7-CPU Programs - manual.pdf”

Din bara de instrumente se adaugă o intrare și o ieșire în format *Bits*.

Avem nevoie de IB 1 pentru intrări și QB 0 pentru ieșiri. Acum se poate simula funcționarea aplicației.



**Figura 2.26.** Modulul de simulare S7-PLCSIM

### 2. *Online*

Pentru lucrul în mod online trebuie închis modul offline (PLCSIM).

Se încarcă programul în CPU prin apasarea butonului *Download*  din bara de meniu sau se poate folosi scurtătura Ctrl+L.

La efectuarea unor modificări într-unul din block-urile programului se poate face download pe CPU numai pentru acel bloc (prin pozitionarea pe bloc și apăsarea butonului *Download* ).

Aceeași funcționalitate a aplicatiei se poate obține folosind elementul SR (SET-RESET) din secțiunea *Bit Logic* a ferestrei *Program elements*.

În cazul în care după încărcarea configurației hardware sau software în automat, acesta are ledul de eroare SF sau BF activat (roșu), înseamnă că un pas din configurație nu a fost făcut corect. Se recomandă verificarea stării automatului. Pentru o funcționare corectă, automatul trebuie să fie în starea *RUN*, și nu *STOP*. Încărcarea unei noi versiuni a configurației hardware poare conduce la disparația erorii.

Cu toate acestea, dacă CPU-ul rămâne blocat în starea *STOP*, sau dacă eroarea din CPU-ul automatului nu a dispărut după comutarea procesoarului în *RUN*, se recomandă ștergerea blocurilor încărcate în automat și/sau resetarea memoriei PLC-ului.

Ștergerea blocurilor încărcate în automat, se realizează doar din modul online al aplicației (meniul *View -> Online* din Simatic Manager), prin conectarea pe cardul de memorie al automatului. La ștergerea blocurilor de instrucțiuni din stațiile de lucru, trebuie avut în vedere că blocurile de sistem nu trebuie șterse. Se selectează blocurile încărcate în cardul automatului, în afară de cele de sistem, după care se apasă tasta delete pentru ștergere. Chiar și după ștergerea blocurilor de instrucțiuni, automatul nu va rămâne fără erori, pentru acest lucru fiind necesară resetarea automatului și reîncărcarea în automat a blocurilor de instrucțiuni.

Resetarea memoriei se face fie prin accesarea opțiunii *Clear/Reset...* din meniul *PLC -> Diagnostic/Setting* al mediului Simatic Manager, fie folosind switch-ul MRES (poziținat pe partea frontală a CPU). Resetarea din switch se face astfel: 1. Se apasă și se ține apasat pe poziția MRES până când led-ul STOP iluminează continuu (aproximativ 9 secunde); 2. În următoarele 3 secunde, switch-ul trebuie eliberat și apoi dus iar pe poziția MRES. Acum led-ul STOP va clipi, ceea ce înseamnă că se relizează procedura de ștergere; 3. Se eliberează switch-ul.

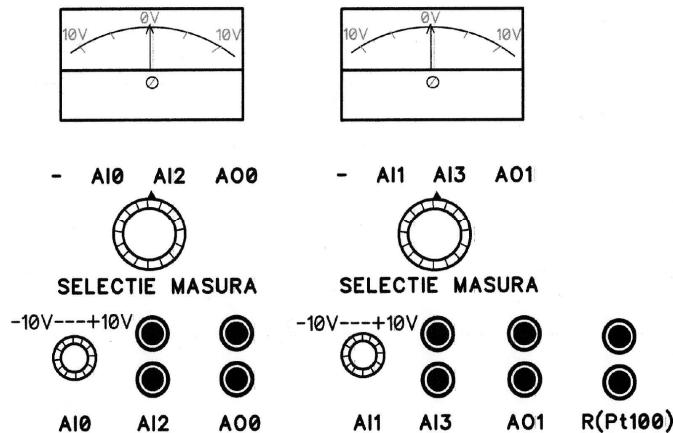
În urma resetării, orice DB care a fost încărcat în memoria automatului, va fi resetat la valorile implicate.

### **3.2.Citirea și afișarea unei mărimi analogice**

Se va face citirea unei mărimi analogice de la panou și se va afișa la ieșire pe un indicator (figura 2.27). Citirea se va face de la canalul AI0 iar scrierea se va face la canalul AO1. Modificarea valorii tensiunii de intrare, în gama -10 ... +10 V, se va face prin intermediul unui potențiometru conectat la canalul AI0 (poziționat sub voltmetrul din stânga).

## Lucrarea 2

 **OBSERVAȚIE:** Unei mărimi analogice în gama: **-10 V ÷ +10 V** îi corespunde ca reprezentare în numeric (WORD sau INT) o valoare în gama **-27648 ÷ +27648**.



**Figura 2.27.** Schema părții analogice a panoului PLC

*Mod de lucru:*

1. Se definește tabela de simboluri care este accesibilă de la nivelul S7 Program în ierarhia aplicației.

Symbol	Address	Data Type	Comment
AI0	PIW 256	WORD	
AO1	PQW 258	WORD	

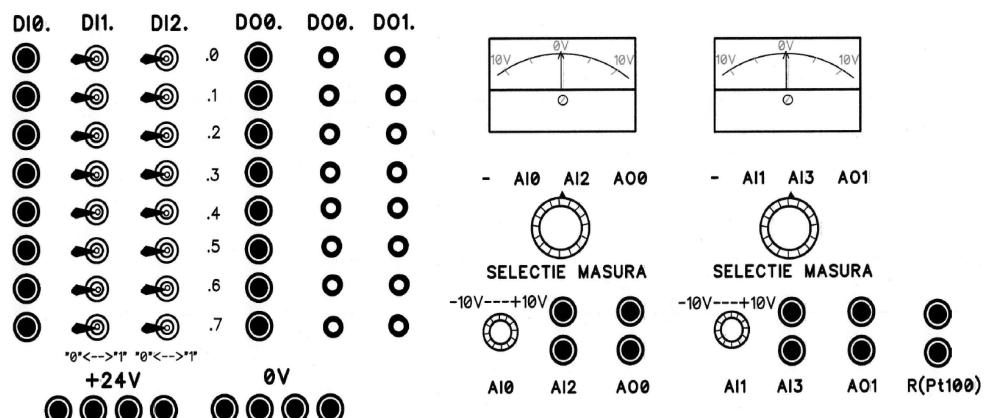
2. Accesând secțiunea Blocks a aplicației se va putea deschide blocul de organizare OB1 unde se va face programarea efectivă. OB1 oferă acces la o librărie de componente, denumită *Program Elements*, cu ajutorul cărora se va edita programul. Aceasta este poziționată în partea stângă a ferestrei delucru. Se folosește blocul *MOVE* din *Program Elements* pentru a înscrie valoarea analogică în memorie. Acesta va trebui poziționat pe prima rețea a programului prin tragere cu mouse-ul pe linia corespunzătoare primei rețele. Adaugarea blocului *MOVE* mai poate fi făcută și selectând rețeaua unde se

dorește a fi poziționat iar apoi se va da dublu-click pe bloc în librăria de componente.

- Blocul *MOVE* are o intrare și o ieșire. Intrarea va fi valoarea analogică de la adresa PIW256, iar ieșirea va fi la adresa PQW256. Astfel, vom scrie numele de variabile asociate anterior adreselor menționate, la intrare AI0 iar la ieșire AO0. Cu blocul *MOVE* se înscrie din memorie la ieșire valoarea analogică citită.

### 3.3.Compararea a două tensiuni

Se vor citi două tensiuni și se vor compara. Atâtă timp cât prima este mai mare sau egală cu a doua se va aprinde intermitent un LED. Dacă a doua este mai mare decât prima se va aprinde intermitent alt LED. Schema panoului se găsește în figura 2.28 și în anexa 8. Se vor folosi intrările AI0 și AI1 deoarece ambele au conțectate căte un potențiometru ce permite modificarea tensiunii de intrare pe canal în gama -10 ... + 10 V. Ieșirile pentru LED-uri vor fi DO 0.0, respectiv DO 0.1. Pentru comanda intermitentă a LED-urilor se va utiliza un bit din byte-ul de clock al procesorului.



**Figura 2.28.** Schema standului cu PLC Siemens S7 300

### **3.4.Regulator P**

Să se implementeze un regulator P și să se testeze funcționalitatea acestuia în buclă deschisă.

*Mod de lucru:*

1. Regulatorul se va realiza cu blocul FB41, care se va configura corespunzător
2. Blocul FB 41 va fi implementat în blocul de organizare OB35 (întrerupere ciclică)
3. Se va implementa un bloc DB, instanțiat pentru FB41, ce va permite configurarea parametrilor regulatorului
4. Se va face monitorizarea parametrilor regulatorului prin DB41.

## **4. Conținutul referatului**

Rezolvarea aplicațiilor de la secțiunea precedentă și redactarea unui referat care să prezinte soluțiile aplicațiilor (diagrame ladder).

## **3. Reglarea nivelului**



### **1. Scopul lucrării**

Obiectivul acestei lucrări de laborator este realizarea unei aplicații de monitorizare și control pentru nivelul de lichid dintr-un rezervor cu ajutorul unui PLC Siemens SIMATIC S7-300 și a unui PC. Lucrarea se va desfășura pe standul didactic FESTO Compact Workstation. Scopul lucrării este familiarizarea cu activitățile ce implică implementarea unui sistem pentru monitorizarea și controlul unui proces, având la dispoziție echipamente de lucru similare cu cele utilizate în practică, pentru conducerea proceselor industriale.



### **2. Considerații teoretice**

#### **2.1. Descriere FESTO Compact Workstation**

Stația de lucru FESTO combină patru bucle de control, fiecare cu senzori digitali și/sau analogici și actuatori. Cu ajutorul unui PLC este posibilă configurarea acestor bucle individual, în cascadă sau feedforward:

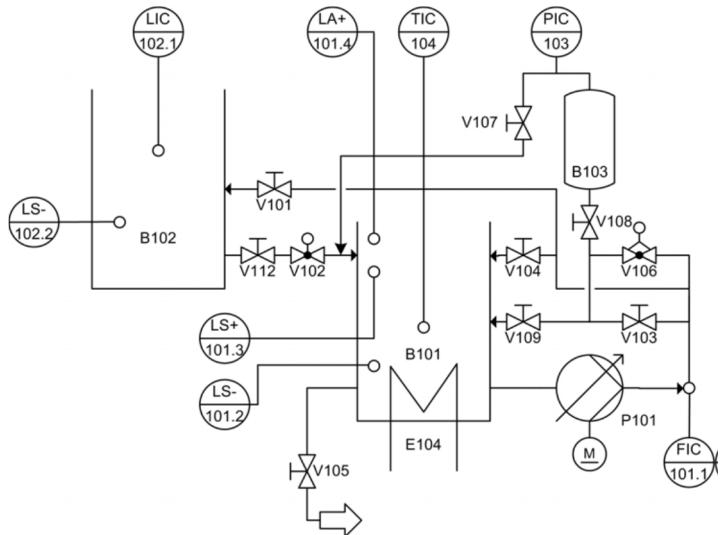
- Sistem de control automat al nivelului
- Sistem de control automat al debitului
- Sistem de control automat al temperaturii
- Sistem de control automat al presiunii

Schema generală a stației FESTO, cu cele patru bucle de control ce pot fi configurate, este prezentată în figura 3.1.

Principalele componente ale stației FESTO sunt:

- Senzor ultrasonic analogic pentru nivel (notat în figura 3.1 cu LIC 102.1)
- Senzor de debit cu semnal de ieșire în frecvență
- Senzor de debit cu semnal de ieșire în tensiune
- Senzor de presiune piezorezistiv

- Indicator presiune pentru 0...1 bar
- Senzor de temperatură PT100
- 2 senzori de capacitive de proximitate pentru nivel min/max în rezervorul de jos
- Comutator plutitor (electromecanic) pentru depăşire nivel în rezervorul de sus
- Comutator plutitor pentru monitorizarea alarmei de depăşire nivel maxim în rezervorul de jos
- Pompă centrifugă
- Controller motor pompă
- Valvă proporțională cu modul de control electronic
- Sistem de încălzire cu releu de control integrat
- Terminale I/O digitale
- Terminal pentru semnale analogice
- Convertor de semnale: din curent în tensiune, din frecvență în tensiune, PT100 în tensiune
- PLC Siemens SIMATIC S7-300 CPU 314C-2DP
- Panou de control
- Sistem de țevi de legatură
- Rezervor de presiune
- 2 rezervoare pentru apă
- Valve manuale



**Figura 3.1.** Schema generală a standului FESTO

Standul FESTO permite configurarea mai multor tipuri de procese, precum reglare de nivel, reglare de presiune, reglare de debit de intrare și reglare de temperatură. Configurarea sistemului se face prin modificarea poziției (închis/deschis) valvelor manuale.

În continuare vor fi descrise componentele stației FESTO utilizate pentru realizarea aplicației de reglare a nivelului.

## 2.2.Componentele stației FESTO

### 2.2.1. Descrierea instalației

Schema instalației pentru controlul nivelului este prezentată în figura 3.2.

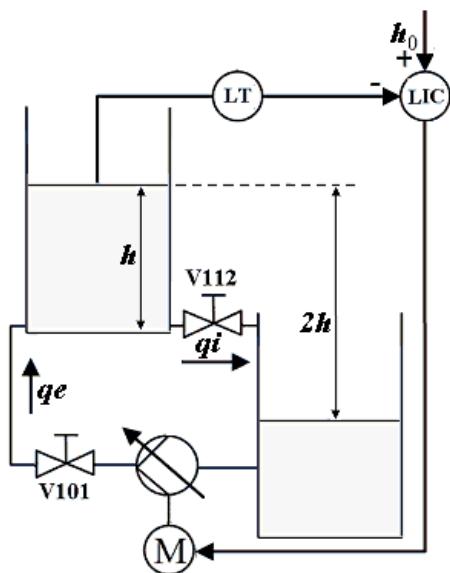
Sistemul conține două rezervoare de apă situate unul deasupra celuilalt.

Rezervorul pentru reglarea nivelului sus iar rezervorul tampon jos. Prin alegerea acestei soluții constructive sarcina pompei este reprezentată de distanța pe verticală dintre suprafața apei în rezervorul de sus și suprafața apei în cel de jos distanță egală cu de două ori înălțimea coloanei de apă în rezervorul principal. Pentru nivel zero în rezervorul principal, rezervorul tampon este plin suprafața apei în cele două rezervoare fiind aproximativ pe același nivel.

### Lucrarea 3

Aria secțiunii transversale a rezervorului pentru reglarea nivelului este de  $332.5\text{cm}^2$ . Pentru ca nivelul apei în rezervorul tampon să nu coboare sub nivelul conductei de aspirație a pompei, nivelul apei în rezervorul pentru reglarea nivelului nu trebuie să depășească 20 cm.

Alte componente ale stației Festo utilizate pentru controlul nivelului sunt: senzor ultrasonic analogic, senzor de debit, amplificatorul pentru comanda motorului pompei, convertor de semnale: din curent în tensiune, din frecvență în tensiune, PLC, panou de control, sistem de țevi de legatură, valve manuale.



**Figura 3.2.** Schema sistemului de reglare a nivelului

#### *2.2.2. Senzor*

Senzorul SIEMENS BI.SI.0193 poate fi utilizat pentru a face două tipuri de măsurători:

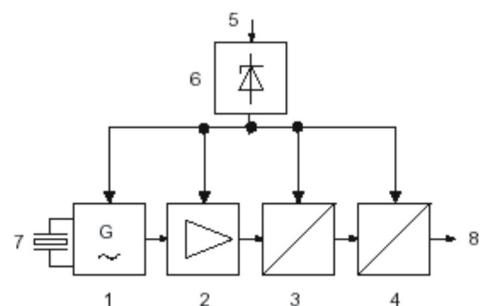
- se poate măsura distanța de la senzor la obiect, senzorul fiind setat pentru acest tip de măsurare încă din fabrică: semnalul de ieșire crește proporțional cu creșterea distanței dintre senzor și obiect.

### Lucrarea 3

- pentru măsurarea nivelului este însă necesară o altă setare pentru senzor deoarece distanța față de senzor scade odată cu creșterea nivelului. Astfel, zona de măsură este schimbată pentru ca senzorul să dea semnal de ieșire maxim pentru un nivel maxim și semnal de ieșire minim pentru un nivel minim; altfel spus, semnalul de ieșire crește invers proporțional cu distanța dintre senzor și suprafața lichidului din rezervor.



**Figura 3.3.** Senzorul ultrasonic SIEMENS BE.SI.0193

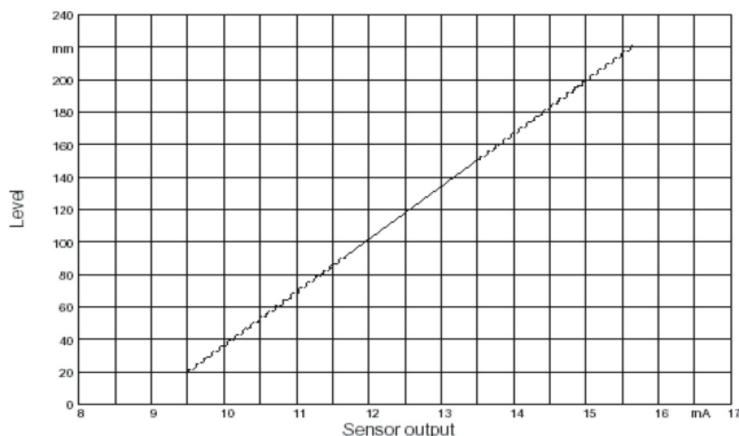


**Figura 3.4.** Diagrama schematică a senzorului ultrasonic

### Date tehnice

Parametrii	Valoare
Clasa de protecție	IP67
Temperatura de operare	-25 la 70 °C

Tensiunea de alimentare	24 VDC
Ieșire analogică	
Gama de curent	4-20 mA
Rezistență	0-300 Ω
Gama de măsurare	Max. 46-346 mm
Domeniul de măsurare	50 – 345 mm

**Tabelul 3.1.** Datele tehnice ale senzorului ultrasonic**Figura 3.5.** Curba caracteristică

Senzorul nu este liniar pe toată gama de măsurare. Este recomandată folosirea acestui senzor la măsurarea nivelului lichidelor cuprins între 80 și 180 mm. În afara acestor limite senzorul poate funcționa diferit de curba caracteristică.

#### 2.2.3. Element de execuție

Pompa centrifugă P101 (1 – figura 3.6) este elementul de execuție folosit în trei dintre buclele de reglare disponibile pe stand. În cadrul sistemului de reglare a nivelului pompa împinge lichidul din rezervorul de jos în cel de sus printr-un sistem de țevi.



**Figura 3.6.** Pompa centrifugă

Pompa nu poate să funcționeze fără lichid. Înainte de punerea în funcțiune rezervorul B101 trebuie să fie umplut cu lichid. Motorul pompei este alimentat de la controller-ul A4 prin releul K1. Controller-ul pompei poate fi comandat atât analogic cât și digital. Cu o ieșire digitală (O2 din XMA1) este posibilă comutarea între controlul digital binar și controlul analogic variabil între 0 și 24 V. Prin controlul digital binar ( $O2 = „0”$ ), pompa este pornită și opriță prin ieșirea suplimentară O3. Când  $O3 = „0”$  pompa este opriță iar când  $O3 = „1”$  pompa este pornită. În modul de control analogic ( $O2 = „1”$ ), tensiunea de comandă a PLC-ului, în gama 0 - 10 V, este aplicată controller-ului pompei, iar acesta variază alimentarea pompei cu o tensiune între 0 și 22 V.

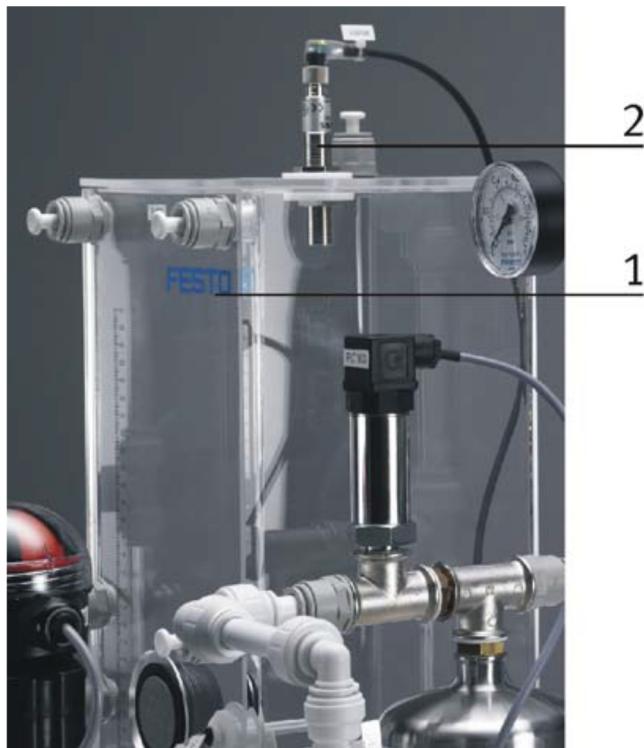
#### *2.2.4. Funcția de control al nivelului*

Funcția de control a nivelului are rolul de a regla nivelul de umplere cu lichid a unui rezervor.

Pompa P101 este folosită pentru a pompa lichidul din rezervorul B101 în rezervorul B102 (1 – figura 3.7) printr-un sistem de țevi de legătură. Nivelul lichidului în rezervorul B102 este monitorizat cu ajutorul unui senzor ultrasonic B101 (2) în punctul de măsurare LIC 102.1. Valoarea reală ar trebui

### Lucrarea 3

păstrată la un anumit nivel chiar dacă apar perturbații de debit de evacuare sau se modifică referința.



**Figura 3.7.** Componente sistem reglare nivel

Variabila manipulată este debitul de fluid al pompei P101. Pentru producerea de perturbații în sistem este posibilă închiderea manuală, total sau parțial, a valvei V102. Pentru a goli lichidul din rezervorul superior se deschid valvele manuale V101 și V104.

Semnalul de curent (4-20 mA) al senzorului ultrasonic este conectat ca semnal standard la terminalul analogic X2.8 (anexa 1), după care intră în convertorul A1, acesta realizând conversia din semnal unificat de curent (4-20 mA) în semnal unificat de tensiune (0-10 V).

### 2.3.Configurarea sistemului

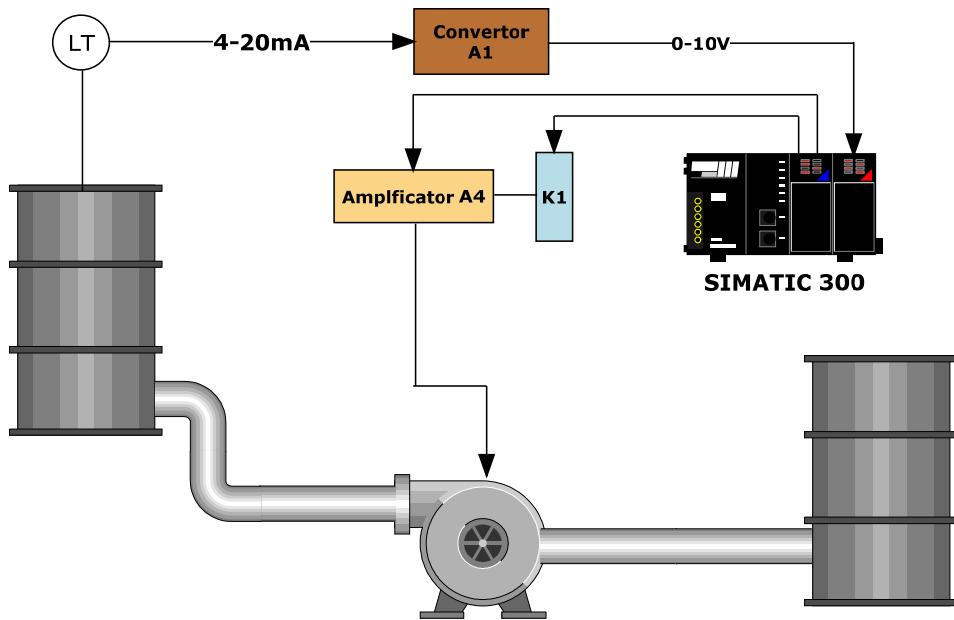
Configurarea standului FESTO pentru a realiza un sistem de control al nivelului trebuie să se facă ținând cont de informațiile prezentate în tabelul 3.2.

Senzor		B101	
Element de execuție		Pompa P101	
Valvă manuală		Deschisă	Închisă
	V101	•	
	V103		•
	V104		•
	V105		•
	V107		•
	V108		•
	V109		•
	V112	•	

**Tabelul 3.2.** Configurarea standului FESTO

### 2.4.Conexiuni

Semnalul provenit de la senzorul ultrasonic pentru măsurarea nivelului, care este convertit din semnal de curent unificat în semnal de tensiune unificată de către convertorul A1 și conectat la canalul AI0 al automatului programabil. Pompa centrifugă este alimentată prin amplificatorul A4, care este conectat la canalul AO0 al automatului programabil. Se va ține cont de faptul că amplificatorul A4 va trebui utilizat pe modul de lucru analogic. Acest lucru se face prin anclansarea releului K1, a cărui comandă este conectată la automatul programabil pe ce-a de-a treia ieșire digitală – DO0.2 (figura 3.8). Pentru detalii vezi anexa 1.



**Figura 3.8.** Schema de conexiuni



### 3. Desfășurarea lucrării

Reglarea nivelului este un proces ce implică senzori pentru măsurarea acestuia, elemente de execuție, o unitate numerică de calcul (în cazul de față un automat programabil) și un dispozitiv pentru monitorizare. În această lucrare de laborator s-a utilizat un senzor ultrasonic, pentru măsurarea nivelului, o pompă centrifugă, ca element de execuție, un automat programabil Siemens din seria S7-300, pentru controlul procesului și un PC pentru achiziție de date și monitorizare. Aplicația va măsura continuu valoarea nivelului de lichid și va asigura menținerea acestuia la o valoare prestabilită. Nivelul din rezervor, măsurat de senzorul ultrasonic constituie mărimea de ieșire a sistemului și semnal de intrare pentru regulatorul implementat în Step 7. Mărimea de comandă analogică este aplicată la intrarea elementului de execuție (pompa centrifugă). Toate componentele aplicației vor fi implementate folosind mediul SIMATIC Step7 pentru programare și SIMATIC WinCC Flexible pentru control și monitorizare.

### **3.1.Configurarea PLC**

1. Se crează un nou proiect utilizând datele corespunzătoare PLC-ului folosit (vezi secțiunea 2.5.2)
2. Se realizează configurația hardware a aplicației. Acest lucru presupune stabilirea modelului unității centrale (CPU) a automatului programabil, modelului sursei de alimentare a acestuia și configurarea modulelor de intrare și ieșire (vezi secțiunea 2.5.2)

### **3.2.Control continuu FB41 „CONT\_C”**

FB41 este o funcție de control continuu implementată de STEP 7. Această funcție este utilizată de PLC-urile SIMATIC S7 pentru a controla procesele tehnice cu intrări continue și ieșiri variabile. În timpul atribuirii parametrilor se pot activa sau dezactiva subfuncții ale regulatorului PID pentru o mai bună adaptare a acestuia la proces.

Pentru configurarea parametrilor regulatorului se poate folosi blocul DB41. Acesta permite, pe lângă setarea parametrilor regulatorului, folosirea câtorva opțiuni de control a blocului regulator precum și monitorizarea grafică a parametrilor regulatorului.

Funcțiile regulatorului se bazează pe algoritmul de control PID al unui regulator cu semnal analogic. Efectele proporțional, integrator și derivativ sunt conectate în paralel, iar acest lucru permite configurarea de regulatoare P, PI, PD, PID. Pot fi implementate și regulatoare pure I sau D. În figura 3.9 este prezentată schema bloc a regulatorului. În aceasta se pot observa toate mărimile de intrare și ieșire precum și funcțiile suplimentare pe care acest bloc le conține. În aplicația curentă funcția FB41 este apelată pentru implementarea unui regulator PI, cu observația că se trece comanda pe manual în cazul în care se dorește realizarea identificării sistemului. În acest scop s-au utilizat următoarele intrări: variabila de intrare (PV\_IN), valoarea presetată (SP\_INT), selectorul pentru control automat MAN\_ON (are valoare FALSE în cazul rulării pentru monitorizare ]n regim automat, respectiv valoarea TRUE în cazul în care se urmărește realizarea identificării sistemului), intrarea MAN ( pe care se aplică o treaptă de identificare),

## Lucrarea 3

valoarea parametrului proportional (GAIN), valoarea parametrului integrator (TI) și timpul dintre apelurile succesive ale blocului (perioada de eșantionare).

Blocul FB41 implementează regulatorul a cărui structură este prezentată în figura 3.5. Se va urmări pe schema din figura 3.5 semnificatia parametrilor de mai jos.

Principalii parametri utilizați, precum și intervalul de valori al acestora, sunt prezențați în cele ce urmează:

SP\_INT – referința procesului (în intervalul -100...100)

PV\_IN – măsura de la proces explimată în procente (în intervalul -100...100)

PV\_PER – măsura de la proces explimată în decimal (în intervalul -27648...27648)

GAIN – factorul de proporționalitate al regulatorului

TI – constanta de integrare a regulatorului (în secunde)

TD – constanta de derivare a regulatorului (în secunde)

LMN – comanda regulatorului (în domeniul 0...100)

LMN\_PER – comanda regulatorului scalată (în domeniul 0...27648)

MAN – comanda manuală (în domeniul 0...100)

PVPER\_ON – comutator măsură proces internă/periferică

MAN\_ON – comutator manual (bucla de control este întreruptă)/automat (bucla de control este activă)

CYCLE – timpul dintre apelurile succesive ale blocului regulatorului (egal cu 100 ms în mod standard)

Ieșirea procesului poate fi conectată pe 2 intrari, PV\_PER sau PV\_IN. Se utilizează PV\_PER dacă se dorește ca blocul FB 41 să facă sclarea intrării de la valoarea în decimal, disponibilă în urma conversiei analog-numerice, la valoarea în procente (necesară calculelor interne din blocul FB41) și respectiv PV\_IN dacă scalarea se va face separat în programul de aplicație.

Măsura din proces PV\_PER este scalată pentru a intra în domeniul de lucru al regulatorului.

$$CRP\_IN = PV\_PER \cdot 100/27648$$

$$PV\_PER \in [0 \dots 27648]$$

$$PV\_IN \in [0 \dots 100]$$

Uneori este necesară amplificarea și deplasarea semnalului primit din preces. Acest lucru se poate realiza în blocul FB41 prin setarea parametrilor PV\_fac și PV\_OFF, astfel:

Ieșirea din PV\_NORM = (ieșirea din CRP\_IN) \* PV\_FAC + PV\_OFF

Implicit PV\_FAC are valoarea 1 și PV\_OFF are valoarea 0.

Comada regulatorului LMN poate să fie scalată înainte de a se aplica elementului de execuție. Dacă este necesară, scalarea se face prin amplificarea semnalui și/sau deplasarea acestuia, după relația:

$$LMN = (\text{ieșirea din LMNLIMIT}) * LMN\_FAC + LMN\_OFF$$

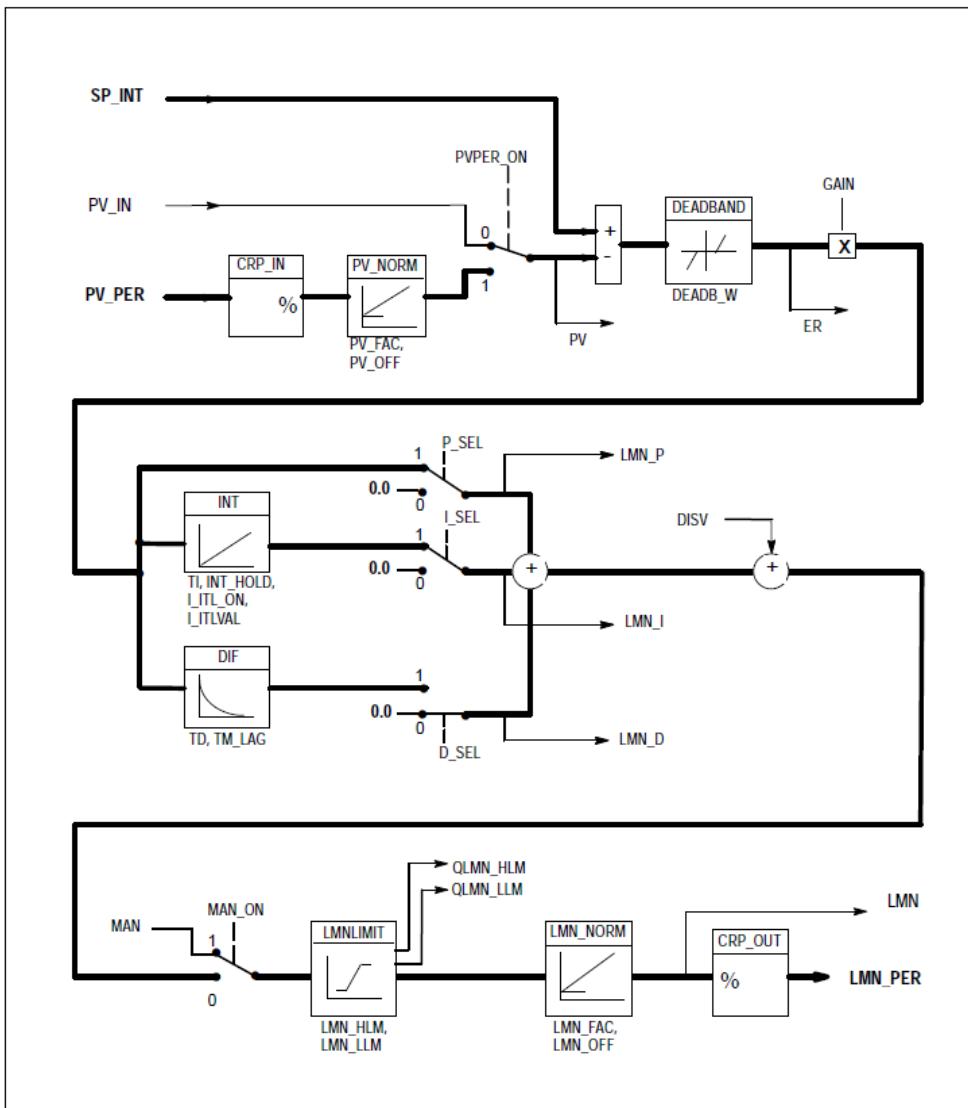
Implicit LMN\_FAC are valoarea 1 și LMN\_OFF are valoarea 0.

$$LMN\_PER = LMN \cdot 27648/100$$

$$LMN\_PER \in [0 \dots 27648]$$

$$LMN \in [0 \dots 100]$$

### Lucrarea 3



**Figura 3.9.** Schema regulatorului implementat în blocul FB41

Funcția de transfer a regulatorului este:

$$H_{PID}(s) = GAIN \left( 1 + \frac{1}{TI} \cdot \frac{1}{s} + TD \cdot s \cdot \frac{1}{TM\_LAG \cdot s + 1} \right)$$

***DB 41 – Bloc de date al regulatorului***

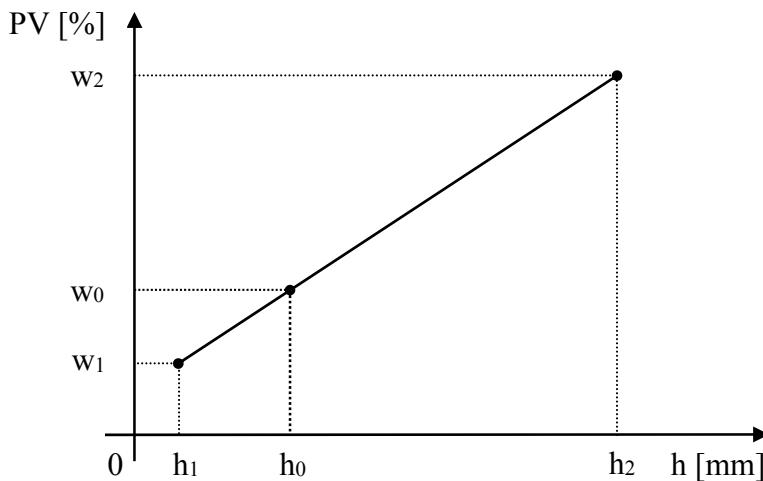
DB 41 este un bloc de date care este instantiat automat la apelarea functiei bloc FB41. În acest bloc se află toți parametrii regulatorului folosiți în funcția bloc FB41 și valorile corespunzătoare acestora.

Etapele realizării lucării de laborator sunt descrise în cele ce urmează.

**3.3.Programarea PLC**

1. Se inserează blocul OB35 (din meniul *Insert -> S7 Block -> Organization Block*) care este un bloc de organizare a căruia execuție ciclică se face la intervale de timp specificate (întreruperi).
2. Se intră în OB 35 și se selectează din fereastra din dreapta: *Libraries ->Standard Libraries ->PID Control Blocks* funcția bloc a regulatorului FB41.
3. Se asociază funcției bloc FB41 blocul de date DB41 ca o adresă a acestuia, apoi se realizează legăturile dintre FB41 și proces (intrarare și ieșire în/din regulator).
4. Anclanșarea releului K1 (asigură controlul analogic al amplificatorului pompei centrifuge) se poate face utilizând ca semnal de intrare linia ENO a blocului FB41. Astfel se va conecta semnalul provenit de la senzorul de nivel la variabila PV\_PER iar comanda regulatorului, LMN\_PER, la pompa centrifugă (PIW256). De asemenea este necesară configurarea parametrului PVPER\_ON al funcției bloc FB41 (PVPER\_ON="1" pentru activarea intrării PV\_PER).
5. Scalarea nivelului pentru referință. Referința de nivel se va stabili la o valoare dorită exprimată în milimetrii. Însă regulatorul va accepta referință de nivel exprimată în procente. Prin urmare avem nevoie de o conversie sau o scalare a valorii exprimate în milimetrii la valoarea exprimată în procente. Procedura de scalare este descrisă în cele ce urmează. Din blocul de date DB41 se vor citi valorile intrării analogice (variabila PV) la nivel minim

(aprox. 50 mm) și maxim (aprox. 200 mm), date ce vor fi folosite în continuare pentru scalarea nivelului de la valoarea în milimetri la cea în procente (figura 3.10).



**Figura 3.10.** Scalarea referinței

6. Referința regulatorului, SP\_INT este o variabilă ce ia valori reale, în procente, în domeniul [-100.0 .. +100.0] %. Aceasta mărime procentuală are însă un corespondent fizic greu de stabilit. Altfel spus, nu există o legătură clară între valoarea procentuală a referinței și valoarea acesteia într-o scală de valori standardizată (în cazul de față, înălțimea coloanei de lichid măsurată în milimetri). Este astfel necesară scalarea referinței din valori în milimetrii în valori procentuale care să fie transmise ulterior la intrarea SP\_INT a blocului regulatorului. Pentru a realiza scalarea se vor folosi valorile determinate la pasul anterior.

O metodă de calcul a valorii procentuale corespunzătoare unei valori a referinței impuse în milimetri este din ecuația dreptei determinate de coordonatele obținute experimental în cele două puncte de masură. Astfel, se va determina ecuația dreptei care trece prin cele două puncte. Se crează o funcție bloc FC1 care implementează relația:

$$w_0 = ah_0 + b$$

unde a și b sunt coeficienți determinați în urma aplicării ecuației dreptei,  $f(x)=ax+b$ , în cele două puncte de măsură. Rezultă astfel un sistem de două ecuații cu necunoscutele a și b:

$$\begin{cases} w_1 = ah_1 + b \\ w_2 = ah_2 + b \end{cases}$$

de unde rezultă:

$$a = \frac{w_1 - w_2}{h_1 - h_2}$$

și

$$b = w_1 - \frac{w_1 - w_2}{h_1 - h_2} h_1$$

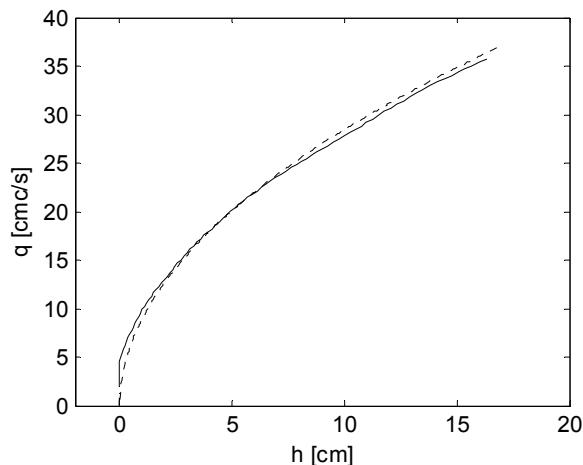
7. Funcția bloc care implementează scalarea referinței va fi FC1 (se crează în mod asemănător cu OB35 cu specificația ca este o funcție bloc). Vom avea nevoie de un parametru de intrare care va fi nivelul exprimat în milimetrii iar parametrul de ieșire al funcției va fi nivelul exprimat în procente. Parametrii se vor declara în secțiunea *Interface* a funcției. După salvare, funcția se apelează din blocul de organizare OB1 cu parametru de intrare valoarea referinței dorite în milimetrii și parametru de ieșire adresa variabilei SP\_INT din DB41, și anume DB41.SP\_INT.

### 3.4. Identificarea experimentală a procesului

1. Dezvoltarea aplicației de monitorizare în WinCC Flexible, care va cuprinde un grafic cu evoluția mărimilor de interes din proces (referință de nivel, nivelul măsurat, comanda regulatorului, eroarea) și va permite modificarea online a referinței de nivel (vezi lucrarea de laborator numărul 4 – Reglarea temperaturii unui letcon, secțiunea 3.5).
2. Se identifică funcția de transfer a procesului folosind ca și sistem de achiziție a datelor programul realizat în WinCC. Regulatorul

PID (FB41) se pune pe modul manual (MAN\_ON='TRUE') și se dă o treaptă procesului

Datorită caracteristicii neliniare a procesului, determinată în mare măsură de variația neliniară a nivelului în funcție de debitul de intrare (figura 3.11), identificarea experimentală a sistemului se va face în jurul unui punct de funcționare.



**Figura 3.11.** Debitul de ieșire din rezervor în funcție de nivel

Pentru a realiza identificarea experimentală a sistemului avem nevoie de un software de achiziție de date. Vom folosi mediul WinCC Flexible pentru a înregistra valoarea nivelului din proces. Se va crea un proiect în WinCC urmând pașii descriși în lucrarea 4 secțiunea 3.5.

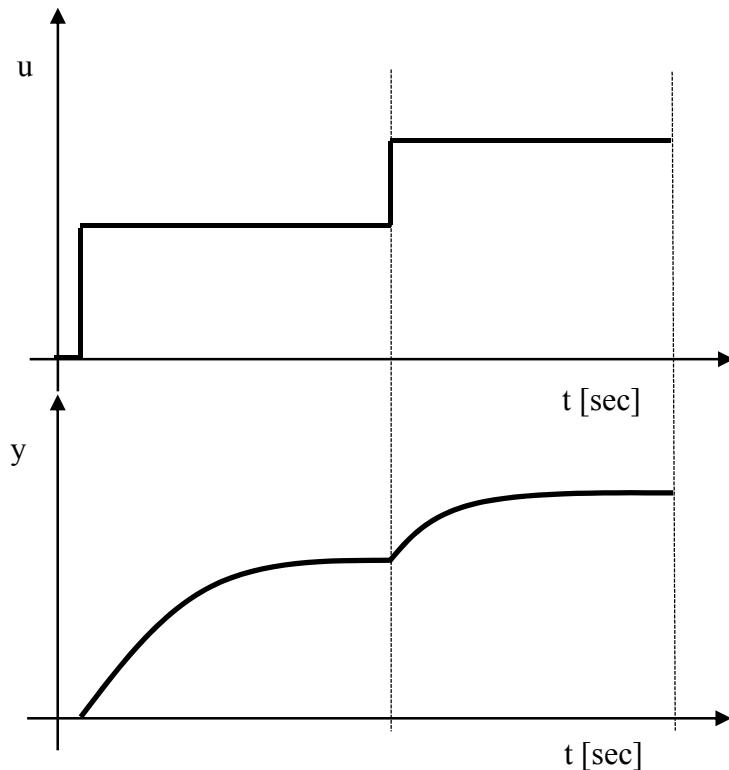
Se va crea un tag care va fi conectat cu semnalul provenit de la sezorul de nivel (AI0). Se vor modifica în proprietățile tag-ului câmpurile „*Data Log*” și „*Logging cicle*”. La „*Data Log*” se modifică formatul fișierului în care se vor salva datele și se alege „*csv*” iar la numărul de înregistrari se scrie o valoare suficient de mare ca să acopere timpul necesar pentru identificarea sistemului cu o perioada de eşantionare de o secundă (de exemplu 5000). La câmpul „*Logging cicle*” se va specifica perioada de eşantionare a datelor alegându-se valoarea de 1 secundă. Se va intra în modul de monitorizare și

### Lucrarea 3

achiziție de date prin startarea aplicației realizate dar acest lucru trebuie făcut înainte de a face modificări asupra intrării procesului (înainte de a da un semnal de intrare folosit la pasul de identificare a sistemului). Pornirea aplicației se face de la butonul „*Start runtime system*” din bara de instrumente.

Astfel, se va aplica o treaptă de tensiune pompei în jurul unei valori de regim staționar a nivelui, de exemplu 150 mm. Tensiunea necesară pompei pentru menținerea nivelui coloanei de lichid la 150 mm este aproximativ 4V (pentru o anumită pozitie a valvelor manuale; de obicei se va deschide valva de evacuare la 1/3 din secțiunea conductei). Se pornește PLC-ul, se încarcă programul și se trece în modul RUN. Din Simatic Manager se accesează DB41, și se trece regulatorul pe modul manual, astfel variabila MAN\_ON va trebui să aibă valoarea „TRUE”. Se dă comandă manuală setând valoarea variabilei MAN la 40.0 [%], valoarea corespunzătoare la ieșire fiind de 4 V. Se asteaptă până când procesul intră în regim stationar, adică nivelul ramane aproximativ constant (o perioadă de ~15 min din momentul aplicării treptei de 40%). Se dă o treaptă la intrarea sistemului, crescând comanda de la 40.0 [%] (valoarea inițială) la 45.0 [%]. Astfel treapta folosită la identificare va avea amplitudinea de 0.5 V (figura 3.12).

Se precizează că datele au fost obținute prin aplicarea unei trepte de tensiune la ieșirea analogică a PLC și măsurarea unei tensiuni la intrarea analogică a PLC. Aceasta permite identificarea întregii părți fixate (rezervor, pompă, senzor, convertor, amplificator).



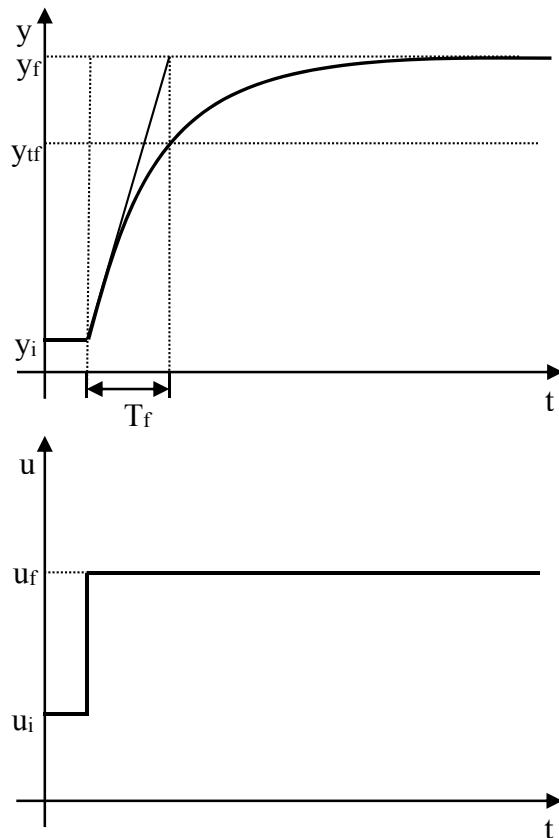
**Figura 3.12.** Identificarea experimentală a sistemului

În urma identificării experimentale se va obține un semnal  $y$  (nivel) în domeniul [DEC]. Se va folosi mediul Matlab pentru a scala acest semnal la valori procentuale. Astfel, se va înmulți vectrul  $y$  cu valoarea  $100/27648$ .

### 3.5.Determinarea parametrilor modelului procesului și acordarea regulatorului

Dinamica procesului de reglare a nivelului se consideră a fi asimilată dinamicii unui sistem de ordinul I. Modelul dinamicii sistemului de ordinul I este determinat ca fiind descris de o funcție de transfer de ordinul I:

$$H_f(s) = \frac{k_f}{T_f s + 1}$$



**Figura 3.13.** Determinarea parametrilor modelului experimental al procesului

Determinarea parametrilor modelului procesului se va face pe baza răspunsului la semnal de tip treaptă al procesului.

Răspunsul procesului caracterizează comportarea sistemului dinamic la o excitație de tip treaptă. Acest tip de excitație este asemănăt cu orice semanal caracterizat de două niveluri de amplitudine care are o duartă suficient de mare. Determinarea modelui procesului se va face utilizand ieșirea corespunzătoare semnalui de excitație aplicat pentru identificare, odata ce sistemul a ajuns în punctul de funcționare ( $\Delta u = 5.0\%$ ).

Parametrii funcției de transfer a procesului vor fi:

Factorul de amplificare al procesului. Acesta se calculeaza ca fiind raportul dintre valoarea ieșirii și cea a intrării în regim statioar.

$$k_f = \frac{\Delta y}{\Delta u} = \frac{y_f - y_i}{u_f - u_i}$$

Constanta de timp a procesului. Constanta de timp al procesului reprezintă timpul necesar ieșirii procesului astfel încât aceasta sa ajungă la 63.4 % din valoarea sa de regin stationar. Aceasta se determină prin proiecția punctului  $y_{tf} = y_i + 0.634 \cdot (y_f - y_i)$ , la intersecția graficului, pe axa de timp (figura 3.13).

### Acordarea regulatorului

Pentru acordarea regulatorului se va utiliza o metoda simplă și anume determinarea funcției de transfer a acestuia prin impunerea de performante în bucla inchisă. Impunem bucla inchisă ca fiind un sistem de ordinul I cu o constantă de timp mai rapidă de 10 ori decât constanta de timp a procesului. Evident și timpul de răspuns al sistemului în bucla inchisă se va micsora de 10 ori. În urma calculelor se va obține un regulator de tipul PI.

Astfel, dacă impunem:

$$H_0(s) = \frac{1}{T_0 s + 1}$$

Unde  $T_0 = T_f / 10$

Putem scrie pentru bucla închisă că

$$H_0(s) = \frac{H_R H_f}{1 - H_R H_f}$$

Astfel, rezultă că funcția de transfer a regulatorului este:

$$H_R(s) = \frac{1}{H_f} \cdot \frac{H_0}{1 - H_0}$$

Pentru a fi implementat cu ajutorul blocului FB41 regulatorul trebuie adus la forma:

$$H_R(s) = GAIN \left( 1 + \frac{1}{TI} \cdot \frac{1}{s} + \frac{TDS}{TM\_LAGs + 1} \right)$$

Se vor urma pașii de mai jos:

1. Se acordează regulatorul și se verifică mai întâi prin simulare (vezi lucrarea 1 secțiunea 3) obținerea performanțelor impuse
2. Valorile obținute pentru parametrii regulatorului se asignează în blocul de date DB41. Pe lângă aceasta mai trebuie făcută configurarea funcției bloc FB41 în concordanță cu specificul procesului
3. Testare sistem de control și monitorizare. Acest pas implică verificarea performanțelor sistemului de control. Parametrii de performanță ai sistemului de control sunt aceeași cu cei impuși la acordarea regulatorului? Dacă nu de ce?

### **Configurare bloc de date DB41**

Parametrii care vor trebui configurați în DB41 sunt:

PVPER\_ON = 1 – selecție pentru intrarea PV\_PER

MAN\_ON = FALSE – switch pentru a selecta modul automat de lucru al regulatorului

P\_SEL – switch pentru activarea componentei proportionale a regulatorului

I\_SEL – switch pentru activarea componentei integrale a regulatorului

IITL\_ON – switch pentru activarea condițiilor intiale

IITVAL = 40.0 specifică condițiile initiale, adică valoarea comenzi aplicate pentru a ajunge cu sistemul în punctul de funcționare stabilit

GAIN – valoarea componentei proportionale a regulatorului

TI – valoarea componentei integrale a regulatorului

Se fixează referință la valoarea dorită în milimetri (recomandabil la o valoare cu 2-3 cm mai mare decât regimul staționar utilizat la identificarea procesului).

#### **4. Conținutul referatului**

Se va descrie modul de realizare al aplicației. Referatul va cuprinde referiri la următoare etape din realizarea aplicației:

1. Identificarea sistemului și determinarea unei model al procesului
2. Acordarea regulatorului
3. Scalarea referinței de nivel din valoarea în milimetrii în domeniul [0...100]%
4. Implementarea controlului automat al nivelului în Step 7
5. Dezvoltarea unei aplicații de monitorizare în WinCC Flexible care va cuprinde un grafic cu evoluția mărimilor de interes din proces (referință de nivel, nivelul măsurat, comanda regulatorului, abaterea) și va permite modificarea online a referinței de nivel
6. Testarea sistemului de control și monitorizare
7. Concluzii

## 4. Reglarea temperaturii unui letcon



### 1. Scopul lucrării

Obiectivul acestei lucrări este realizarea unei aplicații de reglare și monitorizare a temperaturii cu ajutorul unui PLC Simatic S7-300. Scopul este familiarizarea cu activitățile ce implică implementarea unui sistem pentru monitorizarea și controlul unui proces, având la dispoziție echipamente de lucru similare cu cele utilizate în practică, pentru conducerea proceselor industriale.



### 2. Considerații teoretice

La începutul lucrării sunt prezentate o serie de considerații cu caracter general privind sistemele de reglare a temperaturii, după care se continuă cu principalele detalii constructiv funcționale ale unui proces concret (ciocan de lipit termostatat) și ale elementelor din componența sistemului de reglare. Este expusă de asemenea o metodă de modelare analitică a procesului pe baza analizei fenomenelor de transfer și acumulare de căldură.

#### 2.1. Sisteme de reglare a temperaturii

Toate sistemele de reglare a temperaturii sunt de fapt sisteme de reglare a transferului de căldură. Din cauza caracteristicilor acestui transfer, procesele respective au constante de timp mai mari decât ale proceselor în care se regleză debitul, presiunea sau nivelul. Aceste procese prezintă de asemenea o serie de trăsături specifice:

- reprezintă de regulă sisteme stabile în buclă deschisă;
- relațiile dintre mărimele ce caracterizează o instalație termică sunt de obicei funcții continue liniare, care pot fi liniarizate cu ușurință;
- nu conțin de regulă elemente oscilante, adică ecuațiile lor caracteristice, obținute în urma liniarizării, au numai rădăcini reale negative;

- reprezintă sisteme filtrante care atenuează puternic oscilațiile de frecvență înaltă;
- prezența timpului mort se face simțită în special în instalațiile în care se încălzesc fluide.

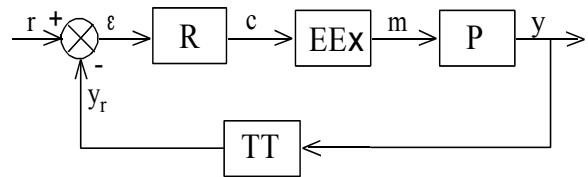
Întârzierile care apar în măsurarea temperaturii sunt de asemenea un factor important. Traductoarele de temperatură sunt de obicei introduse într-un tub de protecție pentru a efectua înlocuiri în caz de defectare fără întreruperea procesului. Viteza de răspuns a elementului de măsurare a temperaturii depinde de felul în care este realizat, de materialul tubului de protecție și de natura și viteza fluidului în care este introdus. Traductorul de temperatură trebuie de asemenea montat în locul unde coeficientul de transfer al căldurii este cel mai mare posibil.

Procesele care au constante de timp și timp mort mici pot fi reglate cu sisteme dotate cu regulatoare bipozitionale. Reglarea bipozitională se utilizează în special la procesele cu încălzire electrică, la cuptoarele cu tuburi radiante și în cuptoarele cu flacără în care nu se regleză presiunea în cupor. Sistemul de reglare bipozitional cu zonă de histerezis poate să realizeze reglarea temperaturii cu o abatere de 1% față de mărimea de referință. Când diferența dintre temperatura maximă dezvoltată de instalație și temperatura de lucru (sau diferența dintre puterea maximă a instalației și sarcina curentă) este mare, transferul de căldură este intens și o reglare bipozitională are oscilații mari de amplitudine. În acest caz se recomandă utilizarea unor regulatoare liniare cu structură PI sau PID.

Regulatoarele de tip PI sunt cel mai frecvent utilizate, în special când perturbațiile de sarcină sunt mari și frecvente. Regulatoarele PID sunt indicate în cele mai multe reglări de temperatură în care constantele de timp sunt mari.

## 2.2. Descrierea elementelor componente ale părții fixate

Sistemul de reglare studiat reprezintă o variantă clasică aşa cum se arată în schema bloc din figura 4.1.



**Figura 4.1.** Structura de reglare

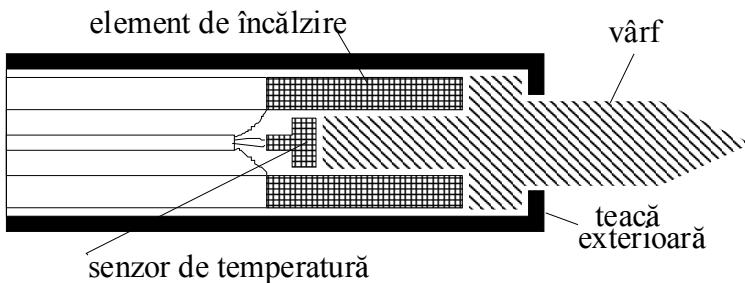
În aplicațiile industriale ale sistemelor de reglare automată este mult vehiculat termenul de parte fixată a instalației de automatizare. Prin parte fixată se înțelege, de regulă, atât procesul propriu-zis, cât și elementul de execuție și elementul de măsurare din sistemul de reglare respectiv.

### 2.2.1. Procesul (*P*)

Procesul este concretizat printr-un ciocan de lipit cu încălzire electrică sau letcon electric. Conform dicționarului limbii române, letconul este o unealtă în formă de ciocan utilizată pentru încălzirea pieselor metalice în vederea lipirii cu cositor, cu plumb etc. Forma de ciocan este specifică variantelor vechi de letcon, care se încălzește în foc până la o temperatură superioară celei de topire a cositorului, de exemplu, și se lipește cu el până temperatura devine prea mică pentru a topi cositorul, după care se încălzește din nou și.a.m.d. Letconul electric nu mai păstrează forma de ciocan, având diverse forme, în funcție de scopul pentru care a fost construit, iar vârful este încălzit electric. Pentru lipire este important să se mențină temperatura vârfului letconului la anumite valori, în funcție de aliajul utilizat la lipire și în funcție de destinația lipiturilor. La ciocanele de lipit uzuale menținerea temperaturii se realizează prin alegerea corespunzătoare a puterii electrice consumate. Ciocanele de lipit performante sunt supradimensionate din punct de vedere al puterii electrice, menținerea temperaturii vârfului la valori de cca 250 – 300 °C realizându-se prin deconectarea alimentării la depășirea acestor valori. Pentru aceasta, vârful letconului are încorporat fie un releu termic, fie

un traductor de temperatură. Pentru ultima variantă, stația de lipire va trebui să fie prevăzută cu un regulator pentru comanda încălzirii vârfului. De regulă acesta este un regulator bipozițional simplu realizat cu un comparator de tensiune și un releu. În această lucrare, pentru scopuri didactice vor fi utilizate regulatoare mult mai complicate, cu algoritmi bipoziționali sau PID.

În figura 4.2. este prezentat schematic vârful unui letcon de lipit:



**Figura 4.2.** Vârf letcon de lipit

Elementul de încălzire este constituit dintr-o înfășurare rezistivă care transformă energia electrică în energie termică. Rezistența de încălzire de are o valoare de cca.  $11 \Omega$ . Puterea electrică este de 42W la o tensiune de alimentare de 24V. Vârful este demontabil și poate fi înălțurat prin defiletarea tecii exterioare de protecție. Prin demontarea vârfului sau montarea lui poate fi simulață o perturbație parametrică asupra procesului. Aceste operații au ca efect modificarea factorului de amplificare și a constantelor de timp ale procesului.

Modelul procesului se poate obține studiind fenomenul de transfer de căldură pe baza ecuațiilor de bilanț energetic. Transferul de căldură diferă după cum acesta are loc prin:

- Conducție - transmiterea căldurii în interiorul unui conductor termic sub formă de flux de cădură, datorită unei diferențe de temperatură;
- Convecție - transferul de căldură între zona caldă și cea rece a unui fluid, prin amestecare, cu ajutorul curenților de fluid naturali (convecție liberă) sau artificiali (convecție forțată);
- radiație - transmiterea căldurii prin emisie de radiații infraroșii.

În aceste cazuri, debitul de căldură are următoarele expresii:

- transfer de cădură prin conducție:

$$q = \frac{k_c S}{h} (\theta_1 - \theta_2)$$

- transfer de căldură prin convecție:

$$q = k_{cv} S (\theta_1 - \theta_2)$$

- transfer de căldură prin radiație:

$$q = k_r S \varepsilon (\theta_1^4 - \theta_2^4)$$

în care:

- $q$  - reprezintă debitul de căldură transferată între două puncte aflate la temperaturile  $\theta_1$  și  $\theta_2$  [W];
- $k_c$  - conductivitatea termică a conductorului [W/m°C];
- $k_{cv}$  - coeficient de convecție [W/m<sup>2</sup>°C];
- $k_r$  - constantă de radiație [W/m<sup>2</sup>°C];
- $h$  - grosimea conductorului (distanța pe care se propagă căldura) [m];
- $S$  - suprafața prin care are loc transferul de căldură [m<sup>2</sup>];
- $\varepsilon$  - factor energetic de emisie (emisivitate);

O altă lege fizică necesară pentru studiul proceselor cu transfer de căldură este cea care determină efectele acumulării de energie. Temperatura în interiorul unui corp care acumulează căldură variază în conformitate cu relația:

$$q = C \frac{d\theta}{dt} = c m \frac{d\theta}{dt}$$

unde:

- $q$  - reprezintă debitul de căldură [W];
- $C$  - capacitatea termică a corpului [J/°C];
- $m$  - masa corpului [Kg];
- $c$  - căldura specifică a corpului la presiune constantă [J/kg°C]
- $\theta$  - temperatura corpului [°C].

Pentru ciocanul de lipit cu încălzire electrică se determină un model cu parametri concentrați, de la variația puterii electrice dezvoltată de elementul de încălzire ( $\Delta q$ ) la variația temperaturii măsurată de senzorul încorporat ( $\Delta\theta$ ). Acceptând că temperaturile sunt uniform distribuite în cuprinsul

elementelor componente, transferul energiei termice se poate aproxima prin relațiile:

$$q = C_i \frac{d\theta_i}{dt} + k_1 (\theta_i - \theta)$$

$$k_1 (\theta_i - \theta) = C_v \frac{d\theta}{dt} + k_2 (\theta - \theta_a)$$

în care notațiile au următoarele semnificații:

- $q$  - debitul de căldură introdus de elementul de încălzire [W];
- $C_i$  - capacitatea termică a elementului de încălzire [J/°C];
- $C_v$  - capacitatea termică a vârfului împreună cu teaca exterioară [J/°C];
- $\theta_i$  - temperatura elementului de încălzire [°C];
- $\theta$  - temperatura vârfului [°C];
- $\theta_a$  - temperatura mediului ambiant [°C];
- $k_1, k_2$  - coeficienți de transfer termic [W/°C].

Întrucât capacitatea termică a elementului de încălzire este foarte mică ( $C_i \ll C_v$ ) se poate neglijă. În aceste condiții aplicând relațiilor (2.5) transformata Laplace pentru condiții inițiale nule și variații mici în imediata vecinătate a punctului de funcționare, rezultă:

$$\Delta q = C_v s \Delta \theta + k_2 \Delta \theta$$

din care se obține funcția de transfer:

$$H_p(s) = \frac{\mathcal{L}\Delta\theta}{\mathcal{L}\Delta q} = \frac{k_p}{T_p s + I}$$

cu

$$k_p = \frac{I}{k_2}, \quad T_p = \frac{C_v}{K_2}$$

### 2.2.2. Traductorul de temperatură (TT)

Traductorul de temperatură este încorporat în vârful letconului (fig 4.2.). Este un termocuplu cupru-constantan (Cu-CuNi), indicat pentru măsurarea temperaturilor între  $-100^{\circ}\text{C}$  și  $+400^{\circ}\text{C}$ . Trebuie menționat că, aşa cum se observă și din figura 4.2, traductorul este montat într-un loc cu transfer intens al căldurii și măsoară temperatura la baza vârfului de lipit. Această

temperatură în regim dinamic este mai mare decât cea din vîrf datorită inerției termice în transmiterea căldurii, în regim staționar temperaturile egalizânduse.

Fiind în contact direct cu teaca elementului de încălzire și fără teacă proprie de protecție, funcția de transfer a termocuplului va fi în acest caz de ordinul întâi:

$$H_t(s) = \frac{k_t}{T_t s + 1}$$

Pentru temperaturi de cca 200 - 300°C inerția unui termocuplu fără teacă este de aproximativ 5 minute în aer liniștit, 3 minute în curent de aer și 0.2 minute în lichid agitat. În condițiile de montare precizate aici se poate aprecia o valoare a constantei de timp de cca 0.2-0.5 minute. Dacă se dorește o precizie ridicată sunt necesare măsuri pentru compensarea temperaturii joncțiunii de referință.

Valoarea coeficientului de proporționalitate  $k_t$  se poate determina din caracteristica statică a traductorului. El va avea o valoare egală cu raportul dintre domeniul semnalului de ieșire și domeniul mărimii de intrare (domeniul de temperatură).

### 2.2.3. Elementul de execuție (EEx.)

Elementul de execuție este un releu comandat prin intermediu unei ieșiri digitale a PLC-ului.

### 2.2.4. Filtrarea numerică a semnalelor

Filtrarea numerică se impune atunci când operația de măsurare este afectată de zgomot, cu un spectru de frecvență în zona frecvențelor relativ joase (raportate la frecvența de eșantionare). Zgomotele de frecvență ridicată se recomandă să fie filtrate cu ajutorul unor circuite RC care trece jos, plasate la intrarea modulului de interfață de intrări analogice.

În mod obișnuit, filtrarea se realizează prin mediere sau cu ajutorul unui filtru numeric, care de cele mai multe ori este varianta discretizată a elementului de întârziere de ordinul I. Există și algoritmi mai sofisticăți de

filtrare, dar în general, în aplicațiile industriale este suficientă utilizarea acestor algoritmi de filtrare simpli.

În cazul continuu se folosesc frecvent pentru filtrare elemente de întârziere de ordinul I, care reprezintă filtre trece-jos și asigură o filtrare de tip exponențial, la un semnal de intrare  $y$  de tip treaptă răspunsul  $y_F$  fiind o exponențială. Funcția de transfer a unui astfel de element are forma:

$$H_F(s) = \frac{Y_F(s)}{Y(s)} = \frac{1}{T_F s + 1}$$

cu factor de amplificare unitar și constanta de timp  $T_F$ .

Din această relație se poate scrie

$$T_F s Y_F(s) + Y_F(s) = Y(s)$$

și trecând în domeniul timp

$$T_F \frac{dy_F}{dt} + y_F = y$$

De la această ecuație se poate trece la o ecuație cu diferențe și acceptând aproximarea derivatei prin intermediul diferențelor finite:

$$\frac{dy_F}{dt} \approx \frac{\Delta y_F}{\Delta t} = \frac{y_{F,k} - y_{F,k-1}}{T_e}$$

( $T_e$  fiind perioada de eșantionare a sistemului discret) rezultă:

$$T_F \frac{y_{F,k} - y_{F,k-1}}{T_e} + y_{F,k} = y_k$$

Această ultimă relație poate fi scrisă sub o formă mai convenabilă pentru realizarea calculelor:

$$y_{F,k} = y_{F,k-1} + \frac{T_e}{T_F + T_e} (y_k - y_{F,k-1})$$

Se poate observa din ultima relație comportarea diferită în momentul inițial față de comportarea elementului continuu de întârziere de ordinul I, răspunsul la un semnal treaptă având valoarea inițială diferită de zero. Această deosebire este determinată de faptul că discretizarea elementului continuu nu s-a efectuat prin intermediul relațiilor riguroase dintre funcțiile de transfer în  $s$  și cele în  $z$ .

### 2.3. Descrierea standului

Instalația de față este compusă din:

- un letcon electric SOLOMON
- un automat programabil din seria Simatic S7-300 dispus pe un panou ce conferă usurință și transparență în utilizarea resurselor acestui PLC.
- conectori izolați ce asigură partea de conexiuni dintre letcon și standul didactic cu PLC Siemens S7-300.
- un calculator ce realizează programarea PLC-ului, interfațarea și monitorizarea procesului.
- un modul MPI/DP ce asigură comunicația dintre calculator și automatul programabil.
- panoul sinoptic ce asigură accesul la resursele și funcțiilor automatului.



## 3. Desfășurarea lucrării

Pentru realizarea acestei aplicații se folosește editorul de programe SIMATIC Manger din cadrul mediului SIMATIC Step7. Prin acestă unealtă, dezvoltatorul are acces la toate adresele de intrare și ieșire, atât analogice cât și digitale, ale PLC-ului și la o serie de adrese ce sunt alocate variabilelor temporare din program.

### 3.1. Configurarea PLC

Etapele realizării lucrării de laborator sunt prezentate în cele ce urmează:

1. Se crează un nou proiect utilizând datele corespunzătoare PLC-ului folosit
2. Se realizează configurația hardware a aplicației. Acest lucru presupune stabilirea modelului unității centrale (CPU) a automatului programabil, modelului sursei de alimentare a acestuia și configurarea modulelor de intrare și ieșire

### **3.2. Control continuu al temperaturii FB58 „TCONT\_CP”**

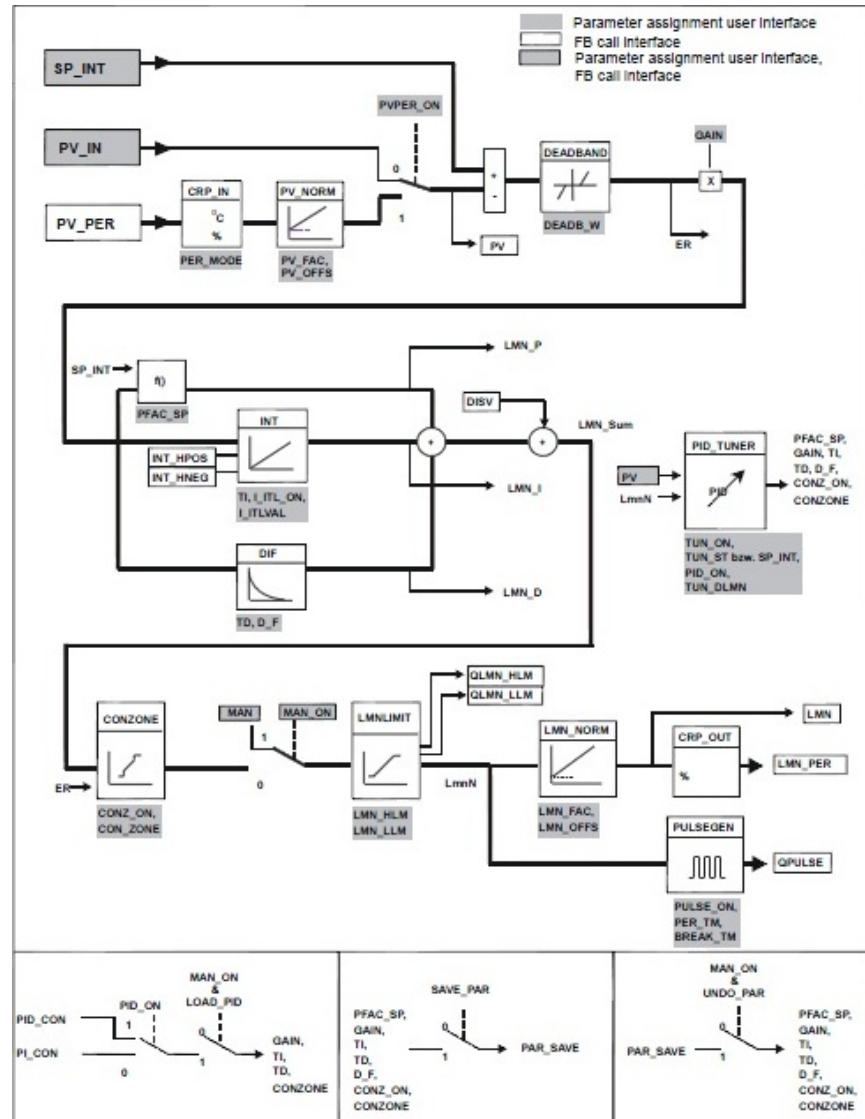
FB58 este o funcție de control continuu implementată de STEP 7. Această funcție este utilizată de PLC-urile SIMATIC S7 pentru a controla procesele tehnice cu intrări continue sau digitale (semnal PWM) și ieșiri variabile. În timpul atribuirii parametrilor se pot activa sau dezactiva subfuncții ale regulatorului PID pentru o mai bună adaptare a acestuia la proces.

Pentru acordarea regulatorului se poate folosi blocul DB58. Acesta permite, pe lângă setarea parametrilor regulatorului, folosirea câtorva opțiuni de control a blocului regulator precum și monitorizarea grafică a parametrilor regulatorului.

Functiile regulatorului se bazează pe algoritmul de control PID al unui regulator cu semnal analogic, și se poate activa dacă este necesar un generator de impuls, pentru a obține semnale de ieșire modulate în timp (PWM). Efectele proporțional, integrator și derivativ sunt conectate în paralel, iar acest lucru permite configurarea de regulatoare P, PI, PD, PID. Pot fi implementate și regulatoare pure I sau D. În figura următoare este prezentată schema bloc a regulatorului. În aceasta se pot observa toate mărurile de intrare și ieșire precum și funcțiile suplimentare pe care această funcție le conține. În program această funcție este apelată pentru implementarea unui regulator PI, cu observația că se trece comanda pe manual în cazul în care se dorește realizarea identificării sistemului. În acest scop s-au utilizat următoarele intrări: variabila de intrare (PV\_IN), valoarea presetată (SP\_INT), selector control automat MAN\_ON (în cazul rulării pentru monitorizare), selector control manual MAN\_ON (în cazul în care se urmărește realizarea identificării sistemului), intrarea MAN (pe care se aplică o treaptă de identificare), valoarea parametrului proporțional (GAIN), valoarea parametrului integrator (TI) și timpul dintre apelurile succesive ale blocului. Ca parametru de ieșire fiind comanda PWM (QPULSE) și va fi aplicată pe relee care comandă încălzitorul.

Se precizează că mai trebuie să se seteze parametrii precum perioada PWM, PER\_TM din blocul asociat DB 58, care în cazul de față este 2 secunde, setarea de activare PWM PULSE\_ON=1.

Blocul FB58 implementează regulatorul a cărui structură este prezentată în figura 4.3.

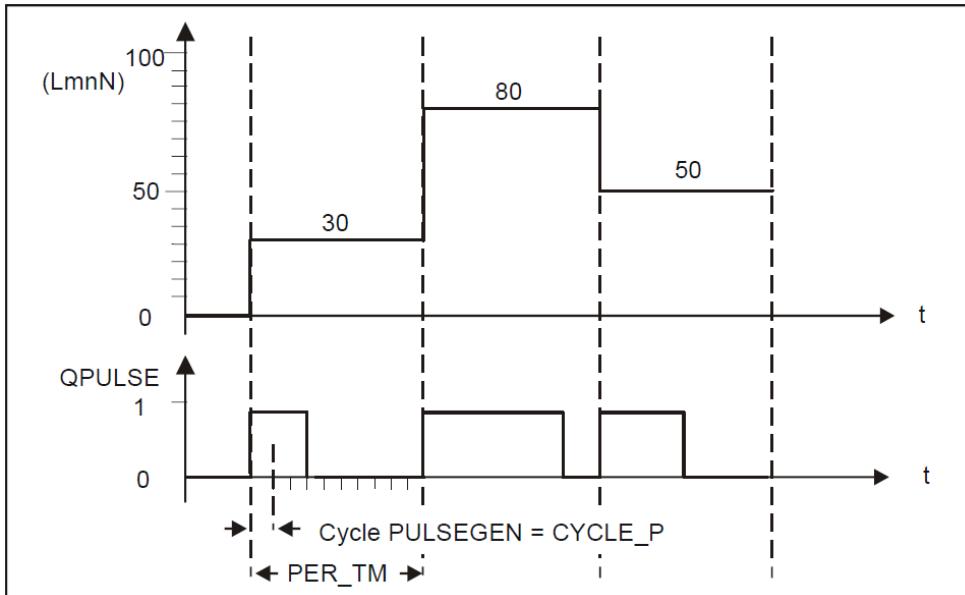


**Figura 4.3.** Schema regulatorului implementat în blocul FB58

Funcția de transfer a regulatorului este:

$$H_{PID}(s) = GAIN \left( 1 + \frac{1}{TI} \cdot \frac{1}{s} + TD \cdot s \cdot \frac{1}{TM\_LAG \cdot s + 1} \right)$$

Modulul de PWM, PULSEGEN, din cadrul acestui bloc realizează conversia intrării analogice într-o succesiune de impulsuri cu perioada PER\_TM, folosind tehnica de modulației semnalelor în durată. În figura următoare este prezentat modul de funcționare al blocului pe trei valori de intrare diferite.



**Figura 4.4.** Modul de funcționare a modulului PULSEGEN

Se precizează că la intrarea în modulul PULSEGEN a valorii analogice este aplicată comanda de la regulator, în figura 4.4, aceasta fiind reprezentată de valoarea LmnN. Sunt luate în considerare 3 perioade de eşantionare diferite în care valorile semnalului analogic sunt pe rând 30,80, respectiv 50. Ieșirea QPULSE este calculată astfel: se împarte intervalul perioadei în 10 părți egale (pentru Cycle PULSGEN setat la valoarea 0.1 secunde) care în cazul valorii analogice de intrare 30, sunt setate 3 dintre ele pe valoarea 1 logic respectiv 7 dintre ele pe valoarea 0 logic. Astfel se obține un factor de umplere de 30% din valoarea perioadei dată de utilizator (PER\_TM).

Principalii parametri utilizați, precum și intervalul acestora sunt prezențați în cele ce urmează:

SP\_INT – referința procesului (în domeniul 0...100)

PV\_IN – măsura internă (în domeniul 0...100)

PV\_PER – măsura de la proces (în domeniul 0...27648)

GAIN – factorul de proporționalitate al regulatorului

TI – constanta de integrare a regulatorului (în secunde)

TD – constanta de derivare a regulatorului (în secunde)

LMN – comanda regulatorului (în domeniul 0...100)

LMN\_PER – comanda regulatorului scalată (în domeniul 0...27648)

QPULSE – comanda regulatorului în semnal PWM

PER\_TM – perioada semnalului PWM

MAN – comanda manuală (în domeniul 0...100)

PVPER\_ON – comutator măsură proces internă/periferică

PULSE\_ON – comutator comandă PWM

MAN\_ON – comutator manual (bucla de control este întreruptă)/automat  
(bucla de control este activă)

CYCLE – timpul dintre apelurile succesive ale funcției bloc a regulatorului – FB58. Valoarea standard este egală cu 100 ms.

CYCLE\_P – perioada de eşantionare cu care lucrează generatorul de semnal PWM. Aceasta trebuie să corespundă cu perioada de întrerupere a blocului de organizare OB35. Durata unui puls generat este întotdeauna multiplu de această valoare.

Măsura din proces PV\_PER este scalată pentru a intra în domeniul de lucru al regulatorului.

$$CRP\_IN = PV\_PER \cdot 100/27648$$

$$PV\_PER \in [0 \dots 27648]$$

Valoarea scalată în procente a valorii variabilei PV\_PER este reținută în variabila PV ce are domeniul de valori:

$$PV \in [0 \dots 100]$$

Ieșirea din PV\_NORM = (ieșirea din CRP\_IN) \* PV\_FAC + PV\_OFF

Implicit PV\_FAC are valoarea 1 și PV\_OFF are valoarea 0.

Comanda regulatorului LMN este scalată pentru a putea fi aplicată elementului de execuție.

$$LMN = (ieșirea din LMNLIMIT) * LMN\_FAC + LMN\_OFF$$

Implicit LMN\_FAC are valoarea 1 și LMN\_OFF are valoarea 0.

$$LMN\_PER = LMN \cdot 27648/100$$

$$LMN\_PER \in [0 \dots 27648]$$

$$LMN \in [0 \dots 100]$$

### **DB 58 – Bloc de date al regulatorului**

DB 58 este un bloc de date care este instantiat automat la apelarea functiei bloc FB58. În acest bloc se află toți parametrii regulatorului folosiți în funcția bloc FB58 cu valorile corespunzătoare.

Pentru compilarea și transferul proiectului program către se PLC se accesează butonul prezentat în Figura 4. 5 din bara de meniu. Trecerea în modul Online se face printr-un click pe butonul descris în figura 4.6.

Modul Online permite monitorizarea și modificarea blocurilor în timpul execuției programului.



**Figura 4.5** Butonul de download



**Figura 4.6** Mod Online

### **3.3. Programarea PLC**

1. Se inserează blocul OB35 (din meniul *Insert -> S7 Block -> Organization Block*) care este un bloc de organizare a căruia execuție ciclică se face la intervale de timp specificate (întreruperi).
2. Se intră în OB 35 și se selectează din fereastra din dreapta: *Libraries ->Standard Libraries ->PID Control Blocks* funcția bloc a regulatorului FB41.
3. Din blocul de date DB58 se vor citi valorile intrării analogice (variabila PV) la temperatura minimă (aprox. 24 °C) și maximă (aprox. 100 °C) date ce vor fi folosite în continuare pentru scalare
4. Se crează un bloc FC1 în care se realizează funcția de scalare a referinție după formula:

$$PV = aT + b$$

unde a și b sunt coeficienți determinați în urma aplicării ecuației dreptei în cele două puncte de măsură iar T este valoarea referinței de temperatură (vezi lucrarea 3, secțiunile 3.2 și 3.3)

5. Funcția de scalare a referinței (FC1) se apelează în interiorul blocului OB1

### **3.4. Identificarea experimentală a procesului și implementarea regulatorului acordat**

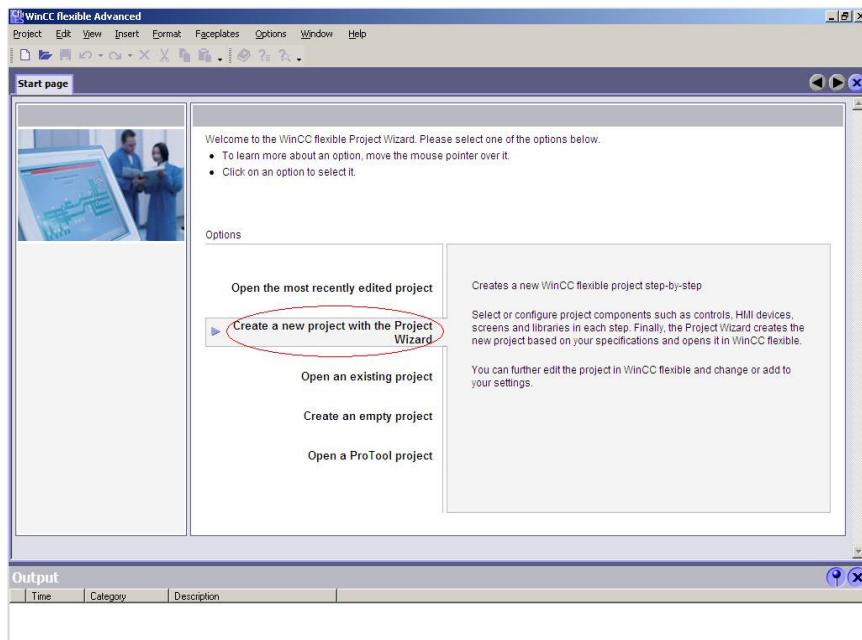
1. Se realizează interfața WinCC. Dezvoltarea aplicației de monitorizare în WinCC Flexible care va cuprinde un grafic cu evoluția mărimilor de interes din proces (referință de temperatură, temperatură măsurată, comanda regulatorului, abaterea) și va permite modificarea online a referinței de temperatură (secțiunea 3.5)
2. Se identifică funcția de transfer a procesului. Achiziția de date se va face eu ajutorul programul realizat în WinCC. Regulatorul PID (FB58) se pune pe modul manual (MAN\_ON = TRUE) și se dă o comandă de 15% (aproximativ 1.5 V) (vezi lucrarea 3 secțiunea 3.4)
3. Se acordează regulatorul – cu ce metoda impunerii de performanțe pentru sistemul în buclă închisă – iar parametrii acestuia se asignează în blocul de date DB58. Pe lângă aceasta mai trebuie făcută configurarea funcției bloc FB58 în concordanță cu specificul procesului (vezi lucrarea 3 secțiunea 3.5)
4. Testare sistemului de control și monitorizare

### **3.5. Crearea unui proiect WinCC Flexible**

În secțiunile de imagini ce urmează sunt prezentate cele 8 pași necesari pentru crearea unui nou proiect. Proiectul WinCC Flexible se integrează unui proiect Step 7. Există și opțiunea creării unui nou proiect WinCC Flexible direct din fereastra SIMATIC Manager.

Crearea unui nou proiect se face prin selectarea opțiunii „*Create a new project with the Project Wizard*” din pagina de start a aplicației.

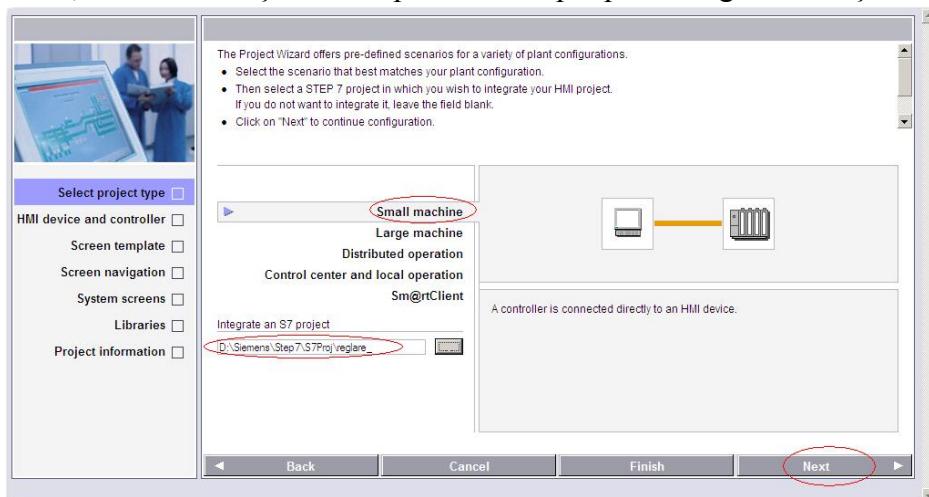
## Lucrarea 4



**Figura 4.7.** Pasul 1 în crearea proiectului

Dintr-o listă se alege opțiunea „*Small Machine*” – reprezintă tipul de proiect pentru o legătură directă PLC - Panou Sinoptic.

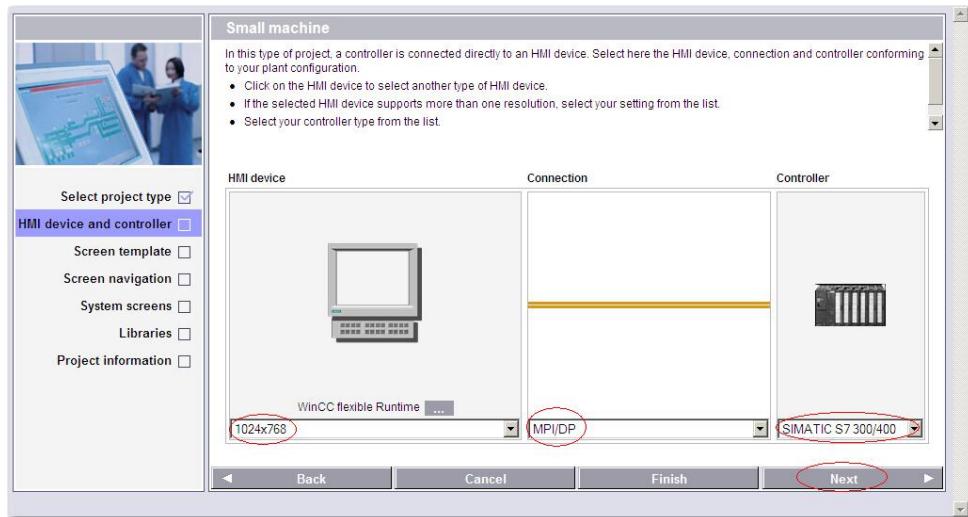
Se integrează proiectul Step 7 pentru care se dorește crearea unei interfețe grafice, în cazul de față s-a ales proiectul Step 7 pentru reglarea turăției.



**Figura 4.8.** Pasul 2 în crearea proiectului

## Lucrarea 4

Se alege tipul de dispozitiv HMI - în acest caz se alege „*WinCC Flexible Runtime*”. Este setată și rezoluția pentru ecran. Tipul de conexiune – MPI/DP. Seria din care face parte automatul programabil – SIMATIC S7 300/400.

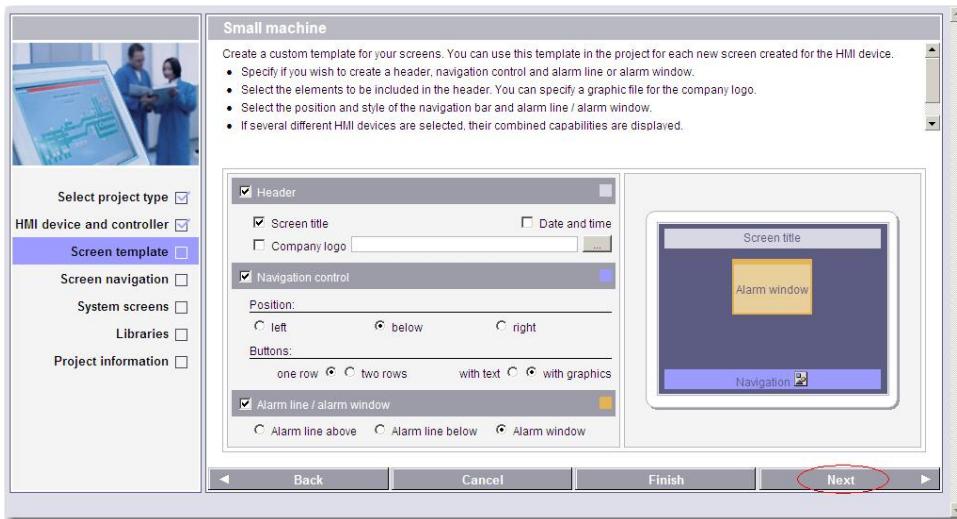


**Figura 4.9.** Pasul 3 în crearea proiectului

Se specifică dacă se dorește crearea unui header, butoane de control pentru navigație, linie pentru alarme/ fereastră pentru alarme. Tipul de conexiune – MPI/DP.

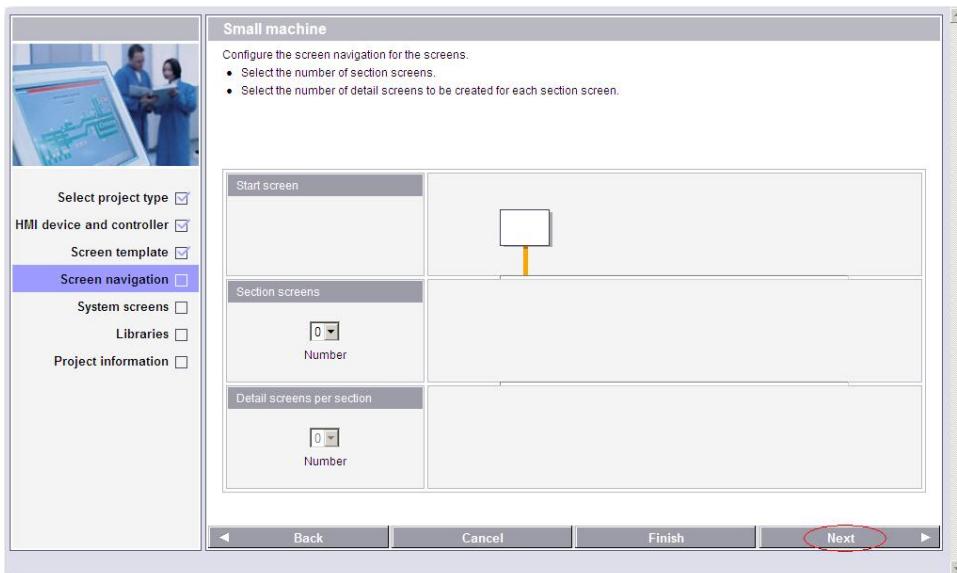
Selectarea elementelor ce vor fi incluse în header. Selectarea poziției și stilul bara de navigație cât și a alarmelor.

## Lucrarea 4



**Figura 4.10.** Pasul 4 în crearea proiectului

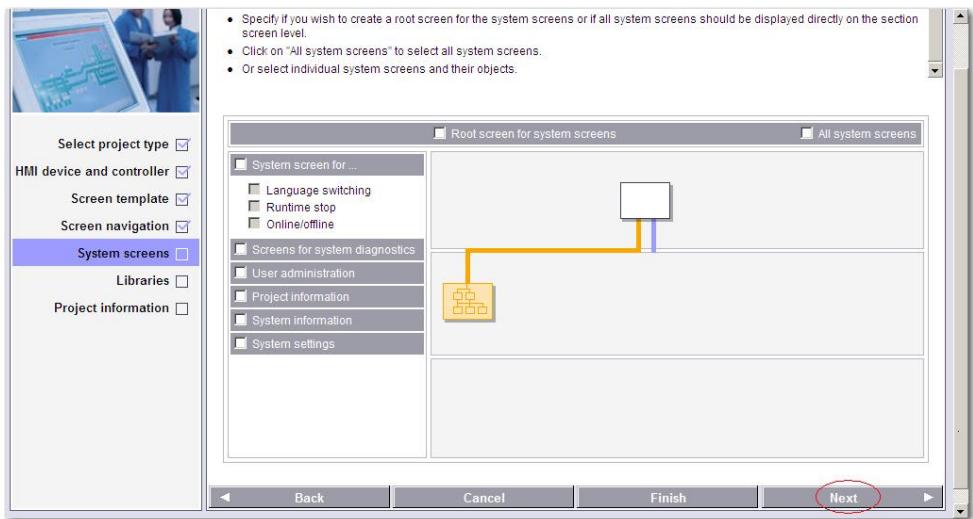
Alegerea numărului de ecrane. Alegerea numărului ecranelor cu detalii corespunzătoare fiecărui ecran.



**Figura 4.11.** Pasul 5 în crearea proiectului

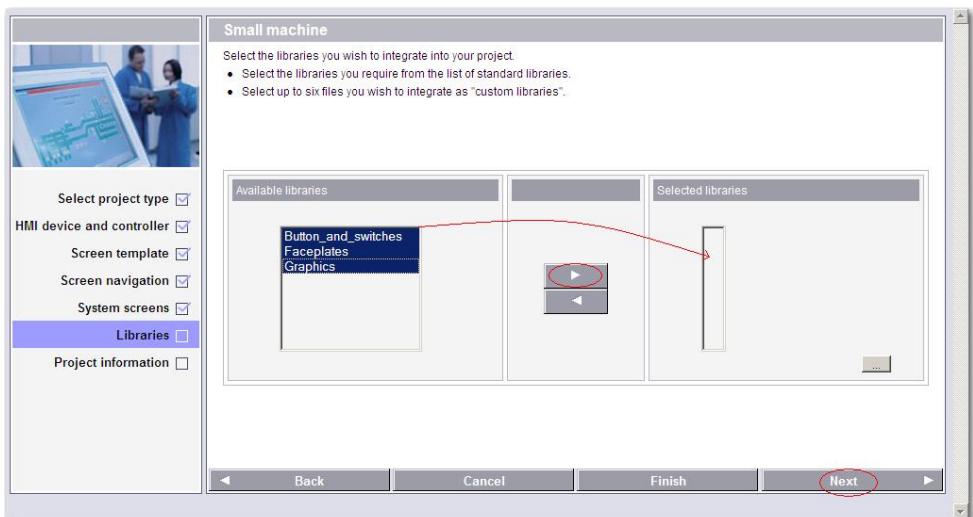
Se specifică dacă se dorește crearea unui ecran principal pentru toate ecranele sistem, sau dacă ecranele ar trebui afișate direct în poziția specifică.

## Lucrarea 4



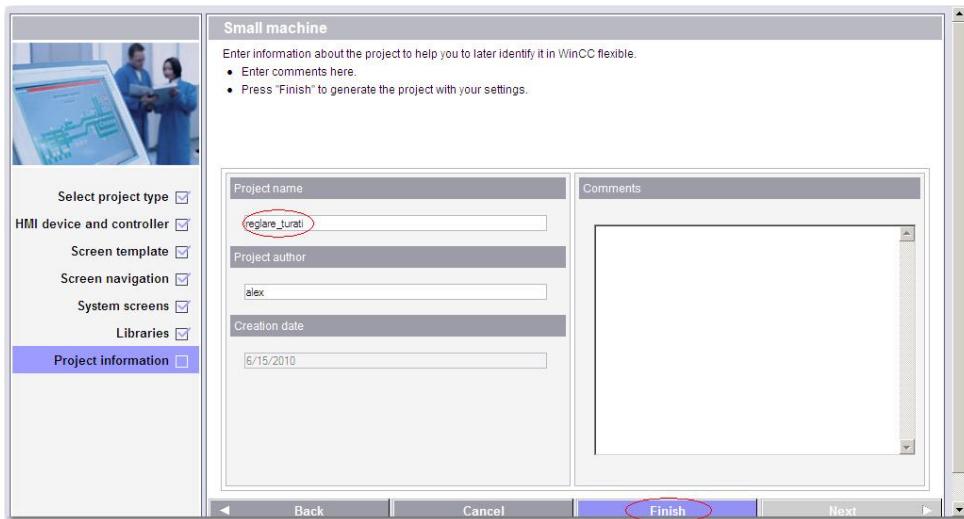
**Figura 4.12.** Pasul 6 în crearea proiectului

Se adaugă librăriile necesare. Sunt selecțiate până la 6 fișiere care pot fi integrate ca librării opționale.



**Figura 4.13.** Pasul 7 în crearea proiectului

Se adaugă comentarii. Apasând butonul „Finish” este generat proiectul ce conține toate setările specificate în pașii anteriori.



**Figura 4.14.** Pasul 8 în crearea proiectului

În continuare sunt prezentate modulele ce alcătuiesc interfața grafică a proiectului și lista de tag-uri. Tag-urile sunt variabile de lucru ce corespund fiecărui obiect creat în WinCC. Prin intermediul tag-urilor se face legătura între obiectele create în WinCC Flexible și adresele utilizate în Step7.

În tabelul următor sunt explicitate funcțiile fiecărui tag, tipul de dată, și sursa(adresa) de la care primește valoarea de instanțiere aşa cu apare în blocul de instanțiere **DB41**.

Numele Tag-ului	Tipul	Adresa	Corespondent în DB41
comanda	REAL	DB41 DBD 72	LMN
MAN_ON	BOOL	DB41 DBX 0.1	MAN_ON
MAN	REAL	DB41 DBD 16	MAN
ref(referință)	REAL	DB41 DBD 6	SP_INT
temperatura	WORD	DB41 DBW 14	PV_IN
abaterea	REAL	DB41 DBD 96	ER

**Tabel 4.1.** Explicitarea tag-urilor

Tot din meniul de declarare a tag-urilor (figura 4.15) se pot și alte caracteristici speciale pentru fiecare tag, cum ar fi: timpul de achiziție, canalul de conexiune, afișarea ciclică, etc.

Ecranul principal al interfeței(start screen) este format din mai multe panel-uri(componente grafice), fiecare având o funcție bine stabilită în controlul procesului. Acest lucru este necesar pentru ca operatorului să-i fie facilă conducerea procesului. Astfel ecran principal cuprinde urmatoarele panel-uri: panoul cu parametrii regulatorului, panoul pentru lucru în regim manual, indicatorul analogic al comenzi, graficul ce reprezintă răspunsul sistemului la referința data și butonul de ieșire.

### **3.6.Achiziția datelor**

Un alt rol foarte important pe care îl are mediul WinCC Flexible este acela, că prin funcțiile speciale pe care le are, se pot face și achiziții de date ale unor parametrii aaproceselor conduse.

În cazul procesului de reglare a temperaturii ne interesează variația în timp răspunsului sistemului la diferite referințe presetate. În secțiunea următoare sunt specificați pașii necesari creării unui astfel de sistem de achiziții pentru aplicația curentă.

Adăugarea unui tag prin care se va face achiziția de date. Aceasta va adresa canalul de intrare al automatului programabil pe care este conectat senzorul de temperatură. Pentru a adauga un tag se merge la secțiunea *Communications -> Tags*. Setați numele tag-ului, conexiunea, tipul de date, adresa, data log (aici se specifică fișierul unde se vor stoca datele, tipul acestuia – csv și numărul de înregistrări, care trebuie să fie suficient de mare – să acopere un timp de 20 de minute cu o perioadă de eşantionare de o secundă), logging cycle (perioada de scriere a datelor în fișier, ceea ce coincide cu perioada de eşantionare – o secundă în cazul de fată).

## Lucrarea 4

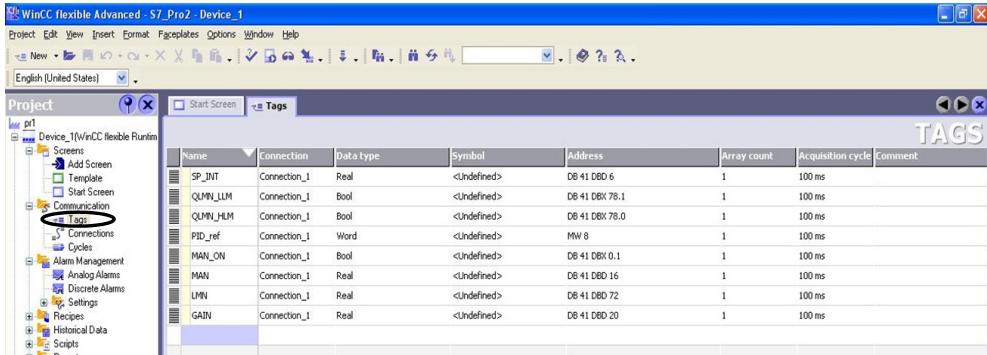


Figura 4.15. Crearea unui tag

Este bine de știut că prin aceste setări, în timpul rulării aplicației valorile răspunsului sistemului este stocat într-un fisier cu extensia „csv” care ulterior v-a putea fi prelucrat în mediul Matlab. Tot aici se selectează și calea în care va fi salvat fișierul (ex. C:\Logs).

Pentru compilarea și startarea aplicației se va accesa butonul *Start runtime system* din bara de instrumente a programului.

## 4. Conținutul referatului

Se va descrie modul de realizare al aplicației. Referatul va cuprinde referiri la urmatoare etape din realizarea aplicației:

1. Identificarea sistemului și determinarea unei model al procesului
2. Acordarea regulatorului
3. Scalarea referinței de temperatură din °C în domeniul [0...100]%
4. Implementarea controlului automat al temperaturii în Step 7
5. Dezvoltarea unei aplicații de monitorizare în WinCC Flexible care va cuprinde un grafic cu evoluția mărimilor de interes din proces (referință de temperatură, temperatura măsurată, comanda regulatorului, abaterea) și va permite modificarea online a referinței de temperatură
6. Testarea sistemului de control și monitorizare
7. Concluzii

## 5. Reglarea temperaturii într-un tub



### 1. Scopul lucrării

Obiectivul acestei lucrări este realizarea unei aplicații de reglare și monitorizare a temperaturii cu ajutorul unui PLC Simatic S7-300. Scopul este familiarizarea cu activitățile ce implică implementarea unui sistem pentru monitorizarea și controlul unui proces lent – reglarea temperaturii.



### 2. Considerații teoretice

La începutul lucrării sunt prezentate o serie de considerații cu caracter general privind sistemele de reglare a temperaturii, după care se continuă cu principalele detalii constructiv funcționale ale unui proces concret și ale elementelor din componența sistemului de reglare.

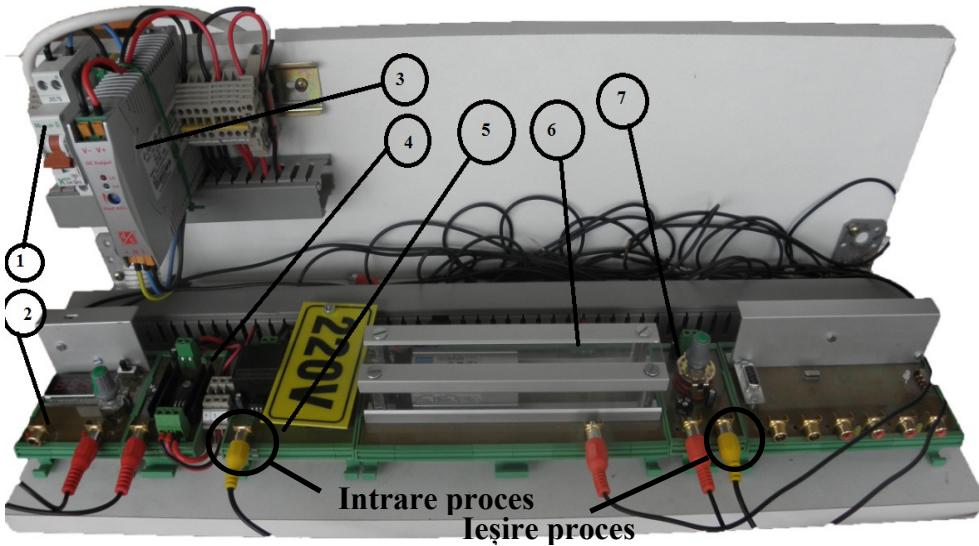
#### 2.1. Descrierea standului didactic

Standul didactic este de fapt un proces adus la scară mică și este ideal pentru înțelegerea elementelor din care este compus un proces pe scară largă și a fenomenelor care au loc. Standul pune la dispoziție toate elementele specifice unui proces real, de la achiziția de date cu ajutorul senzorului până la posibilitatea de a acționa prin element de execuție.

Elementele principale ale standului sunt următoarele:

- senzor de temperatură;
- încălzitor;
- ventilator (folosit pentru a produce un flux controlat de aer în interiorul tubului);
- potențiometru (necesar reglării turăției ventilatorului) ;
- sursă de alimentare a echipamentelor de pe stand;
- cabluri conectare;
- tub;

Figura 5.1, care prezintă standul didactic în întregime.



**Figura 5.1.** Stand didactic pentru reglarea temperaturii în tub

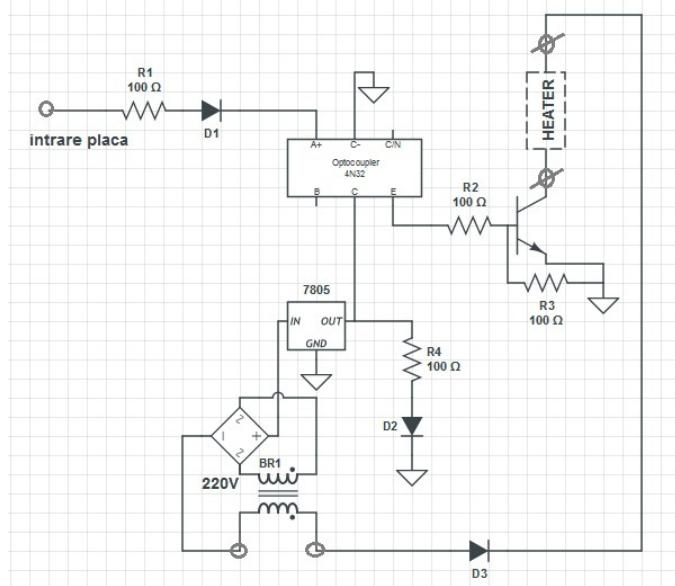
Elementele marcate pe figura 5.1 sunt:

- Siguranță electrică folosită ca intrerupător(1);
- placă circuit imprimat 1 (potențiometru) (2);
- sursă de alimentare (3);
- placă circuit imprimat 2 (alimentare încălzitor) (4);
- placă circuit imprimat 3 (comandă încălzitor) (5);
- tubul în care se controlează temperatura (6);
- placă circuite imprimate 4 (filtrare și amplificare semnal senzor) (7);

Sistemul de control va trebui să mențină o temperatură constantă în interiorul tubului, stabilită printr-o referință. Elementul de execuție este un încălzitor comandat de PLC. Temperatura este măsurată cu senzor LM35 iar semnalul este trimis la PLC pentru procesare. Ventilatorul are două roluri. Primul rol este acela de a produce un flux constant de aer prin interiorul tubului, asigurat printr-o turăție constantă a ventilatorului. Al doilea rol este unul de perturbație pe fluxul de aer și se realizează prin creșterea sau scăderea turăței ventilatorului.

### Scheme electrice

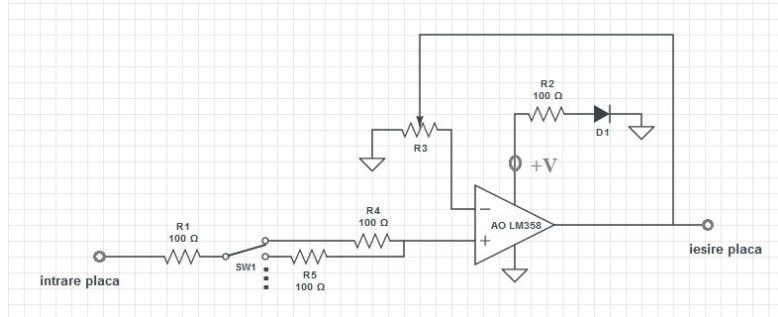
În figura 5.2 este prezentată schema electrică a circuitului imprimat (3).



**Figura 5.2.** Schema electrică a circuitului imprimat (3)

Circuitul prezentat în figura 5.2 realizează comanda încălzitorului. Semnalul de la PLC este aplicat la intrarea circuitului, care printr-un optocupluor comanda alimentarea în curent alternativ a încălzitorului.

În figura 5.3 este prezentată schema electrică a circuitului imprimat (4).



**Figura 5.3.** Schema electrică a circuitului imprimat (4)

Circuitul din figura 5.3 amplifică semnalul provenit de la senzorul de temperatură aplicat la intrare cu factor de amplificare variabil. Factorul de amplificare poate fi selectat cu ajutorul selectorului SW1. De la ieșirea circuitului semnalul este transmis la PLC.

## 2.2. Element de execuție

Elementul de execuție este un încălzitor Cirrus 25 cu convecție forțată bazată pe tehnologia PTC.

### **Tehnologia PTC**

PTC-urile sunt surse de căldură electrice multilaterale. Elementele de încălzire bazate pe această tehnologie sunt aşa numite pietre ceramice cu caracteristică de autolimitare a temperaturii. Pietrele PTC au un timp de răspuns rapid și stagnează după ce temperatura predefinită a fost atinsă. Este posibilă fabricarea lor în formă pătrată, dreptunghiulară, circulară sau în formă de inel. Peste temperatura de referință, proprietățile semiconductoare și feroelectrice ale ceramicii sunt folosite pentru a produce o creștere în rezistență cu câteva ordine de mărime și creând astfel proprietatea de autolimitare.

Creșterea de temperatură poate fi simțită într-o plajă de câteva grade Celsius. Într-o aplicație dată o disipare termică mai mare va rezulta o rezistență mai mică, menținută de element. Acest efect crește puterea de ieșire a încălzitorului în timpul unei utilizări în condiții de temperatură scăzută. Invers, când temperatura ambiantă crește și mai puțină căldură este disipată, rezistența elementului va crește, rezultând un current apropiat de zero.

Rezistența și ieșirea dinamică a încălzoarelor PTC le face o alegere excelentă pentru furnizarea de căldură de la o sursă electrică controlată. Simplificarea în proiectare și reducerea energiei consumate reprezintă economisiri semnificative pentru o aplicație dată. Prin îndepărțarea de componente predispuse să se defecteze, cum ar fi termostatele, crește fiabilitatea unui produs. Siguranța este un mare beneficiu întrucât indiferent de cât curent este aplicat la PTC, nu va depăși niciodată temperatura de suprafață dorită.

Pietrele PTC sunt ceramic policristaline dopate cu titanat de bariu. Piatra este formată după un proces care include amestacare, măcinare, uscare și

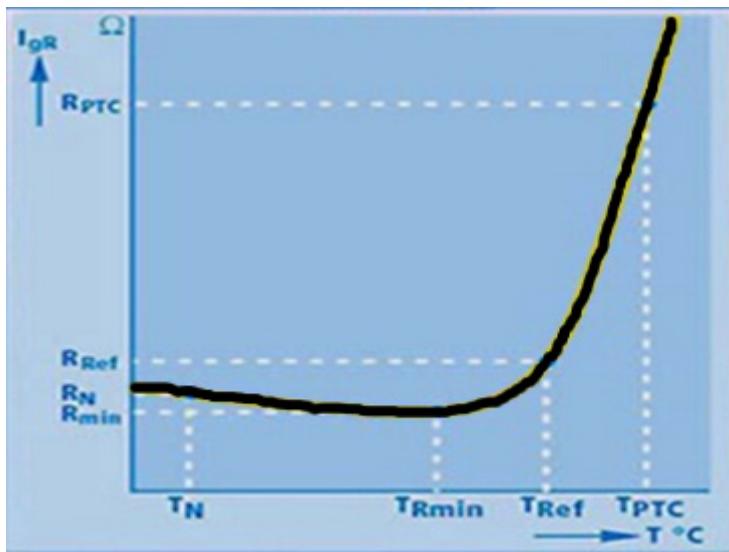
sinterizare. Pentru a facilita conexiunea electrică, pe suprafața PTC-urilor sunt aplicate niște contacte metalizate, cunoscute sub numele de electrozi.

Caracteristici și beneficii ale PTC-urilor:

- autolimitarea de temperatură asigură siguranță;
- este dinamic și eficient din punct de vedere energetic;
- acceptă tensiuni de la 12 până la 24V, aplicații speciale de tensiune până la 800V;
- permite tensiunile duale: 12-24V, 100-240V;
- temperatura de suprafață poate fi setată în domeniul 40-280°C;
- toleranța de temperatură este de +/- 5°C;
- are un design compact;
- densitate de putere mare;
- ciclu de viață lungă;
- fiabile;
- sunt disponibile și configurații personalizate în ceea ce privește mărimea, curentul de rupere, temperatura, tensiunea și puterea de ieșire.

În figura 5.4 putem observa dependența tipică temperatură-rezistență. Elementele din figură sunt următoarele:

- RN: rezistență nominală termistorului (valoarea rezistenței la temperatură nominală, adică 25°C)
- RMin: valoarea minimă a rezistenței (valoarea rezistenței la temperatura TRmin)
- TRMin: temperatura la RMin
- RRef: rezistență de referință
- TRef: temperatura de referință
- RPTC: rezistență la alegere în zona abruptă
- TPTC: temperatura la rezistență RPTC



**Figura 5.4.** Caracteristică rezistență-temperatură la PTC

Caracteristicile constructive ale încălzitorului sunt prezentate în figura figura 5.5.

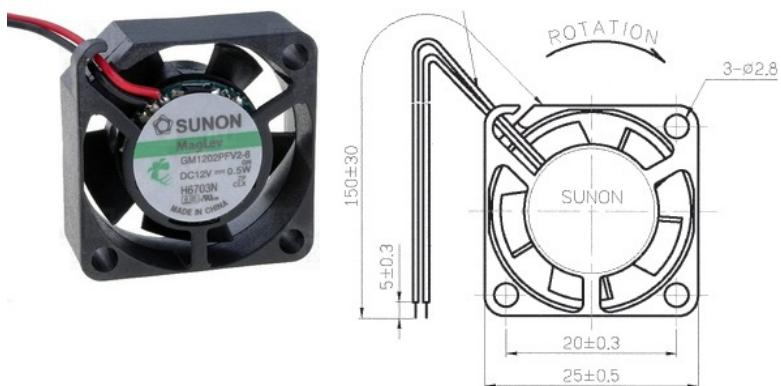


**Figura 5.5.** Încălzitorul Cirrus 25/2

### 2.3. Ventilator

Ventilatorul atașat la încălzitor este un ventilator de curent continuu, produsă de firma SUNON (model GM1202PFV2). Rolul lui constă în asigurarea unui flux de aer constant prin interiorul tubului. Crescând sau scăzând turăția ventilatorului putem obține un efect perturbator.

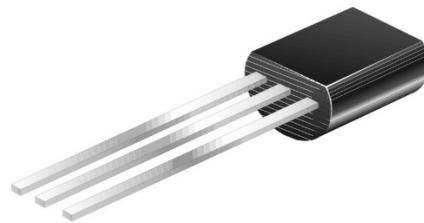
În figura 5.6 se prezintă caracteristicile de construcție ale ventilatorului.



**Figura 5.6** Ventilator

### 2.4. Senzorul LM35

Senzorii de temperatură din seria LM35 sunt circuite integrate de precizie, a căror tensiune de ieșire este liniar proporțională cu temperatura măsurată, în grade Celsius, astfel au avantajul față de senzorii de temperatură, calibrări în grade Kelvin, că nu mai trebuie să scădem o constantă mare de tensiune din ieșirea senzorului pentru a obține o scalare în grade Celsius. LM35 nu necesită o calibrare externă pentru a obține precizie de  $\pm \frac{1}{4}^{\circ}\text{C}$  la temperatura camerei sau  $\pm \frac{3}{4}^{\circ}\text{C}$  în interval  $(-55..+150)^{\circ}\text{C}$ . Impedanța de ieșire mică, ieșirea liniară și precizia fac sezonul LM35 extrem de ușor de utilizat. Pentru că consumă numai  $60\text{ }\mu\text{A}$ , are o autoîncăzire foarte mică, de  $0.1^{\circ}\text{C}$  în aer fără curent. Seria LM35 este disponibilă sub forma unui ambalaj ermetic de tranzistor TO-46 sau TO-92 (figura 5.7).



**Figura 5.7** Senzorul LM35 CZ

Caracteristici:

- calibrat direct în grade Celsius;
- factor de scalare liniară de  $+10V/^{\circ}C$ ;
- precizie garantată de  $0.5^{\circ}C$  (la  $+25^{\circ}C$ );
- evaluat în intervalul de  $(-55, +150)^{\circ}C$ ;
- cost redus;
- funcționează pentru tensiune de alimentare de la 4V până la 30V;
- autoîncălzire foarte scăzută;
- neliniaritate tipică de  $\pm \frac{1}{4}^{\circ}C$ ;
- impedanță de ieșire mică,  $0.1 \Omega$  pentru 1 mA.

În figura figura 5.8 se prezintă diagrama de conexiuni a senzorului LM35 CZ.



**Figura 5.8.** Diagrama de conexiune a senzorului LM35 CZ(integrat cu tranzistor TO-92)

Senzorii LM35 pot fi utilizați ușor, ca și oricare alt senzor de temperatură, sub formă de circuit integrat. Pot fi lipiți sau cimentați pe o suprafață și temperatura lor va urmări cu o precizie de  $0.01^{\circ}C$  temperatura suprafeței.

Se presupune că temperatura ambiantă a aerului este aproape egală cu temperatura suprafeței, dacă acestea ar difera foarte mult, temperatura actuală a senzorului ar fi o valoare intermediară dintre temperatura aerului și a suprafeței.



### **3. Desfășurarea lucrării**

Pentru realizarea acestei aplicații se foloseste editorul de programe SIMATIC Manger din cadrul mediului SIMATIC Step7. Prin acestă unealtă, dezvoltatorul are acces la toate adresele de intrare și ieșire, atât analogice cât și digitale, ale PLC-ului și la o serie de adrese ce sunt alocate variabilelor temporare din program.

Pentru concarea standului la PLC se vor folosi explicațiile din figura 5.1. Pe figură sunt marcate intrarea, respectiv ieșirea procesului. Intrarea procesului se va conecta la o ieșire digitală a PLC-ului, de exemplu DO0.7. Ieșirea procesului se va conecta la un canal analogic de intrare al PLC-ului, de exemplu AI1.

#### **3.1. Configurarea PLC**

Etapele realizării lucrării de laborator sunt prezentate în cele ce urmează:

1. Se crează un nou proiect utilizând datele corespunzătoare PLC-ului folosit
2. Se realizează configurația hardware a aplicației. Acest lucru presupune stabilirea modelului unității centrale (CPU) a automatului programabil, modelului sursei de alimentare a acestuia și configurarea modulelor de intrare și ieșire

#### **3.2. Control continuu al temperaturii FB58 „TCONT\_CP”**

FB58 este o funcție de control continuu implementată de STEP 7. Această funcție este utilizată de PLC-urile SIMATIC S7 pentru a controla procesele tehnice cu intrări continue și ieșiri variabile. În timpul atribuirii parametrilor se pot activa sau dezactiva subfuncții ale regulatorului PID pentru o mai bună adaptare a acestuia la proces.

Pentru acordarea regulatorului se poate folosi blocul DB58. Acesta permite, pe lângă setarea parametrilor regulatorului, folosirea cătorva opțiuni de control a blocului regulator precum și monitorizarea grafică a parametrilor regulatorului.

Funcțiile regulatorului se bazează pe algoritmul de control PID al unui regulator cu semnal analogic, și se poate activa dacă este necesar un generator de impuls, pentru a obține semnale de ieșire modulate în timp (PWM). Efectele proporțional, integrator și derivativ sunt conectate în paralel, iar acest lucru permite configurarea de regulatoare P, PI, PD, PID. Pot fi implementate și regulatoare pure I sau D. În figura următoare este prezentată schema bloc a regulatorului. În aceasta se pot observa toate mărimele de intrare și ieșire precum și funcțiile suplimentare pe care această funcție le contine. În program această funcție este apelată pentru implementarea unui regulator PI, cu observația că se trece comanda pe manual în cazul în care se dorește realizarea identificării sistemului.. În acest scop s-au utilizat următoarele intrări: variabila de intrare (PV\_IN), valoarea presetată (SP\_INT), selector control automat MAN\_ON (în cazul rulării pentru monitorizare), selector control manual MAN\_ON (în cazul în care se urmărește realizarea identificării sistemului), intrarea MAN (pe care se aplică o treaptă de identificare), valoarea parametrului proporțional (GAIN), valoarea parametrului integrator (TI) și timpul dintre apelurile succesive ale blocului. Ca parametru de ieșire fiind comanda PWM (QPULSE) și va fi aplicată pe retelele care comandă încălzitorul. Se precizează că mai trebuie setați parametrii precum perioada PWM, PER\_TM din blocul asociat DB 58, care în cazul de față este 2 secunde, setarea de activare PWM PULSE\_ON=1.

Blocul FB58 implementează regulatorul a cărui structură este prezentată în lucrarea 4, secțiunea 3.2.

### 3.3. Programarea PLC

1. Se inserează blocul OB35 (din meniul *Insert -> S7 Block -> Organization Block*) care este un bloc de organizare a cărui execuție ciclică se face la intervale de timp specificate (întreruperi).
2. Se intră în OB 35 și se selectează din fereastra din dreapta: *Libraries ->Standard Libraries ->PID Control Blocks* funcția bloc a regulatorului FB41.

3. Din blocul de date DB58 se vor citi valorile intrării analogice (variabila PV) la temperatura minimă (aprox. 24 °C) și maximă (aprox. 100 °C) date ce vor fi folosite în continuare pentru scalare
4. Se crează un bloc FC1 în care se realizează funcția de scalare a referinței după formula:

$$PV = aT + b$$

unde a și b sunt coeficienți determinați în urma aplicării ecuației dreptei în cele două puncte de măsură iar T este valoarea referinței de temperatură (vezi lucrarea 3 secțiunile 3.2 și 3.3)

5. Funcția de scalare a referinței (FC1) se apelează în interiorul blocului OB1

#### **3.4. Identificarea experimentală a procesului**

1. Se realizează interfața WinCC. Dezvoltarea aplicației de monitorizare în WinCC Flexible care va cuprinde un grafic cu evoluția mărimilor de interes din proces (referință de temperatură, temperatură măsurată, comanda regulatorului, abaterea) și va permite modificarea online a referinței de temperatură (vezi lucrarea 4 secțiunea 3.5)
2. Se identifică funcția de transfer a procesului folosind programul realizat în WinCC. Regulatorul PID (FB58) se pune pe modul manual (MAN\_ON să fie TRUE) și se dă o comandă de 15% (aproximativ 1.5 V) (vezi lucrarea 3 secțiunea 3.4)
3. Se acordează regulatorul iar parametrii acestuia se asignează în blocul de date DB58. Pe lângă aceasta mai trebuie făcută configurarea funcției bloc FB58 în concordanță cu specificul procesului (vezi lucrarea 3 secțiunea 3.5)
4. Testare sistem de control și monitorizare

#### **4. Conținutul referatului**

Se va descrie modul de realizare al aplicației. Referatul va cuprinde referiri la urmatoare etape din realizarea aplicației:

1. Identificarea sistemului și determinarea unei model al procesului
2. Acordarea regulatorului
3. Scalarea referinței de temperatură din domeniul 25-120 °C în domeniul [0...100]%
4. Implementarea controlului automat al temperaturii în Step 7
5. Dezvoltarea unei aplicații de monitorizare în WinCC Flexible care va cuprinde un grafic cu evoluția mărimilor de interes din proces (referință de temperatură, temperatura măsurată, comanda regulatorului, abaterea) și va permite modificarea online a referinței de temperatură
6. Testarea sistemului de control și monitorizare
7. Concluzii

## 6. Reglarea turației motorului de curent continuu



### 1. Scopul lucrării

Obiectivul acestei lucrări este realizarea unei aplicații de reglare și monitorizare a turației unui motor de current continuu cu ajutorul unui PLC Siemens SIMATIC S7-300. Scopul principal al aplicației este familiarizarea cu un sistem de achiziție de date realizat cu PLC, identificarea experimentală a procesului și implementarea unui sistem de control pentru un proces rapid.



### 2. Considerații teoretice

PLC-ul Simatic S7-300 a jucat un rol major în dezvoltarea aplicației de reglare a turației. Cu ajutorul PLC-ului se poate realiza achiziția datelor în vederea identificării experimentale a procesului, implementarea regulatorului necesar buclei de reglare a procesului și realizarea interfeței grafice HMI, cu ajutorul căreia se va realiza funcția de monitorizare.

În mediul industrial, această soluție este aplicabilă și este una practică, deoarece este des întâlnită în sisteme de reglare a turației motoarelor de curent continuu: viteza unei benzi transportoare dintr-o fabrică, controlul unei stații de mixare, controlul unor pompe industriale, roboți industriali, etc.

#### 2.1. Motorul de curent continuu

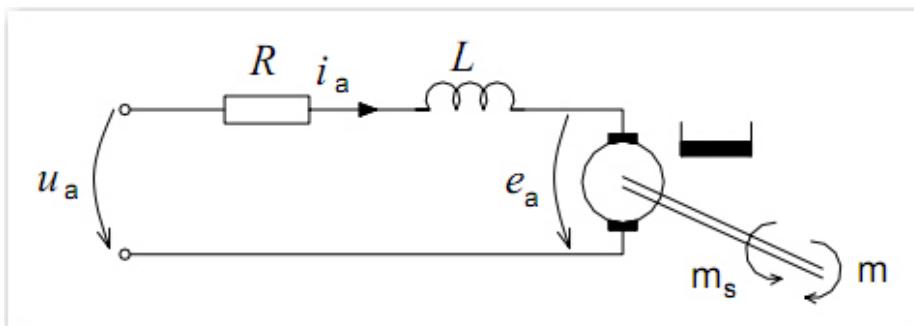
Un motor electric (sau electromotor) este un dispozitiv electromecanic ce transformă energia electrică în energie mecanică. Transformarea în sens invers, a energiei mecanice în energie electrică, este realizată de un generator electric. Nu există diferențe de principiu semnificative între cele două tipuri de mașini electrice, același dispozitiv putând îndeplini ambele roluri în situații diferite. Motorul de curent continuu este alcătuit în principal din două subansambluri: subansamblul statoric (produce campul magnetic inductor) și subansamblul rotoric (indusul). Înfășurarea generatorului care produce

câmpul magnetic inductor, numită înfăşurare de excitaţie, este situată în stator şi este alimentată în curent continuu. Câmpul magnetic inductor statoric poate fi produs şi cu ajutorul unor magneţi permanenţi. Înfăşurarea de excitaţie a maşinii de curent continuu poate fi alimentată în mai multe feluri: de la surse exterioare de curent continuu, când se consideră că avem excitaţie separată, sau chiar de la bornele generatorului de curent continuu, când se zice că avem generator cu autoexcitaţie. În ultimul caz înfăşurarea de excitaţie poate fi conectată în paralel, în serie sau mixt faţă de înfăşurarea rotorică.

Înfăşurarea rotorică a generatorului de curent continuu, numită induşul maşinii, în care se induce tensiunea electromotoare este o înfăşurare specială de curent continuu care se conectează la un colector prevăzut cu mai multe segmente de inel izolate între ele. Rolul colectorului este acela de a redresa mecanic tensiunea electromotoare induşă în rotor şi de a furniza în exterior o tensiune constantă (curent continuu).

## 2.2. Modelarea procesului

Se consideră motorul de curent continuu cu excitaţie cu magneţi permanenţi din Figura 6.1. Mărimea de comandă este tensiunea de alimentare  $u_a$  şi cea de ieşire, viteza unghiulară sau turatia motorului  $\Omega$ .



**Figura 6.1.** Schema echivalentă a motorului de curent continuu

Ecuațiile motorului sunt:

$$U_a = R \cdot i_a + L \frac{di_a}{dt} + e_a$$

$$J \frac{d\Omega_r}{dt} = m - m_s - m_d$$

unde,

$$e_a = k \cdot \Omega$$

$$m = k \cdot i_a$$

$$m_d = F \cdot \Omega,$$

$R$  reprezintă rezistență iar  $L$  inductanță înfășurării rotorice a motorului,  $i_a$  este curentul absorbit de motor,  $e_a$  reprezintă tensiunea contraelectromotoare dezvoltată de motor,  $m$  este cuplul activ dezvoltat de motor. Tensiunea contraelectromotoare este proporțională cu turația motorului, constanta  $k$  depinzând de construcția mașinii. Cuplul dezvoltat de motor este proporțional cu valoarea curentului prin motor, constanta de proporționalitate  $k$  fiind aceeași cu cea de la tensiunea contraelectromotoare.

$m_s$  reprezintă cuplul rezistiv static(constant) care apare de exemplu la sușinerea unei greutăți, iar  $m_d$  este cuplul rezistent dinamic ce crește proporțional cu viteza motorului.

$$m_d = F^* \Omega,$$

unde  $F$  este coeficientul frecării vâscoase.

$J$  – este momentul de inerție cumulate al motorului și sarcinii.

Mărimile de stare sunt curentul prin motor  $i_a$  și viteza unghială sau turația acestuia  $\Omega$ .

Pentru a scoate în evidență mărimile de stare, ecuațiile sistemului se pot scrie sub forma:

$$\begin{cases} L \frac{di_a}{dt} = -R \cdot i_a - k \cdot U_a \\ J \frac{d\Omega}{dt} = k \cdot i_a - F \cdot \Omega - m_s \end{cases}$$

Aplicând transformata Laplace ecuațiilor ce descriu funcționarea motorului, vom avea:

$$Li_a s = -Ri_a - k \cdot \Omega + U_a$$

$$J\Omega s = ki_a - F \cdot \Omega - m_s$$

Dacă eglijăm  $m_s$  obținem,

$$\begin{aligned} i_a &= \frac{J}{k} \Omega \cdot s + \frac{F}{k} \Omega \\ \frac{LJ}{k} \cdot \Omega s^2 + \left( \frac{LF}{k} + \frac{RJ}{k} \right) \Omega s &= -\frac{RF}{k} \Omega - k\Omega + U_a \\ \Omega \left( \frac{LJ}{k} s^2 + \frac{LF + RJ}{k} s + \frac{RF}{k} + k \right) &= U_a \end{aligned}$$

și de aici funcția de transfer de la tensiunea de comandă la viteza unghiulară sau turație :

$$\frac{\Omega}{U_a} = \frac{1}{\frac{LJ}{k} s^2 + \frac{LF + RJ}{k} + \frac{RF + k^2}{k}} = \frac{k}{LJs^2 + (LF + RJ)s + RF + k^2}$$

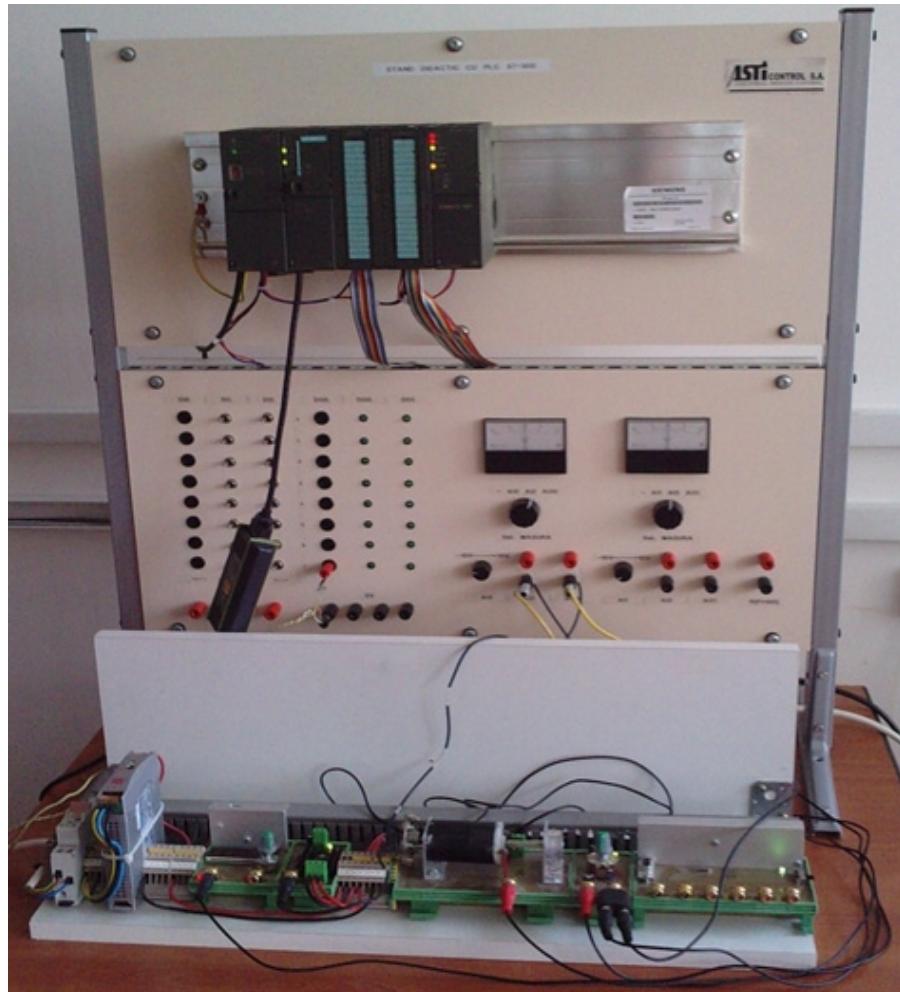
### 2.3. Descrierea standului

În această lucrare se utilizează un stand didactic care include un automat programabil Siemens din seria SIMATIC S7-300, pentru conducerea procesului, un motor de curent continuu și un PC pentru achiziționarea de date și monitorizare (figura 6.2).

Instalația de față este compusă din:

- un stand didactic ce înglobează un motor de current continuu, o sursă de alimentare, redresoare de curent continuu, un circuit de comandă pentru motor, un potențiometru pentru o reglare manuală a turației și un filtru necesar pentru eliminarea perturbațiilor;

- un automat programabil din seria SIMATIC S7-300 încadrat pe un panou ce conferă ușurință și transparentă în utilizarea resurselor acestui PLC;
- conectori izolați ce asigură partea de conexiuni dintre PLC și standul motorului;
- un calculator cu care se realizează programarea PLC-ului, interfațarea și monitorizarea procesului;
- un modul MPI/DP ce asigură comunicația dintre calculator și automatul programabil;
- panoul sinoptic ce asigură accesul la resursele și funcțiilor automatului programabil.



**Figura 6.2.** Stand didactic cu motor de curent continuu

## 2.4. Conexiuni

Comanda motorului se face prin intermediul unuia dintre canalele analogice de ieșire ale PLC-ului. Se va realiza conectarea terminalelor de alimentare ale motoului la bornele, de pe panoul PLC-ului, ale canalului AO0. Măsurarea turatiei motorului se realizează prin conectarea terminalelor tahogeneratorului la o intrare analogică a PLC-ului (de exemplu AI2).



## 3. Desfășurarea lucrării

Turația, măsurată de motorul de curent continuu în regim de generator constituie mărimea de ieșire a sistemului și semnal de intrare pentru regulatorul implementat în Step 7. Mărimea de comandă analogică este amplificată și apoi aplicată motorului de curent continuu. Toate componentele aplicației vor fi implementate folosind mediul SIMATIC Step7 pentru programare și SIMATIC WinCC Flexible pentru control și monitorizare.

Etapele realizării lucării de laborator sunt descrise în cele ce urmează.

1. Se realizează conectarea terminalelor standului experimental la panoul PLC-ului.
2. Se crează un nou proiect utilizând datele corespunzătoare PLC-ului folosit
3. Se realizează configurația hardware a aplicației. Acest lucru presupune stabilirea modelului unității centrale (CPU) a automatului programabil, modelului sursei de alimentare a acestuia și configurarea modulelor de intrare și ieșire
4. Se crează blocul OB35 în care vom avea regulatorul PID – FB41. Acesta se selectează din fereastra din dreapta: *Libraries ->Standard Libraries ->PID Control Blocks* (vezi lucrarea 3 secțiunile 3.2 și 3.3)
5. Scalarea referinței de turație. Se consideră că valoarea măsurată a turației de 25% reprezintă 1000 rot/min. Se va realiza un bloc de conversie ce va permite ca intrarea referinței stabilite de către operator să fie în domeniul rot/min. Ieșirea acestui bloc va fi în

domeniul procentual (va constitui intrare de referință pentru blocul FB41)

6. Se realizează interfața WinCC. Dezvoltarea aplicației de monitorizare în WinCC Flexible care va cuprinde un grafic cu evoluția mărimilor de interes din proces (referință de nivel, nivelul măsurat, comanda regulatorului, abaterea) și va permite modificarea online a referinței de turație
7. Identificarea experimentală a procesului. Se identifică funcția de transfer a procesului folosind o aplicație în Step 7 prezentată în Anexa 3. Se va da o treapta de tensiune de 1.8 V la intrarea procesului și se va înregistra răspunsul acestuia folosind o perioadă de esantionare de 0.01 secunde. Modelul procesului se va aproxima cu o funcție de transfer de ordinul 2 de forma:

$$H_f(s) = \frac{k_f}{(T_{f1}s + 1)(T_{f2}s + 1)}$$

Pentru realizarea identificării modelului procesului din datele obținute pe cale experimentală poate fi utilizat tool-ul Matlab Ident. Va trebui să se țină cont de folosirea unui domeniu unitar de prezentare a semnalelor de intrare și ieșire. Dacă semnalele nu sunt prezentate în același domeniu se va realiza conversia acestora spre exemplu la domeniul procentual.

8. Se facea acordarea regulatorului iar parametrii acestuia se asignează în blocul de date DB41. Pe lângă aceasta mai trebuie făcută configurarea funcției bloc FB41 în concordanță cu specificul procesului (conectarea semnalelor de intrare/ieșire) (vezi lucrarea 3 secțiunea 3.5).
9. Testare sistem de control și monitorizare.

#### **4. Conținutul referatului**

Se va descrie modul de realizare al aplicației. Referatul va cuprinde referiri la următoare etape din realizarea aplicației:

1. Identificarea sistemului și determinarea unei model al procesului
2. Acordarea regulatorului
3. Implementarea controlului automat al turației în Step 7
4. Dezvoltarea unei aplicații de monitorizare în WinCC Flexible care va cuprinde un grafic cu evoluția mărimilor de interes din proces (referință de turatie, turația măsurată, comanda regulatorului, abaterea) și va permite modificarea online a referinței de turație
5. Testarea sistemului de control și monitorizare
6. Concluzii

## 7. Anexa 1. Descriere dispozitive stand FESTO

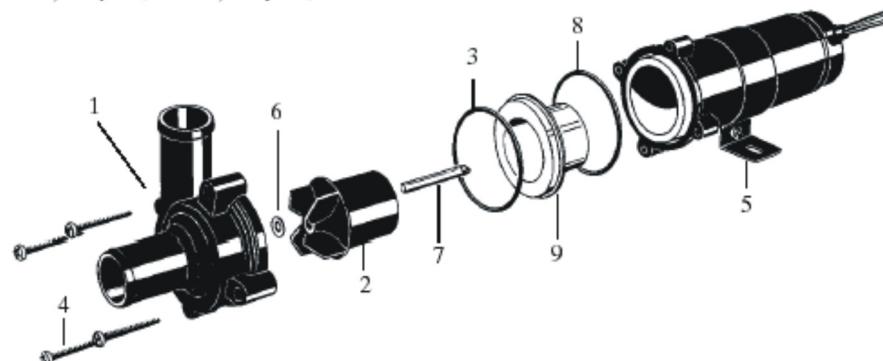
### *Pompa centrifugă*

Standul dispune de o pompă centrifugă autoîncapsulată ce trebuie amorsată înainte de a fi pusă în funcțiune. Se va evita funcționarea ei pe uscat deoarece după o perioadă de 30 de minute acest lucru poate provoca defectarea pompei.



**Figura 7.1.** Pompa centrifugă

Pompa trebuie să funcționeze tot timpul în direcția prescrisă. Motorul permite operarea continuă. Pompa poate fi montată atât în poziție verticală cât și în poziție orizontală. În condițiile în care este montată în poziție orizontală este important ca ieșirea pompei să fie orientată în sus.



**Figura 7.2.** Componentele pompei centrifuge

## Anexa 1

În figura 7.2 este reprezentată pompa centrifugă în detaliu. Este alcătuită din următoarele componente :

1: Carcasă cu diametrul de 20 mm; 2: Lamelă subțire; 3: Sigiliu inelar; 4: Șurub; 5: Garnitura motorului; 6: Pană de fixare; 7: Mâner de control; 8: Șaibă; 9: Solenoid.

Parametru	Valoare
Carcasă	Plastic
Mâner de control	Oțel
Lamela	Corp: plastic Magnet: ferită
Carcasă motor	Oțel
Motor	Motor cu magnet permanent 12/24 V
Conexiune	20 mm (3/4")
Atenuare sunet	EN 55014
Clasă de protecție	IP67 (DIN 40050)
Temperatura de funcționare: Lichid	-40°C la +100°C
Exterior	-40°C la +70°C
Presiune maximă	2,5 bar
Alimentare	24 V
Putere	24 W

**Tabelul 7.1.** Date tehnice pompă

Presiune (bar)	Debit (l/min)	Curent la 24V (A)
0,1	26	1,1
0,2	19,5	1,0
0,3	9,0	0,75

**Tabelul 7.2.** Debitul pompei în funcție de presiune

Valorile au fost măsurate pentru o conexiune de 20 mm.

### ***Controller motor pompă centrifugă***

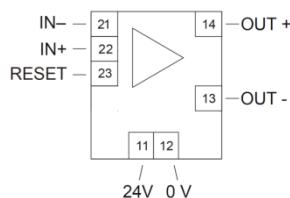
#### *Funcție*

Controller-ul permite varierea tensiunii de ieșire și astfel se poate controla debitul pompei prin modificarea turației motorului. Erorile se pot reseta folosind intrarea de reset (RESET). Aplicând o tensiune de 0V pe intrarea reset eroarea va fi stearsă. Controller-ul este montat pe o shină top-hat.



**Figura 7.3.** Controller pompa centrifugă

#### *Date tehnice:*



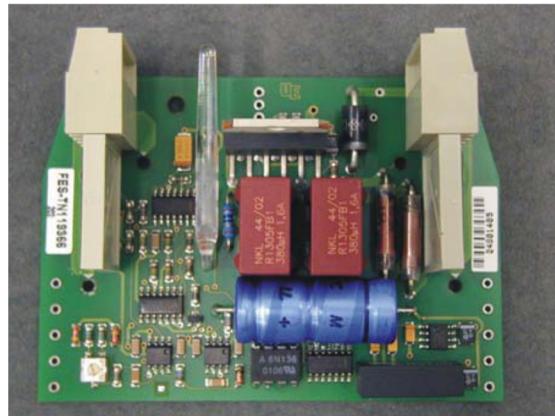
**Figura 7.4.** Alocarea conexiunilor

#### *Ajustarea controler-ului pompei (amplificatorul A4)*

Pe placa controller-ului există un potențiometru (figura 7.5). Acesta se poate ajusta cu ajutorul unei șurubelnite. Scopul ajustării este acela de a seta ieșirea controlerului la 0 V pentru o intrare de 0 V și pentru a avea la ieșire o tensiune de 24 V pentru o intrare de 10 V.

## Anexa 1

---



**Figura 7.5.** PCB controller

Parametru	Valoare
Tensiunea de operare	24 VDC
Intrare	-10...+10 VDC
Ieșire	-24...+24 VDC
Curent de ieșire	Max. 1 A

**Tabelul 7.3.** Date tehnice controler

### **Transformator de măsură curent/tensiune**

#### *Funcție*

Transformatorul de măsură (figura 7.6) convertește valoarea măsurată de senzorului ultrasonic (curent unificat 4 – 20 mA) într-o valoare de tensiune între 0 și 10 V.



**Figura 7.6.** Convertor curent tensiune

## Anexa 1

---

### Date tehnice

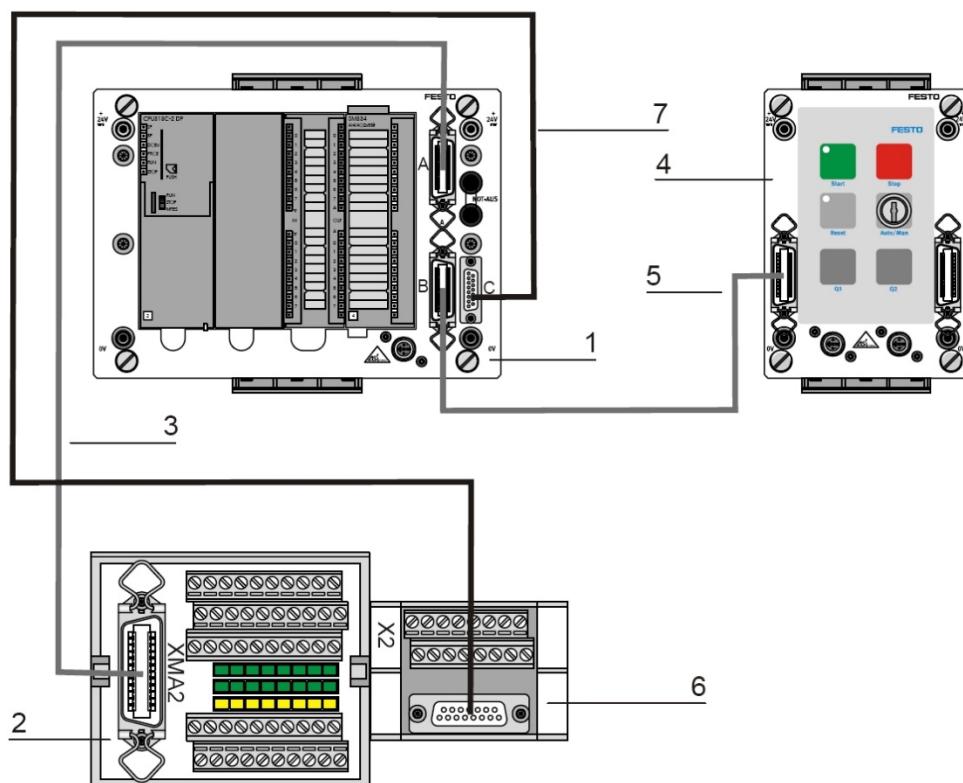
Parametru	Valoare
Semnal de intrare	4...20mA
Curent de intrare	22 mA
Rezistență de intrare	<400 Ω
Tensiunea maxima pe linie	<8 V
Semnal de ieșire	0...10 V
Sarcină	>2 kΩ
Detectie cablu întrerupt	LED verde = off
Eroare transmisie	<0,15 %/<0,1 %
Coeficient de temperatură	<0,02 %/K
Barieră de frecvență	1 kHz
Tensiune de izolare intrare/ieșire	4 kV, 50 Hz, 1min
Sursa de alimentare RW<6%	DC 20 V...30 V
Temperatură de funcționare	0°C...+55°C

**Tabelul 7.4.** Date tehnice pentru transformatorul curent/tensiune

### *Schema de conexiuni*

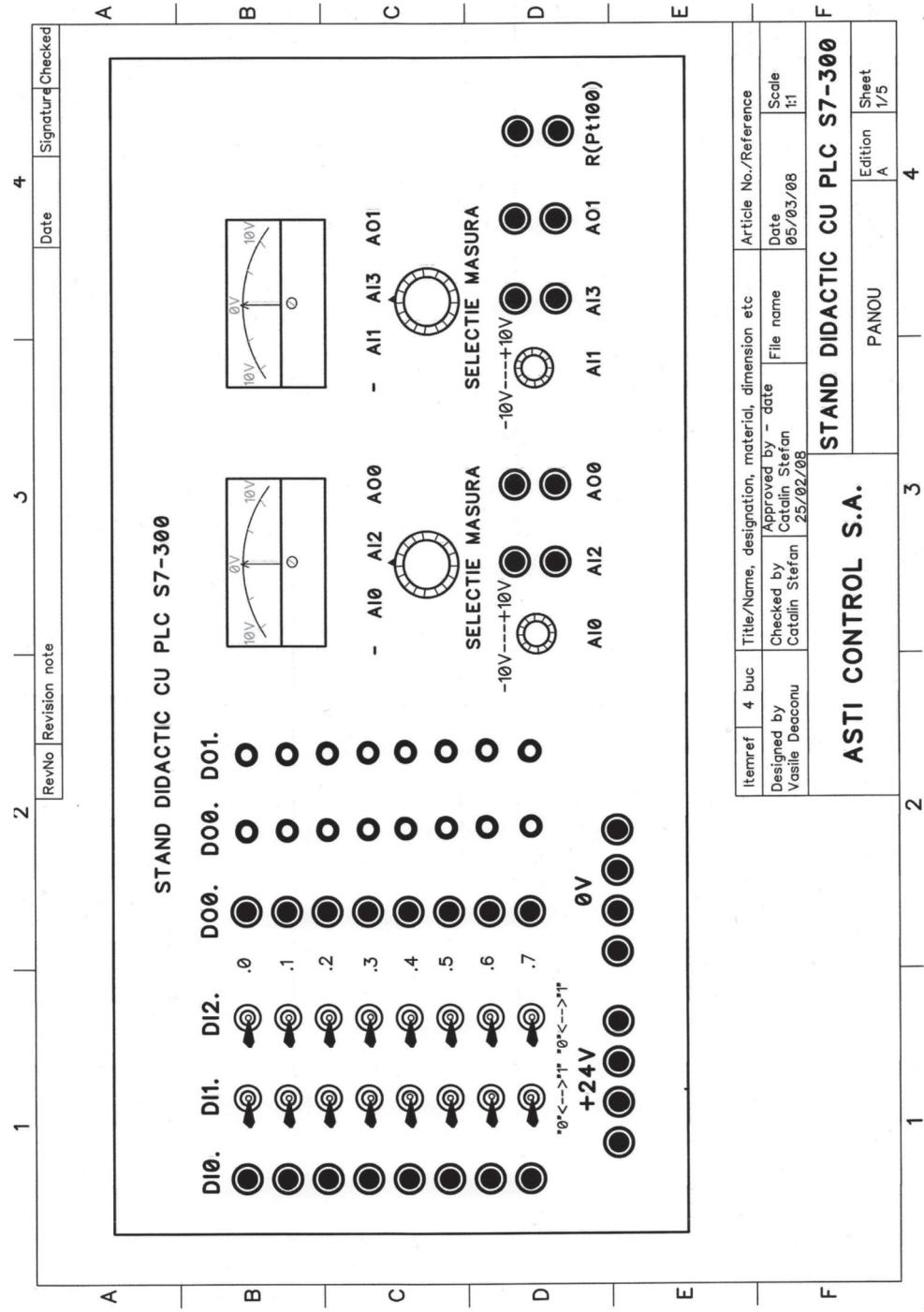
Figura 7.7 prezintă schema conexiunilor electrice dintre PLC, panoul de comandă al standului și conectorii XMA2 și X2.

## Anexa 1

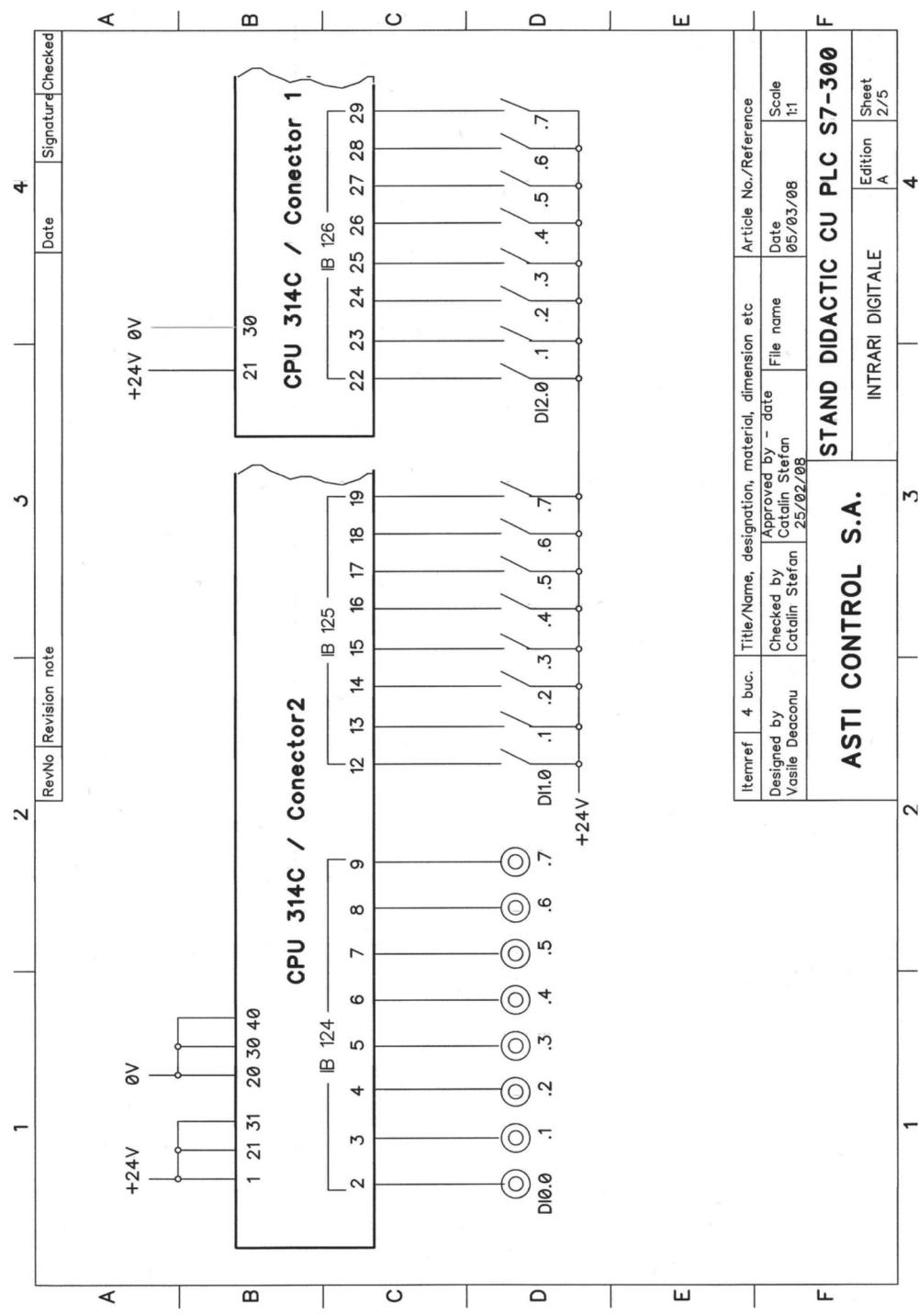


**Figura 7.7.** Conexiuni între PLC, panoul de comandă și conectori

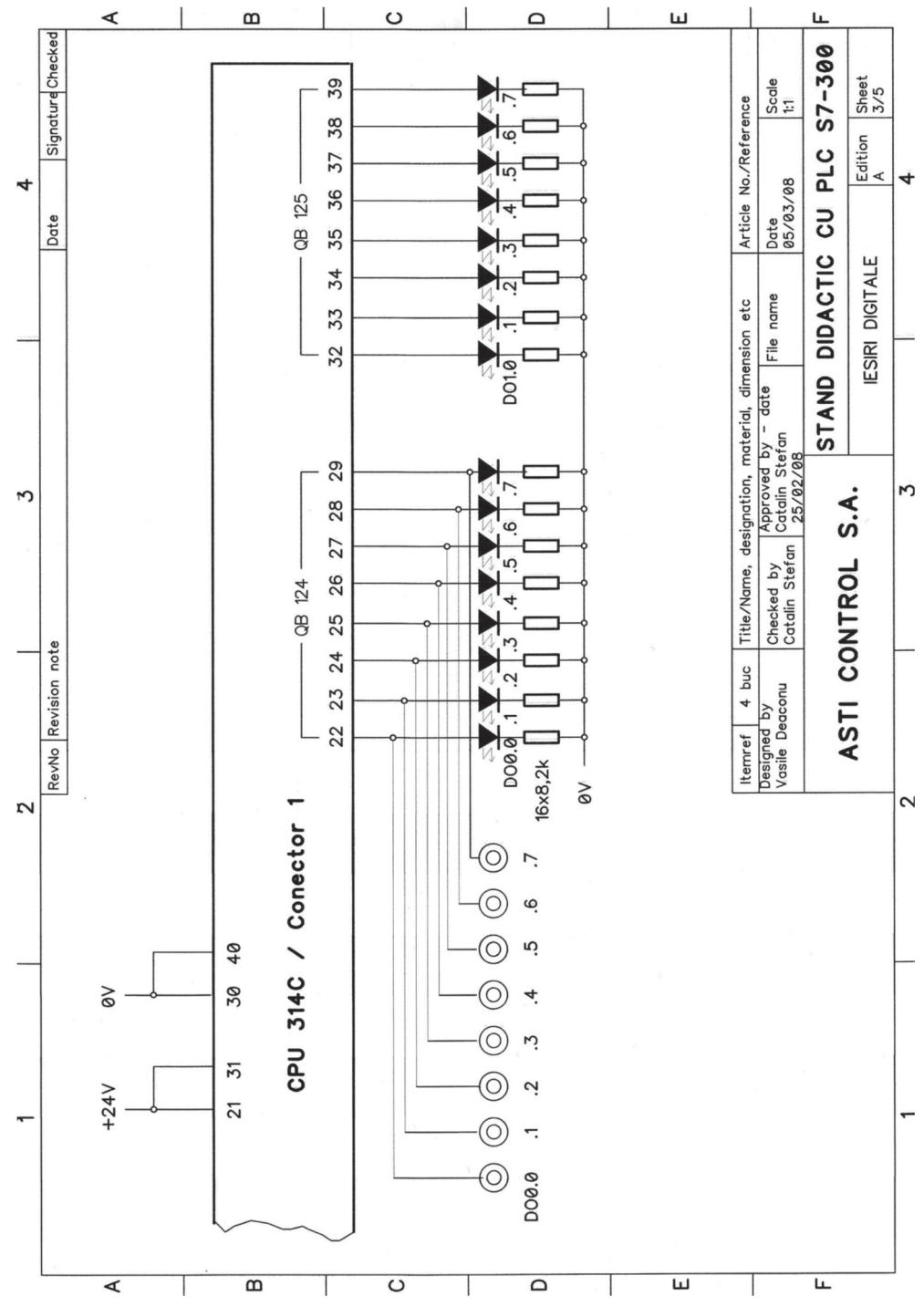
## 8. Anexa 2. Stand cu PLC S7-300

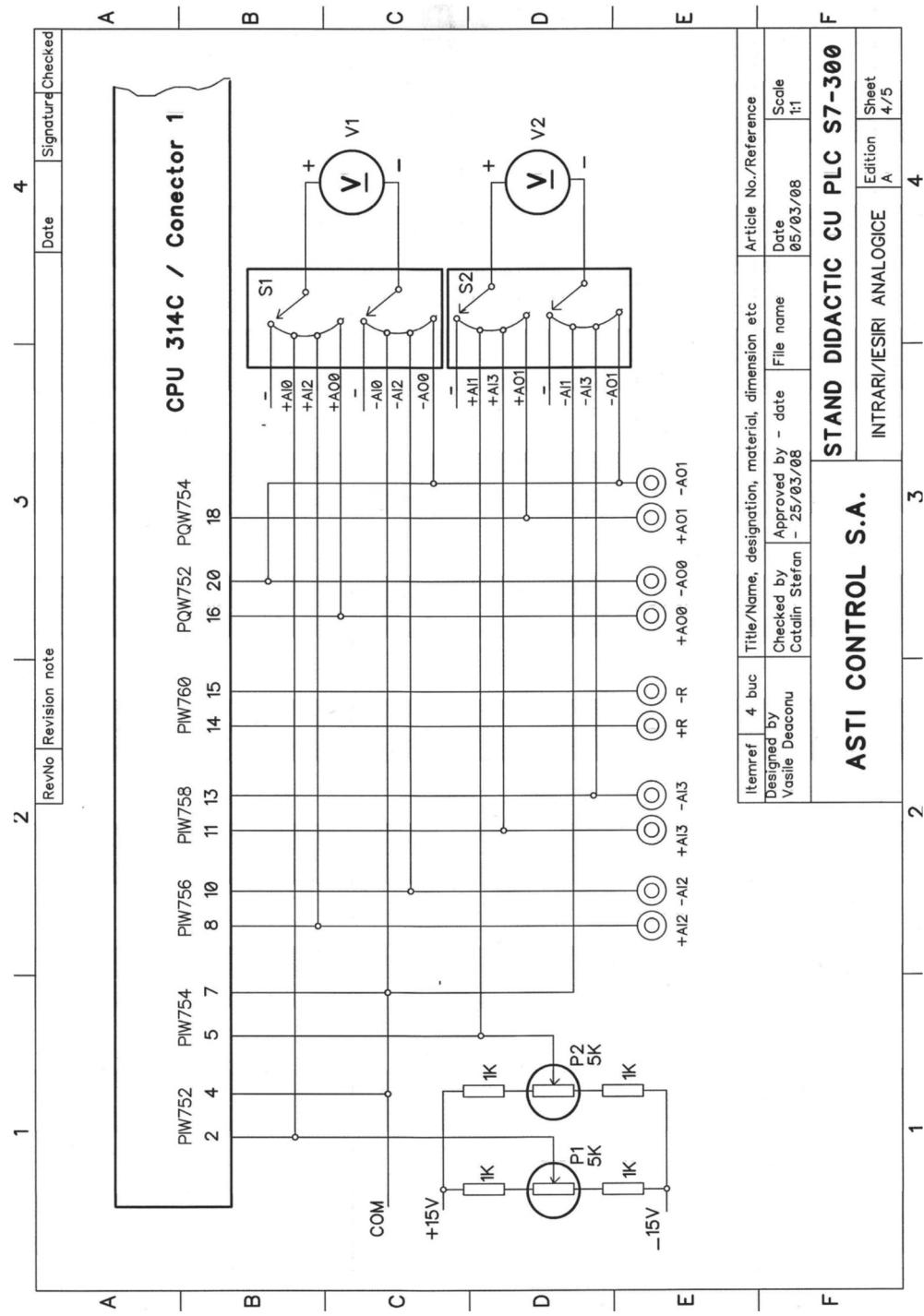


<b>STAND DIDACTIC CU PLC S7-300</b>			PANOU	Edition	Sheet
Itemref	4 buc	Title/Name, designation, material, dimension etc	Article No./Reference		
Designed by		Approved by			
Vasile Deaconu		Checked by	Catalin Stefan	Date	Scale
			25/02/08	05/03/08	1:1
<b>ASTI CONTROL S.A.</b>					
1	2	3	4		

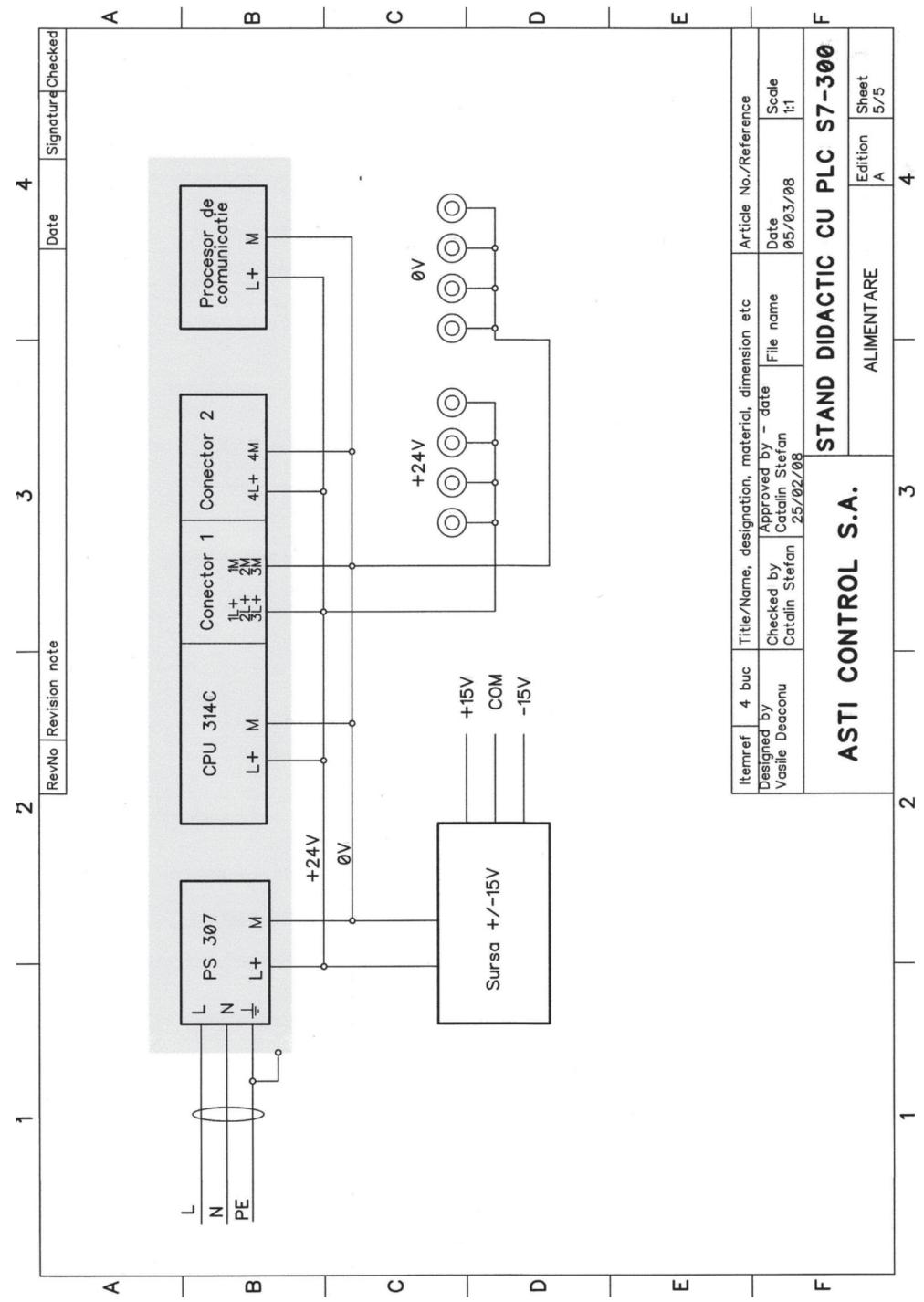


Stand cu PLC S7-300





Stand cu PLC S7-300



## 9. Anexa 3. Achiziție de date

Se va deschide din structura proiectului secțiunea S7 Program și de aici secțiunea Sources.

Se va crea un fișier sursă SCL accesând meniul *Insert->S7 Software->SCL Source* în care se va insera secvența de cod care urmează.

```
DATA_BLOCK DB30
STRUCT
    step: ARRAY [1..2000] OF REAL;
    data: ARRAY [1..2000] OF REAL;
    index: INT;
    step_sample: INT;
    samples: INT;
END_STRUCT
BEGIN
END_DATA_BLOCK

ORGANIZATION_BLOCK OB100
VAR_TEMP
    // Reserved
    info : ARRAY[0..19] OF BYTE;
    // Temporary Variables
END_VAR
DB30.index:=0;
END_ORGANIZATION_BLOCK

FUNCTION FC30 : VOID
TITLE='Data aquisition'
VERSION : '1.0'
AUTHOR: RubenC
NAME: Aq
FAMILY: DataAq
KNOW_HOW_PROTECT
CONST
    n:=2000;
END_CONST
VAR_TEMP
    ind: INT;           // actual sample
    index: INT;         // total number of samples
    aux: INT;           // auxiliary variable
END_VAR
VAR_INPUT
    initial_step: REAL; // initial step
    final_step: REAL;   // final step
```

## Bibliografie

---

```
step_time: INT;           // step time in ms
sampling_time: INT;       // sampling time in ms
step_period: INT;         // step period in ms
PV: INT;                  // process value
Reset: BOOL;              // reset aquisition procedure
END_VAR
VAR_OUTPUT
    MV: WORD;             // manipulated value
END_VAR
BEGIN
index:=DB30.index;
IF (index=0) THEN
    DB30.step_sample:=step_time/sampling_time;
    DB30.samples:=step_period/sampling_time;
END_IF;
ind:=0;
IF (Reset=false AND index<DB30.step_sample) THEN
    aux:=REAL_TO_INT(initial_step*276.48);
    MV:=INT_TO_WORD(aux);
END_IF;
IF (Reset=false AND index<>-1 AND index>=DB30.step_sample AND
index<DB30.step_sample+DB30.samples) THEN
    aux:=REAL_TO_INT(final_step*276.48);
    MV:=INT_TO_WORD(aux);
    ind:=index-DB30.step_sample+1;
    DB30.step[ind]:=final_step-initial_step;
    DB30.data[ind]:=INT_TO_REAL(PV)*100/27648;
END_IF;
IF (Reset=false AND index<>-1 AND index<n) THEN
    DB30.index:=index+1;
ELSE
    DB30.index:=-1;
END_IF;
IF (Reset=true) THEN
    DB30.index:=0;
END_IF;
END_FUNCTION
```

Este necesară stabilirea perioadei de eşantionare și specificarea acestui lucru prin modificarea perioadei de execuție a blocului de organizare OB35 (vezi figura 2.17, pag. 56).

Se crează blocul OB35 accesând meniul *Insert->S7 Block->Organization Block* și se editează conform figurii 9.1.

## Bibliografie

Network 1: Title:

```
Treapta initiala: initial_step [*]
Treapta finala: final_step [*]
Timpul la care este data treapta: step_time [ms]
Perioada de esantionare: sampling_time [ms]
Perioada pe care se face achizitia de date: step_period [ms]
Adresa variabilei masurate: PV [*]
Restartare achizitie de date: Reset [FALSE - functionare normala; TRUE -
resetare]
Adresa la care este aplicata treapta: MW
```

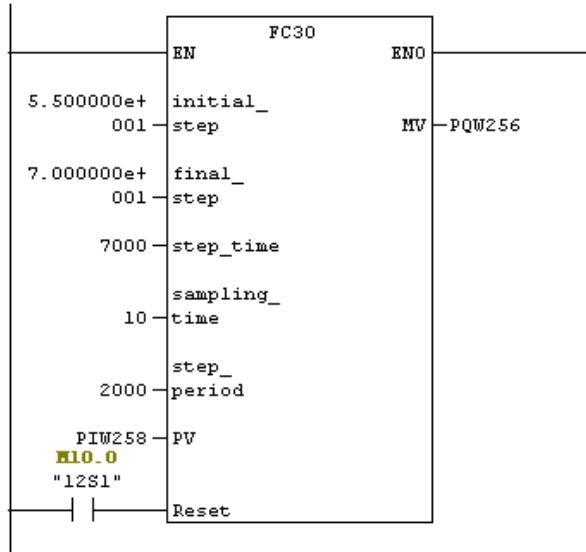


Figura 9.1. Diagrama Leadder corespunzătoare pentru OB 35

În situația prezentată (figura 9.1) comanda inițială este diferită de zero dar ea poate fi nulă în funcție de proces. Aplicarea treptei se face la timpul *step\_time*. Se stabilește amplitudinea treptei care va fi folosită pentru identificare (valoarea în domeniul procentual).

Vectorul de intrare, respectiv ieșire vor fi disponibili în DB30 în modul monitorizare. Acesta va putea fi copiat și importat în mediul Matlab pentru prelucrări ulterioare.

## 10. Bibliografie

- [1] Asavinei I., Niculescu C., Ghid pentru utilizarea termocuplurilor în măsurări industriale, Editura Tehnică, Bucureşti, 1981.
- [2] Constantin P. ş.a., Electronică industrială, EDP, Bucureşti, 1983.
- [3] Călin Sergiu ş.a., Reglarea numerică a proceselor tehnologice, E.T. Bucureşti, 1984.
- [4] SIMULINK, Dynamic System Simulation Software, User's Guide
- [5] Hans Berger, *Automating with STEP 7 in LAD and FBD*, Publicis MCD Corporate Publishing, Erlangen and Munich, 2001.
- [6] Siemens, SIMATIC HMI WinCC flexible 2008, User's Manual.
- [7] Siemens, Programming with Step7, Manual.
- [8] Siemens, Working with Step7, Manual.
- [9] Siemens, PID Control, Manual.
- [10] Siemens, PID Temperature Control, Manual.
- [11] <http://www.automation.siemens.com>