



UNIUNEA EUROPEANĂ



GUVERNUL ROMÂNIEI
MINISTERUL MUNCII, FAMILIEI,
PROTECȚIEI SOCIALE ȘI
PERSOANELOR VÂRSTNICE
AMPOSDRU



Fondul Social European
POSDRU 2007-2013



Instrumente Structurale
2007-2013



MINISTERUL
EDUCAȚIEI
NAȚIONALE



CIPOSDRU



UNIVERSITATEA
TEHNICĂ
DIN CLUJ-NAPOCA

Material suport pentru stagii de practică în domeniul dezvoltării software pentru automatizări

Inițiere în programarea automatelor SIMATIC S7-1200 cu Portalul TIA V10 Volum I



UNIUNEA EUROPEANĂ



GUVERNUL ROMÂNIEI
MINISTERUL MUNCII, FAMILIEI,
PROTECȚIEI SOCIALE ȘI
PERSOANELOR VÂRSTNICE
AMPOSDRU



Fondul Social European
POSDRU 2007-2013



Instrumente Structurale
2007-2013



MINISTERUL
EDUCAȚIEI
NAȚIONALE



CNDIPT
OIPOSDRU



UNIVERSITATEA
TEHNICĂ
DIN CLUJ-NAPOCA

CUPRINS

	PAGINA
1. Introducere.....	3
2. Observații privind programarea automatelor SIMATIC S7-1200	4
2.1 Sistemul de automatizare SIMATIC S7-1200.....	4
2.2 Mediul de programare STEP 7 Basic V10.5 (Portalul TIA V10.5)	5
3. Instalarea mediului de programare STEP 7 Basic V10.5.....	5
4. Conectarea la CPU prin conexiune TCP/IP și revenirea la setările inițiale de fabrică	6
5. Prezentarea automatelor programabile și a utilizării lor	12
5.1 Termenul de automat programabil sau PLC	12
5.2 Modul în care automatul programabil asigură comanda procesului	13
5.3 Modul în care automatul programabil obține informațiile despre starea procesului	13
5.4 Diferența între contacte normal închise și contacte normal deschise	14
5.5 Modul de adresare a semnalelor individuale de intrare/ieșire de către automatul programabil SIMATIC S7-1200	15
5.6 Modul de procesare a programului în automatul programabil	17
5.7 Modul de prezentare a operațiilor logice în programul automatului	19
5.7.1 Operația ȘI (AND).....	19
5.7.2 Operația SAU (OR).....	21
5.7.3 Negația.....	22
5.8 Modul de generare a programului automatului. Modul în care programul ajunge în memoria automatului	24
6. Configurarea și utilizarea automatului programabil SIMATIC S7-1200	25
7. Exemplu: Comanda unei prese	29
8. Programarea funcționării presei la automatul SIMATIC S7-1200	31
8.1. Prezentare tip PORTAL	31
8.2. Prezentare tip PROIECT	32
9. Bibliografie.....	63

1. Introducere

Următoarele simboluri vă vor ghida prin acest manual:



Informație



Instalare



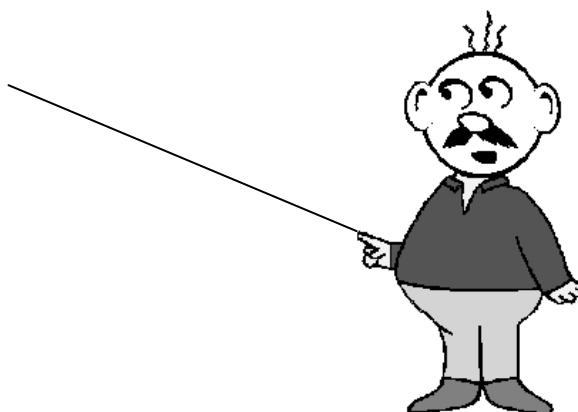
Programare



Exemplu



Atenționare



Obiectivul manualului de instruire:

Acest manual, prezintă o introducere în programarea automatelor programabile (PLC) SIMATIC S7-1200, folosind mediul de programare TIA Portal. Manualul oferă cunoștințele de bază necesare programării, prezentând pașii care trebuie parcurși, utilizând un exemplu detaliat. Pașii de parcurs sunt următorii:

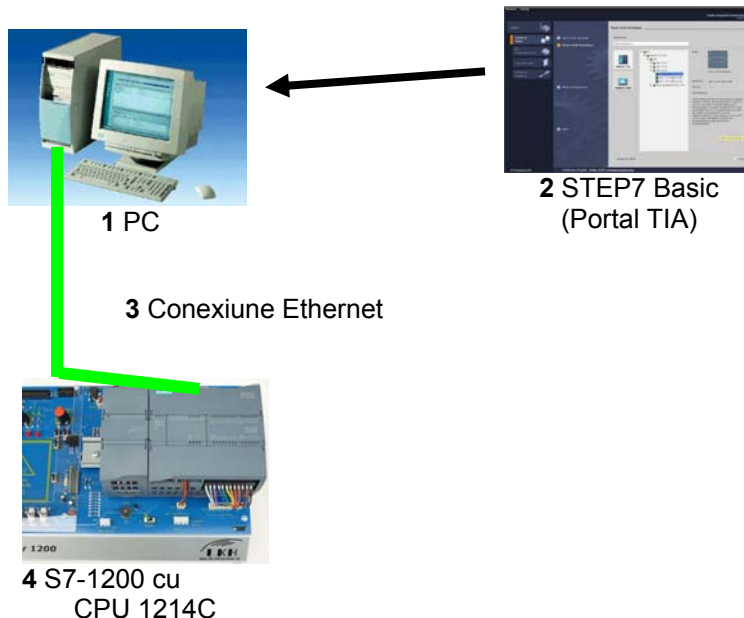
- Instalarea software-ului și setarea interfaței de programare
- Explicație: Ce este un automat programabil și cum funcționează
- Structura și funcționarea automatului programabil SIMATIC S7-1200
- Generarea, încărcarea și testarea unui program de test

Pentru a putea folosi acest manual este necesară cunoașterea sistemului de operare Windows.

Hardware și software necesare:

1. Calculator PC Pentium 4; 1.7 GHz; 1(XP) sau 2 (Vista) GB RAM, zonă liberă pe discul de stocare de cca. 2 GB; Sistemul de operare Windows XP (Home SP3, Professional SP3) sau Windows Vista (Home Premium SP1, Business SP1, Ultimate SP1)
2. Software STEP7 Basic V10.5 SP1 (Totally Integrated Automation (TIA) Portal V10.5)
3. Conexiune Ethernet între calculatorul PC și CPU 1214C

4. Automat programabil SIMATIC S7-1200, de exemplu CPU 1214C. Intrările trebuie să fie accesibile la un panou.



2. Observații privind programarea automatelor SIMATIC S7-1200

2.1 Sistemul de automatizare SIMATIC S7-1200



Sistemul de automatizare SIMATIC S7-1200 este un sistem mini-controler modular din gama de performanțe scăzute.

Familia S7-1200 cuprinde o gamă largă de module pentru adaptarea optimă la necesitățile de automatizare.

Controlerul S7 constă dintr-o unitate centrală de procesare CPU care este echipat cu intrări și ieșiri pentru semnale digitale și analogice.

În cazul în care intrările și ieșirile integrate în CPU nu sunt suficiente pentru aplicația dorită, pot fi instalate modulele de intrare și de ieșire (module IO) suplimentare.

Dacă este necesar, se adaugă procesoare de comunicare pentru RS232 sau RS485. O interfață TCP / IP integrată este obligatorie pentru toate unitățile centrale de procesare CPU.

Automatul programabil (PLC) monitorizează și controlează un utilaj sau un proces cu ajutorul programului S7, care asigură interogarea modulelor IO prin intermediul adreselor de intrare (%I) și comanda procesului prin intermediul adreselor de ieșire (%Q).

Sistemul de automatizare SIMATIC S7-1200 este programat cu software-ul STEP 7 Basic V10.5.



UNIUNEA EUROPEANĂ



GUVERNUL ROMÂNIEI
MINISTERUL MUNCII, FAMILIEI,
PROTECȚIEI SOCIALE ȘI
PERSOANELOR VÂRSTNICE
AMPOSDRU



Fondul Social European
POSDRU 2007-2013



Instrumente Structurale
2007-2013



MINISTERUL
EDUCAȚIEI
NAȚIONALE



CNDIPT
OIPOSDRU



UNIVERSITATEA
TEHNICĂ
DIN CLUJ-NAPOCA

2.2 Mediul de programare STEP 7 Basic V10.5 (Portalul TIA V10.5)



Software-ul STEP 7 Basic V10.5 este mediul de programare pentru sistemul de automatizare - SIMATIC S7-1200

STEP 7 Basic V10.5, permite utilizarea următoarelor funcții pentru a automatiza o instalație:

- Configurarea și parametrizarea hardware
- Definirea comunicației
- Programarea
- Testarea, punerea în funcțiune și service-ul cu funcțiile de operare / diagnosticare
- Elaborarea documentației
- Generarea ecranelor de afișare pentru panourile de operare de bază SIMATIC

Toate funcțiile sunt sprijinite de meniuri help on-line detaliate.

3. Instalarea mediului de programare STEP 7 Basic V10.5



Mediul de dezvoltare STEP 7 Basic V10.5 este livrat pe un DVD.

Pentru a instala mediul de dezvoltare STEP 7 Basic V10.5, se efectuează următoarele operații:

1. Se introduce DVD-ul de STEP 7 Basic V10.5 în unitatea DVD.
2. Programul de instalare se lansează automat. Dacă nu, se pornește prin dublu click pe fișierul '→ **START.exe**'.

Programul de configurare ghidează operatorul pe parcursul întregului proces de instalare a mediului de dezvoltare STEP 7 Basic V10.5.

Pentru a utiliza mediul de dezvoltare STEP 7 Basic V10.5, nu este nevoie de nici o cheie de licență sau dongle de validare atașat la calculator.



UNIUNEA EUROPEANĂ



GUVERNUL ROMÂNIEI
MINISTERUL MUNCII, FAMILIEI,
PROTECȚIEI SOCIALE ȘI
PERSOANELOR VÂRSTNICI
AMPOSDRU



Fondul Social European
POSDRU 2007-2013



Instrumente Structurale
2007-2013



MINISTERUL
EDUCAȚIEI
NAȚIONALE



CNDIPT
OIPOSDRU



UNIVERSITATEA
TEHNICĂ
DIN CLUJ-NAPOCA

4. Conectarea la CPU prin conexiune TCP/IP și revenirea la setările inițiale de fabrică



Pentru a programa automatul programabil SIMATIC S7-1200 de la PC, PG (echipament de programare SIEMENS) sau un laptop, este nevoie de o conexiune TCP / IP între cele două echipamente.

Pentru a comunica unul cu altul, este nevoie , de asemenea, ca adresele calculatorului și automatului programabil SIMATIC S7-1200 să corespundă.

Setarea adresei IP a calculatorului se realizează astfel:

1. Din meniul '**System control**', se apelează '**Network connections**'. Apoi, se selectează '**Properties**' ale conexiunii LAN (→ Start → Settings → System control → Network connections → Local Area Connection → Properties)
2. Se selectează '**Properties**' din cadrul '**Internet Protocol (TCP/IP)**' (→ Internet Protocol (TCP/IP) → Properties)
3. Acum se poate seta '**IP address**' și '**Subnet screen form**', iar apoi se pot confirma cu '**OK**' (→ Use the following IP address → IP address: 192.168.0.99 → Subnet screen form 255.255.255.0 → OK → Close)



Observații cu privire la comunicația pe rețeaua Ethernet:

Adresa MAC:

Adresa MAC conține o parte fixă și o parte variabilă. Partea fixă ("Basic MAC Address") identifică producătorul (Siemens, 3COM, ...). Partea variabilă a adresei MAC diferențiază diferitele posturi Ethernet și trebuie să fie unică (la nivel mondial) pentru fiecare echipament. Fiecare modul are imprimată o adresă MAC specificată de producător.

Gama de valori pentru adresa IP:

Adresa IP constă din 4 numere zecimale din intervalul de valori 0 la 255, separate între ele prin puncte. De exemplu, 141.80.0.16

Gama de valori pentru masca de subrețea:

Această mască este utilizată pentru a recunoaște dacă un post sau adresă IP aparține subrețelei locale, sau pot fi accesată numai prin intermediul unui router.

Masca de subrețea este formată din patru numere zecimale din gama 0 la 255, separate între ele prin puncte. De exemplu, 255.255.0.0

În reprezentarea binară, cele 4 numere zecimale ale măștii de subrețea trebuie să conțină, începând din stânga o serie continuă de valori "1", iar începând din dreapta o serie continuă de valori "0".

Valorile "1", specifică zona adresei IP pentru numărul de rețea. Valorile "0", specifică zona a adresei IP pentru adresa de post din subrețea.

Exemplu:

Valori corecte:

255.255.0.0 zecimal = 1111 1111.1111 1111.0000 0000.0000 0000 binar

255.255.128.0 zecimal = 1111 1111.1111 1111.1000 0000.0000 0000 binar

255.254.0.0 zecimal = 1111 1111.1111 1110.0000 0000.0000 0000 binar

Valoare greșită:

255.255.1.0 zecimal = 1111 1111.1111 1111.0000 0001.0000 0000 binar

Gama de valori pentru adresa de gateway (Router):

Adresa constă din 4 numere zecimale din intervalul de valori 0 la 255, separate între ele prin puncte. De exemplu, 141.80.0.1.

Relația dintre adresa IP, adresa router și masca de subrețea:

Adresa IP și adresa gateway trebuie să difere doar în pozițiile ce se află în dreptul valorilor "0" din masca de subrețea.

Exemplu:

S-au selectat următoarele adrese : pentru masca de subrețea 255.255.255.0, pentru adresa IP 141.30.0.5 și adresa router 141.30.128.1.

Adresa IP și adresa de gateway trebuie să aibă o valoare diferită numai în cel de al 4-lea număr zecimal. Dar, în exemplul nostru, cele două adrese diferă și în cel de al 3-lea număr zecimal.

Asta înseamnă că, în exemplul nostru trebuie să facem una din următoarele schimbări:

- Masca de subrețea să devină : 255.255.0.0 sau
- Adresa IP să fie: 141.30.128.5 sau
- Adresa gateway să fie: 141.30.0.1

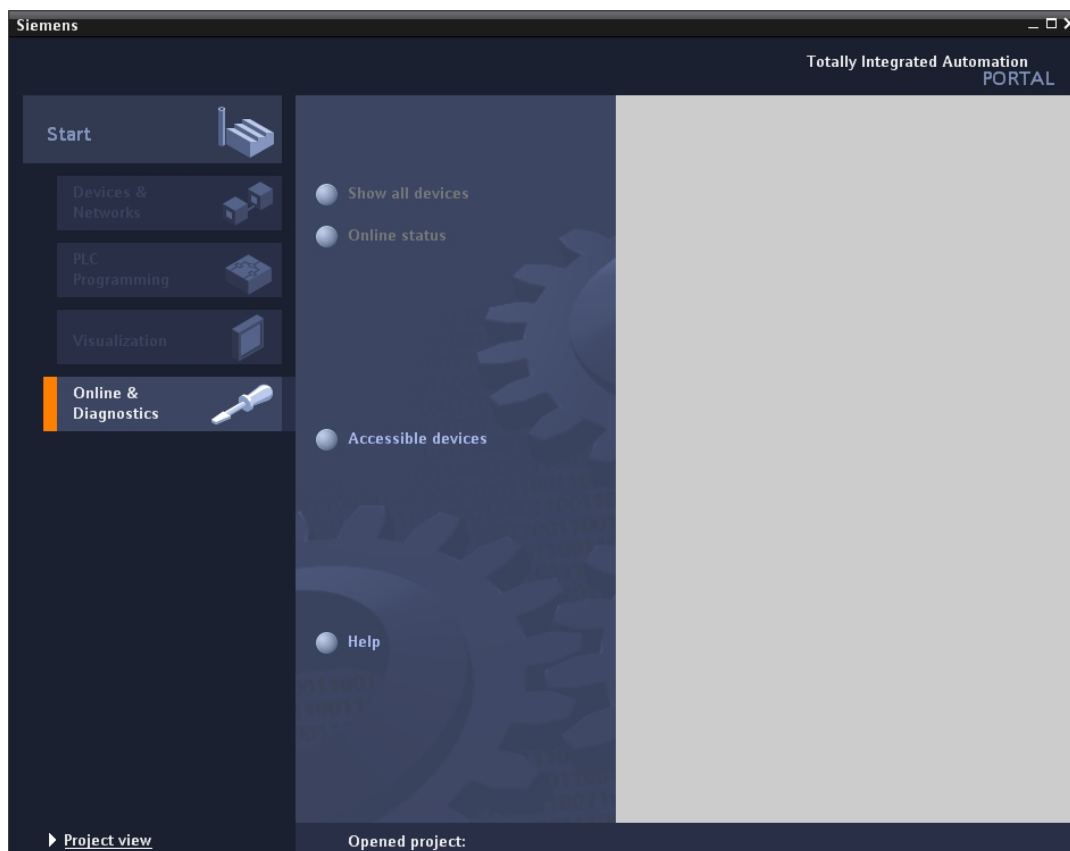


Adresa IP a automatului programabil SIMATIC S7-1200 se setează astfel:

4. Se selectează '**Totally Integrated Automation Portal**' prin dublu click (→ Totally Integrated Automation Portal V10)

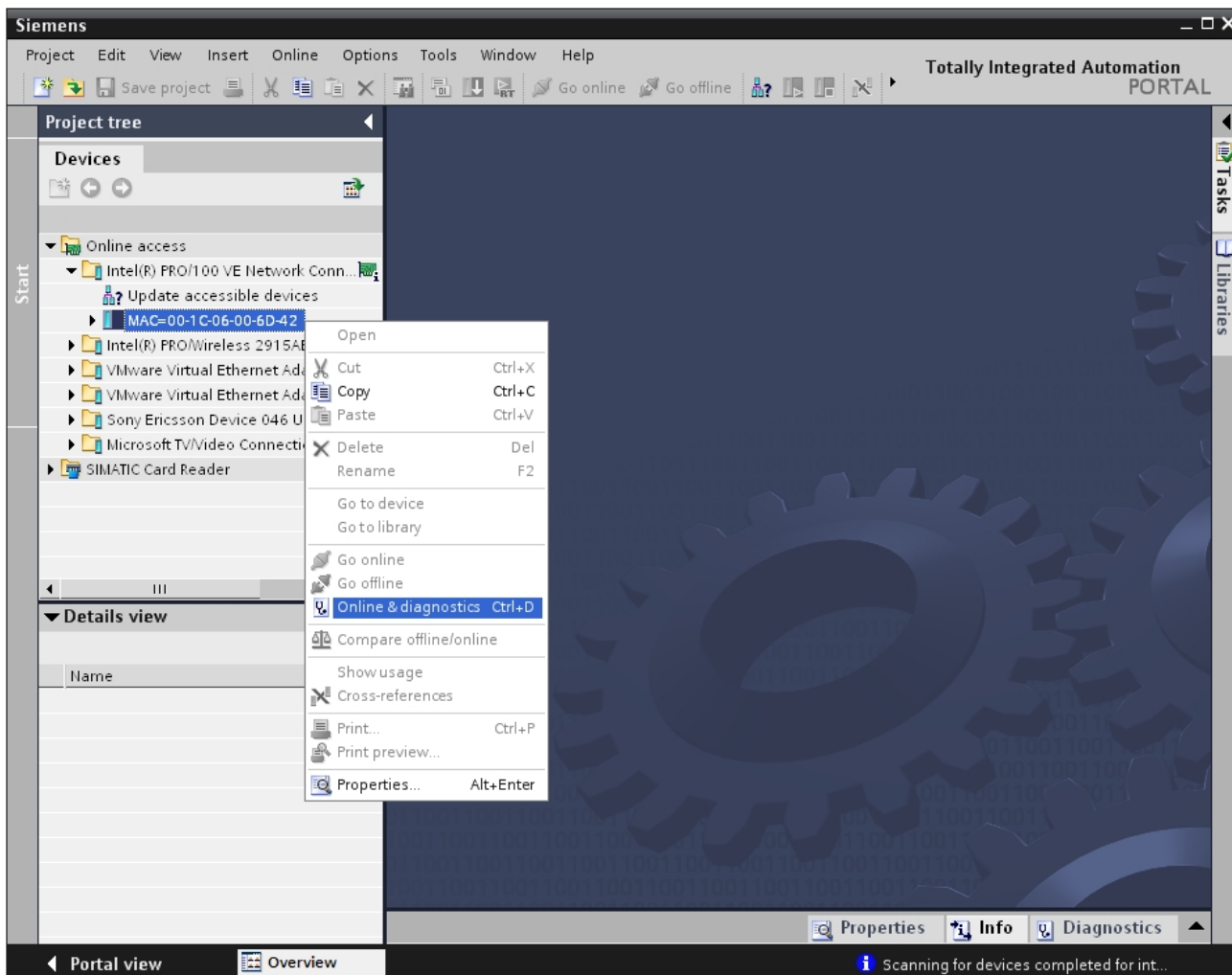


5. Apoi se selectează '**Project View**' din partea stânga jos a ecranului. (→ Project view)





6. Apoi, în meniul de navigație tip proiect, selectăm în '**Online accesses**', placa de rețea care a fost deja instalată în prealabil. Dacă aici se face clic pe '**Update accessible stations**', va fi prezentată adresa MAC a automatului programabil SIMATIC S7-1200 conectat la calculator. Se selectează '**Online & Diagnosis**'. (→ Online accesses → ... Network Connection → Update accessible stations → MAC= → Online & Diagnosis)



Notă: Dacă pentru unitatea centrală CPU a automatului programabil conectată la calculator, s-a setat în prealabil o adresă IP, în loc de adresa MAC va apare această adresă.



7. In submeniul 'Functions' se va putea selecta 'Assign IP address'. Aici se va introduce la 'IP address' adresa IP a automatului iar la 'Subnet screen form' masca de subrețea. După click pe 'Assign IP address', această nouă adresă va fi atribuită automatului programabil SIMATIC S7-1200 (→ Functions → Assign IP address → IP address: 192.168.0.1 → Subnet screen form: 255.255.255.0 → Assign IP address)

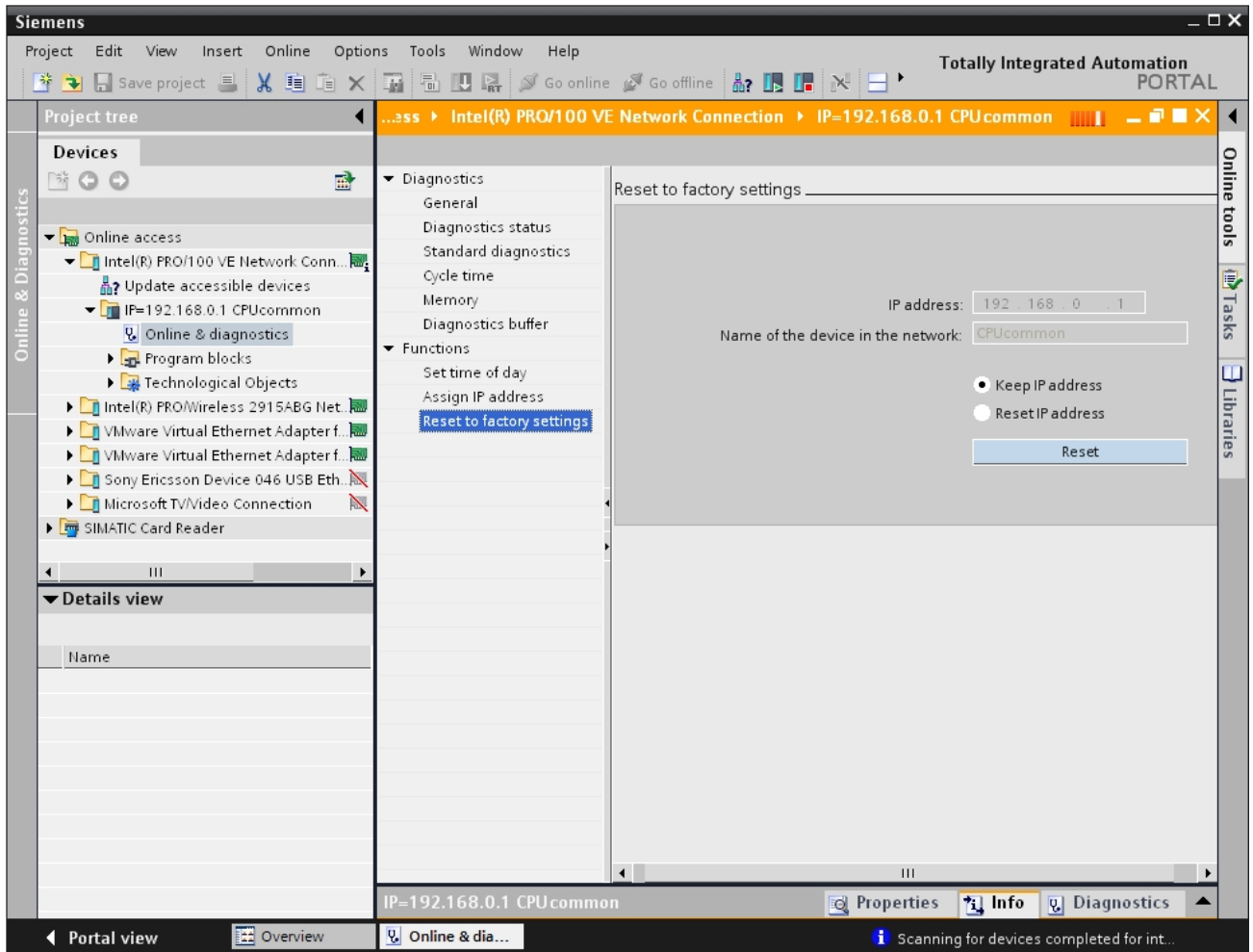
The screenshot displays the Siemens TIA Portal software interface. The main window is titled 'Siemens Totally Integrated Automation PORTAL'. The 'Project tree' on the left shows a project named 'Intel(R) PRO/100 VE Network Connection' with a MAC address of '00-1C-06-00-6D-42'. The 'Diagnosis' pane in the center shows the 'Functions' menu with 'Assign IP address' selected. The 'Assign IP address' dialog box is open, showing the following configuration:

- MAC address: 00-1C-06-00-6D-42
- IP address: 192.168.0.1
- Subnet screen mask: 255.255.255.0
- Use router: 0.0.0.0

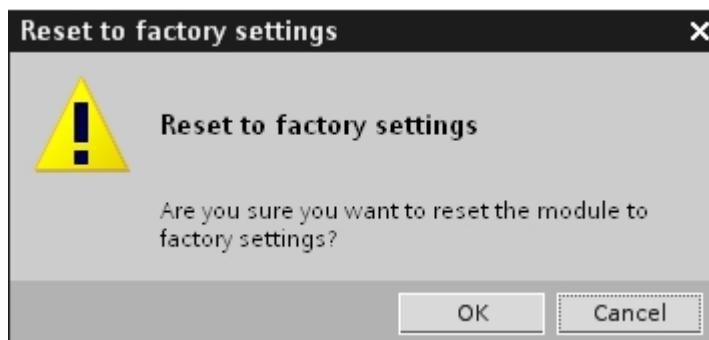
The 'Assign IP address' button is highlighted, and a tooltip explains: 'Assign an IP address to the module. Assigns the IP configuration to the module.'



- In submeniul **'Functions'**, se selectează **'Reset to factory settings'**. Se mențin setările de până acum prin selectarea **'Keep IP address'** și se apasă pe **'Reset'**. (→ Functions → Reset to factory settings → Retain IP address → Reset)



- Se confirmă solicitarea de reset dacă se vrea revenirea la setările din fabrică cu butonul **'OK'** (→ OK)

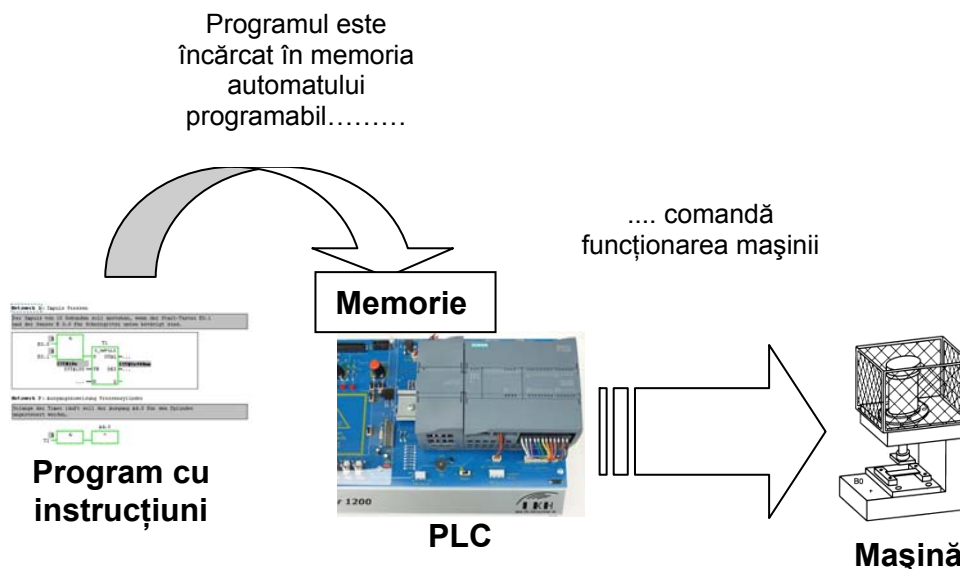


5. Prezentarea automatelor programabile și a utilizării lor

5.1 Termenul de automat programabil sau PLC



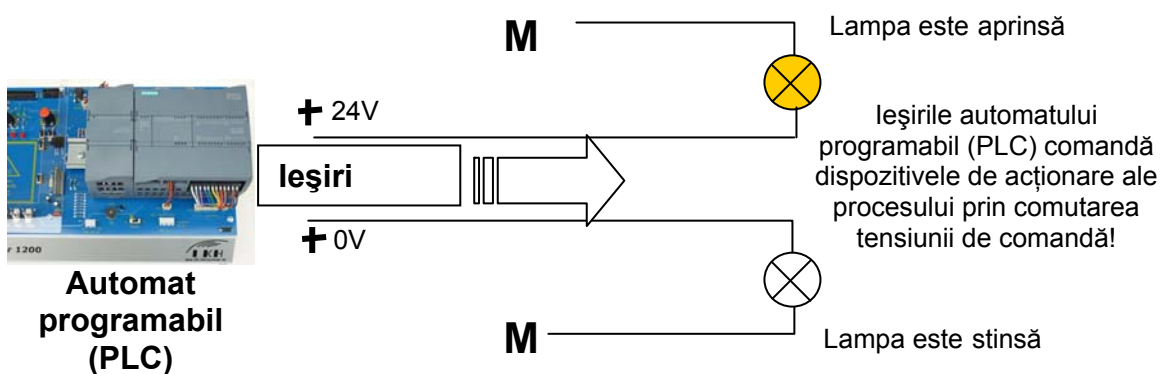
PLC este abrevierea pentru Programmable Logic Controller cunoscut în literatură ca automat programabil . Acesta este un dispozitiv care comandă un proces (de exemplu: o presă de tipărit ziare, o instalație pentru umplerea cu ciment a sacilor, o presă pentru injecția formelor de plastic, etc ...). Acest lucru se realizează în conformitate cu instrucțiunile unui program care se află în memoria dispozitivului.



5.2 Modul în care automatul programabil asigură comanda procesului



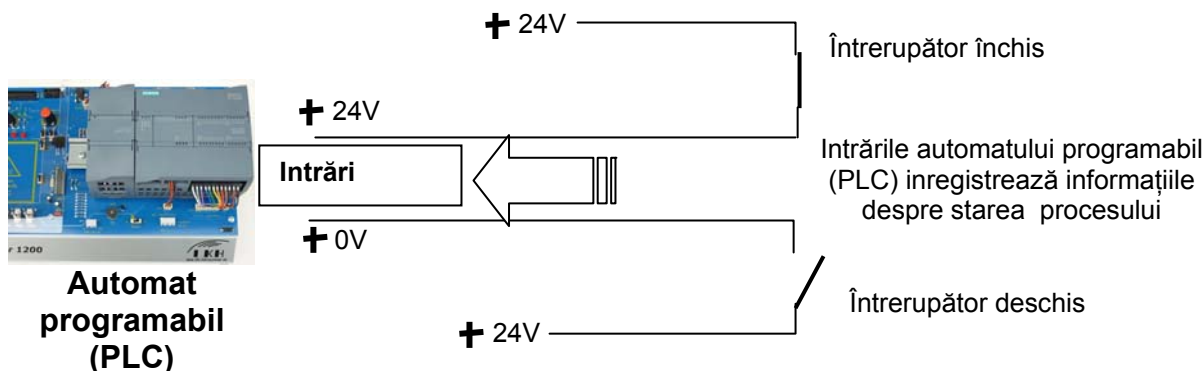
Automatul programabil (PLC) comandă procesul astfel: prin conexiunile automatului programabil (PLC) numite ieșiri, dispozitivele de acționare ale procesului sunt conectate la o tensiune de comandă de 24V, de exemplu. Acest lucru permite pornirea și de oprire motoarelor, deschiderea și închiderea vanelor, aprinderea și stingerea lămpilor de semnalizare.



5.3 Modul în care automatul programabil obține informațiile despre starea procesului



Automatul programabil (PLC) primește informații despre proces de la așa-numiții transmițători de semnal care sunt conectați la intrările automatului. Acești transmițători de semnal pot fi, de exemplu, senzorii care recunosc dacă o piesă de lucru este într-o anumită poziție, sau ele pot fi simple întrerupătoare sau butoane care pot fi doar deschise sau închise. Acestea din urmă pot fi de două categorii: comutatoare cu contacte normal închise, care în stare neacționată sunt în poziția închis și comutatoare cu contacte normal deschise, care în stare neacționată sunt în poziția deschis.

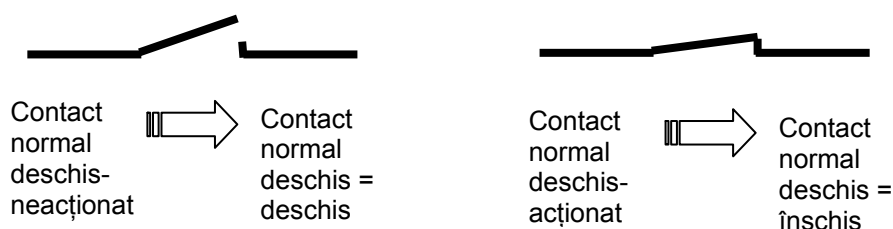


5.4 Diferența între contacte normal închise și contacte normal deschise

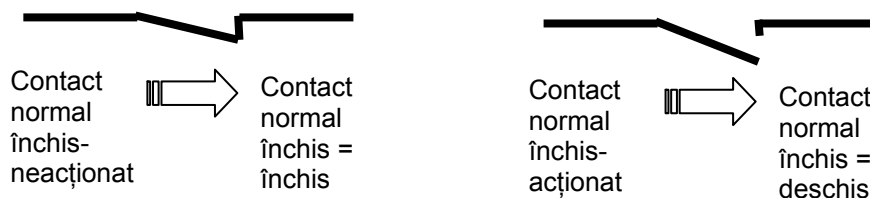


În ceea ce privește transmițătorii de semnal ele pot fi de două categorii: cu contacte normal închise (nî) și cu contacte normal deschise (nd).

Comutatorul de mai jos este cu contact normal deschis, adică, acesta este închis în momentul în care a fost acționat.



Comutatorul de mai jos este cu contact normal închis, adică, acesta este închis când nu a fost acționat.



5.5 Modul de adresare a semnalelor individuale de intrare/ieșire de către automatul programabil SIMATIC S7-1200

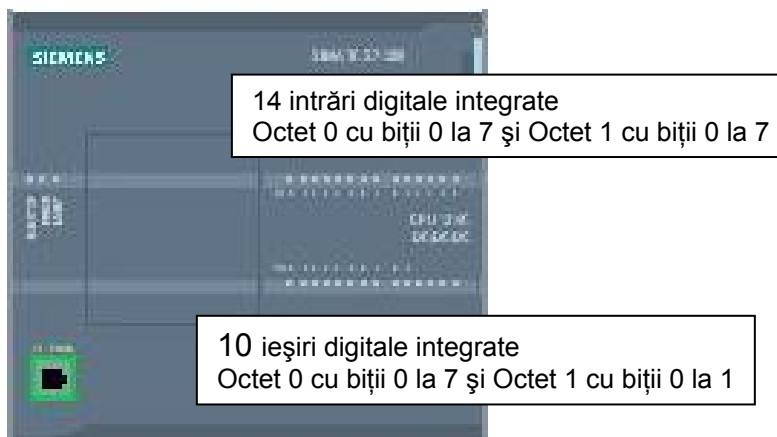


Desemnarea unei anumite intrări sau ieșiri în program se numește adresare.

Intrările și ieșirile automatului programabil (PLC) sunt de obicei combinate în grupuri de câte 8 intrări digitale sau ieșiri digitale. Acest grup de 8 semnale digitale se numește un octet. Fiecărui grup de 8 semnale digitale se asociază un număr, așa-numita adresa de octet.

Pentru a adresa o singură intrare sau ieșire dintr-un octet, fiecare octet este desfăcut pe biți. Aceștia sunt numerotați de la bitul 0 la bitul 7. Astfel se ajunge la adresa de bit.

Automatul programabil (PLC) prezentat mai jos are octeții de intrare 0 și 1, precum și de octeții de ieșire 0 și 1.



Pentru a adresa a cincea intrare digitală, de exemplu, vom specifica următoarea adresă:

%I 0 . 4

%I indică tipul adresei ca fiind intrare, **0** este adresa de octet iar **4** este adresa de bit.
Adresa de octet și adresa de bit sunt întotdeauna separate de un punct.



Notă: Pentru adresa de bit, 4 reprezintă a cincea intrare, pentru că numărarea începe cu 0.



UNIUNEA EUROPEANĂ



GUVERNUL ROMÂNIEI
MINISTERUL MUNCII, FAMILIEI,
PROTECȚIEI SOCIALE ȘI
PERSOANELOR VÂRSTNICE
AMPOSDRU



Fondul Social European
POSDRU 2007-2013



Instrumente Structurale
2007-2013



MINISTERUL
EDUCAȚIEI
NAȚIONALE



CNDIPT
OIPOSDRU



UNIVERSITATEA
TEHNICĂ
DIN CLUJ-NAPOCA

Pentru a adresa a zecea ieșire digitală , de exemplu, vom specifica următoarea adresă:

%Q 1 . 1

%Q indică tipul adresei ca fiind ieșire, **1** este adresa de octet iar următorul **1** este adresa de bit.

Adresa de octet și adresa de bit sunt întotdeauna separate de un punct.



Notă: Pentru adresa de bit, 1 reprezintă a zecea ieșire, fiind a doua ieșire din octetul al doilea, pentru că atât numărarea biților cât și al octeților începe cu 0.



UNIUNEA EUROPEANĂ



GUVERNUL ROMÂNIEI
MINISTERUL MUNCII, FAMILIEI,
PROTECȚIEI SOCIALE ȘI
PERSOANELOR VÂRSTNICE
AMPOSDRU



Fondul Social European
POSDRU 2007-2013



Instrumente Structurale
2007-2013



MINISTERUL
EDUCAȚIEI
NAȚIONALE



CNDIPT
OIPOSDRU



UNIVERSITATEA
TEHNICĂ
DIN CLUJ-NAPOCA

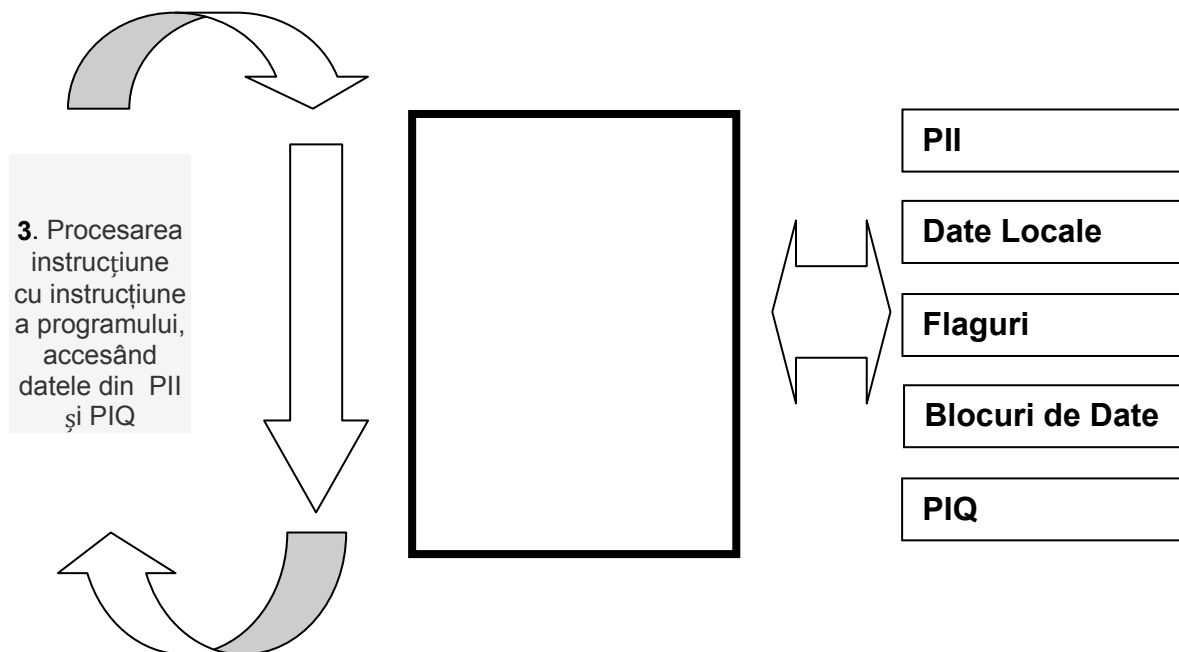
5.6 Modul de procesare a programului în automatul programabil



Programul este procesat în automatul programabil (PLC) în mod ciclic, respectând următoarea succesiune:

1. În primul rând, starea imaginii de proces a ieșirilor (**PIQ**) este transferată la ieșirile efective ale automatului, acestea fiind comutate în poziția închis sau deschis.
2. Apoi **procesorul**-care este practic creierul automatului programabil (PLC)-verifică dacă intrările individuale sunt sau nu sub tensiune. Această stare a intrărilor este stocată în imaginea de proces a intrărilor (**PII**). Pentru intrările care sunt sub tensiune, sunt stocate valori binare de 1 sau "High", iar pentru cele care nu sunt sub tensiune, sunt stocate valori binare de 0 sau "Low".
3. Acest procesor procesează apoi programul stocat în memoria program. Programul constă dintr-o listă de operațiuni logice și instrucțiuni care sunt procesate secvențial (una după alta). Pentru informațiile de intrare necesare, procesorul accesează PII care a fost înmănată în prealabil cu starea intrărilor, iar rezultatul operațiilor logice (RLO) este scrisă într-o imagine proces a ieșirilor (**PIQ**). Dacă este necesar, procesorul accesează, de asemenea, alte zone de memorie în timpul procesării programului; de exemplu, pentru datele locale de sub-programe, blocuri de date și flaguri.
4. Apoi sunt efectuate sarcinile interne ale sistemului de operare, cum ar fi auto-testele și comunicațiile. După aceasta procesorul se întoarce la execuția punctului 1.

1. Transferă starea din PIQ la ieșiri.
2. Stocheză starea intrărilor în PII.



4. Efectuare de sarcini interne ale sistemului de operare (comunicatie, auto-test, etc...)



Notă: Timpul necesar procesorului pentru a parcurge această secvență se numește timp de ciclu. Durata timpului de ciclu depinde de numărul și tipul de instrucțiuni ce se execută și de viteza de procesare a procesorului.

5.7 Modul de prezentare a operațiilor logice în programul automatului



Operațiile logice sunt utilizate pentru a stabili condițiile pentru comutarea unei ieșiri. În programul automatului (PLC), acestea pot fi programate în limbajele de programare Ladder Diagram (**LAD**) sau Function Block Diagram (**FBD**).

Pentru ilustrare vom folosi limbajul FBD.

Există un număr mare de operații logice care pot fi utilizate în programe PLC.

Operațiile logice cele mai frecvent utilizate sunt **ȘI (AND)**, **SAU (OR)** și **NEGAREA** unei intrări. Acestea sunt explicate pe scurt mai jos, folosind exemple.

Notă: Informații despre operații logice suplimentare pot fi obținute rapid în cu ajutorul meniului online help.

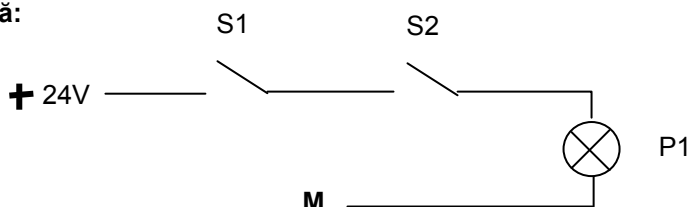
5.7.1 Operația ȘI (AND)



Exemplu de operație ȘI (AND):

O lampă de semnalizare P1 se aprinde atunci când două întrerupătoare normal deschise S1 și S2 sunt acționate simultan.

Schemă:



Explicație:

Indicatorul luminos se aprinde numai atunci când ambele întrerupătoare sunt acționate.

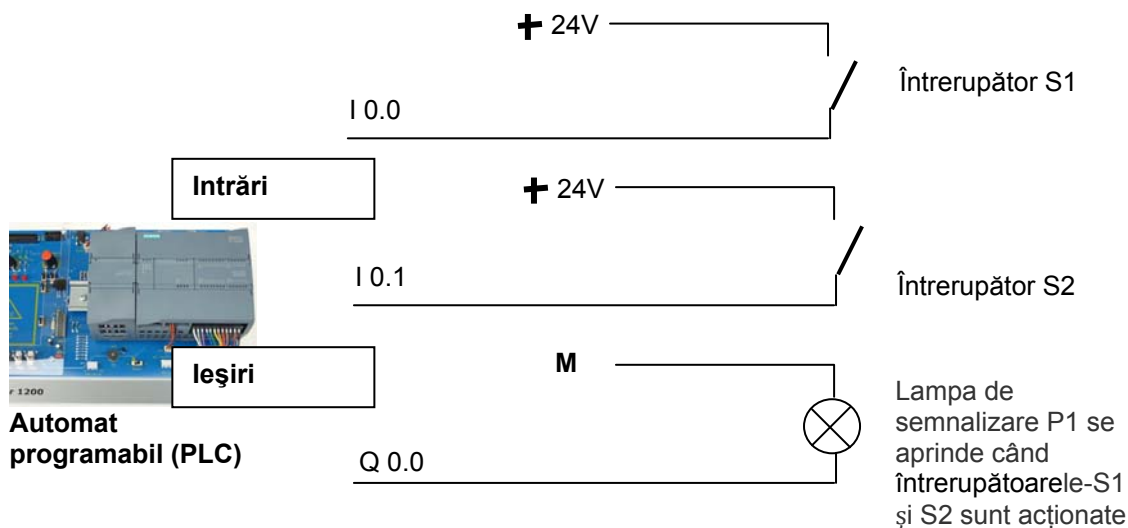
Adică, dacă S1 **ȘI** S2 sunt acționate, lampa P1 este aprinsă.



Cablarea automatului programabil (PLC):

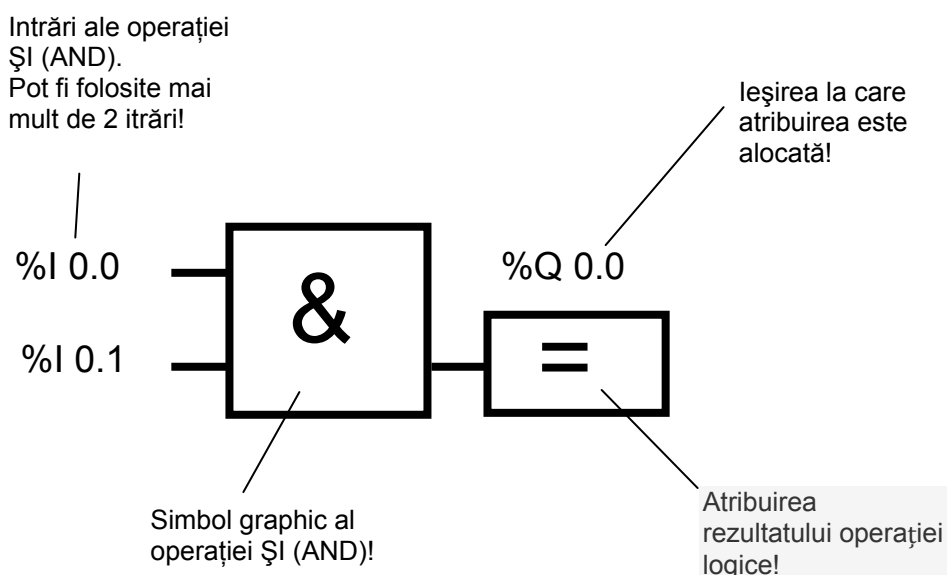
Pentru a aplica această logică a unui program PLC, ambele întrerupătoare trebuie să fie conectate la intrările automatului programabil (PLC). În exemplul nostru, S1 este conectată la intrare I 0.0 și S2 la intrare I 0.1.

În plus, lampa de semnalizare P1 trebuie să fie conectată la o ieșire, de exemplu la Q 0.0.



Operația logică ȘI (AND) in limbaj FBD:

În blocul de funcții cu diagramă FBD, operația ȘI (AND) este programată folosind un simbol grafic, și arată ca mai jos:



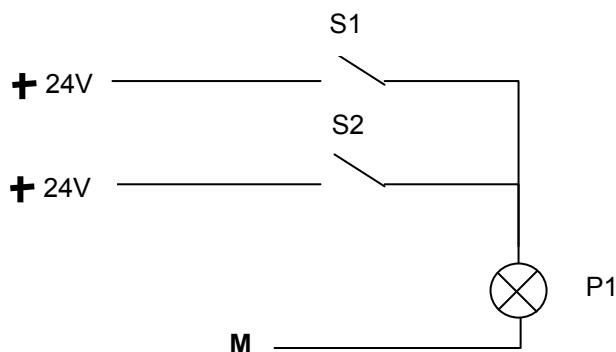
5.7.2 Operația SAU (OR)



Exemplu de operație SAU (OR):

O lampă de semnalizare P1 se aprinde atunci când unul sau ambele întrerupătoare normal deschise S1 și S2 sunt acționate.

Schemă:



Explicație:

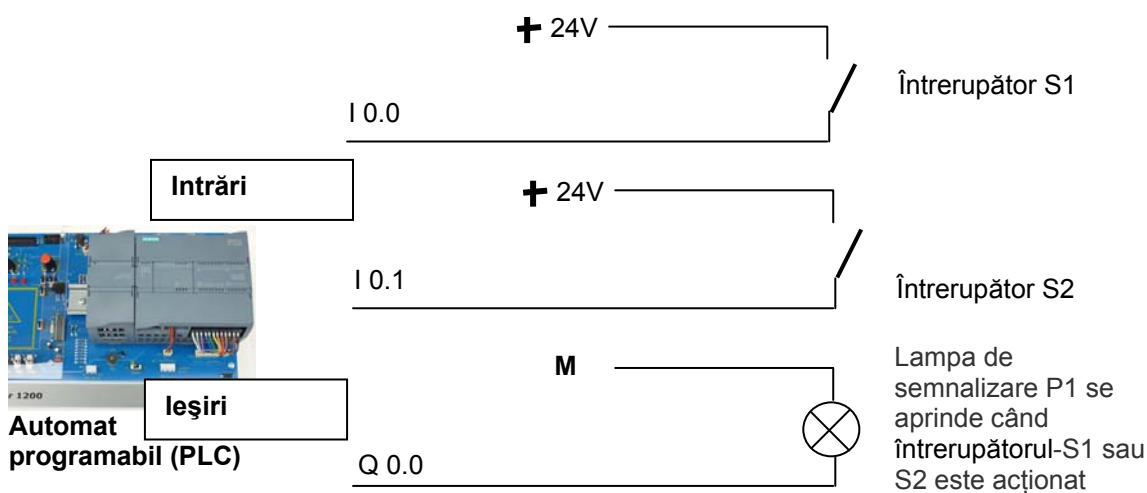
Indicatorul luminos se aprinde numai atunci când unul sau ambele întrerupătoare sunt acționate.

Adică, dacă S1 **SAU** S2 este acționat, lampa P1 este aprinsă.

Cablarea automatului programabil (PLC):

Pentru a aplica această logică a unui program PLC, ambele întrerupătoare trebuie să fie conectate la intrările automatului programabil (PLC). În exemplul nostru, S1 este conectată la intrare I 0.0 și S2 la intrare I 0.1.

În plus, lampa de semnalizare P1 trebuie să fie conectată la o ieșire, de exemplu la Q 0.0.

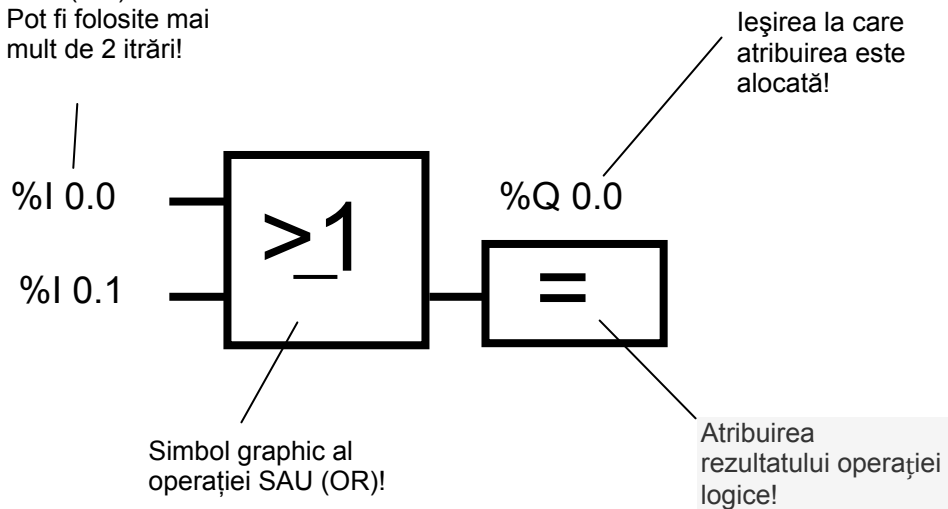


i

Operația logică SAU (OR) in limbaj FBD:

În planul de funcții FBD, operația SAU (OR) este programată folosind un simbol grafic, și arată ca mai jos:

Intrări ale operației
SAU (OR).
Pot fi folosite mai
mult de 2 intrări!



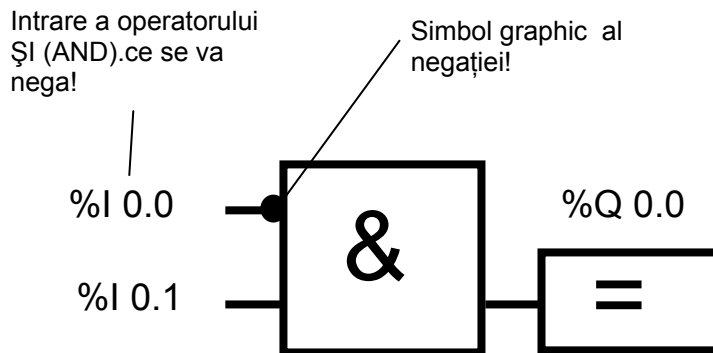
5.7.3 Negația

i

În operațiile logice, este adesea necesar să se afle dacă un contact normal deschis nu a fost acționat sau dacă un contact normal închis fost acționat și ca urmare nu se aplică tensiunea pe intrarea corespunzătoare.

Acest lucru se întâmplă atunci când vom programa o **NEGAȚIE** pe o intrare a operatorilor ȘI sau SAU.

In the function block diagram FBD, the negation of an input at an AND operation is programmed with the following graphic representation: În blocul de funcții cu diagramă FBD, negația unei intrări ale operatorului ȘI (AND) este programată folosind simbolul graphic de mai jos:



Adică, tensiunea este aplicată pe ieșirea %Q 0.0 numai când %I 0.0 **NU** este conectat ȘI %I 0.1 este conectat.

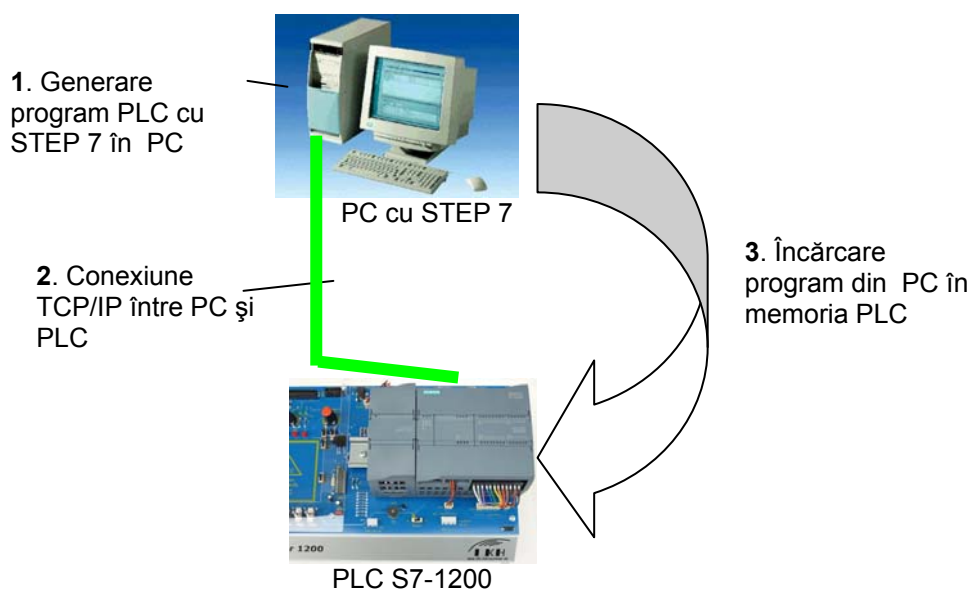
5.8 Modul de generare a programului automatului. Modul în care programul ajunge în memoria automatului



Programul automatului PLC este generat pe un calculator PC folosind mediul de programare STEP 7, și este depozitat temporar pe calculator.

După ce PC-ul este conectat prin interfața TCP / IP la automat, programul poate fi transferat cu ajutorul unei funcții de încărcare în memoria PLC.

Calculatorul PC nu mai este necesar în continuare pentru rularea programului în automat.



Note: Secvențele exacte de operare sunt descrise pas cu pas în capitolele de mai jos

6. Configurarea și utilizarea automatului programabil SIMATIC S7-1200

i

Gama de module :

Sistemul de automatizare SIMATIC S7-1200 este un sistem modular care oferă următoarea gamă de module:

- Module unități centrale CPU de diverse capacități, intrări / ieșiri integrate și interfață PROFINET (de exemplu, CPU1214C)



- Sursă de alimentare PM cu intrare de 120/230V curent alternativ, 50Hz/60Hz, 1.2A/0.7A și ieșire de 24V/2.5A current continuu



- Plăci de semnal SB pentru a suplimenta numărul de intrări / ieșiri analogice și digitale, fără ca dimensiunea procesorului să se schimbe

(plăcile de semnal pot fi utilizate cu modulele de unitate central CPU: 1211C, 1212C și 1214C)





- Module de extensie de semnal SM pentru intrări și ieșiri analogice și digitale suplimentare

(împreună cu unitățile centrale CPU 1212C pot fi utilizate un număr de maxim 2 module de extensie de semnal SM, în timp ce cu CPU 1214C se pot utiliza maximum 8 module de extensie de semnal SM)



- Module de comunicație CM pentru comunicație serială RS 232/RS 485 (atât cu CPU 1211C/1212C cît și cu CPU 1214C, pot fi utilizate până la 3 module de comunicație CM)



- Modul Switch Compact CSM cu 4 conectori priză de tip RJ45 pentru conexiune Ethernet de 10/100 MBit/s



- Carduri de memorie SIMATIC de 2MB sau 24MB pentru stocarea datelor de program și pentru facilitarea înlocuirii modulelor CPU în timpul operațiilor de întreținere



Notă: Pentru încercarea exemplurilor propuse în acest manual este suficientă utilizarea unui modul de unitate centrală CPU cu intrări / ieșiri digitale integrate.

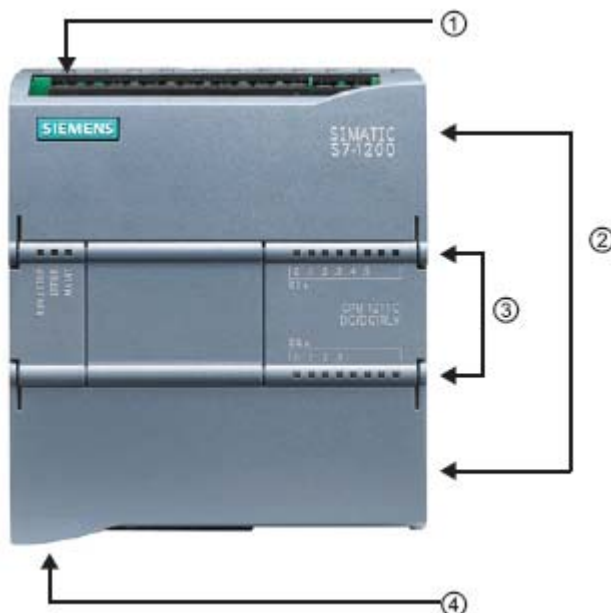


Elemente de bază ale modulelor unitate centrală:

Cu o tensiune de alimentare integrată (prin conexiunea de 24V) și intrările și ieșirile integrate, modulul unitate centrală CPU S7-1200 este gata de funcționare, fără alte componente suplimentare.

Pentru comunicația cu un dispozitivul de programare, modulul unitate centrală CPU este echipat cu un port TCP/ IP integrat.

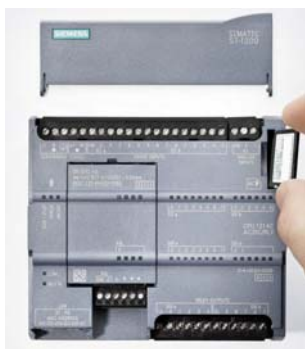
Printr-o rețea Ethernet, modulul unitate centrală CPU este capabil să comunice cu dispozitive de tip panou operator HMI (human machine interface) sau cu alte module CPU.



- ① Conexiunea de alimentare de 24V
- ② Bloc terminal de conexiuni pentru cablarea cu procesul (sub clapeta de protecție)
- ③ LED-uri de stare a intrărilor / ieșirilor integrate și regimurilor de operare a modulul unitate centrală CPU
- ④ Conexiune TCP/IP (în partea de a CPU)

Cardul de memorie SIMATIC (MC) stochează programul , datele, date de sistem, fișiere și proiecte. Acesta poate fi folosit pentru următoarele:

- Transferul unui program la mai multe module unitate centrală CPU
- Actualizarea firmware-ului modulelor unitate centrală CPU, al modulelor de semnale SM și al modulelor de comunicație CM

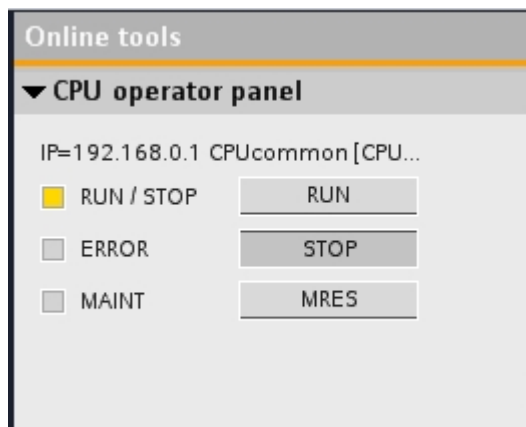


Regimurile de operare al modulelor unitate centrală CPU

Modulul unitate centrală CPU are următoarele regimuri de operare:

- În regimul de operare **STOP**, modulul unitate centrală CPU nu execută programul, și permite încărcarea unui proiect
- În regimul de operare **STARTUP**, modulul unitate centrală CPU efectuează pornirea.
- În regimul de operare **RUN** (de funcționare), programul este executat ciclic. Proiectele nu pot fi încărcate în modulul unitate centrală CPU în timpul regimului de operare RUN.

Modulul unitate centrală CPU nu are un comutator fizic pentru schimbarea regimului de lucru. Regimul de operare (**STOP sau RUN**) este schimbat cu ajutorul butonului de pe panoul de operare al mediului de programare STEP7 Basic. În plus, panoul de operare STEP7 Basic este prevăzut cu butonul **MRES** pentru a efectua o resetare generală a memoriei și are prevăzute LED-uri de afișare a stării CPU.



Culoarea LED-ului de stare **RUN / STOP** de pe panoul frontal al modului indică regimul de operare curent.



- lumină **galbenă** indică regimul **STOP**.
- lumină **verde** indică regimul **RUN**.
- lumină **intermitentă** indică regimul **STARTUP**.

În plus, există LED-ul **ERROR** pentru a indica erorile și LED-ul **MAINT** pentru a indica faptul că sunt necesare operațiuni de mentenanță.

7. Exemplu: Comanda unei prese



Acest program exemplu realizează comanda unei prese.

O presă, având un sistem de protecție a mâinilor operatorului, va fi pornit cu butonul **START - S3** numai când ecranul de protecție este închis. Starea ecranului este supravegheată de un senzor „Ecran de protecție închis” **B1**.

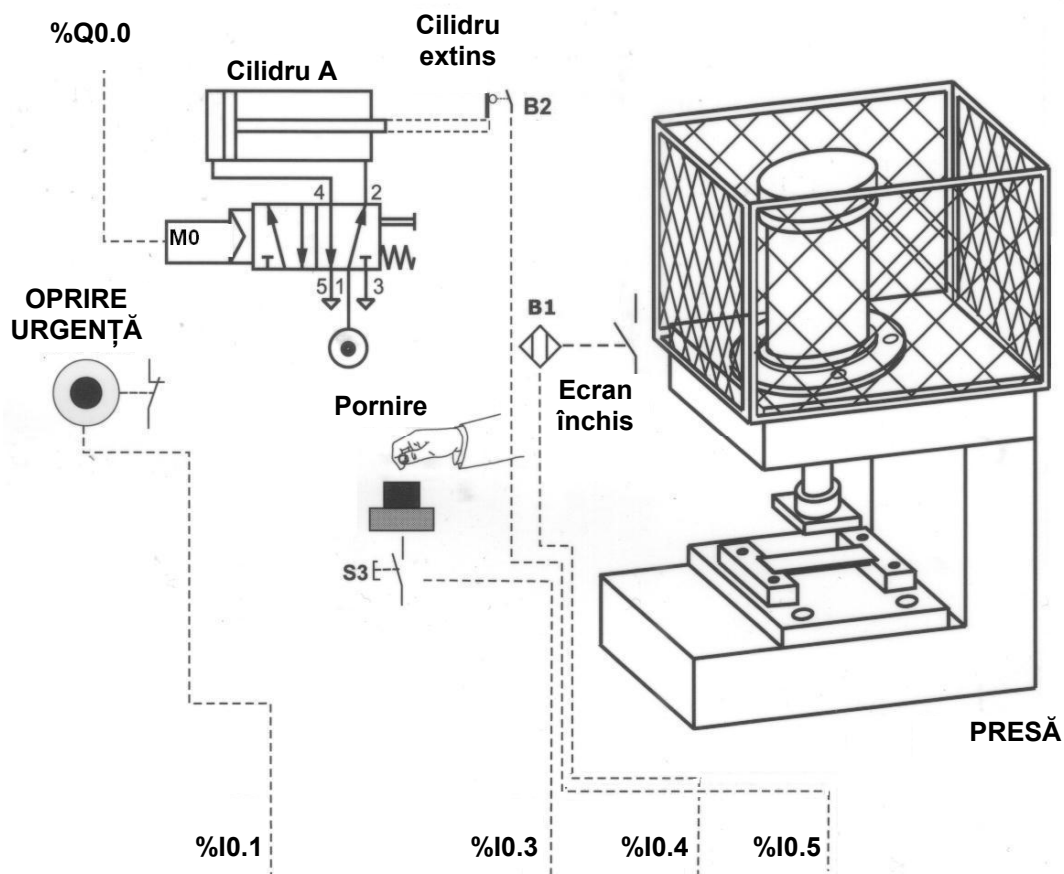
La comanda **START**, este acționat un sertar pneumatic cu 5/2 căi, **M0** pentru a asigura presarea unei piese din plastic.

Presă se retrage din nou în cazul în care:

- se apasă butonul **OPRIRE URGENTĂ** (contact normal închis)
- sau senzorul „Ecran de protecție închis” **B1** nu mai este activ
- sau senzorul „Cilindru extins” **B2** se activează .

Listă de atribuire:

Adresă	Simbol	Observații
%I 0.1	OPRIRE URGENȚĂ	buton (contact normal închis)
%I 0.3	S3	buton Start S3 (contact normal deschis)
%I 0.4	B1	Senzor "Ecran de protecție închis" (contact n.d.)
%I 0.5	B2	Senzor „Cilindru extins” (contact normal deschis)
%Q 0.0	M0	Extinde Cilindru A



8. Programarea funcționării presei la automatul SIMATIC S7-1200



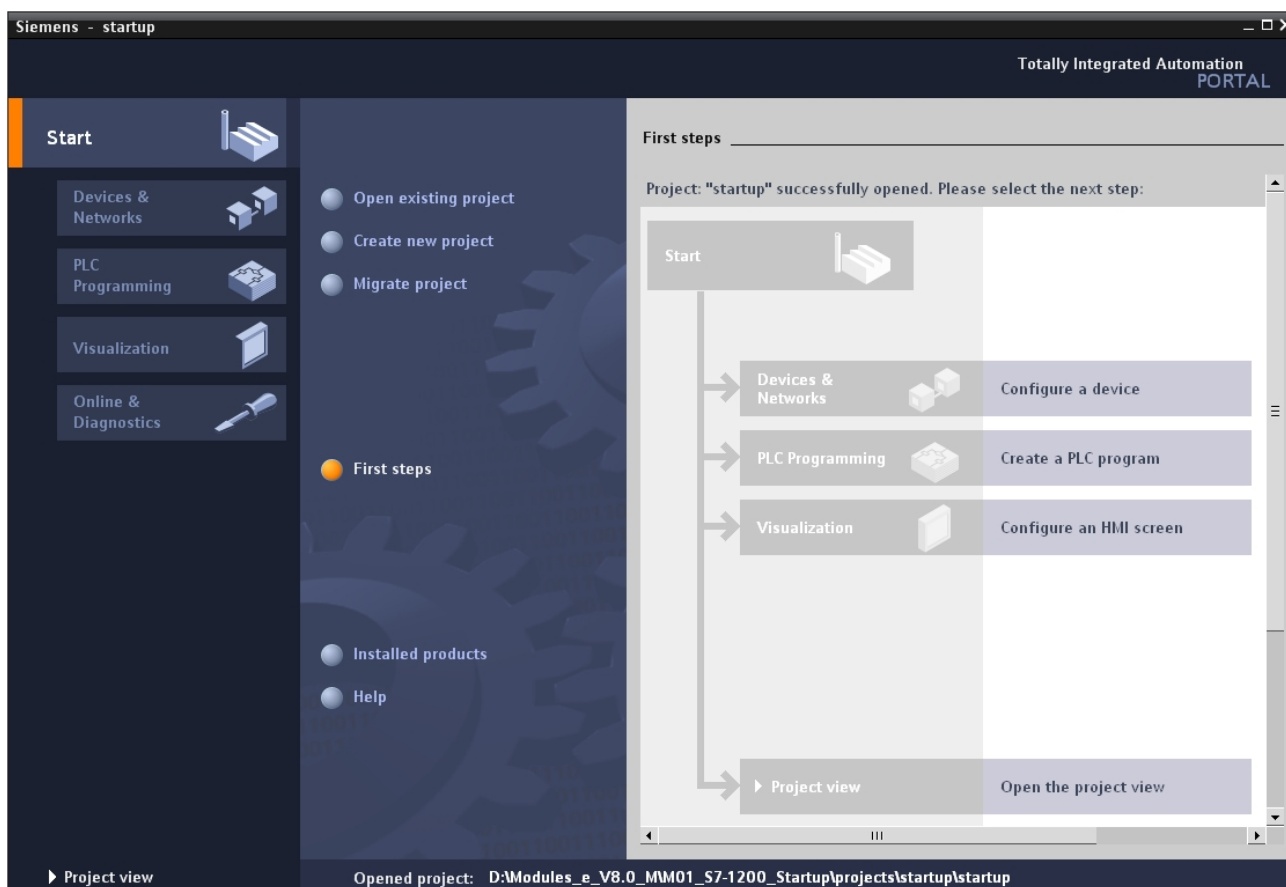
Mediul de dezvoltare **'Totally Integrated Automation Portal'** gestionează proiectul și asigură elaborarea programului.

Printr-o interfață unitară, portalul asigură configurarea, parametrizarea și programarea componentelor cum ar fi: controlerul, dispozitivele de vizualizare și elementele de rețea. Sunt asigurate și instrumente online de diagnosticare a erorilor.

Mediul de dezvoltare **'Totally Integrated Automation Portal'** are două moduri de prezentare: prezentare tip portal și prezentare tip proiect.

8.1. Prezentare tip PORTAL

Prezentarea tip portal oferă o prezentare orientată spre activități a instrumentelor pentru procesarea proiectului. Prin aceasta prezentare, se poate decide rapid ceea ce se dorește să se facă și se apelează instrumentul potrivit pentru sarcina respectivă. Dacă este necesar, trecerea la prezentarea tip portal are loc în mod automat, pentru activitatea selectată. pentru sarcina selectată. In continuare, sunt prezentate noțiuni de bază și sprijiniți primii pași în programare.



Notă: In colțul din stânga jos, puteți trece de la prezentarea tip portal la prezentarea tip proiect!

8.2. Prezentare tip PROIECT



Prezentarea tip proiect este o prezentare structurată a tuturor părților constitutive ale proiectului. Ca o regulă de bază, bara de meniu cu barele de funcții este situată în partea de sus, meniul de navigare cu toate părțile componente ale unui proiect - în stânga, iar pe elementele de acțiune (cu instrucțiuni și biblioteci, de exemplu), în dreapta.

Dacă un element (în imaginea următoare de exemplu, blocul program FC1) este selectat în meniul de navigație proiect, aceasta este afișat în centru și poate fi prelucrat acolo.

The screenshot displays the Siemens TIA Portal interface. On the left, the 'Project tree' shows a hierarchy: 'start-up' > 'Control press' > 'Program blocks' > 'Program press'. The 'Details view' is currently empty. The main workspace shows the 'Interface' table for the selected 'Program press' block:

Name	Data type	Comment
1 Input		
2 Emerg_OFF	Bool	Emergency OFF
3 Start	Bool	pushbutton START
4 B_safety_fence	Bool	sensor safety fence closed
5 B_cylinder	Bool	sensor cylinder moved out
6 Output		
7 M_press	Bool	press cylinder

Below the table, the 'Block title' is 'program to control pressing of plastic moldings'. 'Network 1' is titled 'control_press cylinder' and contains a ladder logic diagram with the following elements:

- Inputs: #Start, #B_safety_fence, #Emerg_OFF, #B_safety_fence, #B_cylinder.
- Logic: An AND gate (&) with inputs #Start and #B_safety_fence. Its output is connected to the S (Set) input of an SR flip-flop block labeled #HM01.
- Reset: An OR gate (>=1) with inputs #Emerg_OFF and #B_safety_fence. Its output is connected to the R1 (Reset) input of the #HM01 flip-flop.
- Output: The Q output of the #HM01 flip-flop is connected to an output coil (=) labeled #M_press.



Notă: În colțul din stânga jos, puteți trece de la prezentarea tip proiect la prezentarea tip portal!

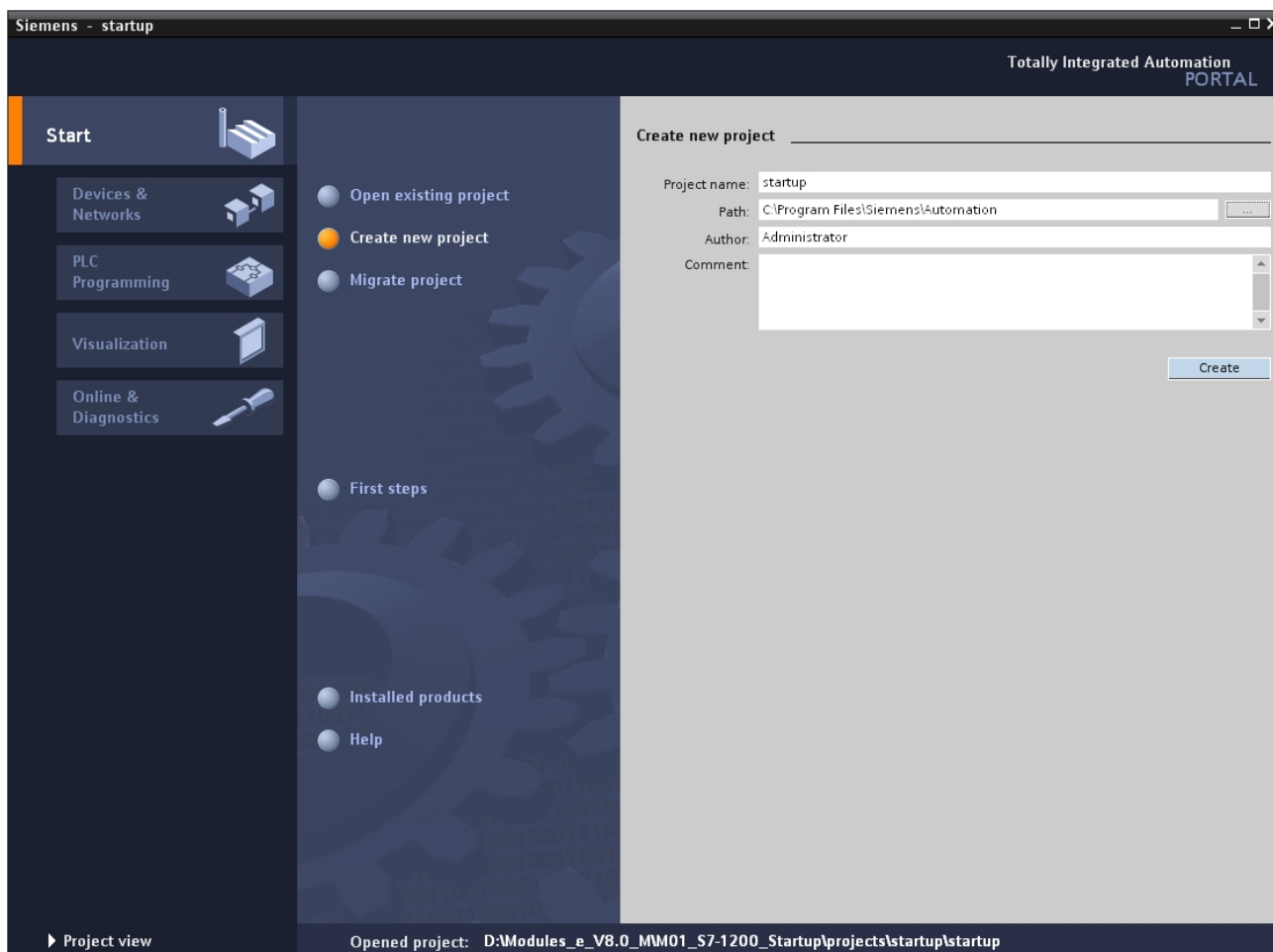


Cu pașii care urmează, se crează un proiect pentru SIMATIC S7-1200 și se programează modul de rezolvare al aplicației:

PASUL 1. Instrumentul principal este mediul de dezvoltare **'Totally Integrated Automation Portal'** care se deschide cu un dublu click (→ Totally Integrated Automation Portal V10)



PASUL 2. Programele pentru automatele programabile SIMATIC S7-1200 sunt gestionate sub formă de proiecte. Creerea unui astfel de proiect este prezentată mai jos în formatul de prezentare tip portal (→ Generate new project → startup → Create)





PASUL 3. Aici sunt prezentați primii pași de configurare a proiectului realizați cu **'First Steps'**. Mai întâi se configurează dispozitivul cu **'Configure a device'** (→ First steps → Configure a device)

Siemens - startup

Totally Integrated Automation
PORTAL

Start

- Devices & Networks
- PLC Programming
- Visualization
- Online & Diagnostics

● Open existing project

● Create new project

● Migrate project

● First steps

● Installed products

● Help

First steps

Project: "startup" successfully opened. Please select the next step:

- Start
- Devices & Networks → Configure a device
- PLC Programming → Create a PLC program
- Visualization → Configure an HMI screen
- Project view → Open the project view

▶ Project view

Opened project: C:\Program Files\Siemens\Automation\startup\startup



PASUL 4. Apoi, inserăm un dispozitiv nou cu 'Add new device' scriind la 'Device name' numele dispozitivului **Control Press (Comanda Presă)**. Pentru aceasta alegem din catalog unitatea central de comandă '**CPU1214C**' cu codul de comandă corespunzător cu CPU de care dispunem pentru experimentări (ex:6S7 214-1AE30-0XB0) (→ Insert new device → Control press → CPU1214C → 6ES7 → Add)

Siemens - startup

Totally Integrated Automation
PORTAL

Start

Devices & Networks

PLC Programming

Visualization

Online & Diagnostics

Show all devices

Add new device

Configure networks

Help

Add new device

Device name: Control press

SIMATIC PLC

SIMATIC HMI

FLC

SIMATIC S7-1200

CPU

CPU 1211C

CPU 1212C

CPU 1214C

6ES7 214-1AE30-0XB0

6ES7 214-1BE30-0XB0

6ES7 214-1HE30-0XB0

Unspecific CPU 1200

Device: CPU 1214C DC/D0DC

Order no.: 6ES7 214-1AE30-0XB0

Version: V1.0

Description:

50KB work memory, 24VDC power supply with DI14 x 24VDC SINK/SOURCE, DO10 x 24VDC and AI2 onboard; 6 high speed counters and 2 pulse outputs onboard; signal board expands onboard I/O; up to 3 communication modules for serial communication; up to 8 signal modules for I/O expansion; 0.1ms/1000 instructions; PROFINET connection for programming, HMI and PLC to PLC communication

Open device view

Add

Project view

Opened project: C:\Program Files\Siemens\Automation\startup\startup



PASUL 5. Acum , mediul comută automat din prezentarea tip portal în prezentarea tip proiect deschisă la configurarea hardware. In acest ecran se pot adăuga la automatul programabil module suplimentare preluate din catalog (din partea dreaptă) și în **'Device overview'** (stânga jos), se pot seta adresele intrărilor / ieșirilor . Tot în acest ecran se poate vedea că intrările integrate ale unității central CPU au adresele %I0.0 la %I1.5 iar ieșirile integrate au adresele %Q0.0 to %Q1.1 (→ Device overview → DI14/DO10 → 0...1)

Module	Slot	I address	Q address	Type	Order no.	Firmware	Comment
	102						
	101						
Control press	1			CPU 1214C DC/DC/DC	6ES7 214-1AE30-0XB0	V1.0	
DI14/DO10	1.1	0...1	0...1	DI14/DO10			
AI2	1.2	Value range: 0...1023					
	1.3						
HSC_1	1.16			High speed counter.			
HSC_2	1.17			High speed counter.			
HSC_3	1.18			High speed counter.			
HSC_4	1.19			High speed counter.			



PASUL 6. Pentru ca mediul de dezvoltare să acceseze mai târziu unitatea centrală CPU corectă, adresa IP și masca de subrețea trebuie să fie setate.

(→ Properties → General → Ethernet addresses → IP address: 192.189.0.1 → Subnet screen form: 255.255.255.0)

(vezi capitolul 4 pentru setarea interfeței de programare)

The screenshot displays the Siemens TIA Portal software interface. The main window shows a rack configuration for a SIMATIC S7-1200. The 'Control press' device is highlighted. The 'Device overview' pane is open, showing the 'Ethernet addresses' tab. The 'IP protocol' section is visible, with the IP address set to 192.168.0.1 and the subnet mask set to 255.255.255.0. The 'Subnet' dropdown is set to 'Not connected'. The 'Router address' is set to 192.168.0.1. The status bar at the bottom indicates 'Project startup created.'



PASUL 7. Deoarece în programarea modernă, nu se mai progrează cu adrese absolute ci cu variabile simbolice, **variabilele globale ale automatului** programabil trebuie specificate acum.

Aceste variabile globale ale automatului programabil sunt numele descriptive cu comentarii pentru acele intrări și ieșiri, care sunt utilizate în cadrul programului. Mai târziu, în timpul programării, variabilele globale ale automatului programabil pot fi accesate prin intermediul acestor nume.

Aceste variabile globale pot fi utilizate în întregul program în toate blocurile.

În acest scop, selectați mai întâi în meniul de navigare '**Control Press[CPU1214C DC/DC/DC]**' și apoi '**PLC tags**'. Cu un dublu click, deschideți tabelul '**PLC tags**' și introduceți numele pentru intrări și ieșiri, așa cum se arată mai jos (→ Control Press[CPU1214C DC/DC/DC] → PLC tags → PLC tags)

The screenshot shows the Siemens TIA Portal software interface. The main window displays the 'PLC tags' configuration table for the 'Control press [CPU 1214C]' device. The table lists several tags with their names, data types, addresses, and comments.

Name	Data type	Address	Retain	Comment
Emerg-OFF	Bool	%I0.1	<input type="checkbox"/>	Emergency-OFF (nc contact)
S3	Bool	%I0.3	<input type="checkbox"/>	pushbutton START S3 (no contact)
B1	Bool	%I0.4	<input type="checkbox"/>	sensor safety/fence closed (no contact)
B2	Bool	%I0.5	<input type="checkbox"/>	sensor cylinder A moved out (no contact)
M0	Bool	%Q0.0	<input type="checkbox"/>	move out cylinder A

Below the table, the 'Emerg-OFF' tag is selected, and its properties are shown in the 'Tag' configuration panel. The 'General' tab is active, showing the following details:

- Name:** Emerg-OFF
- Data type:** Bool
- Address:** %I0.1
- Retained:**
- Comment:** Emergency-OFF (nc contact)
- Time stamp:**
 - Date created: 11/28/2009 10:50 PM
 - Last modified: 7/12/2010 1:36 PM



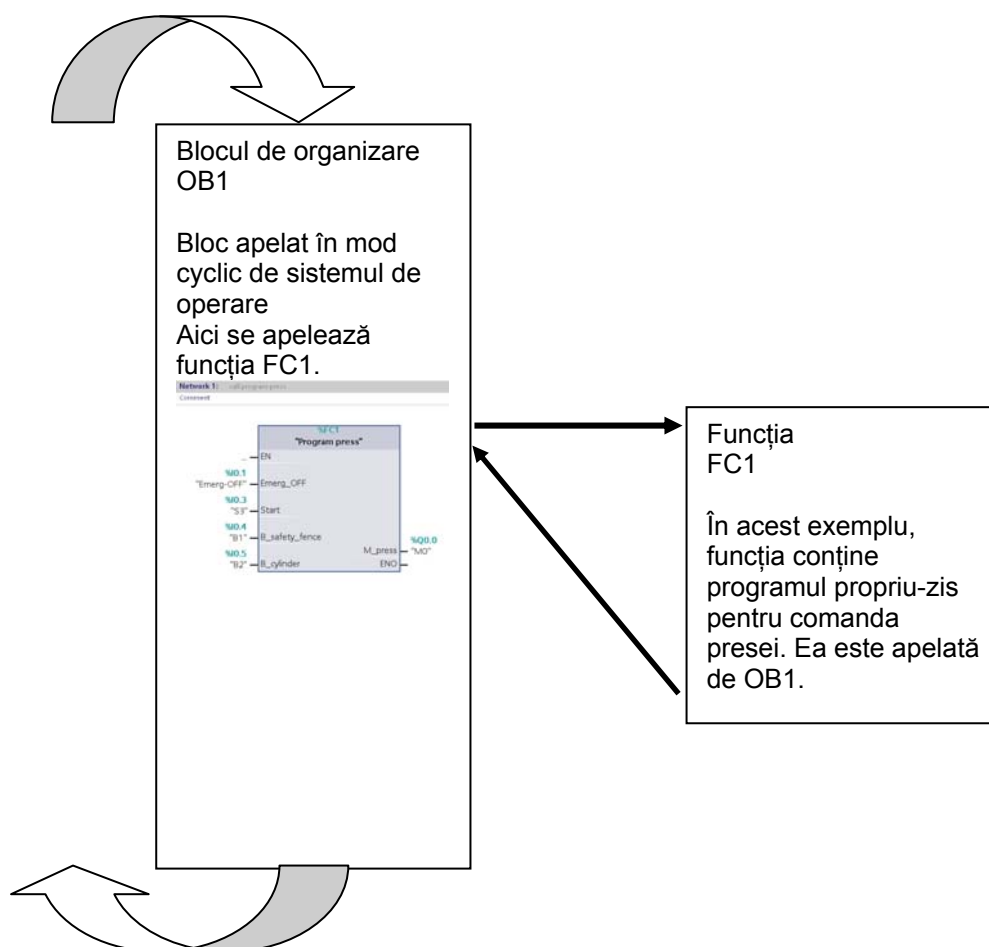
PASUL 8. Secvențele de program sunt scrise în așa-numitele blocuri. În mod obligatoriu, blocul de organizare OB1 există deja.

El reprezintă interfața cu sistemul de operare al CPU, fiind apelat de acesta în mod automat și procesat în mod ciclic.

Din acest bloc de organizare, pot fi apelate la rândul lor blocuri suplimentare, pentru o programare structurată, ca de exemplu funcția FC1.

Scopul acestora este de a descompune o sarcină generală în sarcini parțiale. Acestea pot fi rezolvate mult mai ușor iar funcționalitatea lor poate fi testată.

Structura programului exemplu:





UNIUNEA EUROPEANĂ



GUVERNUL ROMÂNIEI
MINISTERUL MUNCII, FAMILIEI,
PROTECȚIEI SOCIALE ȘI
PERSOANELOR VÂRSTICE
AMPOSORU



Fondul Social European
POSDRU 2007-2013



Instrumente Structurale
2007-2013



OIPOSORU



UNIVERSITATEA
TEHNICA
DIN CLUJ-NAPOCA



PASUL 9. Pentru a genera funcția FC1, in Meniul de Navigare se selectează mai întâi **'Control Press[CPU1214C DC/DC/DC]'** și apoi **'Program blocks'**. Apoi se face dublu click pe **'Add new block'** (→ Control Press[CPU1214C DC/DC/DC] → Program blocks → Add new block)

The screenshot shows the Siemens TIA Portal interface. The 'Project tree' on the left shows the project structure, with 'Add new block' highlighted under 'Program blocks'. The main window displays a table of PLC tags for the 'Emerg-OFF' block.

	Name	Data type	Address	Retain	Comment
1	Emerg-OFF	Bool	%I0.1	<input type="checkbox"/>	Emergency-OFF (nc contact)
2	S3	Bool	%I0.3	<input type="checkbox"/>	pushbutton START S3 (no contact)
3	B1	Bool	%I0.4	<input type="checkbox"/>	sensor safety fence closed (no contact)
4	B2	Bool	%I0.5	<input type="checkbox"/>	sensor cylinder A moved out (no contact)
5	M0	Bool	%Q0.0	<input type="checkbox"/>	move out cylinder A
6				<input type="checkbox"/>	

The 'Details view' at the bottom shows the configuration for the 'Emerg-OFF' tag:


- General**
 - Name: Emerg-OFF
 - Data type: Bool
 - Address: %I0.1
 - Retained:
 - Comment: Emergency-OFF (nc contact)
- Time stamp**
 - Date created: 11/28/2009 10:50 PM
 - Last modified: 7/12/2010 1:36 PM





PASUL 10. Din zona de selecție se selectează o funcție '**Function (FC)**' căruia i se atribuie numele '**Program press**'. Ca limbaj de programare se alege limbajul '**FBD**'. Numerotarea este automată. Pentru că, oricum, această funcție FC1 este apelată mai târziu cu numele său simbolic, numărul funcției nu mai are importanță. Se confirmă selecțiile efectuate cu '**OK**'. (→ Function (FC) → Program Press → FBD → OK)


Add new block

Name:
Program press


Organization block (OB)


Function block (FB)


Function (FC)


Data block (DB)

[more...](#)

Language: FBD

Number: 1

Manual
 Automatic
 Symbolic access only

Description:
Functions are code blocks or subroutines without dedicated memory.

► **Further information**

Add new and open

OK Cancel



PASUL 11. Blocul 'Program Press[FC1]' va fi deschis în mod automat. Cu toate acestea, înainte de scrierea programului, interfața blocului trebuie declarată.

Atunci când interfața este declarată, sunt specificate variabilele locale recunoscute numai în cadrul acestui bloc.

Variabilele locale constau din două grupuri:

- Grupul de parametri care realizează interfața blocului pentru apelul din program.

Tip	Name	Funcție	Disponibil în
Parametri de intrare	Input	Parametri ai căror valori sunt citite de bloc	Funcții, blocuri funcționale și unele tipuri de blocuri de organizare
Parametri de ieșire	Output	Parametri ai căror valori sunt scrise de bloc	Funcții și blocuri funcționale
Parametri InOut	InOut	Parametri ai căror valori sunt citite de bloc când este apelat iar după procesare sunt suprascrise	Funcții și blocuri funcționale

- Datele locale, care sunt folosite pentru stocarea rezultatelor intermediare.

Tip	Name	Funcție	Disponibil în
Date locale temporare	Temp	Variabile care sunt utilizate la stocarea rezultatelor intermediare temporare. Datele temporare sunt reținute doar pentru un ciclu.	Funcții, blocuri funcționale și blocuri de organizare
Date locale statice	Static	Variabile care sunt utilizate la stocarea rezultatelor intermediare statice în blocul de date momentan. Datele statice sunt reținute pe parcursul a mai multe cicluri, până când se suprascriu.	Blocuri funcționale



PASUL 12. In exemplul nostru este nevoie de declararea următoarelor variabilelor locale:

Intrare (Input):

Emerg_OFF	Aici se introduce butonul de OPRIRE DE URGENȚĂ
Start	Aici se introduce butonul de START
B_safety_fance	Aici se introduce starea senzorului “Ecran de protecție deschis”
B_Cylinder	Aici se introduce starea senzorului “Cilindru extins”

Ieșire (Output):

M_Press	Aici se scrie starea ieșirii de comandă “Cilindru presă”
---------	--

Temporar (Temp):

HM01	Flag auxiliar 01 pentru bistabilul RS
------	---------------------------------------

În cazul nostru toate variabilele sunt de tip boolean 'Bool'; aceasta înseamnă că aceste variabile pot avea doar două stări '0' (false) sau '1' (true).

Pentru mai multă claritate în înțelegerea programului, toate variabilele locale trebuie să fie însoțite de comentarii suficiente.



Siemens - startup

Project Edit View Insert Online Options Tools Window Help

Totally Integrated Automation PORTAL

Project tree: startup > Control press > Program blocks > Program press

PLC Programming

Devices

- startup
 - Add new device
 - Devices & Networks
 - Control press [CPU 1214C DC/DC/DC]
 - Device configuration
 - Online & diagnostics
 - Program blocks
 - Add new block
 - Main [OB1]
 - Program press [FC1]
 - Technological Objects
 - PLC tags
 - Watch tables
 - Text lists
 - Local modules
 - Common data
 - Languages & Resources
 - Online access
 - SIMATIC Card Reader

Interface

Name	Data type	Comment
1 Input		
2 Emerg_OFF	Bool	Emergency OFF
3 Start	Bool	pushbutton START
4 B_safety_fence	Bool	sensor safety fence closed
5 B_cylinder	Bool	sensor cylinder moved out
6 Output		
7 M_press	Bool	press cylinder
8 InOut		
9		
10 Temp		
11 HM01	Bool	temporary memory bit 01
12 Return		
13 Ret_Val	Void	

& >=1 [??] -ol [-]

Block title: program to control pressing of plastic moldings

Comment

Properties Info Diagnostics

General

General

Attributes

Name: Emerg_OFF Data type: Bool

Default value: false



Initial value:

Comment: Emergency OFF

Portal view Overview Program press Project startup opened.



PASUL 13. După declararea variabilelor locale, se poate începe programarea. Pentru a avea o perspectivă mai clară asupra programului, se programează folosind unități logice

numite “networks”. Se poate insera un “network” nou apăsând pe simbolul  ‘Insert network’. Ca și în cazul blocurilor, fiecare “network” trebuie să fie documentat prin comentariu text de descriere, în linia de titlu. Pentru o descriere cu un text mai lung, se poate utiliza câmpul ‘Comment’ (→ 

Pentru implementarea logicii de comandă e nevoie de un bistabil RS ‘SR Flipflop’. Acesta se află în meniul ‘Instructions’ în subdirectorul ‘Bit logic’. Punând cursorul mouse-ului pe un obiect de tip bistabil RS ‘SR Flipflop’, vor fi afișate informații despre acel obiect (→ Instructions → Bit logic → SR)

The screenshot displays the Siemens TIA Portal software interface. The main window shows the 'Program press' block configuration. The 'Interface' table lists the following variables:

Name	Data type	Comment
1 Input		
2 Emerg_OFF	Bool	Emergency OFF
3 Start	Bool	pushbutton START
4 B_safety_fence	Bool	sensor safety fence closed
5 B_cylinder	Bool	sensor cylinder moved out
6 Output		

The 'Block title' is 'program to control pressing of plastic moldings'. The 'Network 1' is 'control_press cylinder'. The 'Properties' pane shows the 'General' tab with the following details:

- Name: Program press
- Type: FC

The right-hand pane shows the 'Instructions' menu, with 'Bit logic' expanded to show 'SR Set reset flip-flop' highlighted.



PASUL 14. Dacă se selectează un anumit obiect și apoi se apasă tasta 'F1' de pe calculatorul PC, vor apare informații online despre acel obiect într-o fereastră situată în partea dreaptă (→ F1)

The screenshot shows the Siemens TIA Portal interface. The main window displays a PLC program for 'Program press'. On the right, the 'Information System' window is open, showing the help page for the 'SR: Set reset flip-flop' instruction. The help page includes a symbol diagram, a parameter table, and a detailed description.

Parameter	Data type	Memory area	Description
<Operand>	BOOL	I, Q, M, D, L	Operand that is specified.
S	BOOL	I, Q, M, D, L	Enabled set
R1	BOOL	I, Q, M, D, L	Enabled reset
Q	BOOL	I, Q, M, D, L	Signal state of the specified operand

Description
 You can use the "Set reset flip-flop" operation to set or reset the bit of the specified operand based on the signal state of the inputs S and R1. If the signal state at input S is "1" and is "0" at input R1, the specified operand is set to "1". If the signal state at input S is "0" and is "1" at input R1, the specified operand is reset to "0".
 Input R1 takes priority over input S. If the signal state is "1" at the two inputs S and R1, the signal state of the specified operand is reset to "0".
 The operation is not executed if the signal state at the two inputs R and S1 is "0". The signal state of the operand then remains unchanged.
 The current signal state of the operand is transferred to output Q and can be queried there.



Notă: În fereastra din dreapta, de online help, sunt furnizate informații detaliate despre funcționarea și cablarea bistabilului RS 'SR Flipflop'.





PASUL 15. Bistabilul RS 'SR Flipflop' se poate trage acum cu mouse-ul sub linia de comentariu din Network 1. (→ SR)

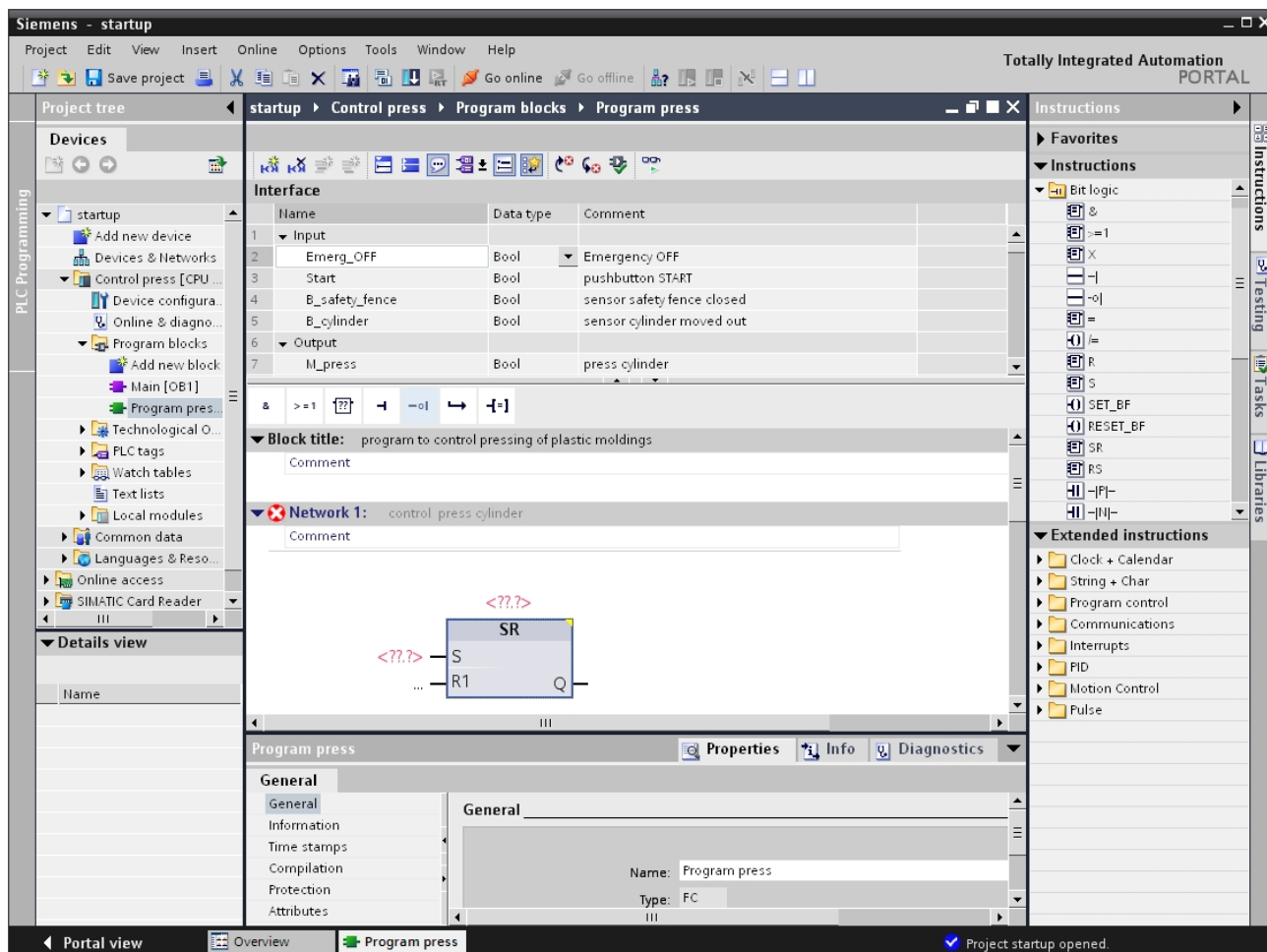
The screenshot displays the Siemens TIA Portal software interface. The main window shows the 'Program press' block in the 'Network 1' editor. The interface table is as follows:

Name	Data type	Comment
Input		
1		
2	Bool	Emergency OFF
3	Bool	pushbutton START
4	Bool	sensor safety fence closed
5	Bool	sensor cylinder moved out
Output		
7	Bool	press cylinder

The network diagram shows a normally open contact labeled 'SR' being added to Network 1. The right-hand sidebar shows the 'Instructions' library with 'SR' selected under 'Bit logic'.



PASUL 16. În pasul următor, se selectează intrarea Set a bistabilului RS 'SR Flipflop' și se face and click pe obiectul  AND (ȘI) din meniul Favorites (→ S → Favorites →  AND)







The screenshot displays the Siemens TIA Portal interface for configuring a PLC program. The main window shows the 'Program press' network, which includes an SR Flipflop block. The interface is divided into several panes:

- Project tree:** Shows the project structure, including 'start-up', 'Control press', and 'Program blocks'.
- Interface:** A table listing the variables used in the program:

Name	Data type	Comment
1	Input	
2	Emerg_OFF	Emergency OFF
3	Start	pushbutton START
4	B_safety_fence	sensor safety fence closed
5	B_cylinder	sensor cylinder moved out
6	Output	
7	M_press	press cylinder
- Network 1:** Shows the configuration of the SR Flipflop block, with the Set (S) input and Reset (R) input connected to the M_press variable.
- Instructions:** A list of available instructions, including the AND symbol selected in the Favorites section.
- Properties:** Shows the general properties of the 'Program press' block, including its name and type (FC).



PASUL 17. În același fel plasăm obiectul  OR (SAU) pe intrarea R1 și obiectul 
 atribuire la ieșirea Q a a bistabilului RS 'SR Flipflop' (→ R1 →  OR → Q → 

The screenshot shows the Siemens TIA Portal interface for a PLC program. The main workspace displays a ladder logic network with the following components:

- Interface Table:**

Name	Data type	Comment	
1	Input		
2	Emerg_OFF	Bool	Emergency OFF
3	Start	Bool	pushbutton START
4	B_safety_fence	Bool	sensor safety fence closed
5	B_cylinder	Bool	sensor cylinder moved out
6	Output		
7	M_press	Bool	press cylinder
- Network 1:** control press cylinder. The logic consists of:
 - An AND gate (&) with two inputs labeled <??.?>.
 - An SR Flipflop (SR) with its S input connected to the output of the AND gate.
 - An OR gate (>=1) with its R1 input connected to the output of the AND gate.
 - The Q output of the SR Flipflop is connected to an assignment instruction (=).
- Right Panel:** Shows the 'Instructions' library with 'Bit logic' instructions expanded, including &, >=1, X, -, -o|, =, /=, R, S, SET_BF, RESET_BF, SR, RS, -|F|-, and -|I|-.
- Bottom Panel:** Shows the 'Program press' properties window with the 'General' tab selected.



PASUL 18. Acum se pot introduce variabile locale în program. Este suficient să se introducă prima literă a numelui variabilei locale în câmpul de introducere. După care se poate selecta variabila dorită din lista care apare. Variabilele locale sunt identificate cu simbolul '#' care precede numele variabilei (→ #M_Press).

The screenshot shows the Siemens TIA Portal software interface for PLC programming. The main window displays a ladder logic network for a control press cylinder. The network consists of two parallel normally open contacts (represented by red boxes with '>=1' and '&') leading to an SR (Set-Reset) coil. The SR coil is labeled 'SR' and has a reset input 'R1' and a set input 'S'. The output of the SR coil is connected to a coil labeled 'M' with the variable name '#M_press'.

Name	Data type	Comment	
1	Input		
2	Emerg_OFF	Bool	Emergency OFF
3	Start	Bool	pushbutton START
4	B_safety_fence	Bool	sensor safety fence closed
5	B_cylinder	Bool	sensor cylinder moved out
6	Output		
7	M_press	Bool	press cylinder





PASUL 19. În mod asemănător se adaugă și celelalte variabile locale. La obiectul OR (SAU), trebuie adăugată o intrare suplimentară . Pentru aceasta, se selectează intrarea cea mai de jos cu click dreapta și apoi se apasă 'Insert input' (→ Insert input)

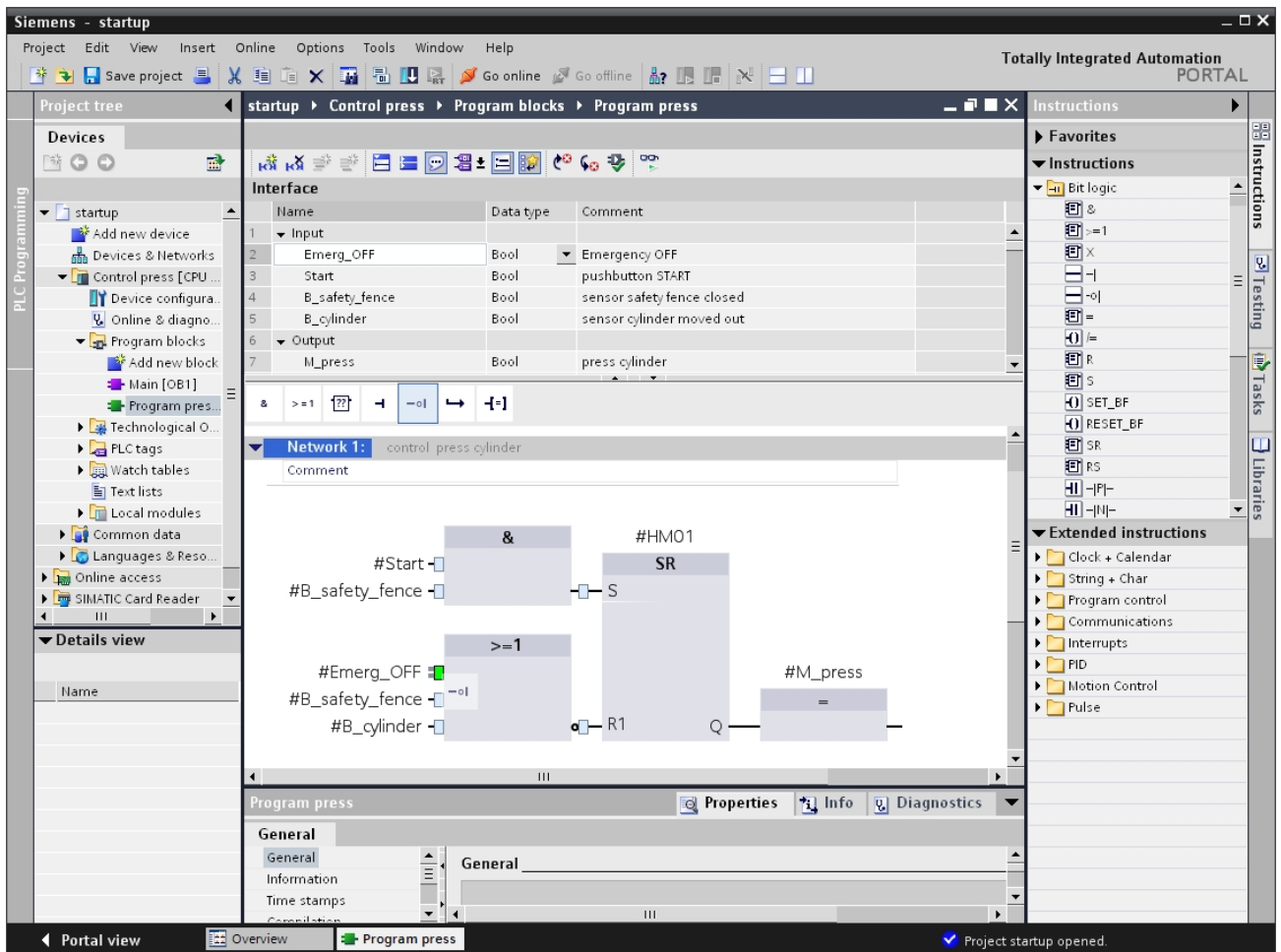
The screenshot shows the Siemens TIA Portal interface. The main window displays a ladder logic network for 'control_press.cylinder'. The network contains an AND instruction (&) with inputs #Start and #B_safety_fence, followed by an SR (Set Reset) instruction with input #Emerg_OFF and output R1. A comparison instruction (>=1) is connected to the SR instruction. The output of the network is connected to an equals sign (=) instruction, which is connected to the output #M_press. A context menu is open over the '>=1' instruction, with the 'Insert input' option highlighted. The interface includes a Project tree on the left, a menu bar at the top, and a right-hand sidebar with various toolbars and panels.



PASUL 20. Se va atribui o variabilă locală și la intrarea suplimentară.

Dacă o intrare trebuie negată, de trage simbolul de negație  din meniul 'Favorites'

la acea intrare. (→ Favorites → )



Name	Data type	Comment	
1			
2	Input		
3	Emerg_OFF	Bool	Emergency OFF
4	Start	Bool	pushbutton START
5	B_safety_fence	Bool	sensor safety fence closed
6	B_cylinder	Bool	sensor cylinder moved out
7	Output		
8	M_press	Bool	press cylinder

Network 1: control press cylinder

Logic: (#Start & #B_safety_fence) SR #HMO1; (#B_safety_fence NOT) & #Emerg_OFF >=1 R1; #M_press =



PASUL 21. În continuare se configurează blocul procesat în mod ciclic 'Main[OB1]' prin meniul 'Properties'. Proprietățile blocului pot fi modificate (→ Properties → Main[OB1])

The screenshot displays the Siemens TIA Portal interface for configuring a cyclic program block. The main window shows a ladder logic diagram with the following components:

- Interface Table:**

Name	Data type	Comment	
1	Input		
2	Emerg_OFF	Bool	Emergency OFF
3	Start	Bool	pushbutton START
4	B_safety_fence	Bool	sensor safety fence closed
5	B_cylinder	Bool	sensor cylinder moved out
6	Output		
7	M_press	Bool	press cylinder
- Ladder Logic Diagram:**
 - Input **#Start** is connected to the **S** (Set) input of an **SR** (Set-Reset) block.
 - Input **rg_OFF** is connected to the **R** (Reset) input of the **SR** block.
 - Input **B_safety_fence** is connected to the **R1** (Reset) input of an **=** (Assignment) block.
 - Input **B_cylinder** is connected to the **R1** input of the **=** block.
 - The output of the **=** block is connected to the **Q** (Output) of the **SR** block.
 - The output of the **SR** block is connected to the **Q** input of the **#M_press** output block.
- Context Menu:** A right-click context menu is open over the **Properties** button at the bottom of the diagram, with the **Properties** option highlighted.
- Project Tree:** Shows the project structure: **startup** > **Control press** > **Program blocks** > **Program press**.
- Instructions Panel:** Lists various instructions such as **&**, **>=1**, **SR**, **RS**, **SET_BF**, **RESET_BF**, **R**, **S**, **OR**, **AND**, **NOT**, **EQ**, **NEQ**, **GT**, **LT**, **GE**, **LE**, **EQV**, **NEQV**, **ANDV**, **ORV**, **NOTV**, **ANDI**, **ORI**, **NOTI**, **ANDD**, **ORD**, **NOTD**, **ANDC**, **ORC**, **NOTC**, **ANDB**, **ORB**, **NOTB**, **ANDW**, **ORBW**, **NOTW**, **ANDL**, **ORL**, **NOTL**, **ANDH**, **ORH**, **NOTH**, **ANDS**, **ORS**, **NOTS**, **ANDM**, **ORM**, **NOTM**, **ANDN**, **ORN**, **NOTN**, **ANDR**, **ORR**, **NOTR**, **ANDT**, **ORT**, **NOTT**, **ANDF**, **ORF**, **NOTF**, **ANDD**, **ORD**, **NOTD**, **ANDC**, **ORC**, **NOTC**, **ANDB**, **ORB**, **NOTB**, **ANDW**, **ORBW**, **NOTW**, **ANDL**, **ORL**, **NOTL**, **ANDH**, **ORH**, **NOTH**, **ANDS**, **ORS**, **NOTS**, **ANDM**, **ORM**, **NOTM**, **ANDN**, **ORN**, **NOTN**, **ANDR**, **ORR**, **NOTR**, **ANDT**, **ORT**, **NOTT**, **ANDF**, **ORF**, **NOTF**.



UNIUNEA EUROPEANĂ



GUVERNUL ROMÂNIEI
MINISTERUL MUNCII, FAMILIEI,
PROTECȚIEI SOCIALE ȘI
PERSOANELOR VÂRSTNICE
AMPOSORU



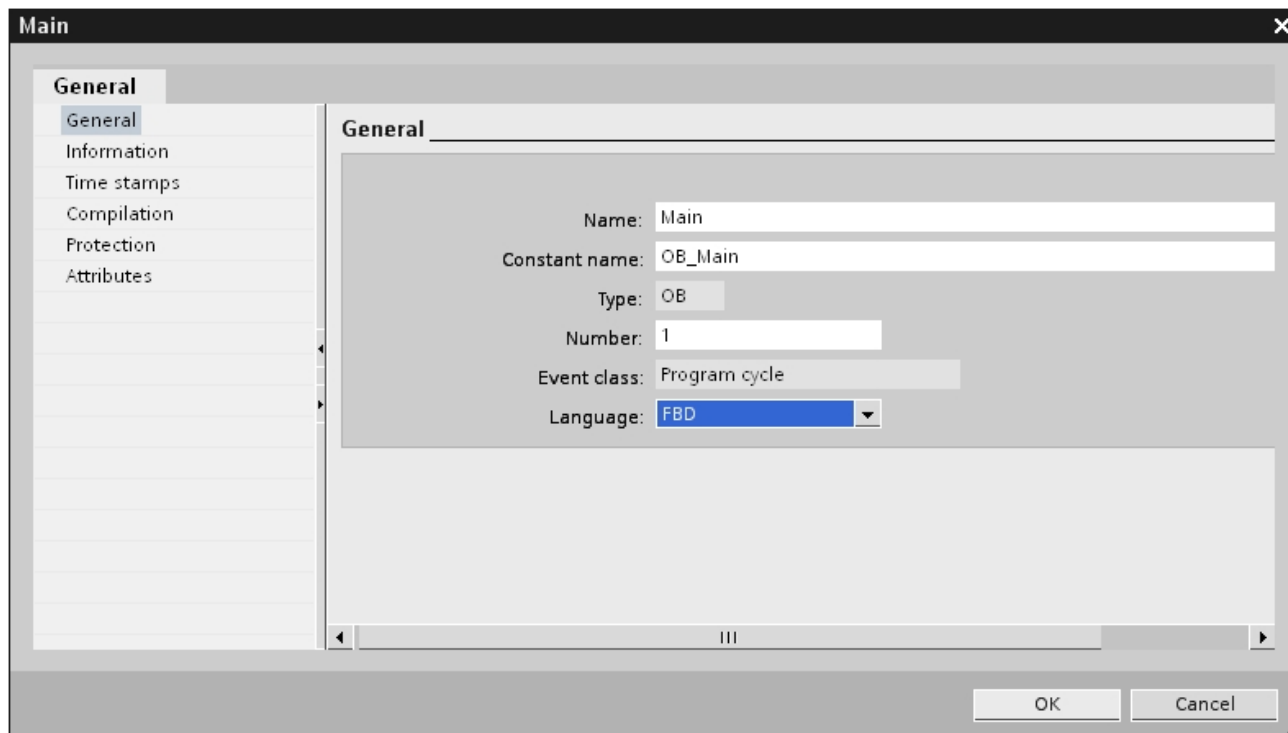
Fondul Social European
POSDRU 2007-2013



Instrumente Structurale
2007-2013



PASUL 22. In cadrul proprietăților se selectează la '**Language**' limbajul de programare , acesta fiind diagram de blocuri funcționale'**FBD**'. (→ FBD → OK)





PASUL 23. Așa cum s-a menționat mai devreme blocul funcție "Program Press" trebuie apelat din blocul de programului principal Main[OB1]. Altfel acest bloc va fi ignorat. Se deschide blocul programului principal cu dublu click pe 'Main[OB1]' (→ Main[OB1])

The screenshot displays the Siemens TIA Portal software interface for configuring a PLC program. The main window shows the 'Program press' block configuration for 'Network 1: control_press_cylinder'. The interface includes a menu bar, a toolbar, and several panels:

- Project tree:** Shows the project structure with 'Main [OB1]' selected.
- Interface table:** Lists the block's inputs and outputs.
- Network 1:** Shows the ladder logic diagram for the 'control_press_cylinder' network.
- Properties panel:** Shows the 'General' properties for the 'Program press' block.
- Instructions panel:** Lists available instructions and extended instructions.

Name	Data type	Comment
Input		
1		
2	Bool	Emergency OFF
3	Bool	pushbutton START
4	Bool	sensor safety fence closed
5	Bool	sensor cylinder moved out
Output		
6		
7	Bool	press cylinder

Network 1: control_press_cylinder

Comment

The ladder logic diagram shows the following logic:

- Inputs: #Start, #B_safety_fence, #Emerg_OFF, #B_safety_fence, #B_cylinder.
- Logic: An AND gate (&) combines #Start and #B_safety_fence. The output of the AND gate is connected to the S (Set) input of an SR (Set-Reset) flip-flop.
- Logic: An OR gate (>=1) combines #Emerg_OFF, #B_safety_fence, and #B_cylinder. The output of the OR gate is connected to the R1 (Reset) input of the SR flip-flop.
- Output: The Q output of the SR flip-flop is connected to the #M_press output.

Program press Properties

General

Information

Time stamps

General



PASUL 24. Blocul "Program Press" poate fi tras simplu, cu Drag&Drop, din meniul de navigare în Network 1 al blocului program principal Main[OB1]. Acest network va fi de asemenea documentat prin comentarii text în blocul Main[OB1] (→ Program Press)

The screenshot displays the Siemens TIA Portal software interface. The main window is titled "Siemens - startup" and shows the "Totally Integrated Automation PORTAL" environment. The interface is divided into several panes:

- Project tree (left):** Shows the project structure, including "startup", "Control press", and "Program blocks". The "Program press [FC1]" block is highlighted.
- Interface (center):** Shows the ladder logic network for "Network 1: call program press". A "Program press [FC1]" block is being dragged into the network.
- Properties (bottom right):** Shows the "General" properties of the "Main" block, including "Name: Main", "Constant name: OB_Main", "Type: OB", and "Number: 1".
- Instructions (right):** Shows a list of available instructions, including "General", "Bit logic", "Timers", "Counters", "Compare", "Math", "Move", and "Convert".

The status bar at the bottom indicates "Project startup opened."



PASUL 25. În continuare, parametrii de interfață (intră-ieșiri) ai blocului "Program Press" trebuie conectați la variabilele globale ale automatului PLC. Este suficient să tastăm prima literă din numele variabilei globale dorite în câmpul din dreapta variabilei locale a blocului. Apoi din lista ce apare se selectează operandul dorit (→ "EMERGENCY OFF")

The screenshot displays the Siemens TIA Portal software interface. The main window shows the 'Program press' block configuration. The 'Interface' section contains a table with columns for Name, Data type, and Comment. The first row is labeled 'Temp'. Below the table, the 'Block title' is set to 'main program'. A 'Network 1' is defined as 'call program press'. The main workspace shows a ladder logic diagram with a normally open contact labeled '%FC1' connected to the 'EN' (Enable) input of the 'Program press' block. The output of the block is connected to a coil labeled 'Emerg_OFF'. The 'Emerg-Off' block is shown in a separate window below the main workspace. The left sidebar shows the 'Project tree' with the 'Program press [FC1]' block selected. The right sidebar shows the 'Instructions' list, including 'General', 'Bit logic', 'Timers', 'Counters', 'Compare', 'Math', 'Move', 'Convert', 'Extended instructions', 'Clock + Calendar', 'String + Char', 'Program control', 'Communications', 'Interrupts', 'PID', 'Motion Control', and 'Pulse'. The bottom status bar indicates 'Project startup opened'.



PASUL 26. În mod asemănător se vor conecta intrările 'Start', 'B_safety_fence' și 'B_cylinder' precum și ieșirea 'M_press' a blocului "Program press" la variabilele globale ale automatului PLC prezentate mai jos. Cu un click pe Save project proiectul va fi salvat. (→ "S3" → "B1" → "B2" → "M0" → Save project)

The screenshot shows the Siemens TIA Portal interface for a project named 'startup'. The main window displays a ladder logic network (Network 1) where a function block 'Program press' (FC1) is being configured. The block's inputs and outputs are connected to global variables as follows:

- Input: "Emerg-OFF" (address %I0.1)
- Input: "S3" (address %I0.3)
- Input: "B1" (address %I0.4)
- Input: "B2" (address %I0.5)
- Output: "M_press" (address %Q0.0)

The interface also shows a project tree on the left with 'Program press [FC1]' selected, and a right-hand pane with various instruction libraries. The status bar at the bottom indicates 'Project startup opened.'



UNIUNEA EUROPEANĂ



GUVERNUL ROMÂNIEI
MINISTERUL MUNCII, FAMILIEI,
PROTECȚIEI SOCIALE ȘI
PERSOANELOR VÂRSTNICE
AMPOSORU



Fondul Social European
POSDRU 2007-2013



Instrumente Structurale
2007-2013





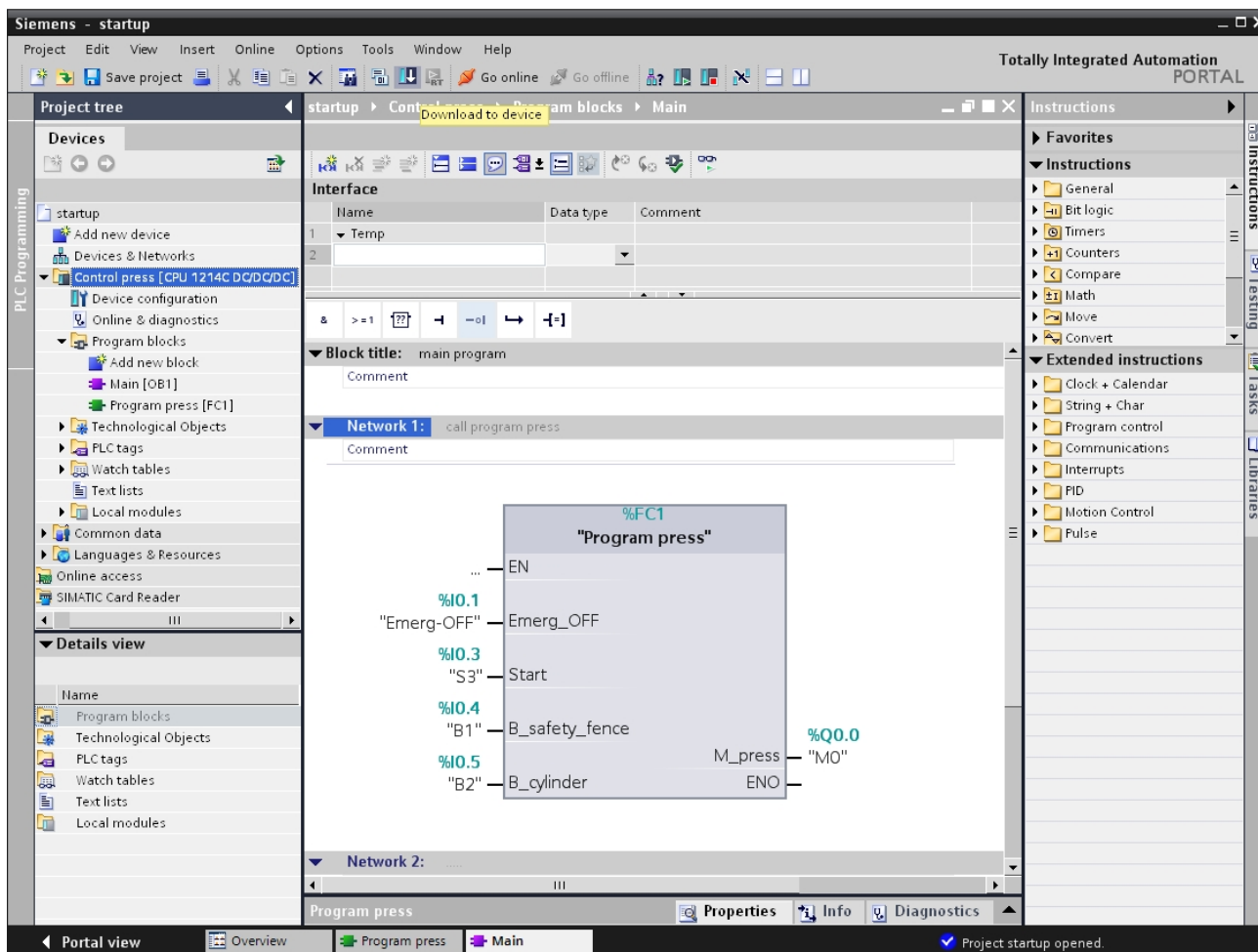
OIOSORU



UNIVERSITATEA
TEHNICA
DIN CLUJ-NAPOCA



PASUL 27. Pentru a încărca întregul program în unitatea centrală CPU, se selectează mai întâi subdirectorul **'Control Press'** apoi se apasă simbolul  "încarcă în dispozitiv" din bara de meniu de sus. (→ Control Press → )



The screenshot displays the Siemens TIA Portal software interface for configuring a PLC program. The main window shows the 'Control Press' program configuration for a SIMATIC 1214C CPU. The interface includes a 'Project tree' on the left, a central workspace for the ladder logic program, and a right-hand sidebar with various toolbars and libraries.

Project tree (Left):

- startup
 - Add new device
 - Devices & Networks
 - Control press [CPU 1214C DC/DC/DC]
 - Device configuration
 - Online & diagnostics
 - Program blocks
 - Add new block
 - Main [OB1]
 - Program press [FC1]
 - Technological Objects
 - PLC tags
 - Watch tables
 - Text lists
 - Local modules
 - Common data
 - Languages & Resources
 - Online access
 - SIMATIC Card Reader

Interface (Center):

Name	Data type	Comment
1	Temp	
2		

Block title: main program

Network 1: call program press

Network 2: ...

Program press (FC1) Ladder Logic:

```

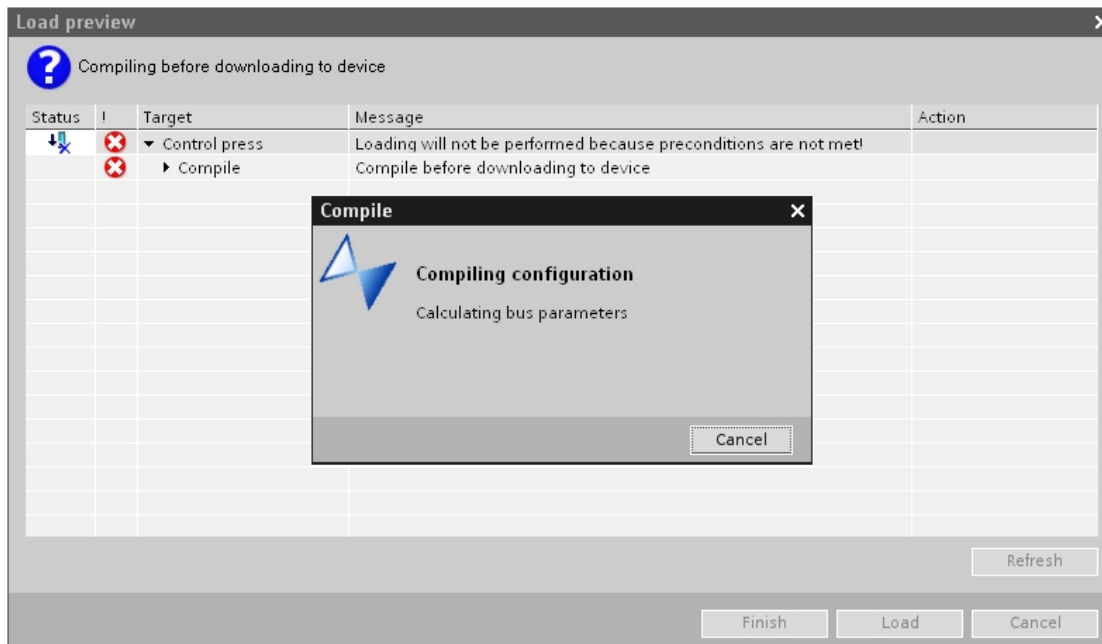
graph TD
    EN((...)) --> Emerg_OFF["%I0.1  
\"Emerg-OFF\""]
    Emerg_OFF --> Start["%I0.3  
\"S3\""]
    Start --> B_safety_fence["%I0.4  
\"B1\""]
    B_safety_fence --> B_cylinder["%I0.5  
\"B2\""]
    B_cylinder --> M_press["M_press"]
    M_press --> MO["%Q0.0  
\"MO\""]
    ENO((...))
  
```

Right-hand sidebar:

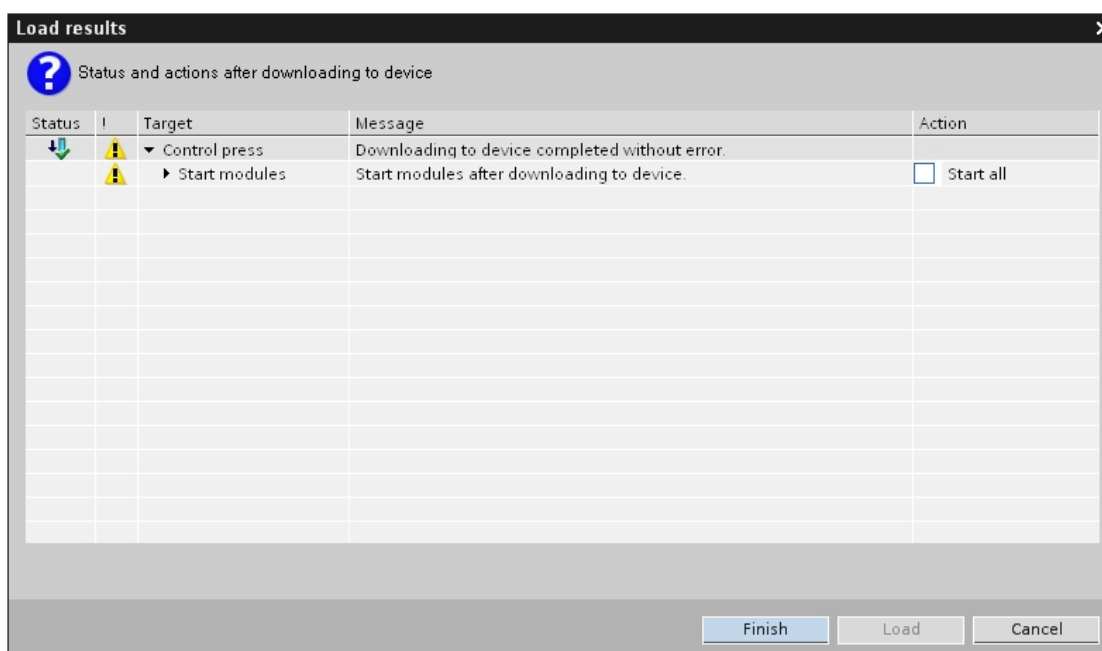
- Instructions
 - Favorites
 - Instructions
 - General
 - Bit logic
 - Timers
 - Counters
 - Compare
 - Math
 - Move
 - Convert
 - Extended instructions
 - Clock + Calendar
 - String + Char
 - Program control
 - Communications
 - Interrupts
 - PID
 - Motion Control
 - Pulse
- Testing
- Tasks
- Libraries





PASUL 28. Pe durata încărcării, starea procesului este afișată într-o fereastră.

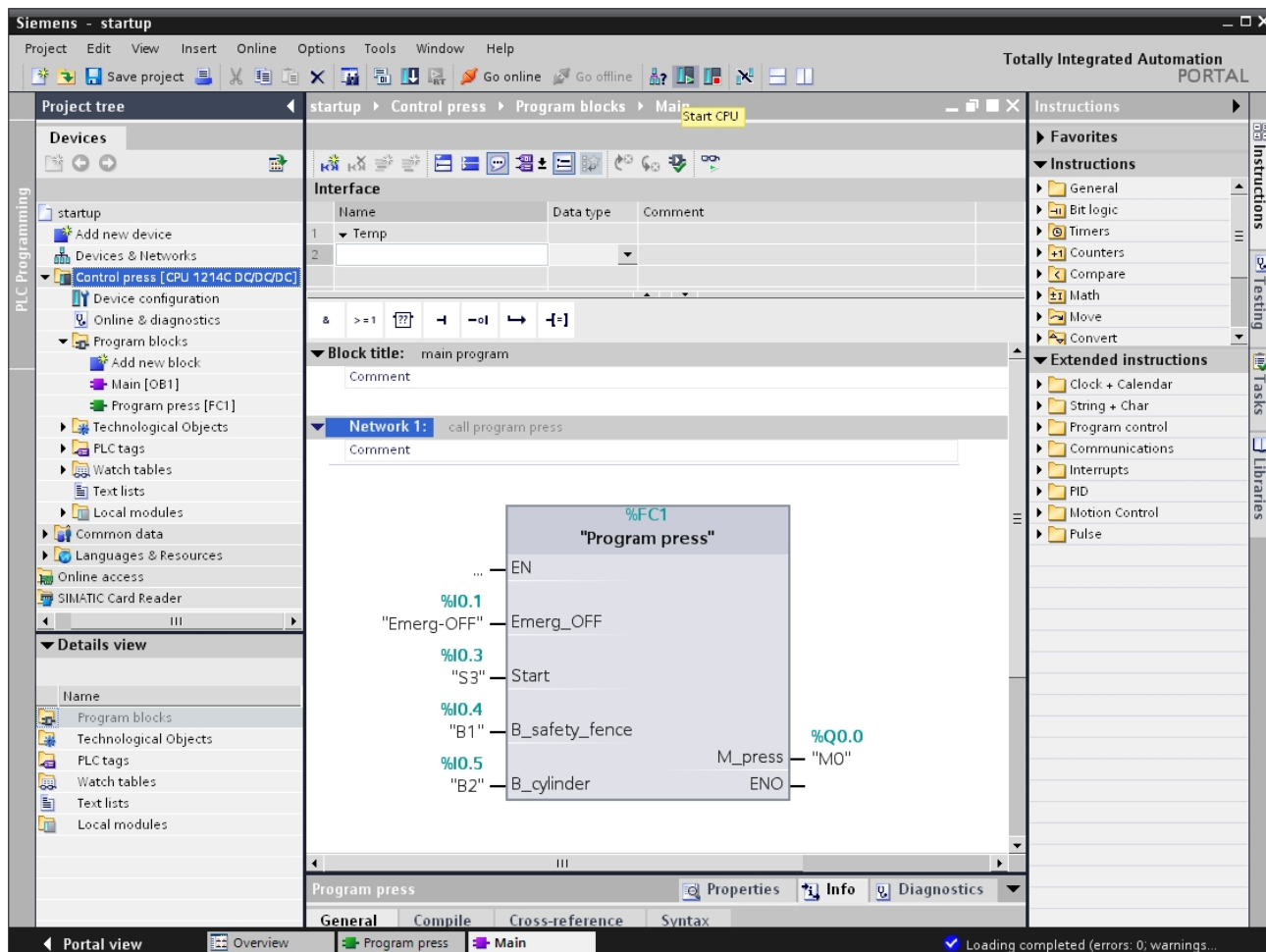


PASUL 29. Dacă încărcarea a fost reușită, acest lucru este arătat într-o fereastră. Luarea la cunoștință se face apăsând pe '**Complete**' (→ Complete)

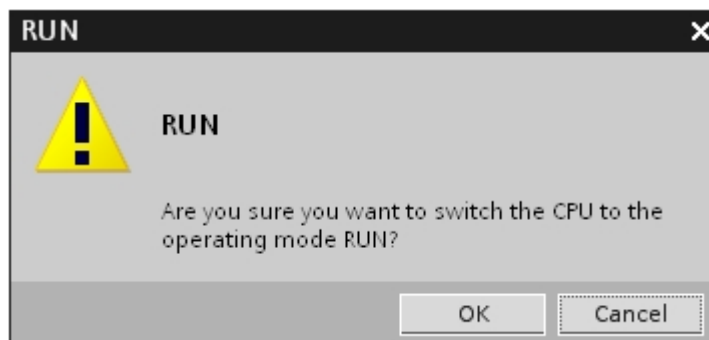




PASUL 30. Acum se lansează în execuție programul din unitatea centrală CPU prin apăsarea simbolului  din bara de meniu de sus. (→ )



PASUL 31. Cu 'OK', se confirmă dorința de lansare a programului din unitatea centrală CPU.
(→ OK)





PASUL 32. Cu un click pe simbolul  "Monitorizare on/off", se poate monitoriza starea variabilelor de intrare și ieșire pe durata testării programului din blocul "Program Press" ()

The screenshot displays the Siemens TIA Portal software interface. The main window shows the configuration of the "Program press" block (FC1) within the "Control press" project. The block is connected to several input and output variables:

- Inputs: EN, "Emerg-OFF" (I0.1), "S3" (I0.3), "B1" (I0.4), "B2" (I0.5), and M_press (Q0.0).
- Output: ENO.

The interface includes a Project tree on the left, a main workspace for the ladder logic diagram, and a right-hand panel with a "CPU operator panel" and a "Call hierarchy" section. The status bar at the bottom indicates "Loading completed (errors: 0; warnings: ...)".



UNIUNEA EUROPEANĂ



GUVERNUL ROMÂNIEI
MINISTERUL MUNCII, FAMILIEI,
PROTECȚIEI SOCIALE ȘI
PERSOANELOR VÂRSTNICHE
ȘI
AMPOSDRU



Fondul Social European
POSDRU 2007-2013



Instrumente Structurale
2007-2013



9. Bibliografie

[1]	Siemens Automation Cooperates with Education	M01-S7-1200 Programming Startup	TIA Training Document 05/2010
[2]	Siemens Automation Cooperates with Education	M02-S7-1200 Function Blocks	TIA Training Document 05/2010
[3]	Siemens Automation Cooperates with Education	M03-S7-1200 IEC-Timer and IEC- Counter	TIA Training Document 05/2010
[4]	Siemens Automation Cooperates with Education	M04-S7-1200 Programming Startup	TIA Training Document 05/2010
[5]	Siemens Automation Cooperates with Education	M05-S7-1200 Analog value Precessing	TIA Training Document 06/2010
[6]	Siemens Automation Cooperates with Education	M06-S7-1200 Closed Loop control	TIA Training Document 06/2010
[7]	Siemens Automation Cooperates with Education	M07-S7-1200 Networking	TIA Training Document 06/2010
[8]	Siemens Automation Cooperates with Education	M08-S7-1200 Human Machine Interface (WinCC Basc)	TIA Training Document 06/2010
[9]	SIEMENS	SIMATIC S7-1200 Programmable controller-System Manual	A5E02486680-06 04/2012
[10]	SIEMENS	SIMATIC TIA Portal STEP 7 Basic V10.5-Getting Started	A5E02651459-01 12/2009



UNIUNEA EUROPEANĂ



GUVERNUL ROMÂNIEI
MINISTERUL MUNCII, FAMILIEI,
PROTECȚIEI SOCIALE ȘI
PERSOANELOR VÂRSTNICE
AMPOSDRU



Fondul Social European
POSDRU 2007-2013



Instrumente Structurale
2007-2013



MINISTERUL
EDUCAȚIEI
NAȚIONALE



CNDIPT
OIPOSDRU



UNIVERSITATEA
TEHNICĂ
DIN CLUJ-NAPOCA

Material suport pentru stagii de practică în domeniul dezvoltării software pentru automatizări

Utilizarea blocurilor în programarea automatelor SIMATIC S7-1200 cu Portalul TIA V10 Volum II



UNIUNEA EUROPEANĂ



GUVERNUL ROMÂNIEI
MINISTERUL MUNCII, FAMILIEI,
PROTECȚIEI SOCIALE ȘI
PERSOANELOR VÂRSTNICE
AMPOSORU



Fondul Social European
POSORU 2007-2013



Instrumente Structurale
2007-2013



MINISTERUL
EDUCAȚIEI
NAȚIONALE



CNDIPT
OIPOSDRU



UNIVERSITATEA
TEHNICĂ
DIN CLUJ-NAPOCA

CUPRINS

	PAGINA
1. Introducere.....	3
2. Observații privind programarea automatelor SIMATIC S7-1200.....	5
2.1 Sistemul de automatizare SIMATIC S7-1200	5
2.2 Mediul de programare STEP 7 Basic V10.5 (Portalul TIA V10.5).....	5
3. Tipuri de blocuri utilizate de SIMATIC S7-1200.....	6
3.1 Programare liniară	6
3.2 Programare structurată.....	7
3.3 Blocuri utilizator pentru SIMATIC S7-1200.....	8
3.3.1 Blocuri de organizare	9
3.3.2 Funcții	10
3.3.3 Blocuri Funcționale	10
3.3.4 Blocuri de Date	11
4. Exemplu de aplicație: comandă bandă rulantă.....	12
5. Programarea comenzii de bandă rulantă cu SIMATIC S7-1200	13
6. Bibliografie.....	39

1. Introducere

Următoarele simboluri vă vor ghida prin acest manual:



Informație



Instalare



Programare



Exemplu



Atenționare



Obiectivul manualului de instruire:

În acest manual, se prezintă diferitele blocuri utilizate la programarea automatelor programabile (PLC) SIMATIC S7-1200, folosind mediul de programare TIA Portal. Manualul explică diferitele tipuri de blocuri, prezentând în capitolele enumerate mai jos cum se crează un program într-un bloc funcțional.

- Generarea blocului funcțional
- Definirea variabilelor interne
- Programarea cu variabile interne în blocul funcțional
- Apelarea și parametrizarea blocului funcțional din OB1

Cunoștințe prealabile:

Pentru a putea utiliza acest manual, sunt necesare următoarele cunoștințe prealabile:

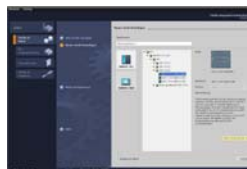
- cunoașterea sistemului de operare Windows
- cunoștințe de bază cu privire la programarea automatelor SIMATIC S7-1200 cu mediul de programare TIA Portal V10

Hardware și software necesare:

1. Calculator PC Pentium 4; 1.7 GHz; 1(XP) sau 2 (Vista) GB RAM, zonă liberă pe discul de stocare de cca. 2 GB; Sistemul de operare Windows XP (Home SP3, Professional SP3) sau Windows Vista (Home Premium SP1, Business SP1, Ultimate SP1)
2. Software STEP7 Basic V10.5 SP1 (Totally Integrated Automation (TIA) Portal V10.5)
3. Conexiune Ethernet între calculatorul PC și CPU 1214C
4. Automat programabil SIMATIC S7-1200, de exemplu CPU 1214C. Intrările trebuie să fie accesibile la un panou.



1 PC



2 STEP7 Basic
(Portal TIA)

3 Conexiune Ethernet



4 S7-1200 cu
CPU 1214C



UNIUNEA EUROPEANĂ



GUVERNUL ROMÂNIEI
MINISTERUL MUNCII, FAMILIEI,
PROTECȚIEI SOCIALE ȘI
PERSOANELOR VÂRSTNICE
AMPOSDRU



Fondul Social European
POSDRU 2007-2013



Instrumente Structurale
2007-2013



MINISTERUL
EDUCAȚIEI
NAȚIONALE



CNDIPT
OIPOSDRU



UNIVERSITATEA
TEHNICĂ
DIN CLUJ-NAPOCA

2. Observații privind programarea automatelor SIMATIC S7-1200

2.1 Sistemul de automatizare SIMATIC S7-1200



Sistemul de automatizare SIMATIC S7-1200 este un sistem mini-controler modular din gama de performanțe scăzute.

Familia S7-1200 cuprinde o gamă largă de module pentru adaptarea optimă la necesitățile de automatizare.

Controlerul S7 constă dintr-o unitate centrală de procesare CPU care este echipată cu intrări și ieșiri pentru semnale digitale și analogice.

În cazul în care intrările și ieșirile integrate în CPU nu sunt suficiente pentru aplicația dorită, pot fi instalate modulele de intrare și de ieșire (module IO) suplimentare.

Dacă este necesar, se adaugă procesoare de comunicare pentru RS232 sau RS485.

O interfață TCP / IP integrată este obligatorie pentru toate unitățile centrale de procesare CPU.

Automatul programabil (PLC) monitorizează și controlează un utilaj sau un proces cu ajutorul programului S7, care asigură interogarea modulelor IO prin intermediul adreselor de intrare (%I) și comanda procesului prin intermediul adreselor de ieșire (%Q).

Sistemul de automatizare SIMATIC S7-1200 este programat cu software-ul STEP 7 Basic V10.5.

2.2 Mediul de programare STEP 7 Basic V10.5 (Portalul TIA V10.5)



Software-ul STEP 7 Basic V10.5 este mediul de programare pentru sistemul de automatizare - SIMATIC S7-1200

STEP 7 Basic V10.5, permite utilizarea următoarelor funcții pentru a automatiza o instalație:

- Configurarea și parametrizarea hardware
- Definirea comunicației
- Programarea
- Testarea, punerea în funcțiune și service-ul cu funcțiile de operare / diagnosticare
- Elaborarea documentației
- Generarea ecranelor de afișare pentru panourile de operare de bază SIMATIC

Toate funcțiile sunt sprijinite de menuri help on-line detaliate.

3. Tipuri de blocuri utilizate de SIMATIC S7-1200



Pentru programarea automatelor programabile SIMATIC S7-1200, programul este scris în așa- numitele blocuri.

În mod obligatoriu, blocul de organizare OB1 există deja.

El reprezintă interfața cu sistemul de operare al CPU, fiind apelat de acesta în mod automat și procesat în mod ciclic.

Dacă sarcina de comandă este complexă, programul este divizat în blocuri de program mai mici, realizați pe criteriul funcțional, pentru o derulare ușoară a programului.

Aceste blocuri sunt apelate din blocurile de organizare. La sfârșitul blocului se revine înapoi în blocul de organizare care a efectuat apelul.

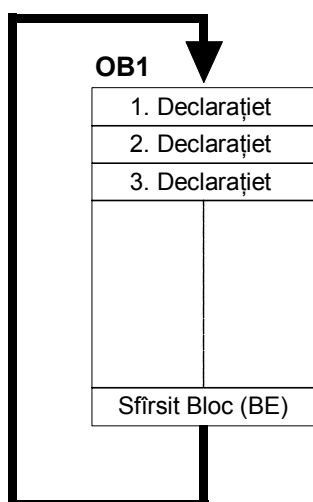
3.1 Programare liniară

Pentru programarea liniară, instrucțiunile sunt stocate într-un bloc și procesate în ordinea în care sunt stocate în memoria program. Când se ajunge la sfârșitul programului (sfârșitul blocului), procesarea programului reîncepe din nou de la început.

Aceasta se numește procesare ciclică.

Timpul necesar unui dispozitiv pentru o singură procesare a tuturor instrucțiunilor numește timp de ciclu.

Procesarea liniară a unui program se utilizează de obicei pentru aplicații de comandă simple, care nu sunt prea mari; acestea pot fi implementate într-un singur bloc de organizare OB.

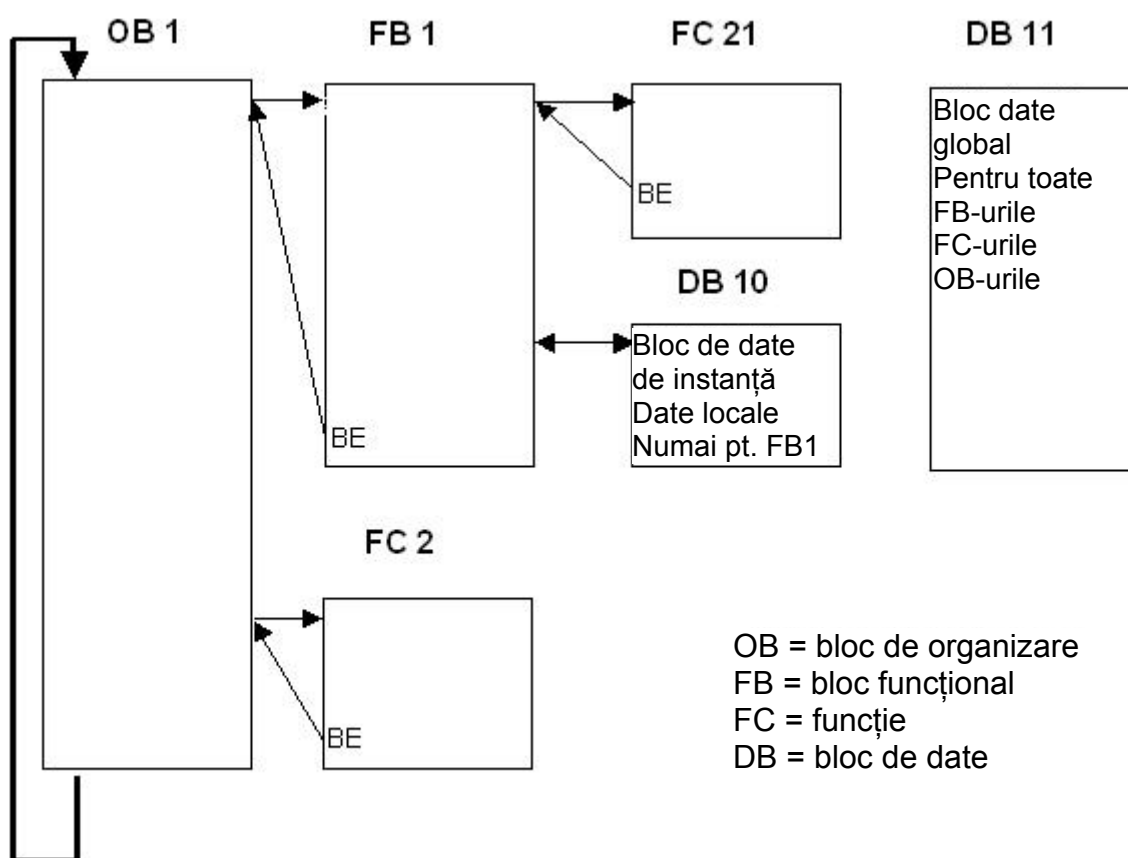


3.2 Programare structurată

i

Dacă sarcina de comandă este complexă, programul este divizat în blocuri de program mai mici, realizate pe criteriul funcțional, pentru o urmărire ușoară a programului. Avantaje: părțile de program pot fi testate individual și în momentul în care funcționează corect pot fi unite într-o funcție totalizatoare.

Blocul principal trebuie să apeleze blocurile program. Când se atinge indicatorul de sfârșit bloc (BE), programul continuă să fie procesat în blocul apelant, cu instrucțiunile ce urmează după instrucțiunea de apel.



3.3 Blocuri utilizator pentru SIMATIC S7-1200



Există următoarele blocuri utilizator pentru programare structurată:

- bloc de organizare OB (organization block):

OB-urile sunt apelate de sistemul de operare în mod ciclic și constituie interfața între programul utilizator și sistemul de operare. Prin OB unitatea de control a automatului programabil este informată prin intermediul instrucțiunilor de apel blocuri, care din blocurile de program trebuie procesate.

- bloc funcțional FB (function block):

Pentru fiecare instanță de apel a FB, trebuie să fie alocată o zonă de separată de memorie. Când FB este apelată, trebuie să fie alocată, de exemplu, un bloc de date (DB) cu rol de bază de dată de instanță.

Datele din această de bază de dată de instanță sunt accesate prin intermediul variabilelor bazei de date.

Dacă un FB este apelat de mai multe ori, trebuie alocate acestuia zone de memorie diferite.

La rândul lui un FB poate apela alte FB-uri și FC-uri.

- funcție FC (function):

Un FC nu are alocată zonă de memorie. Datele locale ale unei funcții sunt pierdute după procesarea acelei funcții.

La rândul lui o funcție FC poate apela alte FB-uri și FC-uri.

- bloc de date DB (data block):

DB-urile sunt utilizate pentru a furniza memorie pentru variabilele de date. Există două tipuri de blocuri de date :

- DB-uri globale, de unde toate OB-urile, FB-urile și FC-urile pot citi date stocate sau unde pot scrie date
- și DB-uri de instanță care sunt alocate unor anumite FB-uri.



Notă:

Dacă în timpul programării într-o FC sau într-un FB sunt utilizate doar variabile interne, aceste blocuri pot fi utilizate de mai multe ori sub formă de blocuri standard.

Ele pot fi apelate de oricâte ori e nevoie. Totuși la FB-uri trebuie asociată o zonă de memorie, o așa numită zonă de instanță (de exemplu un DB), pentru fiecare apel.



UNIUNEA EUROPEANĂ



GUVERNUL ROMÂNIEI
MINISTERUL MUNCII, FAMILIEI,
PROTECȚIEI SOCIALE ȘI
PERSOANELOR VÂRSTNICI
AMPOSDRU



Fondul Social European
POSDRU 2007-2013



Instrumente Structurale
2007-2013



MINISTERUL
EDUCAȚIEI
NAȚIONALE



CNDIPT
OIPOS DRU



UNIVERSITATEA
TEHNICĂ
DIN CLUJ-NAPOCA

3.3.1 Blocuri de organizare



Blocurile de organizare (OB) constituie interfața dintre sistemul de operare și programul utilizator. Ele sunt apelate de sistemul de operare și controlează următoarele procese:

- Comportamentul la pornire al sistemului de automatizare
- Procesarea ciclică a programului
- Procesarea controlată prin alarme a programului
- Tratarea erorilor

Blocurile de organizare pot fi programate în funcție de necesități, stabilind în acest fel comportamentul unității centrale de procesare CPU.

Există o mare varietate de opțiuni de utilizare a blocurilor de organizare în cadrul programelor de aplicație:

- Bloc de organizare pornire **Startup OB**, Bloc de organizare ciclare **Cycle OB**, Bloc de organizare eroare temporizare **Timing Error OB** și Bloc de organizare de diagnoză **Diagnosis OB**:

Acestea pot fi pur și simplu inserate în proiect și programate. Ele nu necesită alocarea unor parametri și nu trebuie apelate.

- Bloc de organizare procesare alarmă **Process Alarm OB** și Bloc de organizare întrerupere temporizată **Time Interrupt OB**:

Aceste blocuri de organizare trebuie parametrizate după ce au fost inserate în proiect. În plus, blocul de organizare procesare alarmă poate fi atașată la un eveniment prin program, utilizând instrucțiunea ATTACH, sau separată din nou de acest eveniment cu instrucțiunea DETACH.

- Bloc de organizare întrerupere pe perioadă de timp **Time Delay Interrupt OB**:

Blocul time delay interrupt OB se poate insera în proiect și programată. În plus, acesta trebuie apelat în programul utilizator cu instrucțiunea SRT_DINT. Parametrizarea nu este necesară.

Informații de pornire

La pornirea unor blocuri de organizare, sistemul de operare înregistrează unele informații care pot fi evaluate în programul utilizator.

Aceast lucru poate fi foarte util în diagnosticarea erorilor.

Dacă aceste informații sunt sau nu înregistrate și care dintre ele este înregistrată se precizează în descrierea blocurilor de organizare.

3.3.2 Funcții



O funcție conține un program care este executat în momentul în care un alt bloc de coduri apelează funcția respectivă.

Funcțiile (FC) sunt blocuri de coduri fără memorie. Datele din variabilele temporare sunt pierdute după procesarea funcției. Pentru a stoca datele funcțiilor, pot fi utilizate blocurile de date globale.

Funcțiile pot fi utilizate, de exemplu, pentru următoarele scopuri:

- Returnarea valorii funcției către blocul apelant; de exemplu în cazul funcțiilor matematice
- Executarea unor funcții tehnologice; De exemplu, comenzi individuale cu operații binare

O funcție poate fi apelată de mai multe ori din diferite locații din cadrul programului utilizator. Aceasta ușurează programarea funcțiilor repetitive în cadrul programului.

3.3.3 Blocuri Funcționale

Blocurile funcționale conțin subprograme care sunt executate de fiecare dată când un bloc funcțional este apelat de un alt bloc de coduri.

Blocurile funcționale sunt blocuri de coduri care își stochează valorile variabilelor în blocuri de date de instanță, astfel că aceste valori sunt disponibile și după încheierea procesării blocului funcțional.

Stocând intrările, ieșirile și parametri in/out în mod permanent în blocuri de date de instanță, valoarea acestora va fi disponibilă și după terminarea procesării blocului. De aceea aceste blocuri funcționale se numesc 'cu memorie'.

Blocurile funcționale sunt utilizate pentru realizarea unor sarcini ce nu pot fi implementate cu funcții:

- În toate cazurile când se utilizează temporizatoare și numărătoare în cadrul blocului
- În toate cazurile când informația trebuie stocată în program; De exemplu când selectăm un regim de operare cu un buton.

Ca și funcțiile, blocurile funcționale pot fi apelate de mai multe ori din diferite locații din cadrul programului utilizator. Aceasta facilitează programarea funcțiilor complicate repetitive.

Instanțe ale blocurilor funcționale

Un apel a unui bloc funcțional se numește o instanță.

Pentru fiecare instanță a unui bloc funcțional se alocă o zonă de memorie care conține datele utilizate de blocul funcțional pentru procesare. Zona de memorie este asigurată de blocuri de date pe care mediul de programare le generează în mod automat.

Este posibilă și asigurarea de zone de memorie comune pentru mai multe instanțe de către un singur bloc de date, ca memorie multi-istanță.



UNIUNEA EUROPEANĂ



GUVERNUL ROMÂNIEI
MINISTERUL MUNCII, FAMILIEI,
PROTECȚIEI SOCIALE ȘI
PERSOANELOR VÂRSTNICE
AMPOSDRU



Fondul Social European
POSDRU 2007-2013



Instrumente Structurale
2007-2013



MINISTERUL
EDUCAȚIEI
NAȚIONALE



CNDIPT
OIPOSDRU



UNIVERSITATEA
TEHNICĂ
DIN CLUJ-NAPOCA

3.3.4 Blocuri de Date



Spre deosebire de blocurile de coduri, blocurile de date nu conțin instrucțiuni, ci sunt utilizate pentru stocarea datelor utilizator. Adică, blocurile de date conțin date variabile utilizate de programul utilizator pentru procesare.

Blocurile de date globale sochează datele ce pot fi utilizate de toate celelalte blocuri.

Dimensiunea maximă a blocurilor de date variază în funcție de tipul unității centrale de prelucrare CPU. Structura blocurilor de date globale se poate stabili după nevoie.

Exemple de utilizare:

- Stocarea informației unui sistem de depozitare. "Care produs unde este depozitat"
- Stocarea rețetei pentru un anumit produs

Oricare bloc funcțional, fiecare funcție sau oricare bloc de organizare, poate citi date din blocul de date global sau poate înscrie date în blocul de date global. Aceste date sunt memorate în blocul de date și după ce blocul de date este închis.

Apelul unui bloc funcțional este numit instanță. Pentru fiecare apel al unui bloc funcțional, cu transfer de parametri, i se alocă un **bloc de date de instanță**, care servește ca depozit de date.

În acest bloc de date sunt stocați parametri efectivi și datele statice ale blocului funcțional.

Dimensiunea maximă a blocurilor de date de instanță variază în funcție de tipul unității centrale de prelucrare CPU. Structura blocurilor de date de instanță este determinată de variabilele declarate în blocul funcțional.

Un bloc de date globale și un bloc de date de instanță pot fi deschise simultan.

4. Exemplu de aplicație: comandă bandă rulantă



Dacă se dorește ca blocurile să lucreze ca o “cutie neagră”, la creerea lor trebuie avut în vedere ca ele să fie programate utilizând variabile. În aceste caz se va aplica următoarea regulă că în cadrul blocului nu se utilizează intrări / ieșiri, flaguri, etc.... cu adrese absolute. În cadrul blocului sunt utilizate doar variabile și constante.



În exemplul de mai jos, trebuie generat un bloc funcțional, utilizând declarații de variabile, care conține comanda unei bande rulante cu două regimuri de funcționare: 'Manual' și 'Automat'.

Regimul de operare 'Manual' se selectează cu ajutorul butonului 'S1' iar regimul de operare 'Automat' se selectează cu ajutorul butonului 'S2' .

În regimul de operare 'Manual', motorul este pornit cât timp butonul 'S3' este acționat fără ca butonul 'S4' să fie acționat.

În regimul de operare 'Automat', motorul benzii este pornit cu butonul 'S3' și oprit cu butonul 'S4' (contact normal închis).

Listă de atribuiri:

Adresă	Simbol	Observații
%I 0.0 deschis)	S1	Buton trecere în regim Manual S1 (contact normal
%I 0.1 deschis)	S2	Buton trecere în regim Automat S2 (contact normal
%I 0.2	S3	Buton pornire S3 (contact normal deschis)
%I 0.3	S4	Buton oprire S4 (contact normal închis)
%Q 0.2	M01	Motor bandă M01



Notă: Butonul de oprire S4 este cu contact normal închis pentru a asigura protecție la întreruperea legăturilor. În cazul întreruperii accidentale a legăturii la acest buton sistemul se oprește în mod automat. Folosind un contact normal deschis, sistemul nu s-ar mai putea opri în cazul întreruperii accidentale a legăturii la acest buton. Din această cauză, în sistemele de automatizare, toate butoanele de oprire de avarie și butoanele / comutatoarele de oprire trebuie să fie proiectate cu contact normal închis.



UNIUNEA EUROPEANĂ



GUVERNUL ROMÂNIEI
MINISTERUL MUNCII, FAMILIEI,
PROTECȚIEI SOCIALE ȘI
PERSONELOR VÂRSTNICE
AMPOSORU



Fondul Social European
POSDRU 2007-2013



Instrumente Structurale
2007-2013



CNDIPT
OIPOSDRU



UNIVERSITATEA
TEHNICA
DIN CLUJ-NAPOCA

5. Programarea comenzii de bandă rulantă cu SIMATIC S7-1200



Proiectul este administrat și componentele sunt programate cu ajutorul mediului de dezvoltare '**Totally Integrated Automation Portal**'.

Printr-o interfață unitară, portalul asigură configurarea, parametrizarea și programarea componentelor cum ar fi: controlerul, dispozitivele de vizualizare și elementele de rețea. Sunt asigurate și instrumente online de diagnosticare a eroilor.

În pașii care urmează, se crează un proiect pentru SIMATIC S7-1200 și se programează soluția de rezolvare al aplicației:

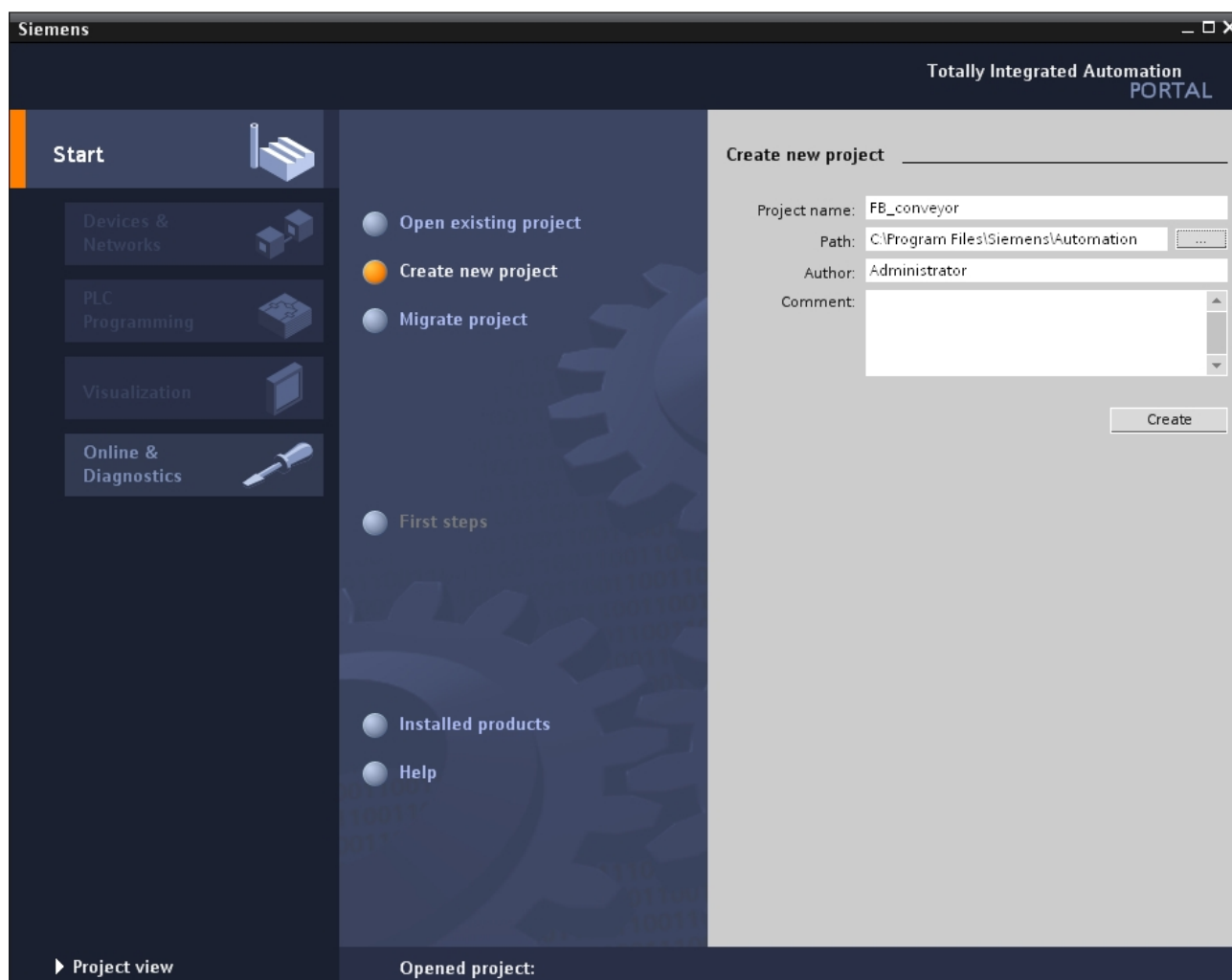


PASUL 1. Instrumentul principal este mediul de dezvoltare '**Totally Integrated Automation Portal**' care se deschide cu un dublu click (→ Totally Integrated Automation Portal V10)





PASUL 2. Programele pentru automatele programabile SIMATIC S7-1200 sunt gestionate sub formă de proiecte. Creerea unui astfel de proiect este prezentată mai jos în formatul de prezentare tip portal (→ Generate new project → FB_conveyor → Create)





PASUL 3. Aici sunt prezentați primii pași de configurare a proiectului realizați cu **'First Steps'**. Mai întâi se configurează dispozitivul cu **'Configure a device'** (→ First Steps → Configure a device)

The screenshot displays the Siemens TIA Portal interface for a project named 'FB_conveyor'. The main window is titled 'Siemens - FB_conveyor' and 'Totally Integrated Automation PORTAL'. On the left, there is a navigation pane with 'Start' selected. Below 'Start', there are four categories: 'Devices & Networks', 'PLC Programming', 'Visualization', and 'Online & Diagnostics'. In the center, there are three options: 'Open existing project', 'Create new project', and 'Migrate project'. Below these, 'First steps' is highlighted with an orange circle. At the bottom of the center pane, there are 'Installed products' and 'Help' options. The right side of the window shows a 'First steps' dialog box with the message: 'Project: "FB_conveyor" successfully opened. Please select the next step:'. Below the message, there are four options with arrows pointing to them: 'Start', 'Devices & Networks', 'PLC Programming', and 'Visualization'. Each option has a corresponding icon and a description: 'Configure a device', 'Create a PLC program', and 'Configure an HMI screen'. At the bottom of the dialog box, there is a 'Project view' option with the description 'Open the project view'. The status bar at the bottom of the window shows 'Opened project: C:\Program Files\Siemens\Automation\FB_conveyor\FB_conveyor'.



PASUL 4. Apoi, inserăm un dispozitiv nou cu **'Add new device'** scriind la **'Device name'** numele dispozitivului **Control conveyor (Comandă bandă rulantă)**. Pentru aceasta alegem din catalog unitatea central de comandă **'CPU1214C'** cu codul de comandă corespunzător cu CPU de care dispunem pentru experimentări (ex:6S7 214-1AE30-0XB0) (→ Add new device → Control conveyor → CPU1214C → 6ES7 → Add)



PASUL 5. Acum , mediul comută automat din prezentarea tip portal în prezentarea tip proiect, deschisă la configurarea hardware. In acest ecran se pot adăuga la automatul programabil module suplimentare preluate din catalog (din partea dreaptă) și în '**Device overview**' (stânga jos), se pot seta adresele intrărilor / ieșirilor . Tot în acest ecran se poate vedea că intrările integrate ale unității central CPU au adresele %I0.0 la %I1.5 iar ieșirile integrate au adresele %Q0.0 to %Q1.1 (→ Device overview → DI14/DO10 → 0...1)

The screenshot shows the Siemens TIA Portal interface for configuring a SIMATIC S7-1200 rack. The 'Device overview' table is as follows:

Module	Slot	I address	Q address	Type	Order no.	Firmware	Comme
Control conve...	1			CPU 1214C DC/DC/DC	6ES7 214-1AE30-0XB0	V1.0	
DI14/DO10	1.1	0...1	0...1	DI14/DO10			
AI2	1.2	64.. 67		AI2			
	1.3						
HSC_1	1.16			High speed counter..			
HSC_2	1.17			High speed counter..			
HSC_3	1.18			High speed counter..			
HSC_4	1.19			High speed counter..			
HSC_5	1.20			High speed counter..			



PASUL 6. Pentru ca mediul de dezvoltare să acceseze mai târziu unitatea centrală CPU corectă, adresa IP și masca de subrețea trebuie să fie setate.

(→ Properties → General → Ethernet addresses → IP address: 192.189.0.1 → Subnet screen form: 255.255.255.0)

The screenshot displays the Siemens TIA Portal interface for configuring a SIMATIC S7-1200 rack. The main window shows a rack diagram with a SIMATIC S7-1200 CPU unit installed in slot 1. The 'PROFINET interface' properties are open, showing the 'Ethernet addresses' tab. The IP address is set to 192.168.0.1 and the subnet mask is 255.255.255.0. The interface is currently 'Not connected'. The status bar at the bottom indicates 'Project FB_conveyor created'.



PASUL 7. Deoarece în programarea modernă, nu se mai programează cu adrese absolute ci cu variabile simbolice, **variabilele globale ale automatului programabil trebuie specificate acum..**

Aceste variabile globale ale automatului programabil sunt nume descriptive, împreună cu comentarii, pentru acele intrări și ieșiri, care sunt utilizate în cadrul programului. Mai târziu, în timpul programării, variabilele globale ale automatului programabil pot fi accesate prin intermediul acestor nume. Aceste variabile globale pot fi utilizate în întregul program, în toate blocurile.

În acest scop, selectați mai întâi în meniul de navigare '**Control Press[CPU1214C DC/DC/DC]**' și apoi '**PLC tags**'. Cu un dublu click, deschideți tabelul '**PLC tags**' și introduceți numele pentru intrări și ieșiri, așa cum se arată mai jos (→ Control convezoar[CPU1214C DC/DC/DC] → PLC tags → PLC tags)

The screenshot shows the Siemens TIA Portal interface for configuring PLC tags. The main window displays a table of PLC tags with the following data:

	Name	Data type	Address	Retain	Comment
1	S1_conv1	Bool	%I0.0	<input type="checkbox"/>	conveyor1 pushbutton manual mode S1 (no contact)
2	S2_conv1	Bool	%I0.1	<input type="checkbox"/>	conveyor1 pushbutton automatic mode S2 (no cont...
3	S3_conv1	Bool	%I0.2	<input type="checkbox"/>	conveyor1 pushbutton ON S3 (no contact)
4	S4_conv1	Bool	%I0.3	<input type="checkbox"/>	conveyor1 pushbutton OFF S4 (nc contact)
5	M01_conv1	Bool	%Q0.2	<input type="checkbox"/>	conveyor1 motor conveyor belt M01
6				<input type="checkbox"/>	

The 'S1_conv1' tag is selected, and its properties are shown in the 'Properties' window below:

- General**
 - Name: S1_conv1
 - Data type: Bool
 - Address: %I0.0
 - Retained:
 - Comment: conveyor1 pushbutton manual mode S1 (no contact)
- Time stamp**
 - Date created: 1/21/2010 11:33 PM
 - Last modified: 7/12/2010 1:54 PM



PASUL 8. Pentru a genera blocul funcțional FB1, în Meniul de Navigare se selectează mai întâi **'Control conveyor[CPU1214C DC/DC/DC]'** și apoi **'Program blocks'**. Apoi se face dublu click pe **'Add new block'** (→'Control conveyor [CPU1214C DC/DC/DC]' → Program blocks → Add new block)

The screenshot shows the Siemens TIA Portal software interface. The main window displays the 'PLC tags' configuration for the 'M01_conv1' tag. The interface includes a menu bar, a toolbar, and a project tree on the left. The 'PLC tags' table lists several tags with their names, data types, addresses, and comments.

	Name	Data type	Address	Retain	Comment
1	S1_conv1	Bool	%I0.0	<input type="checkbox"/>	conveyor1 pushbutton manual mode S1 (no contact)
2	S2_conv1	Bool	%I0.1	<input type="checkbox"/>	conveyor1 pushbutton automatic mode S2 (no contact)
3	S3_conv1	Bool	%I0.2	<input type="checkbox"/>	conveyor1 pushbutton ON S3 (no contact)
4	S4_conv1	Bool	%I0.3	<input type="checkbox"/>	conveyor1 pushbutton OFF S4 (no contact)
5	M01_conv1	Bool	%Q0.2	<input type="checkbox"/>	conveyor1 motor conveyor belt M01
6				<input type="checkbox"/>	

The 'M01_conv1' tag properties are shown in the 'General' tab, including the name, data type (Bool), address (%Q0.2), and comment (conveyor1 motor conveyor belt M01). The 'Time stamp' section shows the date created (1/21/2010 11:34 PM) and last modified (7/12/2010 1:53 PM).



PASUL 9. Din zona de selecție se selectează un bloc funcțional '**Function block (FB)**' căruia i se atribuie numele '**Conveyor**'. Ca limbaj de programare se alege limbajul '**FBD**'. Numerotarea este automată. Pentru că, oricum, această funcție FB1 este apelată mai târziu cu numele său simbolic, numărul blocului funcțional nu mai are importanță. Se confirmă selecțiile efectuate cu '**OK**'. (→ Function block (FB1) → Conveyor → FBD → OK)

Add new block [X]

Name:

Organization block (OB)

Function block (FB)

Function (FC)

Data block (DB)

Language:

Number:

Manual

Automatic

Symbolic access only

Description:

Function blocks are code blocks or subroutines that store their values permanently in instance data blocks, so that they remain available after the block has been executed.

[more...](#)

Further information

Add new and open



PASUL 10. Blocul '**Conveyor[FB1]**' va fi deschis în mod automat. Cu toate acestea, înainte de scrierea programului, interfața blocului trebuie declarată. Atunci când interfața este declarată, sunt specificate variabilele locale recunoscute numai în cadrul acestui bloc.

Variabilele locale constau din două grupuri:

- Grupul de parametri care realizează interfața blocului pentru apelul din program.

Tip	Name	Funcție	Disponibil în
Parametri de intrare	Input	Parametri ai căror valori sunt citite de bloc	Funcții, blocuri funcționale și unele tipuri de blocuri de organizare
Parametri de ieșire	Output	Parametri ai căror valori sunt scrise de bloc	Funcții și blocuri funcționale
Parametri InOut	InOut	Parametri ai căror valori sunt citite de bloc când este apelat iar după procesare sunt suprascrise	Funcții și blocuri funcționale

- Datele locale, care sunt utilizate pentru stocarea rezultatelor intermediare.

Tip	Name	Funcție	Disponibil în
Date locale temporare	Temp	Variabile care sunt utilizate la stocarea rezultatelor intermediare temporare. Datele temporare sunt reținute doar pentru un ciclu.	Funcții, blocuri funcționale și blocuri de organizare
Date locale statice	Static	Variabile care sunt utilizate la stocarea rezultatelor intermediare statice în blocul de date momentan. Datele statice sunt reținute pe parcursul a mai multe cicluri, până când se suprascriu.	Blocuri funcționale



UNIUNEA EUROPEANĂ



GUVERNUL ROMÂNIEI
MINISTERUL MUNCII, FAMILIEI,
PROTECȚIEI SOCIALE ȘI
PERSONELOR VÂRSTNICE
AMPOSORU



Fondul Social European
POSDRU 2007-2013



Instrumente Structurale
2007-2013



CNDIPT
OIPOSDRU



UNIVERSITATEA
TEHNICA
DIN CLUJ-NAPOCA



PASUL 11. In exemplul nostru este nevoie de declararea următoarelor variabilelor locale:

Intrare (Input):

man	Aici se introduce butonul de selectare al regimului
Manual (contact n.d.)	
auto	Aici se introduce butonul de selectare al regimului
Automat (contact n.d.)	
on	Aici se introduce butonul de pornire (contact normal deschis)
off	Aici se introduce butonul de oprire (contact normal închis)

Ieșire (Output):

motor	Aici se scrie starea ieșirii de comandă "Motor bandă"
-------	---

Static (există doar în blocurile funcționale FB):

automan	Aici este stocat regimul de funcționare preselectat
motorauto	Aici este un flag care arată că motorul a fost pornit în regim
Automat	

În cazul nostru toate variabilele sunt de tip boolean 'Bool'; aceasta înseamnă că aceste variabile pot avea doar două stări '0' (false) sau '1' (true).

În acest exemplu, e important de observat că starea variabilelor 'automan' și 'motorauto' trebuie stocată pe o perioadă mai lungă de timp. Din această cauză trebuie utilizate variabile de tip '**Static**'.

Dar acest tip de variabile există doar în blocurile funcționale FB.

Pentru mai multă claritate în înțelegerea programului, toate variabilele locale trebuie să fie însoțite de comentarii suficiente.



UNIUNEA EUROPEANĂ



GUVERNUL ROMÂNIEI
MINISTERUL MUNCII, FAMILIEI,
PROTECȚIEI SOCIALE ȘI
PERSONELOR VÂRSTICE
AMPOSORU



Fondul Social European
POSDRU 2007-2013



Instrumente Structurale
2007-2013



OIPOSORU



UNIVERSITATEA
TEHNICA
DIN CLUJ-NAPOCA

Siemens - FB_conveyor

Project Edit View Insert Online Options Tools Window Help

Totally Integrated Automation PORTAL

Project tree: FB_conveyor > Control conveyor > Program blocks > Conveyor

Devices

- FB_conveyor
 - Add new device
 - Devices & Networks
 - Control conveyor [CP...]
 - Device configurati...
 - Online & diagnost...
 - Program blocks
 - Add new block
 - Main [OB1]
 - Conveyor [FB1]
 - Technological Obj...
 - PLC tags
 - Watch tables
 - Text lists
 - Local modules
 - Common data
 - Languages & Resour...

Details view

Name

Name	Data type	Default value	Retain	Comment	
1	Input				
2	man	Bool	false	Non-Retain	pushbutton manual mode(no contact)
3	auto	Bool	false	Non-Retain	pushbutton automatic mode (no conta...
4	on	Bool	false	Non-Retain	pushbutton motor ON (no contact)
5	off	Bool	false	Non-Retain	pushbutton motor OFF (nc contact)
6	Output				
7	motor	Bool	false	Non-Retain	motor conveyor
8	InOut				
9	Static				
11	automan	Bool	false	Non-Retain	memory bit mode
12	motorauso	Bool	false	Non-Retain	memory bit motor started in auto mode
13	Temp				
14					

Block title: program conveyor belt

Comment

Network 1: memory mode manual/automatic selected

Comment

Properties Info Diagnostics

Portal view Overview Conveyor

Project FB_conveyor opened.



PASUL 12. După declararea variabilelor locale, se poate începe scrierea programului utilizând numele variabilelor (variabilele sunt identificate prin simbolul '#'). În limbajul Function Block Diagram (FBD), programul va arăta în felul următor:





PASUL 13. În continuare se configurează blocul procesat în mod ciclic 'Main[OB1]' prin meniul 'Properties'. Proprietățile blocului pot fi modificate (→ Properties → Main[OB1])

Siemens - FB_conveyor

Totally Integrated Automation PORTAL

Project tree: FB_conveyor > Control conveyor > Program blocks > Conveyor

Block title: program conveyor belt

Network 1: memory mode manual/automatic selected

Context menu options: Open, Cut (Ctrl+X), Copy (Ctrl+C), Paste (Ctrl+V), Delete (Del), Rename (F2), Go to device, Go to library, Compile, Download to device, Go online, Go offline, Compare offline/online, Show usage, Assignment list, Call structure, Resources, Cross-references, Print... (Ctrl+P), Print preview..., Properties... (Alt+Enter)

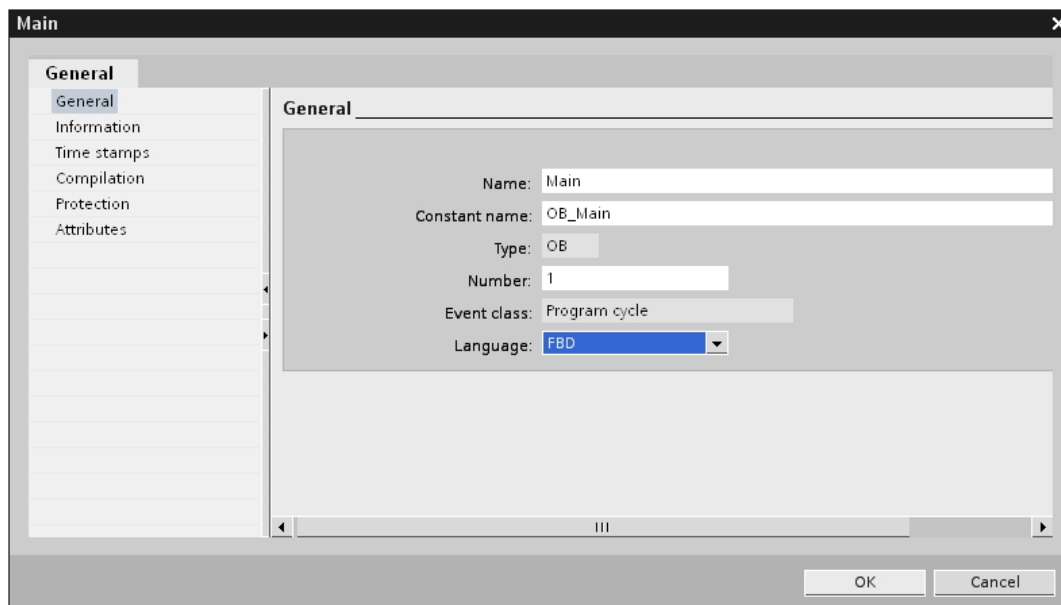
Diagram elements: #automan SR block, #auto S, #man R1, #motorauto SR block, #off, #automan, #on S, R1, #motorauto

Networks: 2: memory motor started in automatic mode, 3: control motor conveyor

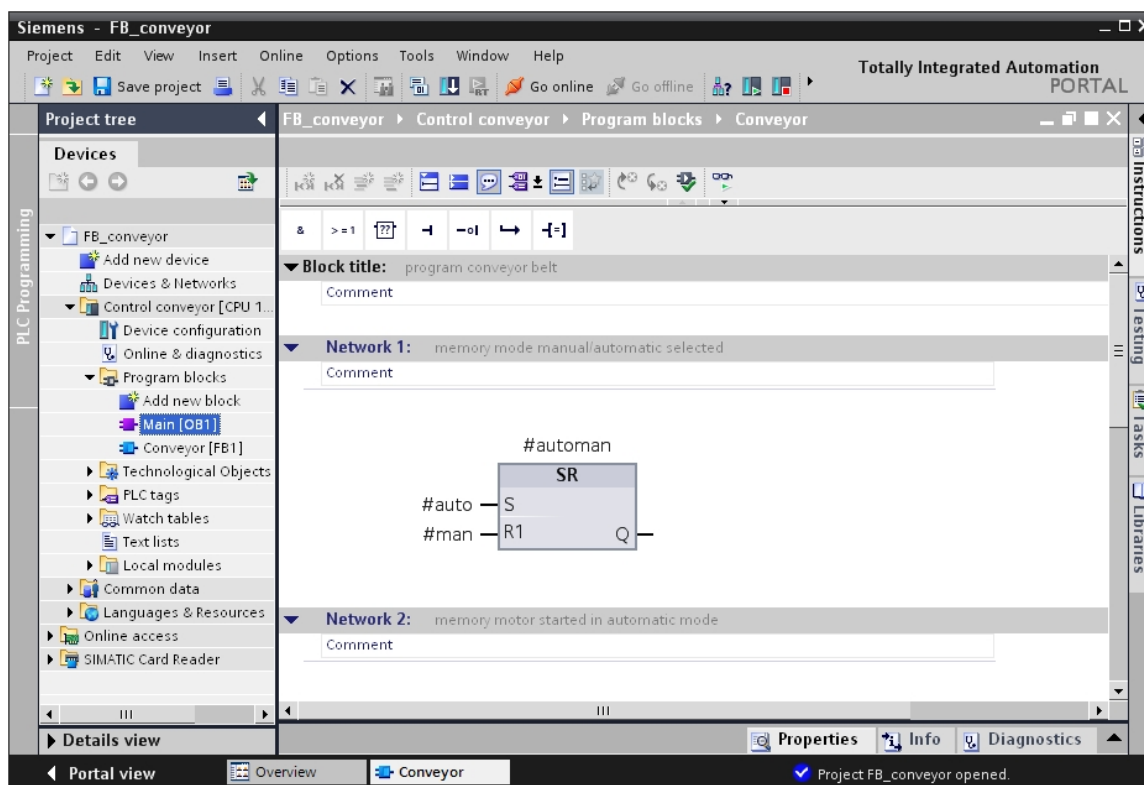
Bottom bar: Portal view, Overview, Conveyor, Project FB_conveyor opened.



PASUL 14. In cadrul proprietăților se selectează la '**Language**' limbajul de programare , acesta fiind diagram de blocuri funcționale'**FBD**'. (→ FBD → OK)



15. Acum blocul funcțional "Conveyor" trebuie apelat din blocul de organizare a programului principal Main[OB1]. Altfel acest bloc va fi ignorant . Se deschide blocul programului principal cu dublu click pe '**Main[OB1]**' (→ Main[OB1])





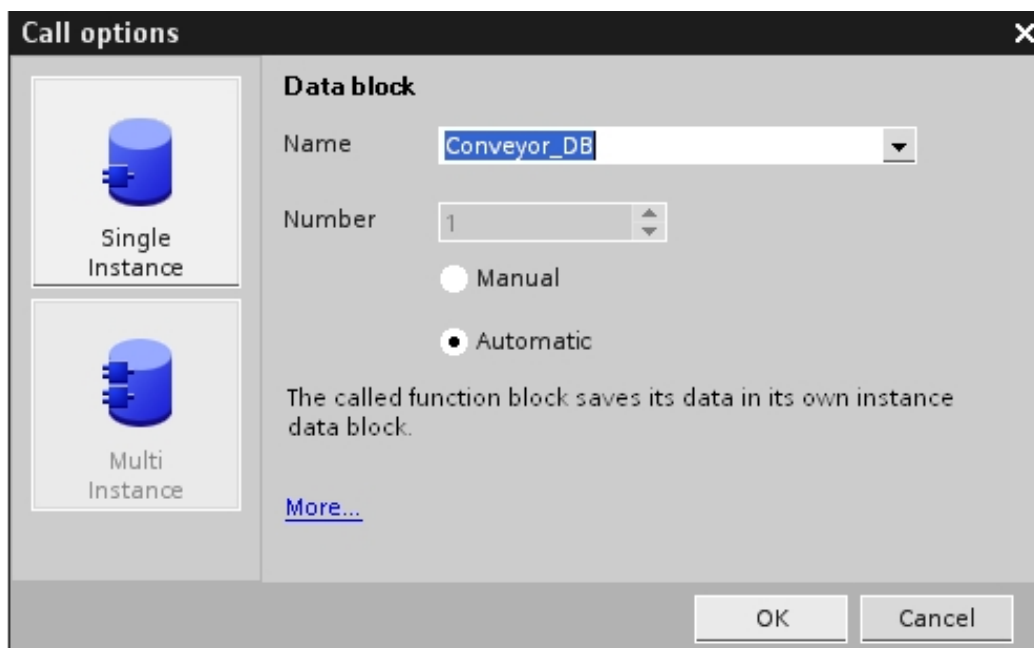
PASUL 16. Blocul funcțional **"Conveyor[FB1]"** poate fi tras, cu Drag&Drop, din meniul de navigare în Network 1 al blocului program principal Main[OB1]. Acest network va fi de asemenea documentat prin comentarii text în blocul Main[OB1] (→ Conveyor[FB1])

The screenshot displays the Siemens TIA Portal software interface for configuring a functional block (FB). The main window is titled "Siemens - FB_conveyor" and shows the "Control conveyor" project structure. The "Main" program block is selected, and the "Network 1" is visible, containing a call to the "Conveyor [FB1]" block. The "Interface" table shows two parameters: "Temp" and an empty row. The "Block title" is "main program". The "Network 1" comment is "call program conveyor belt1". The "Main" block properties are shown in the "General" tab, with the following details:

Property	Value
Name	Main
Constant name	OB_Main
Type	OB
Number	1
Event class	Program cycle



PASUL 17. Pentru că avem de-a face cu un bloc funcțional, trebuie să îl dotăm cu memorie. În cazul automatelor programabile SIMATIC S7-1200, blocurile de date asigură zonele de memorie. Un astfel de bloc de date alocat blocului funcțional se numește **Bloc de Date de Instanță** . În exemplul de mai jos acest Bloc de Date de Instanță este specificat și generat în mod automat (→ automatic→ OK)







PASUL 18. Conectarea variabilelor de intrare și a variabilelor de ieșire cu variabilele globale ale automatului PLC, în OB1, este prezentată mai jos. Cu un click pe Save project proiectul va fi salvat. (→ "S1" → "S2" → "S3" → "S4" → "M01" → Save project)

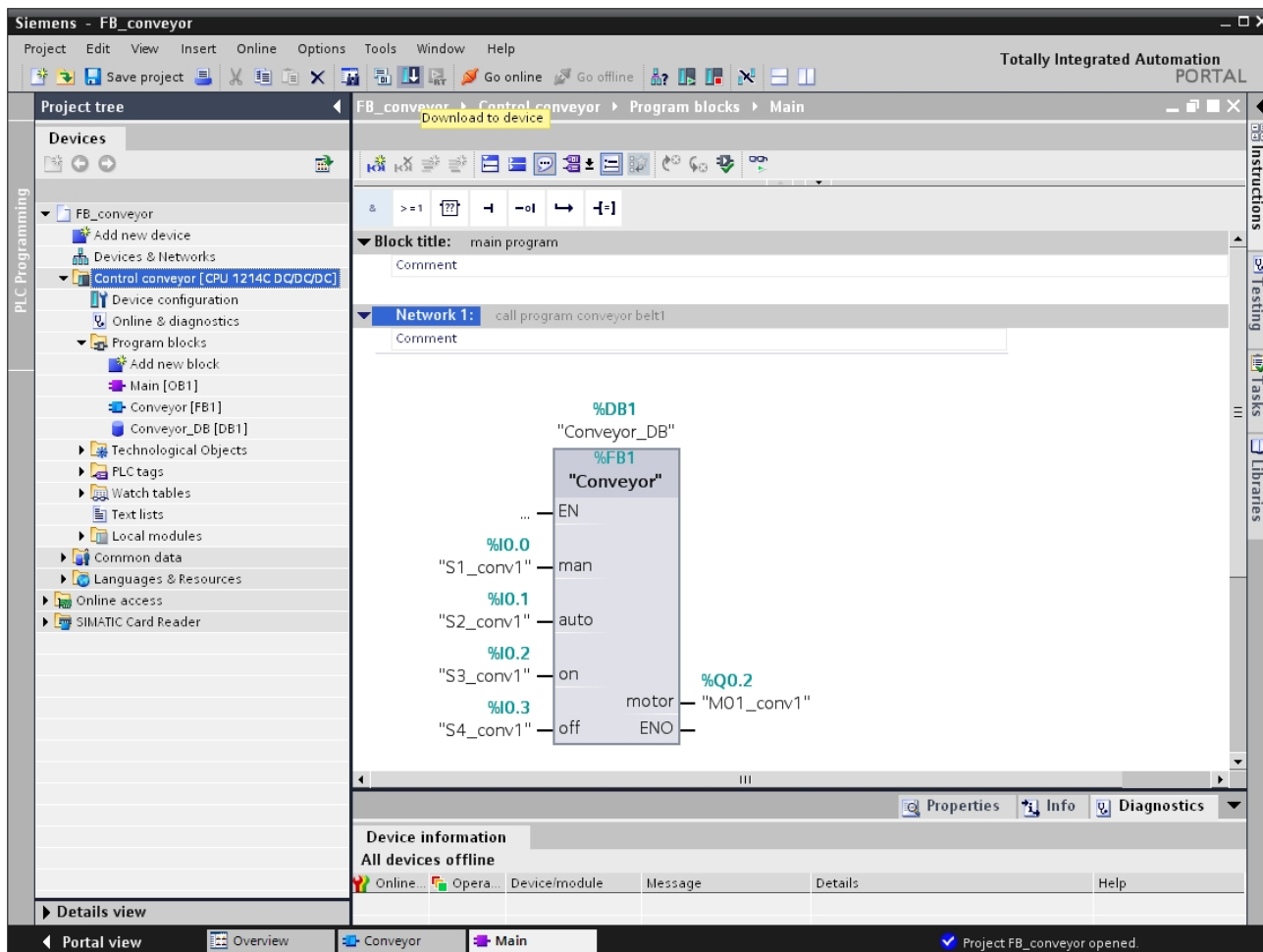
The screenshot displays the Siemens TIA Portal software interface for configuring a conveyor control block. The main workspace shows a ladder logic diagram with the following components:

- Block title:** main program
- Network 1:** call program conveyor belt
- Block:** "Conveyor" (FB1) with the following connections:
 - EN: ...
 - Inputs: "%I0.0" (man), "%I0.1" (auto), "%I0.2" (on), "%I0.3" (off)
 - Output: "M01_conv1" (motor)
 - ENO: ...
- Network 2:** (empty)

The bottom status bar shows "Project FB_conveyor opened." and navigation tabs for Overview, Conveyor, and Main.



PASUL19. Pentru a încărca întregul program în unitatea centrală CPU, se selectează mai întâi subdirectorul '**Control conveyor**' apoi se apasă simbolul  "încarcă în dispozitiv" din bara de meniu de sus. (→ Control conveyor → )



The screenshot displays the Siemens TIA Portal software interface for configuring a conveyor control program. The main window shows the 'Control conveyor' block with the following connections:

- DB1:** "Conveyor_DB"
- EN:** ...
- %I0.0:** "S1_conv1" (man)
- %I0.1:** "S2_conv1" (auto)
- %I0.2:** "S3_conv1" (on)
- %I0.3:** "S4_conv1" (off)
- %Q0.2:** "M01_conv1" (motor)
- ENO:** ...

The interface includes a 'Project tree' on the left, a 'Block title' field set to 'main program', and a 'Network 1' field set to 'call program conveyor belt1'. The bottom status bar indicates 'Project FB_conveyor opened.'



PASUL 20. Dacă s-a omis până acum configurarea interfeței calculatorului PG/PC cu automatul programabil, se va deschide o fereastră în care se poate face această configurare (→ PG/PC interface for loading → Load)

Extended download to device

Configured access nodes of "Control conveyor"

Device	Device type	Type	Address
Control conveyor	CPU 1214C DC/D...	TCP/IP	192.168.0.1

PG/PC interface for loading:

Connection to subnet:



1st gateway:

Accessible devices in target subnet: Show all accessible devices

Device	Device type	Type	Address	Target device
CPUcommon	CPU 1214C DC/D...	TCP/IP	192.168.0.1	CPUcommon
—	—	TCP/IP	Access address	—

Flash LED



PASUL 23. Acum se lansează în execuție programul din unitatea centrală CPU prin apăsarea simbolului  din bara de meniu de sus. (→ )

Siemens - FB_conveyor

Totally Integrated Automation PORTAL

Project tree: FB_conveyor > Control conveyor > Program > Start CPU > Main

Block title: main program

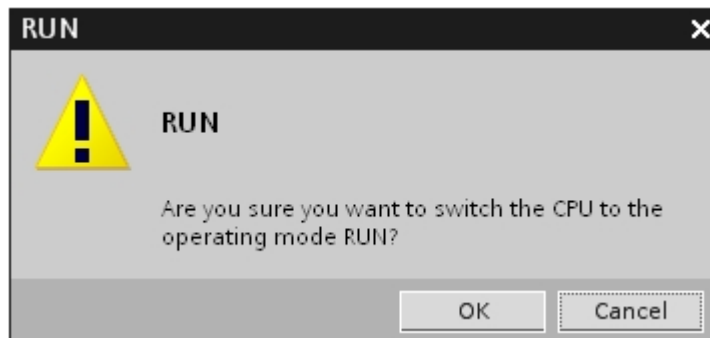
Network 1: call program conveyor belt

Inputs: %DB1 "Conveyor_DB", %FB1 "Conveyor", %I0.0 "S1_conv1" man, %I0.1 "S2_conv1" auto, %I0.2 "S3_conv1" on, %I0.3 "S4_conv1" off, %Q0.2 "M01_conv1" motor, ENO



Message: Loading completed (errors: 0; warnings: 0) 7/23/2010 11:05:26 PM



PASUL 24. Cu 'OK', se confirmă dorința de lansare a programului din unitatea centrală CPU.
(→ OK)





PASUL 25. Cu un click pe simbolul  "Monitorizare on/off", se poate monitoriza starea variabilelor de intrare și ieșire pe durata testării programului din blocul "Conveyor" (→ Conveyor[FB1] → )

The screenshot displays the Siemens TIA Portal software interface for a project named "FB_conveyor". The main workspace shows three ladder logic networks:

- Network 1:** A set-reset (SR) block for variable #automan. The set coil (S) is connected to #auto, and the reset coil (R) is connected to #man.
- Network 2:** A normally open contact for #off and a normally closed contact for #automan are connected to the set coil (S) of an SR block for #motorauto.
- Network 3:** A normally open contact for #motorauto, a normally open contact for #on, and a normally closed contact for #off are connected to the set coil (S) of an SR block for #motor.



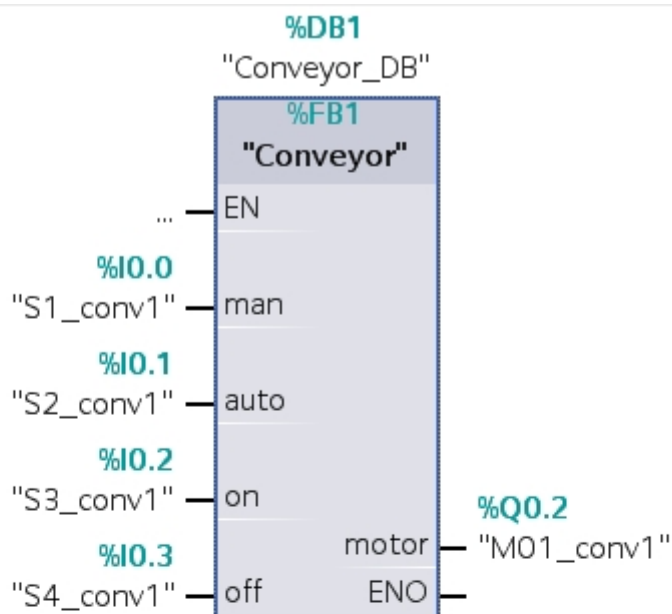
PASUL 26. Dacă blocul funcțional “Conveyor” a fost realizat respectând regulile blocurilor standard (fără utilizarea variabilelor globale în interiorul blocului !!!!!), el poate fi utilizat și apelat de oricâte ori e necesar.

Mai jos este prezentat un tabel cu variabilele automatului programabil, conținând intrările și ieșirile aferente la doua benzi rulante.

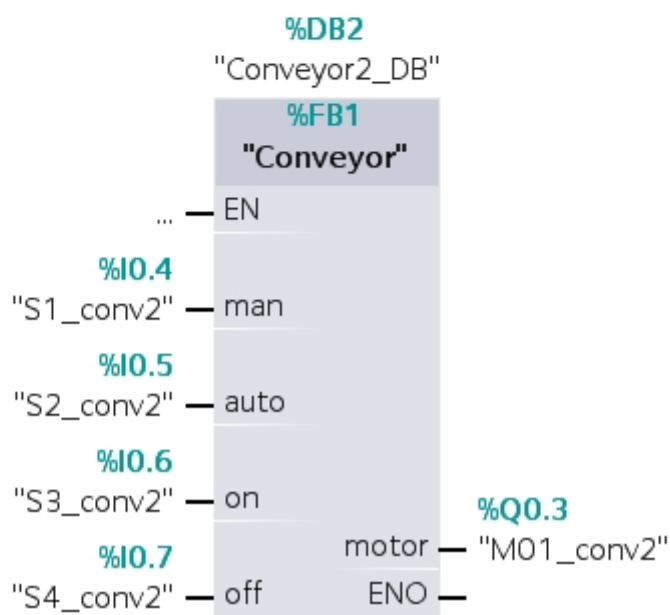
PLC tags					
	Name	Data type	Address	Retain	Comment
1	S1_conv1	Bool	%I0.0	<input type="checkbox"/>	conveyor1 pushbutton manual mode S1 (no contact)
2	S2_conv1	Bool	%I0.1	<input type="checkbox"/>	conveyor1 pushbutton automatic mode S2 (no contact)
3	S3_conv1	Bool	%I0.2	<input type="checkbox"/>	conveyor1 pushbutton ON S3 (no contact)
4	S4_conv1	Bool	%I0.3	<input type="checkbox"/>	conveyor1 pushbutton OFF S4 (nc contact)
5	M01_conv1	Bool	%Q0.2	<input type="checkbox"/>	conveyor1 motor conveyor belt M01
6	S1_conv2	Bool	%I0.4	<input type="checkbox"/>	conveyor2 pushbutton manual mode S1 (no contact)
7	S2_conv2	Bool	%I0.5	<input type="checkbox"/>	conveyor2 pushbutton automatic mode S2 (no contact)
8	S3_conv2	Bool	%I0.6	<input type="checkbox"/>	conveyor2 pushbutton ON S3 (no contact)
9	S4_conv2	Bool	%I0.7	<input type="checkbox"/>	conveyor2 pushbutton OFF S4 (nc contact)
10	M01_conv2	Bool	%Q0.3	<input type="checkbox"/>	conveyor2 motor conveyor belt M01
11				<input type="checkbox"/>	



PASUL 27. Blocul funcțional **"Conveyor"** poate fi apelat de două ori în OB1, având cablate semnale diferite la terminale. Pentru fiecare apel, se specifică un bloc de date de instanță diferit.



▼ **Network 2:** call program conveyor belt2
Comment





UNIUNEA EUROPEANĂ



GUVERNUL ROMÂNIEI
MINISTERUL MUNCII, FAMILIEI,
PROTECȚIEI SOCIALE ȘI
PERSONELOR VÂRSTNICHE
AMPOSORU



Fondul Social European
POSDRU 2007-2013



Instrumente Structurale
2007-2013



CNDIPT
OIPOSORU



UNIVERSITATEA
TEHNICA
DIN CLUJ-NAPOCA

6. Bibliografie

[1]	Siemens Automation Cooperates with Education	M01-S7-1200 Programming Startup	TIA Training Document 05/2010
[2]	Siemens Automation Cooperates with Education	M02-S7-1200 Function Blocks	TIA Training Document 05/2010
[3]	Siemens Automation Cooperates with Education	M03-S7-1200 IEC-Timer and IEC- Counter	TIA Training Document 05/2010
[4]	Siemens Automation Cooperates with Education	M04-S7-1200 Programming Startup	TIA Training Document 05/2010
[5]	Siemens Automation Cooperates with Education	M05-S7-1200 Analog value Precessing	TIA Training Document 06/2010
[6]	Siemens Automation Cooperates with Education	M06-S7-1200 Closed Loop control	TIA Training Document 06/2010
[7]	Siemens Automation Cooperates with Education	M07-S7-1200 Networking	TIA Training Document 06/2010
[8]	Siemens Automation Cooperates with Education	M08-S7-1200 Human Machine Interface (WinCC Basc)	TIA Training Document 06/2010
[9]	SIEMENS	SIMATIC S7-1200 Programmable controller-System Manual	A5E02486680-06 04/2012
[10]	SIEMENS	SIMATIC TIA Portal STEP 7 Basic V10.5-Getting Started	A5E02651459-01 12/2009