SIMATIC
STEP 7

Cuprins

SIMATIC STEP 7

1. Introducere

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 şi module de intrare şi ieşire (module I/O).

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

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

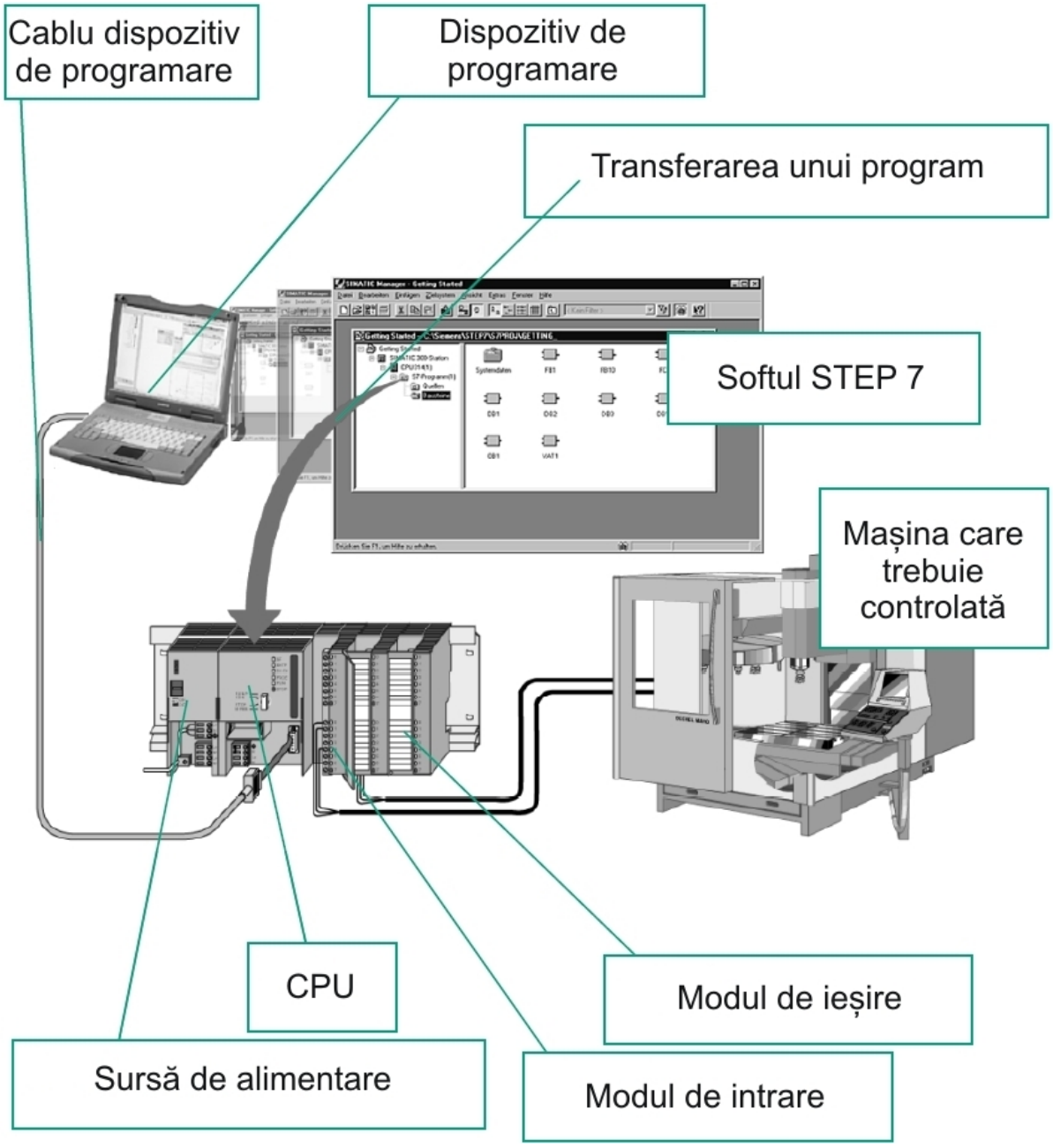


Figura 1. Viziune de ansamblu asupra unui proiect de automatizare

2. Procedura de bază folosind STEP 7

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

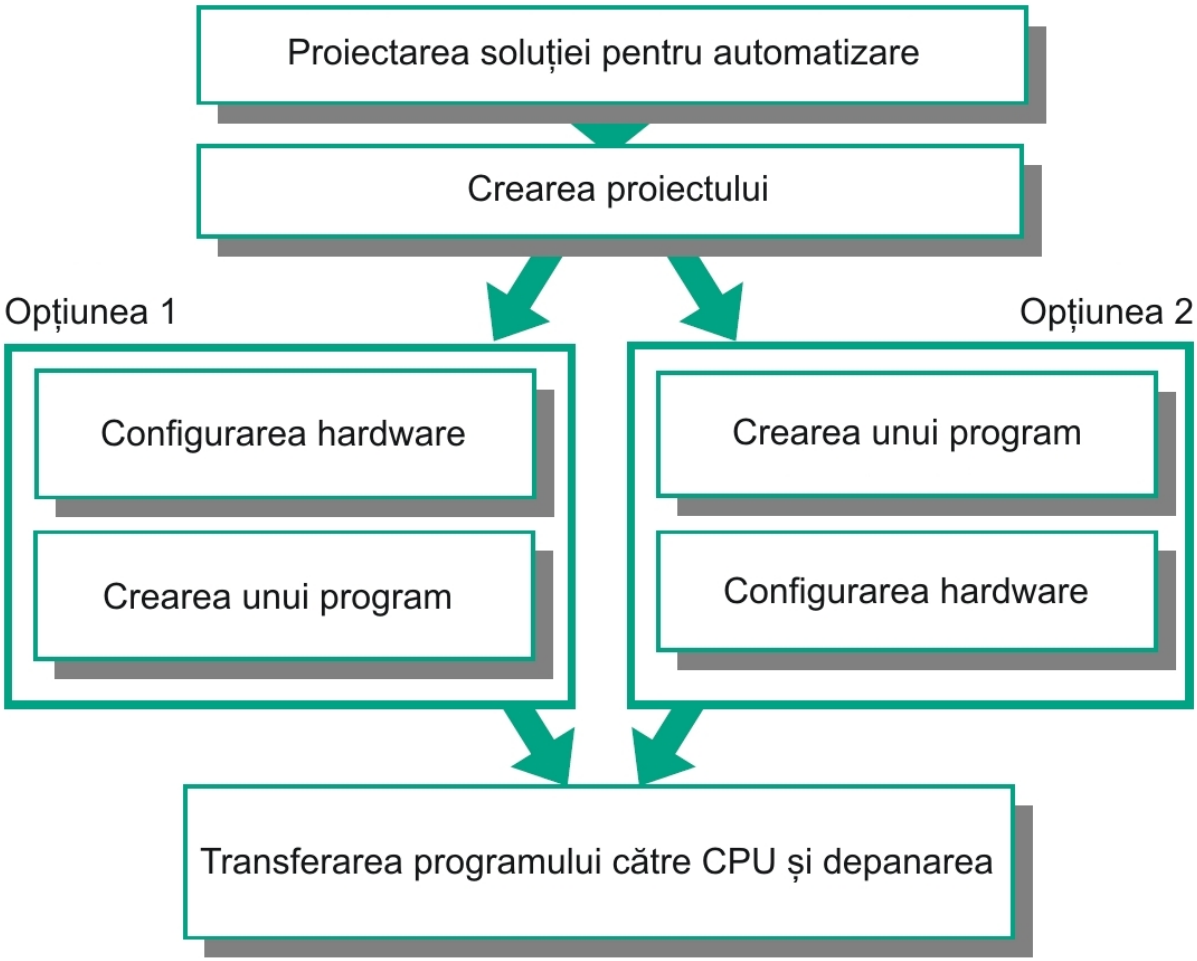


Figura 2. Crearea unei aplicații STEP 7

Dacă aplicaţia este una mai complexă, cu mai multe intrări și ieșiri, se recomandă realizarea configurării hardware la început. Avantajul este ca 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 configurarea hardware se pot defini adrese și se pot schimba parametrii și proprietățile modulelor.

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

3. Instrumente în STEP 7

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

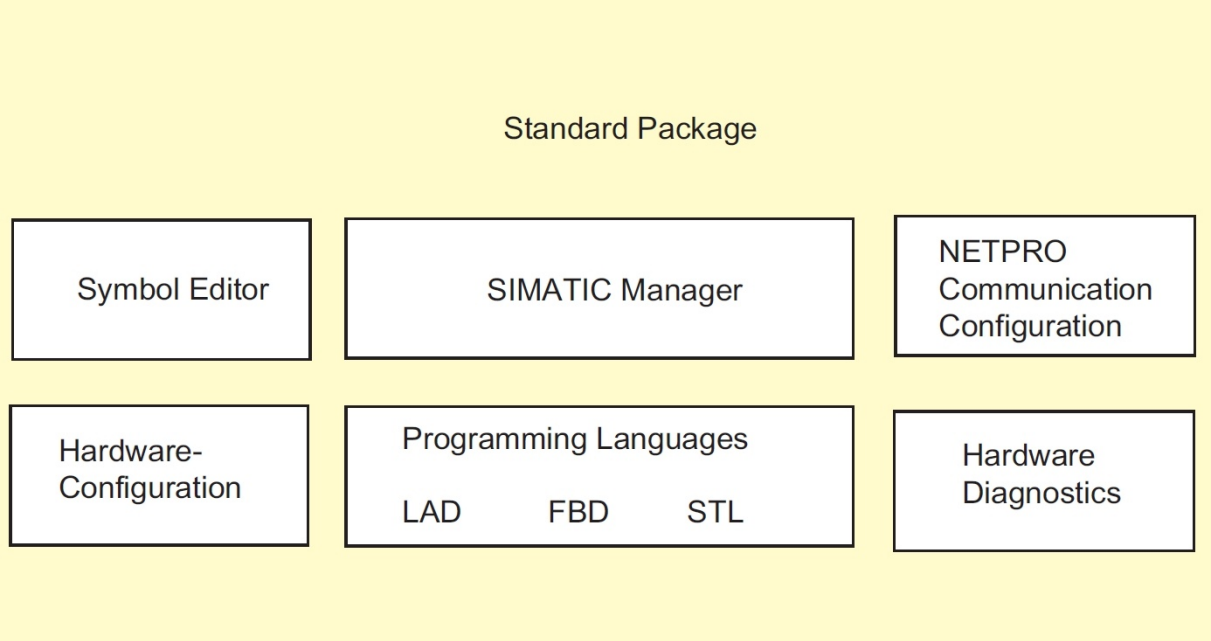


Figura 3. Instrumente STEP 7

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

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.

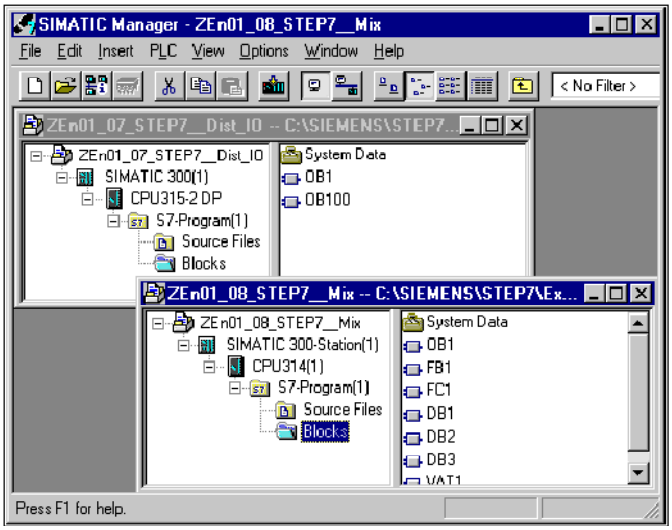


Figura 4. SIMATIC Manager

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 așadar recunoscută automat de toate celelalte instrumente.

Hardware Diagnostics

Aceste funcții oferă o imagine de ansamblu asupra stării controllerului programabil. Se poate localiza un defect și se pot obține informatii detaliate despre el.

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

Pentru CPU afișază următoarele informații suplimentare:

- Cauzele defecţiunilor la rularea unui program utilizator
- Afișează durata ciclului (al celui mai lung, mai scurt, şi a ultimului ciclu)
- Posibilităţi de comunicare pe MPI şi de încărcare
- Date de performanță (număr de intrări/ieșiri posibile, bit memorie, numărătoare, timere, și blocuri)

Limbaje de programare

Este permisă programarea în:

- Ladder Logic (LAD)
- Statment List (STL)
- Function Block Diagram (FBD)

- ✓ ***Ladder Logic*** este un limbaj grafic cu sintaxa similară cu o schema cu relee.
- ✓ ***Statment List*** este un limbaj textual. Dacă un program este scris în STL, instrucțiunile individuale corespund pașilor cu care procesorul execută programul. STL cuprinde construcții de limbaj de nivel înalt (de exemplu acces la date structurate și la parametrii blocurilor).

- ✓ ***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.

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

NetPro (Network Configuration)

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

4. 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)

STEP 7 este pachetul software standard pentru configurarea și programarea controller-elor 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.

Mediul SIMATIC Manager se pornește din Windows apăsând butonul Start, apoi se alege Siemens Automation → SIMATIC → SIMATIC Manager.

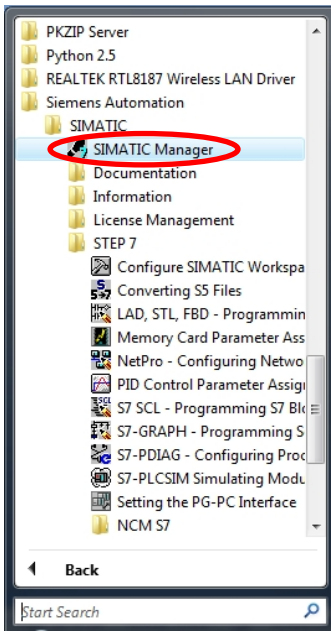


Figura 5. Pornirea programului SIMATIC Manager

4.1. Organizarea generală a programului

Modul de organizare a unui program STEP 7 are anumite particularităti 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 OB care va apărea obligatoriu în orice proiect dezvoltat în STEP 7 este OB1, un bloc care se execută într-o buclă infinită atâta timp cât CPU-ul se afla 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 şi 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 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 functie 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 instanțieri 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;

Pentru a putea proiecta o nouă aplicație va trebui să urmăm pașii următori:

- Se lansează în execuție SIMATIC Manager
- Se selectează din File comanda Wizard "*New Project*" apoi se activează butonul *Next*. Se va deschide un proiect nou.

În figura 6 sunt prezentați cei 4 pași pentru crearea unui proiect nou. Se pot observa în cele 4 ferestre alegerea tipului de unitate centrală (CPU 314C-2 DP), a blocurilor de organizare ce se dorește a fi incluse în proiect (aici OB1 și OB35) și respectiv a modului de programare: Ladder Logic (LAD), Statement List (STL) sau Function Block Diagram (FBD) . Se va alege LAD.

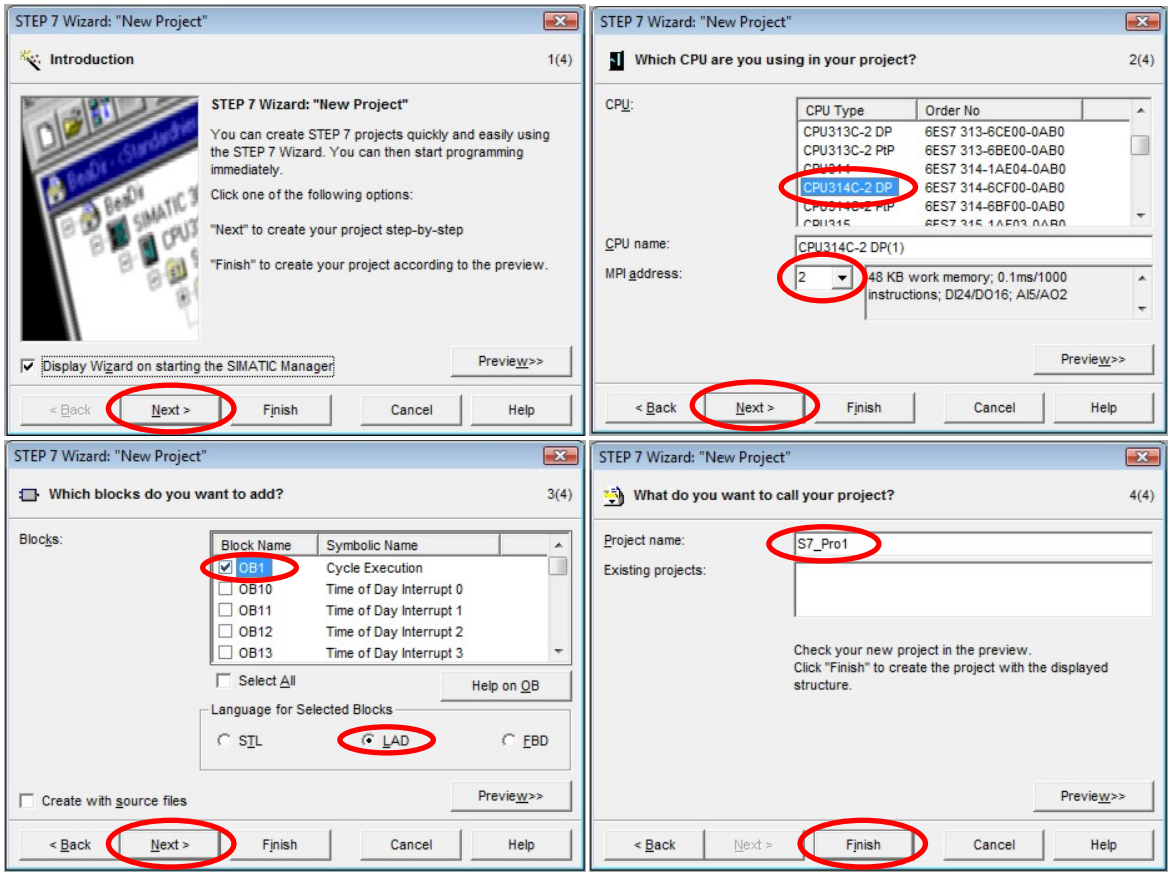


Figura 6. Creare proiect STEP 7

Primul lucru care trebuie efectuat este configurația hardware.

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. 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 proramare: **L**adder **L**ogic (**LAD**), **S**tatement **L**ist (**STL**) sau **F**unction **B**lock **D**iagram (**FBD**). Se alege **L**adder **L**ogic apoi se apasă butonul *Next*. Limbajul de programare poate să fie schimbat și ulterior.

La *Project name* se atribuie proiectului numele dorit, apoi se apasă pe butonul *Finish* care va genera noul proiect.

4.2. Configurarea hardware

Pentru orice proiect dezvoltat în STEP 7 stabilirea configuraţiei hard este primul pas care trebuie făcut. Pentru a realiza configurarea se selectează *SIMATIC 300 Station* din partea stângă a ecranului şi se dă dublu click pe simbolul *Hardware* care apare în partea dreaptă a ecranului. Se va deschide o fereastră cu numele *HW Config – SIMATIC 300 Station*.

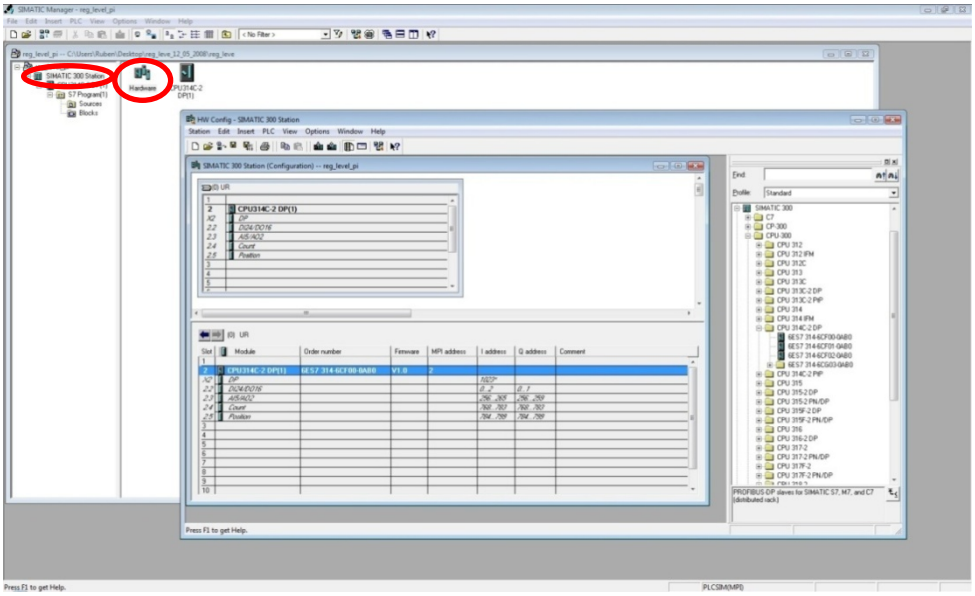
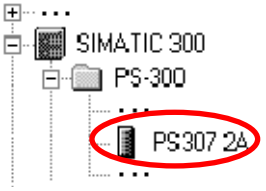


Figura 7. Realizarea configuraţiei hardware

În figura 7 se poate observa realizarea configuraţiei hardware. În partea din dreapta a ecranului se găsesc librăriile cu componentele necesare configurării. În partea din stânga vor fi afişate două tabele cu componentele hardware selectate din librării. CPU-ul care a fost selectat la crearea proiectului este afişat în ferestrele din stanga, în cazul nostru acesta este CPU 314C-2DP. Fiind un CPU care are conţine şi intrări/ieşiri analogice/numerice, vor apare şi acestea în cele două tabele.

Pentru a realiza configurarea, primul lucru de care avem nevoie este o şină, *„rack”*, pe care vor fi ataşate toate componentele. Aceasta, ca de altfel toate celelalte componente, depind de staţia 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, şi deci toate componentele se vor selecta din librăria corespunzătoare. Aşa cum apare specificat în ferestrele din partea stângă a ecranului vom folosi o singură şină care are alocat numărul 0. În slotul 2 este fixat CPU-ul cu modulele de intrare/ieşire. Numărul slotului apare scris şi pe panoul frontal al CPU, o cifră în partea stânga 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 şi 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.

Urmează apoi să verificăm corespondenţa între codul înscris pe CPU (pe panoul frontal în partea stângă jos) şi cel din coloana *Order number* tabelul de jos. Trebuie verificată deasemenea şi versiunea, înscrisă în coloana *Firmware*, ce se va compara cu versiunea CPU-ului, care se poate citi ridicând capacul de pe panoul fontal ce acoperă conectorii de comunicaţie (în cazul nostru V 2.6). Dacă seria și/sau versiunea CPU nu corespund cu cele înscrise pe panoul frontal al CPU atunci se va șterge CPU-ul adaugăt implicit și se va înlocui cu cel corespunzător din libraria de componente, pe care îl luăm cu drag and drop și îl punem în slotul 1 (sau activăm slotul 1 şi apoi dublu click pe CPU-ul aleas).

După alegerea CPU, se vor configura modulele de intrare/ieşire corespunzătoare. În cazul nostru este vorba despre un modul cu 24 de intrări digitale, 16 ieşiri digitale (DI24/DO16), 5 intrări analogice şi respectiv 2 ieşiri analogice (AI5/AO2).

Adresele modulelor analogice şi digitale pot fi configurate după dorintă, dar respectând condiţia ca ele să nu se suprapună. Pentru modificarea adreselor astfel încât cele ale modulului analogic să înceapă de la 256, iar cele ale modulului digital de la 0 se va da dublu click pe fiecare modul şi se va alege tab-ul Addresses, după care se va deselecta *System default* şi se vor modifica adresele.

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

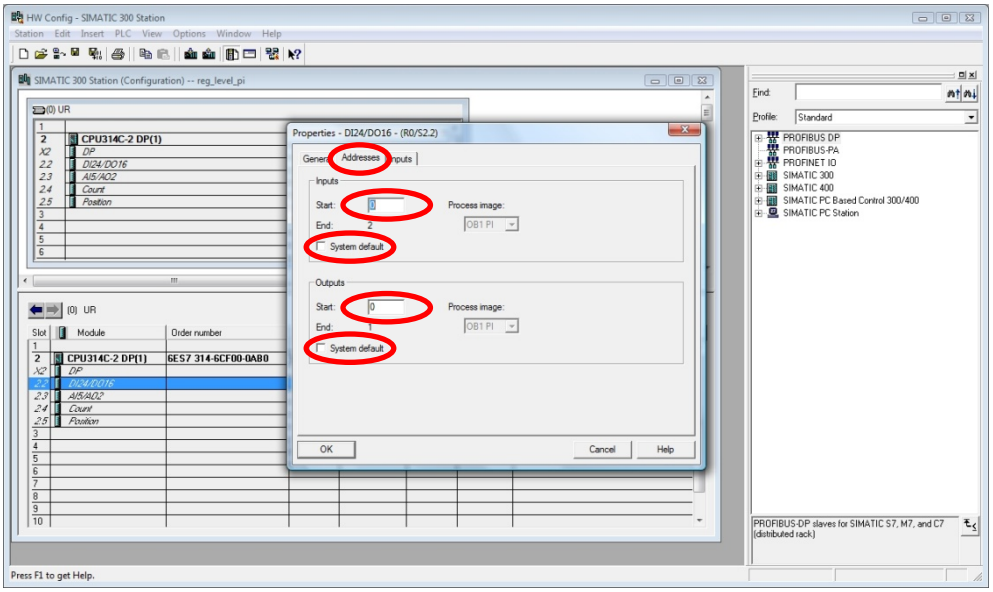


Figura 8. Modificarea adreselor modulului digital

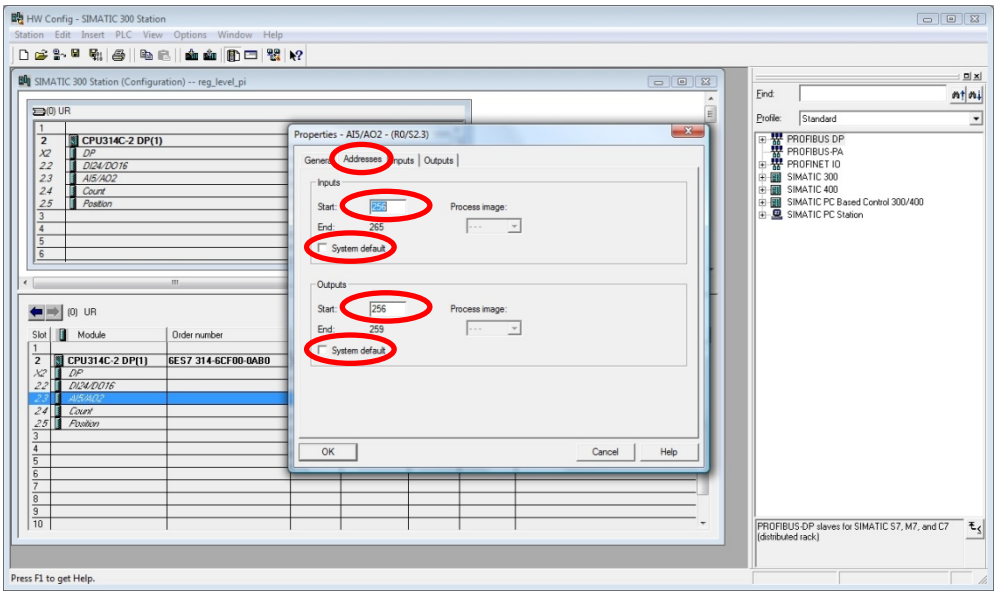


Figura 9. Modificarea adreselor modulului analogic

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

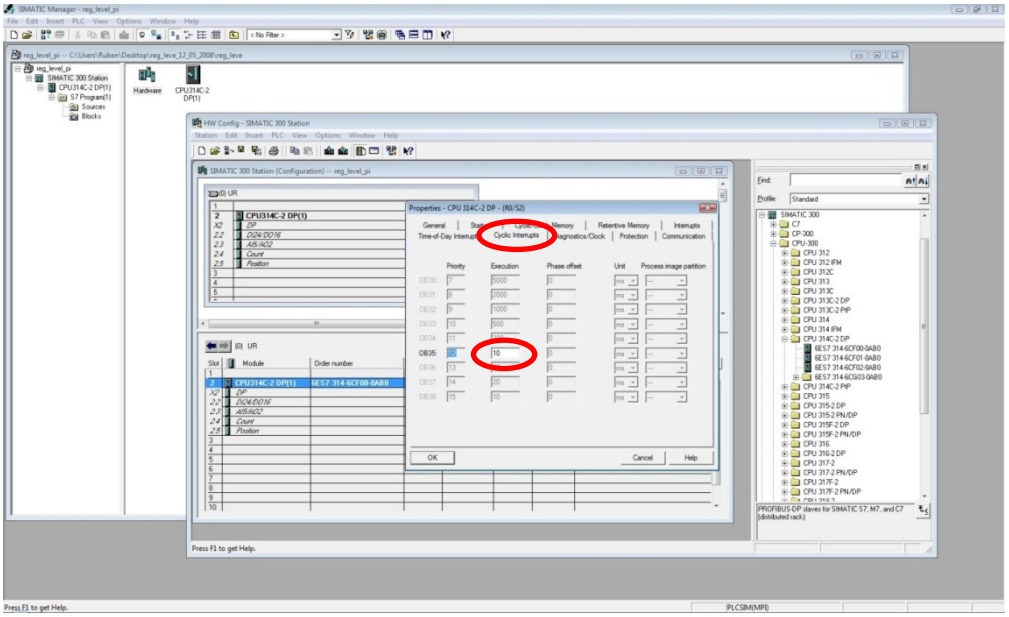


Figura 10. Modificarea timpului pentru întreruperea ciclică OB35

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

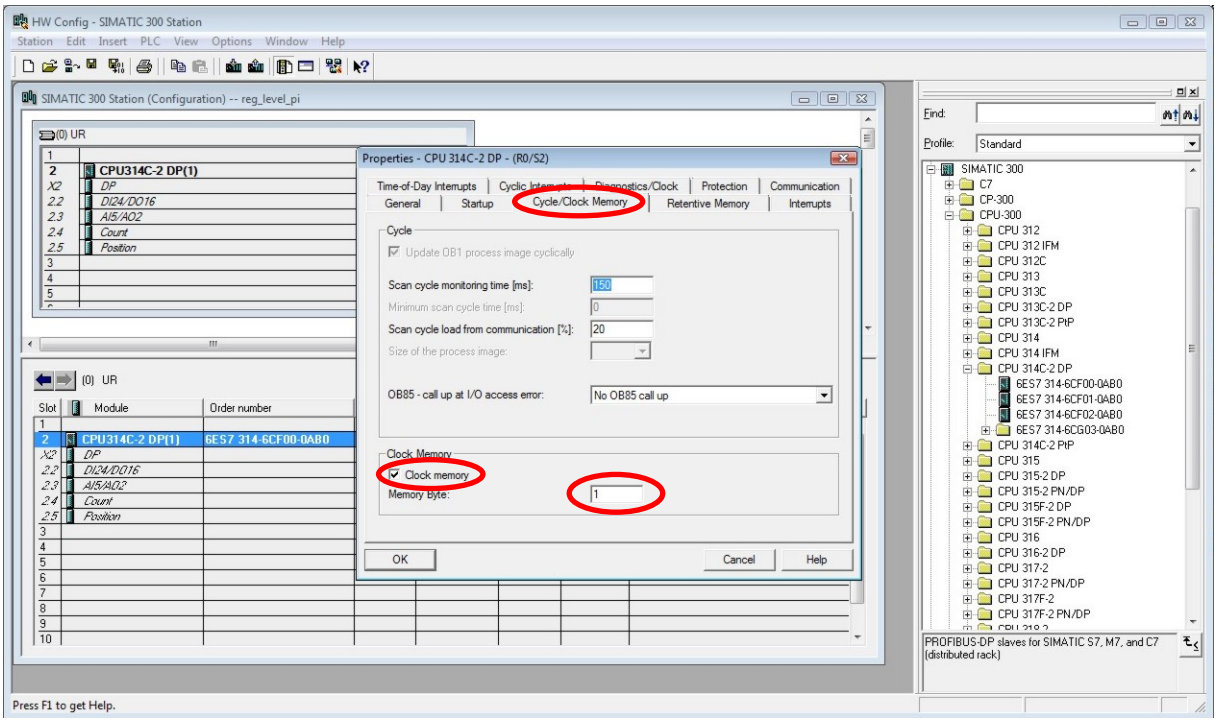


Figura 11. Setarea octetului de memorie pentru ceas

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

Astfel, datele sunt pregatite 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*.

4.3. Definirea simbolurilor

4.3.1. 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.

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 simblolice în adresele absolute cerute în mod automat.

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

4.3.2. Adrese și tipuri de date permise în Simbolic Table

IEC	Description	Data Type	Address Range
I	Input bit	BOOL	0.0 to 65535.7
IB	Input byte	BYTE, CHAR	0 to 65535
IW	Input word	WORD, INT, S5TIME, DATE	0 to 65534
ID	Input double word	DWORD, DINT, REAL, TOD, TIME	0 to 65532
Q	Output bit	BOOL	0.0 to 65535.7
QB	Output byte	BYTE, CHAR	0 to 65535
QW	Output word	WORD, INT, S5TIME, DATE	0 to 65534
QD	Output double word	DWORD, DINT, REAL, TOD, TIME	0 to 65532
M	Memory bit	BOOL	0.0 to 65535.7
MB	Memory byte	BYTE, CHAR	0 to 65535
MW	Memory word	WORD, INT, S5TIME, DATE	0 to 65534
MD	Memory double word	DWORD, DINT, REAL, TOD, TIME	0 to 65532
PIB	Peripheral input byte	BYTE, CHAR	0 to 65535
PQB	Peripheral output byte	BYTE, CHAR	0 to 65535
PIW	Peripheral input word	WORD, INT, S5TIME, DATE	0 to 65534
PQW	Peripheral output word	WORD, INT, S5TIME, DATE	0 to 65534
PID	Peripheral input double word	DWORD, DINT, REAL, TOD, TIME	0 to 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

5. Posibilităţi 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

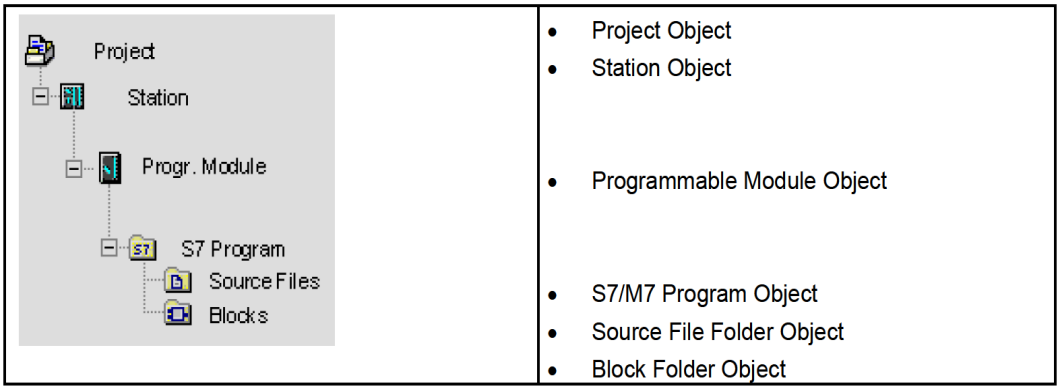


Figura 12. Structura proiectului

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.

6. Programarea funcțiilor

STEP 7 pune la dipoziție 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
✓ Variabilele statice declarate în funcțiile bloc (FB) se păstrează când blocul este închis
✓ FB și FC sunt apelete în cadrul blocurilor de organizare (OB)

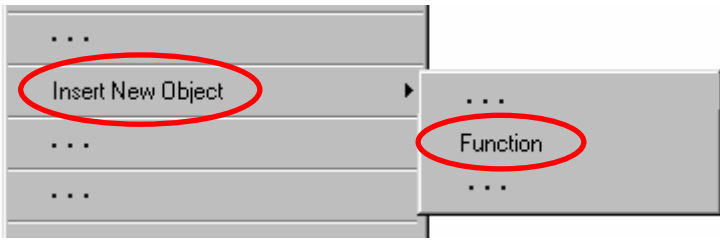


Figura 13. Inseratea unei funcții din meniul pop-up

În caseta de dialog *Properties – Function* se acceptă numele funcției și se selectează limbajul de programare. Se confirmă cu *OK*.

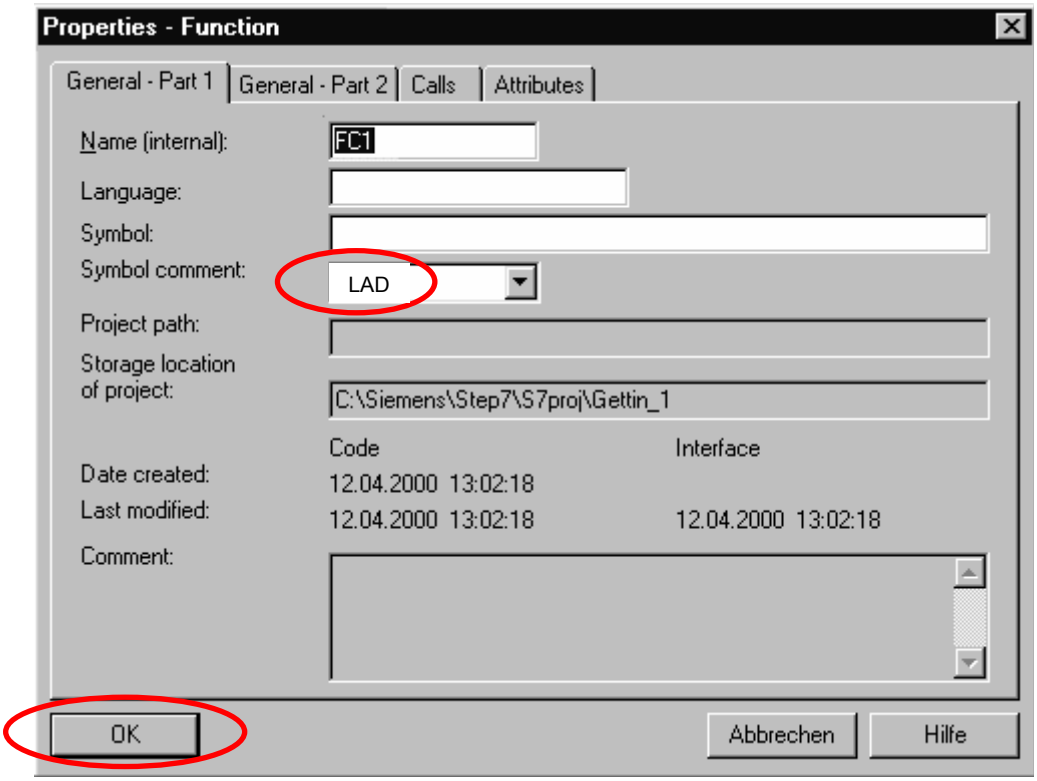


Figura 14. Function Properties

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

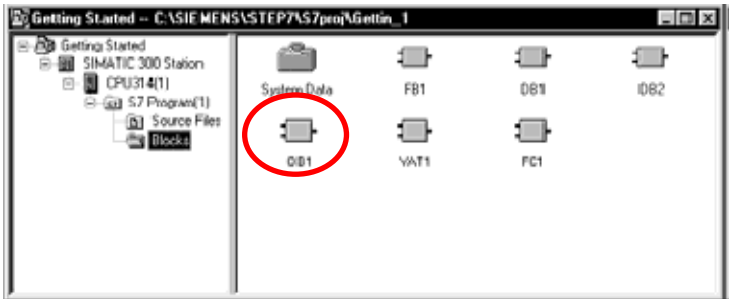


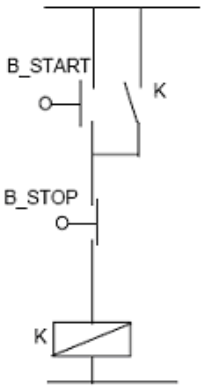
Figura 15. Funcția FC1

7. Aplicații

7.1. Releu cu automenținere

Pentru exemplificare ne propunem realizarea unui program pentru comanda cu automenţinere a unui releu. Programul are ca efect comanda unui releu prin apăsarea butoanelor B_START (pentru anclanşarea releului – închiderea contactelor N.O.) şi B_STOP (pentru dezanclanşarea releului – întreruperea contactelor N.O.).

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 şi S400.



Circuit de comandă cu automenținere

Crearea tabelei 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 şi tipul de date adreselor absolute care vor putea fi folosite apoi în program. Aceste nume se pot folosi în toate părtile programului şi se numesc variabile globale. Pentru aplicaţia 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

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 adauga astfel: la coloana *Symbols* se scrie *Main Program* şi la coloana *Adresses* se scrie OB1. Poate fi adăugat şi un comentariu. Apoi se va realiza tabela de simboluri de mai sus. În acest fel se pot atribui denumiri simbolice tuturor adreselor absolute ale intrărilor şi ieşirilor necesare în program. La final se salvează folosind optiunea *Save* din meniul *File*.

Programarea blocului OB1 în Ladder Logic

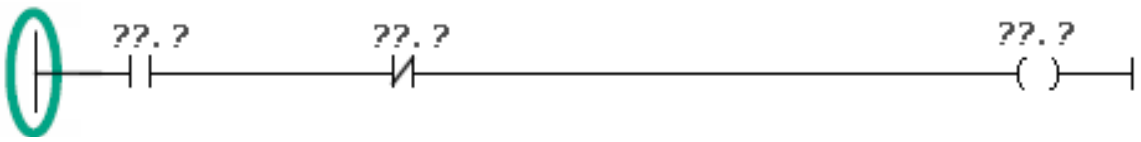
Vom programa acum circuitul START/STOP în Ladder Logic. În sectiunea *Blocks* activăm OB1. Va apare o nouă fereastră. Dacă nu a fost selectat corect limbajul de programare, acesta se mai poate selecta de la View, în cazul nostru vom alege **LAD**.

Dăm click în sectiunea de titlu a lui OB1 şi scriem *Programul principal*.

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

Apăsăm butonul ce se afla în toolbar şi introducem un contact normal deschis , apoi un contact normal închis şi 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 şi selectăm apoi începutul de ramificaţie . Introducem un contact normal deschis şi sfârşitul ramificaţiei .

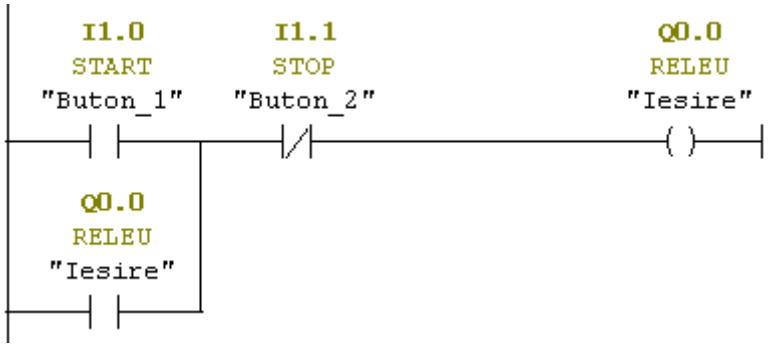


Din meniul *Options*, submeniul *Customize* şi apoi *View* verificăm dacă *Symbolic Representation* şi *Block/network comments* sunt activate.

Apăsăm semnul "??..?" corespunzător primului element al treptei şi 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 roşu înseamnă că acesta nu există în tabela de simboluri sau există o eroare de sintaxă.

Se poate da un nume treptei (Network1:) şi 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 şi realizarea unui download se permite încarcarea programului ca şi cum am avea un automat programabil ataşat, având astfel posibilitatea de a simula funcţionarea programului.

Simulatorul este disponibil în SIMATIC Manager, meniul *Options*, *Simulate Modules*. După activarea acestei opţiuni apare o fereastră cu S7-PLCSIM şi fereastra de dialog *Open Project.* Se selectează *Select CPU access node* şi *OK*. S7-PLCSIM va afişa o nouă fereastră de dialog *Select CPU access node* şi denumirea proiectului deschis în STEP 7. Se va selecta *MPI* şi apoi *OK*. Din SIMATIC Manager se activează folderul *Blocks* şi descarcă apoi programul activând *Download*  . În S7-PLCSIM se deschide fereastra de simulare (figura 16). Modul de lucru cu simulatorul este prezentat în fişierul „s7wsvhdb.pdf" sau „S7-PLCSIM - Testing Your S7-CPU Programs - manual.pdf"

Din bara de instrumente se adaugă o intrare şi o ieşire în format *Bits*.

Avem nevoie de IB 1 pentru intrări şi QB 0 pentru ieşiri. Acum se poate simula funcţionarea aplicaţiei.

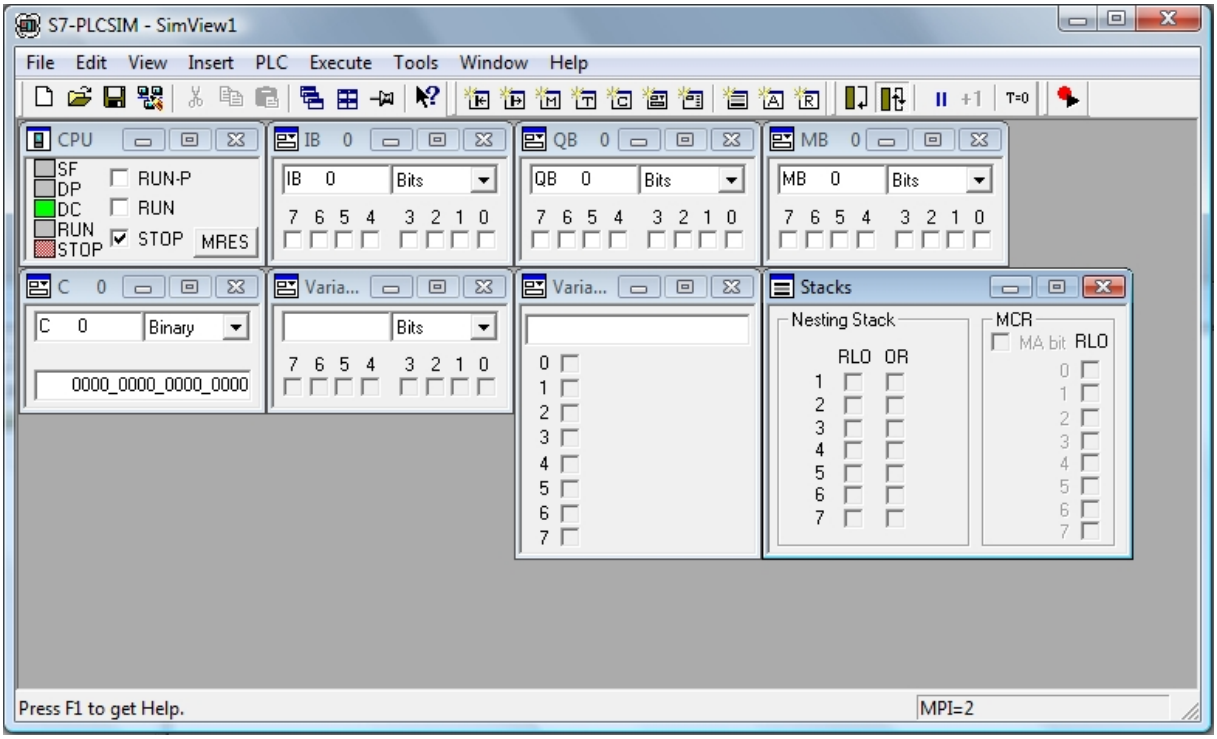


Figura 16. Modulul de simulare S7-PCLSIM

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 şi apăsarea butonului Download ).

Aceeaşi funcţionalitate a aplicatiei se poate obţine folosind elementul SR (SET-RESET) din sectiunea *Bit Logic* a ferestrei *Program elements*.

7.2. 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.

- **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***.

Mod de lucru:

1. Se definește tabela de simboluri

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

2. Se folosește blocul *MOVE* din *Program Elements* pentru a înscrie valoarea analogică în memorie
3. Cu blocul *MOVE* se înscrie la ieșire valoarea analogică citită

7.3. Comapararea a două tensiuni

Se vor citi două tensiuni și se vor compara. Atâta 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.

7.4. Regulator P

Să se impementeze 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. Scalarea referinței se va implemeta cu o funcție (FC)
4. Se va implementa un bloc DB pentru memorarea datelor
5. Se va face monitorizarea parametrilor regulatorului prin DB41