



**UNIVERSITATEA TEHNICĂ**  
DIN CLUJ-NAPOCA

**CLAUDIU ALEXANDRU OPREA**  
**HORIA HEDEȘIU**



# **APLICAȚII ALE AUTOMATIZĂRILOR DISCRETE ÎN SISTEME DEZVOLTATE CU PROGRAMARE GRAFICĂ**

**UTPRESS**  
**CLUJ-NAPOCA, 2023**  
**ISBN 978-606-737-647-0**

**Claudiu Alexandru OPREA**

**Horia HEDEȘIU**

**APLICAȚII ALE AUTOMATIZĂRILOR DISCRETE ÎN MEDII  
DE PROGRAMARE GRAFICĂ**

**- volumul 1 -**



**UTPRESS**

**Cluj - Napoca, 2023**

**ISBN 978-606-737-647-0**



Editura UTPRESS  
Str. Observatorului nr. 34  
400775 Cluj-Napoca  
Tel.:0264-401.999  
e-mail: [utpress@biblio.utcluj.ro](mailto:utpress@biblio.utcluj.ro)  
<http://biblioteca.utcluj.ro/editura>

Director:        Ing. Dan Colțea

Recenzia:        Conf.dr.ing. Silviu Ștefănescu  
                      Conf.dr.ing. Gabriel Chindriș

Pregătire format electronic on-line: Gabriela Groza

Copyright © 2023 Editura UTPRESS

Reproducerea integrală sau parțială a textului sau ilustrațiilor din această carte este posibilă numai cu acordul prealabil scris al editurii UTPRESS.

ISBN 978-606-737-646-3

**ISBN 978-606-737-647-0 vol. 1**

Bun de tipar: 10.06.2023

## Cuprins

1. Introducere.....	4
1.1 Context global.....	4
1.2 Industry 4.0 .....	8
1.3 Industrial Internet of Things .....	10
2. Automatul programabil Siemens Logo! .....	16
2.1 Prezentare generală.....	16
2.2 Programarea Logo!: Scheme logice, diagrame bloc.....	21
2.3 Funcții de bază.....	24
2.3.1: Combinarea unei funcții „ȘI” cu o funcție „SAU” .....	32
2.3.2: Realizarea unui circuit cu automenținere.....	41
2.4 Funcții speciale.....	43
2.5 Logo! Soft Comfort .....	70
2.5.1 Transferul de date între Logo! și PC.....	70
2.5.2 Automatizarea unui sistem de irigație al unei sere .....	76
2.5.3 Automatizarea unui sistem de distrugere a microbilor cu gaz .....	88
3. Automatul programabil Siemens S7-1200.....	90
3.1 Introducere .....	90
3.2 Prezentare generală .....	91
3.3. Siemens TIA Portal.....	104
3.4 Instrucțiuni uzuale utilizate pentru programarea tip LAD .....	115
3.5 Exemple de programe .....	138
Exemplul 1 – nivel lichid în bazin.....	138
Exemplul 2 – Detectarea tranziției pozitive a unui semnal .....	146
Exemplul 3 –Realizarea unui sistem automatizat de găurire .....	150
4. Concluzii și perspective.....	160
5. Bibliografie.....	164



# 1. Introducere

## 1.1 Context global

Ritmul schimbărilor majore la nivel mondial a cunoscut o accelerare fără precedent în ultimele decenii, schimbările fiind resimțite în toate domeniile vieții cotidiene, putând fi pe termen scurt, mediu sau lung. Dintre acestea, influența cea mai puternică o au acele schimbări structurale, pe termen lung, considerate ireversibile, cunoscute sub denumirea generică de mega-tendențe (megatrends).

Cunoașterea trendurilor mondiale reprezintă un avantaj major pentru toate companiile, indiferent de domeniul de activitate în care acestea activează, fie că este vorba de inginerie, psihologie, construcții, transporturi sau altele. În același timp această informație este foarte importantă și pentru persoanele implicate în aceste domenii, pentru că poate influența în mod pozitiv luarea de decizii în plan profesional sau personal. Din acest motiv există numeroase companii de analiză a pieței sau firme de consultanță care analizează în mod continuu noutățile în domeniul cercetării, tendințele industriale, modificările demografice, etc., în încercarea de a identifica megatrend-urile curente și a oferi consultanță companiilor sau persoanelor interesate să dezvolte produse noi sau să investească în domenii profitabile. Principalele mega-tendențe identificate în acest moment sunt prezentate în cele ce urmează.

- **Urbanizarea**

Viitorul omenirii pare să fie urban, mai mult din jumătatea populație lumii trăiește azi în orașe, Organizația Națiunilor Unite, prin ESA (Department of Economic and Social Affairs), estimând că populația rezidentă în orașe crește săptămânal cu 1.5 milioane [1]. Aproximativ 90% din această creștere se petrece în țări din Asia și Africa. Această urbanizarea accelerată pune presiune pe dezvoltarea infrastructurii, serviciilor, crearea de noi locuri de muncă, climă și mediu. Acest trend aduce cu sine și oportunități, orașele în dezvoltare având un potențial imens de a deveni modele de dezvoltare pentru restul comunităților.

Orașele reprezintă motorul economiei globale, Banca Mondială estimând că 80% din PIB-ul mondial fiind generat în orașe [2]. În cadrul orașelor se dezvoltă companii, idei, cunoștințe și reprezintă un magnet pentru talente din toate domeniile, reușind astfel să devină mai productive decât orice altă formă de organizare.

Urbanizarea accentuată are loc preponderent în emisfera sudică, existând o legătură directă între creşterea economică şi urbanizare: în ultimii 30 de ani, odată cu dezvoltarea clasei de mijloc în China, peste jumătate de miliard de persoane au trecut de pragul sărăciei, o dovadă directă a legăturii între bunăstarea populaţiei şi urbanizare. Ţările în dezvoltare pot beneficia de avantajele aduse de urbanizare: noi locuri de muncă, creşterea productivităţii, scăderea costurilor cu infrastructura şi a impactului asupra mediului, dezvoltarea de noi companii, etc. Drept urmare, dezvoltarea în zona sub-sahariană şi pe continentul asiatic a următoarelor mega-oraşe, "Next 10", prezintă o oportunitate majoră pentru mediul de afaceri, dar există riscul ca infrastructura existentă, deficitară, să încetinească sau chiar să oprească această dezvoltare.

Dubai este un exemplu de oraş care a identificat oportunitatea dată de locaţia sa geografică şi care, puternic finanţat din banii generaţi din exploatarea rezervelor imense de ţiţei, a de venit un model de dezvoltare şi un pionier în multe domenii: cea mai mare reţea de metrou fără operatori umani, Burj Khalifa, cel mai înalt zgârie-nor la momentul construcţiei, centrul spaţial Rashid, aeroportul internaţional care o punte între est şi Vest în contextul mobilităţii moderne, etc.

#### • Schimbări climatice şi lipsa resurselor

După o perioadă lungă de controverse pe tema schimbărilor climatice, ultimul deceniu pare să aducă consensul la nivel mondial referitor la efectul pe care industrializarea accentuată îl are asupra evoluţiei climei. Printre primii paşi concreţi au fost Acordul de la Paris (Paris Agreement) semnat în 2015, care implica asumarea unor măsuri şi politici care să limiteze creşterea temperaturii medii globale la 2°C faţă de valorile pre-industriale sau, mai curând, Pactul Climatic adoptat în cadrul întâlnirii din 2021 din Glasgow (Glasgow Climate Change), care încearcă limitarea creşterii temperaturii la 1,5°C. Aceste modificări semnificative şi probabil ireversibile ale climei, ce implică creşterea frecvenţei de apariţie a fenomenelor meteorologice extreme sau modificări ale anotimpurilor şi regimurilor de precipitaţii, duc la nevoia de adaptare a modului în care sunt gestionate afacerile, a modului în care se realizează agricultura sau pur şi simplu a stilului de viaţă a oamenilor.

Modificările climatice, corelate cu cantitatea finită de resurse disponibile pe Pământ, trebuie privite în contextul creşterii continue a nevoii de energie, hrană şi

apă, generate de creșterea accelerată a populației globului (atingerea pragului de 8 miliarde locuitori în 15 Noiembrie 2022), urbanizare și creșterea nivelului de trai.

- **Schimbări în distribuția puterii economice globale**

Economiile statelor emergente sunt pe punctul de a trece în recesiune (sau au făcut-o deja), ceea ce va duce la o modificare a modelului de business actual. Criza cauzată de pandemia "Covid" și războiul din Ucraina pun presiune suplimentară pe lanțurile de aprovizionare, prin creșterea costurilor de transport, creșterea duratei de livrare sau chiar întreruperea rutelor de aprovizionare. Modelele de previziune valabile în perioada pre-pandemică nu mai sunt valabile.

- **Modificări demografice și sociale**

În ultimul deceniu se înregistrează o creștere accelerată a populației la nivel mondial, în special datorită creșterii speranței de viață în țările slab dezvoltate. În plus, scăderea numărului de copii/familie, în special în țările dezvoltate și în economiile emergente duce la modificarea structurii populației (îmbătrânirea), fiind necesară angrenarea în rândul populației active a unor categorii sociale mai puțin active în trecut: femei și persoane în vârstă. O populație îmbătrânită, inactivă pe piața muncii, nu mai poate fi susținută prin sistemul social actual, fiind nevoie de noi politici sociale.

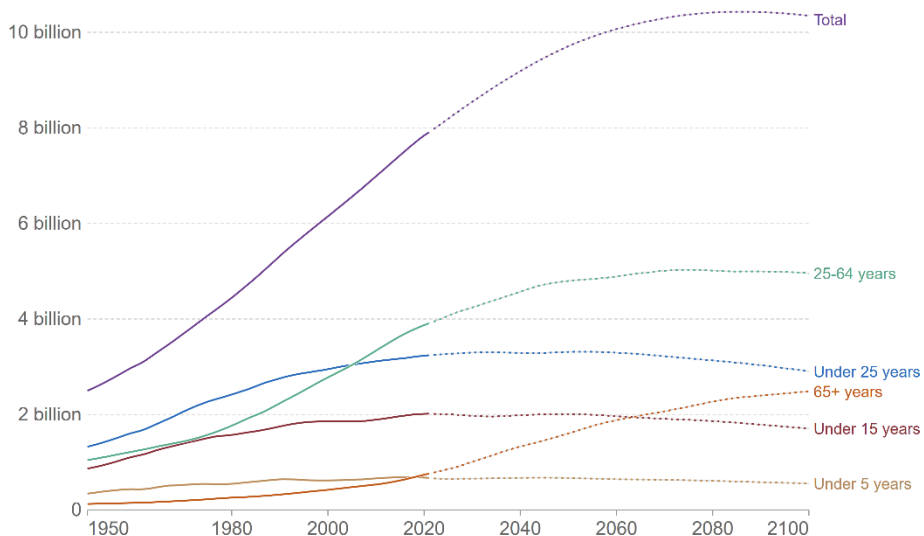
Studiile realizate arată schimbările fundamentale în distribuția grupelor de vârstă în perioada 1950 – 2021, pentru prima dată observându-se că numărul persoanelor de peste 65 de ani, considerate inactivă în majoritatea zonelor, a depășit numărul copiilor cu vârsta de sub 5 ani. În același timp, vârsta medie a populației a crescut de la 20 ani în 1970 la peste 30 de ani în 2022 [3]. Același studio prezintă două țări aflate în capete opuse ale spectrului repartiției vârstei și al dezvoltării economice, Japonia și Nigeria: în 2021 aproximativ 43% din populația Nigeriei avea sub 15 ani, în timp ce în Japonia procentul era de doar 12%, în timp ce mai mult de un sfert din populația Japoniei avea mai mult de 65 de ani, procentul fiind de sub 3% în Nigeria.

În același timp, previziunile făcute de Organizația Națiunilor Unite pentru acest secol arată o păstrare a trendului actual de îmbătrânire al populației în paralel cu depășirea unui total de 10 miliarde persoane în jurul anului 2060, după cum se poate observa în Figura 1.1.



### Population by age group, including UN projections, World

Historic estimates from 1950 to 2021, and projected to 2100 based on the UN medium-fertility scenario. This is shown for various age brackets and the total population.



Source: United Nations, World Population Prospects (2022)

OurWorldInData.org/world-population-growth • CC BY

Fig. 1.1 – Evoluția repartiției populației în funcție de vârstă [4]

Aceste schimbări, pe lângă problemele legate de sistemul de pensii și asistență socială aduc în prim plan o serie de aspecte care vor influența modul în care societățile trebuie să se organizeze, atât la nivel de spații de locuit, sistem de învățământ sau sanitar, cât și la nivel de obiceiuri și preferințe. Noua generație este mai puternic ancorată în tehnologie, având altă abordare în ceea ce privește intimitatea și viața publică, nivelul de acceptare al noilor tehnologii și integrarea acestora în viața de zi cu zi. Companiile care reușesc și vor reuși să își adapteze oferta la obiceiurile noii generații au devenit noii lideri mondiali în ceea ce privește cifra de afaceri.

- **Avans (salturi) tehnologic**

Revoluția digitală nu are granițe sau limitări, implicând o schimbare a abordării în majoritatea domeniilor comerciale sau industriale. Companiile vechi (și mari), dezvoltate într-o perioadă în care tehnologia era scumpă, rigidă și greu accesibilă se vor adapta mai greu noului trend tehnologic, spre deosebire de firmele noi, apărute în epoca digitală.

Pandemia cauzată de virusul gripal "Covid-19" a adus, printre altele, o accelerare a digitalizării la nivel mondial, datorită transformărilor imense ale modului de viață în anii 2020-21. Adoptarea sistemului de lucru "remote" a determinat dezvoltarea de unelte software și hardware care să permită simularea prezenței fizice la locul de muncă, mergând de la programe de video-call, file-sharing, la soluții pentru consultații medicale online, livrare la domiciliu, etc.

În acest context, apariția pe piața largă a primelor programe de Inteligență Artificială capabile să poarte conversații complexe, să realizeze secvențe de cod în limbajele de programare uzuale sau să realizeze elemente artistice vizuale sau audio, un domeniu unde s-a considerat până de curând ca oamenii dețin exclusivitatea, a reprezentat un cutremur major pentru sectorul IT și nu numai. Perspectiva utilizării de astfel de programe de IA pentru a înlocui operatori umani în activități din ce în ce mai complexe a determinat un val de concedieri în marile companii IT, automatizarea proceselor devenind o perspectivă imediată și în alte sectoare, considerate anterior intangibile, cum ar fi contabilitatea și finanțele, grafică, media, drept, industrie și agricultură, medicină etc. O serie de companii au fost prinse pe picior greșit de lansarea acestor programe AI de modelare și procesare a limbajului uman, încercând ulterior să introducă pe piață propriile modele. Pe lângă problemele de etică și moralitate într-un domeniu în care avansul tehnologic este mai rapid decât dezvoltarea cadrului legislativ, apariția acestor tehnologii va schimba din temelii distribuția pe piața muncii, dar și modul în care va fi abordată educația, cercetarea, medicina, artele, zona financiar-bancară și multe alte domenii.

### **1.2 Industry 4.0**

De-a lungul timpului au fost înregistrate momente în care dezvoltarea și adoptare pe scară largă a unei noi tehnologii în domeniul industrial a reprezentat o transformare radicală a acestui domeniul. Primul astfel de moment a venit în Anglia secolului al XVIII-lea odată cu dezvoltarea primului motor cu aburi, invenție ce a determinat o schimbare a modului în care erau realizate exploatările miniere, prelucrările industriale (industria textilă în prima fază) și transportul. Treptat, în decursul unei jumătăți de secol, țările europene au văzut beneficiile aduse de utilizarea motorului cu aburi și au început utilizarea sa, chiar cu opoziția inițială a industriașilor englezi care ar fi dorit să-și păstreze monopolul, secolul al XIX-lea fiind marcat de extinderea treptată spre sudul și estul Europei. Aceste evenimente au

rămas în istorie ca ”prima revoluție industrială”, creșterea productivității muncii și a vitezei de transport fiind ceva nemaîntâlnit până în acel moment.

Finalul secolului al XI-lea și începutul secolului XX au adus o nouă etapă în dezvoltarea industrială odată cu descoperirea electricității și dezvoltarea de echipamente care converteau acest tip de energie în energie mecanică (motoarele electrice), lumină (corpurile de iluminat), energie termică (încălzitoare cu rezistențe), etc. Adoptarea în procesele industriale, suprapusă cu tehnologia de producție pe linii de montaj, dezvoltată inițial în fabricile Ford, a reprezentat începutul celei de-a doua revoluții industriale, care s-a întins până la începutul primului Război Mondial. Perioada e caracterizată prin dezvoltări ale industriei manufacturiere, dezvoltarea comunicațiilor pe distanțe lungi (telegraful), extindere rețelei de transport feroviar, dar și a rețelelor de apă, gaz, canalizare și electricitate în zonele urbane.

Cea de-a treia revoluție industrială a debutat după încheierea celui de-al Doilea Război Mondial și este marcată de tranziția de la sistemele mecanice și analogice la sistemele digitale, fiind perioada în care au apărut primele sisteme de calcul digital și sisteme de stocare a informațiilor în format digital. În paralel se dezvoltă electronica de putere cu ajutorul materialelor semiconductoarelor, producția de masă și sistemele de calcul bazate pe procesoare. Perioada, cunoscută și drept ”Era informațională” a culminat cu apariția și dezvoltarea sistemelor de comunicații mobile (telefoanele mobile), a liniilor de producție automatizate și a internetului.

La mijlocul anilor 2010 este folosit pentru prima dată termenul ”Industry 4.0”, care definește a patra revoluție industrială sau 4IR (4th Industrial Revolution). Perioada este una extrem de dinamică, marcând apariția unor tehnologii noi, cum ar fi inteligența artificială, fabricația aditivă, blockchain, cloud, simulări, big data and analytics și (Industrial) Internet of Things [5]. Toate acestea au schimbat modul în care sunt organizate și funcționează industria, accentul fiind acum pus pe optimizarea procesului tehnologic și a stocurilor de materie primă și produse finite, dezvoltarea de modele digitale ale proceselor reale (digital twin), conducerea autonomă a autovehiculelor, tranziția spre surse de energie regenerabile și autovehicule electrice, etc. La acestea se adaugă creșterea exponențială a numărului de echipamente conectate la internet și a volumului de date transferat între acestea, cunoscut generic ca Internet of Things, în a doua jumătate a decadei anterioare numărul de echipamente conectate la rețea depășind populația Pământului. Echivalentul din zona industrială este cunoscut ca IIoT (Industrial Internet of Things) și permite transformarea unităților de producție în sisteme cu un grad de conectivitate

și autonomie extrem de ridicat, produsele putând fi urmărite cu exactitate pe toată durata ciclului de producție, orice problemă apărută putând fi cu ușurință identificată și corectată cu intervenție umană și costuri minime. Industry 4.0 și IIoT au schimbat în același modul de abordare a mentenanței, modelele digitale ale echipamentelor, împreună cu softurile de Inteligență Artificială, realizând previziuni foarte precise ale stării de funcționare și duratei de viață rămase pentru sistemele de producție, costurile de intervenție fiind reduse semnificativ.

### **1.3 Industrial Internet of Things**

International Telecommunication Union definește IoT ca infrastructura globală pentru societatea informațională, care facilitează servicii avansate, specific, ce sunt realizate prin interconectarea de articole fizice și virtuale funcționând pe bază de tehnologii existente, respectiv în evoluție, de interoperabilitate a informațiilor și datelor. Fiecare producător de echipamente electrice potențate prin electronica dedicată, având capacități de conectare la rețele de date, deplasează în fapt centrul de greutate al afacerii sale spre componenta software [6]. Conceptul IoT a fost introdus în 1999 de britanicul Kevin Ashton, un vizionar al tehnologiei ce a creat un standard global, deschis, care reglementează utilizarea identificatorilor cu radio frecvență (RFID).

IoT dezvoltă sistemul nervos central a unei lumi mai ‘deștepte’, construindu-se pe baza unor echipamente și aparate care pot măsura și interacționa cu oamenii. Este vorba despre sisteme inteligente, interconectate, indiferent că este vorba despre expresorul de cafea, îmbrăcăminte sau multe alte bunuri. Potrivit IEEE Standards Association, IoT va deveni unul din factorii motrici ai creșterii într-o zonă largă de tehnologii. Ca în orice zonă cu potențial imens de business, evaluat la 14.2 mii de miliarde de dolari în 2030, se remarcă o absență a reglementărilor în fazele incipiente ale procesului de dezvoltare. Cum este de așteptat, abundența de idei, respectiv a cazurilor specifice de utilizare, conduc la o fragmentare care este în esență contra-productivă.

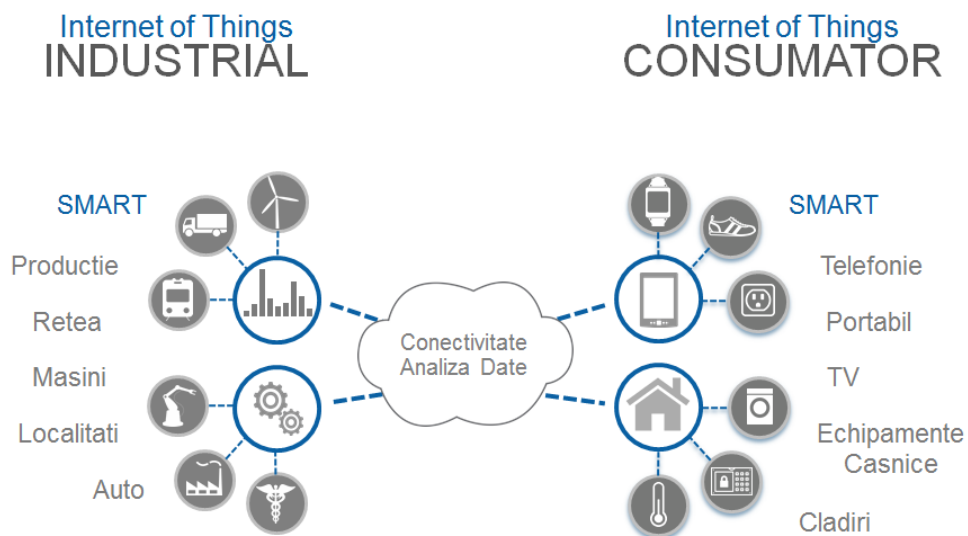


Fig. 1.2 – Internet of Things (sursa: National Instruments)

Stricto sensu, Internet of Things înseamnă ‘lucruri’ sau ‘obiecte’ ce sunt conectate între ele, respectiv conectate la Internet. Varietatea e nelimitată, fiecare din aceste obiecte având un număr unic de identificare, respectiv o adresă IP.

Într-o accepțiune mai largă, IoT adresează aplicațiile la nivelul consumatorului, eterogene în esență – telefonie mobilă, accesorii portabile și purtabile (ce monitorizează diverși parametri și indici de performanță), echipamente și bunuri casnice (instalații audio-video, frigidere, mașini de spălat, etc), respectiv zona de domotică (automatizări casnice/domestice). O evoluție similară se derulează în industrie, unde concepte similare sunt adoptate sau sunt pe cale de a fi adaptate, și ale cărei particularități vor fi analizate ulterior în acest capitol.

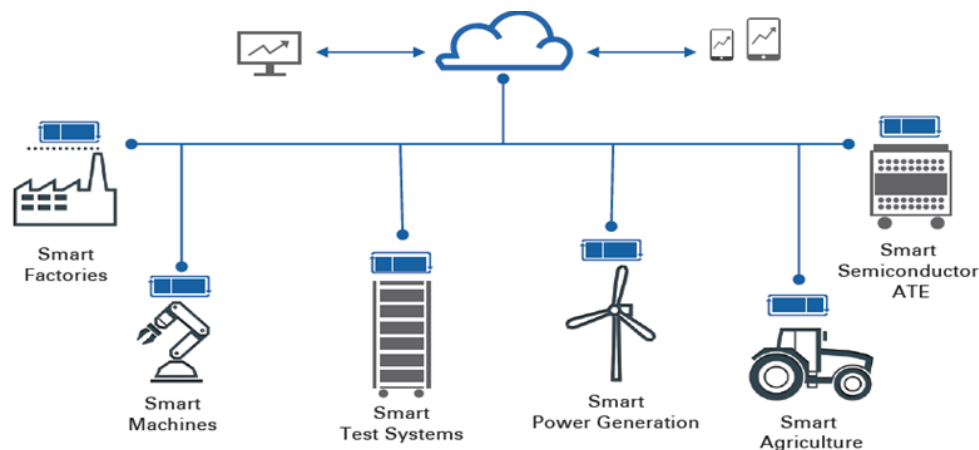


Fig. 1.3 – Internet of Things (sursa: National Instruments)

Cele două zone de acoperire, Consumator respectiv Industrial, sunt caracterizate de o serie de diferențe, dar ambele alcătuiesc IoT. Caracteristicile comune au în vedere crearea și conectarea de echipamente înzestrate cu inteligență ('things' – lucrurile, obiectele). Pe de altă parte, există două diferențe esențiale. În primul rând, IoT Consumator are în vedere cele utilizate în mod curent, în scop personal, în viața de zi cu zi: telefoane inteligente, electrocasnice, îmbrăcăminte, brățări pentru monitorizarea activităților sportive, termostate, și altele.

Samuel Greengard, în 'The Internet of Things'[7], face o distincție clară între aceste obiecte conectate și interconectate, anume physical-first (primar-fizic), respectiv digital-first (primar-digital), preluând la rândul său această idee dintr-o lucrare publicată de ABI Research. Prima categorie consistă în obiecte și procese care nu generează sau vehiculează conținut digital în mod uzual, ci numai în circumstanțe speciale în care sunt prevăzute cu dispozitive dedicate care să îndeplinească aceste funcțiuni. Cea de-a doua categorie se referă la dispozitive care sunt capabile să genereze sau să producă date în mod inerent fiind proiecte în acest sens, și livrând mai departe aceste date spre utilizare unor terțe entități.

Alt termen care intră în sfera IoT este 'Industrial Internet', care gravitează în jurul mașinărilor echipate cu senzori, ceea ce le face inteligente. Industrial IoT (IIoT) are în vedere echipamentele industriale, infrastructura, sistemele de transport, automobilele, rețeaua electrică, echipamentele de producție din fabrici, sistemele industriale de testare, și multe altele. Focalizarea pe zona industrială aduce constrângeri specifice legate de securitate, calitate, latență/timing și durata ciclului de viață. Sistemele industrial IIoT sunt mai complexe și mai dificil de proiectat din

aceste motive. Cu toate acestea, ambele laturi ale IoT, consumator și industrial, sunt dezvoltate pe aceleași tehnologii care implică măsurare, calcul și procesare – pentru a asigura inteligența, execuție, conectivitate – pentru a asigura distribuția informației, analiză, afișare/prezentare a informației.

În cadrul IIoT, comunicarea datelor urmărește trei trasee distincte: mașină-mașină (M2M), om-mașină (H2M), respectiv mașină-smartphone (M2S), unde smartphone-ul poate fi oricare alt echipament cu interfață tactilă, spre exemplu tabletă.

Eficiența IoT vine din modul simplu în care toate aceste elemente se conectează între ele, fie că e vorba de cele care aparțin categoriei primar-fizic, fie respectiv acestea conectate la cele aparținând categoriei primar-digital. Mai mult decât atât, în momentul în care în acest tablou apare și factorul uman ce utilizează o diversitate de mijloace de calcul, putem vorbi de Internet of Humans (IoH).

Suma celor menționate anterior este Internet of Everything (IoE),[8] , termen creat de Cisco Systems, care reprezintă o stare evoluată și avansată, unde lumea fizică se întrepătrunde cu lumea digitală într-un spațiu unic ce dislocă un potențial enorm prin inteligența artificială, învățare automată, diminuând în consecință nevoia de implicare și intervenție a factorului uman.

Un sumar, simplificat, al acestor componente este ilustrat mai jos, potrivit Adrian McEwen și Hakim Cassimally, autorii unei cărți interesante, ‘Designing the Internet of Things’, [9].

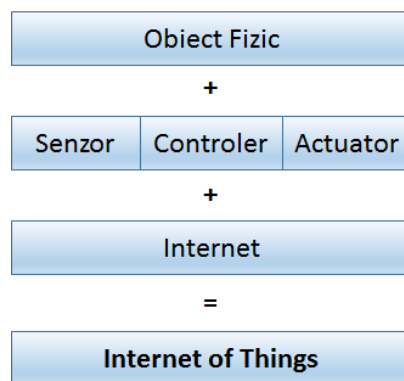


Fig. 1.4 – Componentele IoT

Dezvoltarea IIoT este influențată de trei tendințe majore:

- Costurile de măsurare și procesare a datelor scad în mod continuu, eficiența acestora operațiuni fiind în același timp în creștere, concomitent cu implementarea la nivelul echipamentelor a ceea ce putem numi ‘inteligența locală’, și respectiv a ceea ce conduce la abilitatea de a fi ‘smart’.

- Conectivitatea și operarea în rețea între echipamentele ‘smart’, care la nivel superior interacționează sub o umbrelă organizațională distinctă (firmă, fabrică, etc), mai departe integrate într-o abordare și mai largă prin servicii de tip cloud/internet

- Analiza integrală a datelor, interpretarea și valorificarea acestora fiind realizată de sisteme inteligente, având ca scop esențial elaborarea de prognoze ce permit luarea de decizii eficiente

Sistemele IIoT au la bază aceeași arhitectură, indiferent că este vorba despre monitorizarea unui hidrogenerator, respectiv dezvoltarea și implementarea unui sistem de fabricație inteligent sau controlul unei rețele de producere și distribuție a energiei de tip microgrid. Cerințele care trebuie îndeplinite de acestea pot fi grupate în trei categorii distincte:

- Calcul numeric efectuat de către echipamente dedicate, având ca scop analiza datelor, luarea de decizii, respectiv derularea de secvențe de control. În cazul de față, se vor analiza implementările ce presupun procesare de timp real pe procesoare FPGA;

- Conectivitate, ce are ca scop vehicularea de date și informații între noduri, spre cloud, spre sistemele ERP ale companiei. În cele ce urmează, conectivitatea va fi abordată prin prisma protocoalelor industriale de comunicații, respectiv implementarea acestora utilizând soluții standardizate client/server de tip OPC UA, care au devenit implementările de facto în platformele eterogene de monitorizare și control distribuit;

- Control, care include atât partea de măsurare cât și partea de execuție, ambele fiind instanțe I/O ale IIoT. În cele ce urmează, partea de execuție va fi abordată mai detaliat, prin prisma rolului pe care-l are în IIoT, respectiv a abordării integrate în sisteme dezvoltate prin programare grafică, mai specific, mașina electrică în ipostaza versatilă de actuator.-





Fig. 1.5 –Elementele ce compun IIoT (sursa: National Instruments)

Toate aceste elemente contribuie la generarea unui volum imens de date și informație, cunoscut sub numele de ‘Big Data’, sau mai specific în cazul nostru, ‘Big Analog Data’. Sursele sunt extrem de variate, plecând de la diversitatea de senzori atașați sau înglobați în aparate și echipamentele care devin și datorită acestora ‘smart’, continuând cu sisteme industriale de testare care generează la rândul lor masive de date în timpul validării, verificării, producției și testării funcționale. Analiza efectivă a acestor date în scopul luării unor decizii inteligente nu este o sarcină deloc simplă.

În acest context mondial cunoștințele în domeniul utilizării automatelor programabile și al integrării lor în noua abordare a sistemelor de producție sunt vitale oricărui inginer. Această carte își propune să prezinte în primul volum unele dintre cele mai utilizate modele de PLC-uri de pe piață: Siemens Logo! și Siemens S7-1200. Se va prezenta modul în care acestea se programează și se integrează în exemple simple de automatizări industriale. Cel de-al doilea volum își propune să prezinte modul în care aceste modele și echipamente produse de compania National Instruments pot fi utilizate în aplicații mai complexe, de tip IIoT, funcționând în rețele locale și conectate la cloud.

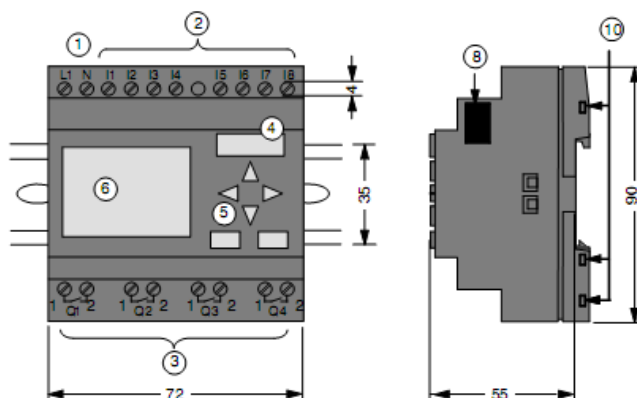
## 2. Automatul programabil Siemens Logo!

### 2.1 Prezentare generală

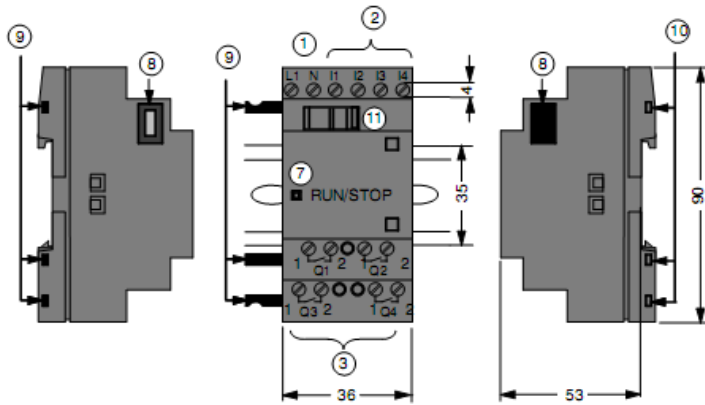
Automatele programabile (Programmable Logic Controllers) se găsesc într-o mare varietate pe piață, în funcție de producător și complexitate. Firma Siemens oferă ca model de bază familia de PLC-uri **Logo!**, cu număr mic de porți de intrare/ieșire și memorie redusă. În cadrul acestui capitol se propune familiarizarea cu acest tip de automate programabile și cu terminologia specifică PLC-urilor în general.

#### Caracteristici tehnice

Clasa **Logo!** cuprinde două categorii în funcție de tipul de alimentare, prima având tensiune de alimentare  $< 24V$  (12 sau 24V c.c. sau 24V c.a.), specifică circuitelor de comandă, iar clasa a doua având tensiunea de alimentare  $> 24V$ , în general 115..240V c.a./c.c. PLC-urile sunt diferențiate prin numărul intrări/ieșiri, memoria internă disponibilă, posibilitatea de extensie a modului, prezența display-ului, tipul intrărilor (analogice sau digitale), tipul funcțiilor interne, etc. În figura 2.1 se prezintă dimensiunile fizice și componentele unui modul central și ale unui modul de extensie **Logo!**.



- |                     |                     |                               |
|---------------------|---------------------|-------------------------------|
| 1 Alimentare        | 4 Conector extensii | 8 Interfață pentru expansiune |
| 2 Conectori intrări | 5 Taste control     |                               |
| 3 Conectori ieşiri  | 6 Display LCD       | 10 Cleme pentru aliniere      |



- |                     |                               |                          |
|---------------------|-------------------------------|--------------------------|
| 1 Alimentare        | 7 Indicator RUN/STOP          | 9 Cleme pentru montare   |
| 2 Conectori intrări | 8 Interfață pentru expansiune | 10 Cleme pentru aliniere |
| 3 Conectori ieşiri  | 11 Slide                      |                          |

Fig. 2.1 – Dimensiunile geometrice ale **Logo!**

### **Punerea în funcțiune (comisionare )**

În general automatele programabile sunt construite pentru a permite montarea ușoară în cadrul dulapurilor de alimentare prevăzute cu șine de 35mm. În funcție de modelul ales, alimentarea se va face cu tensiune continuă de 12-24V, specifică circuitelor de comandă sau cu tensiune alternativă 230V (sau 110 pentru anumite zone). În continuare se va exemplifica modalitatea de alimentare cu energie electrică pentru cel de-al doilea caz, montaj specific modelului **Logo! 230RC**.

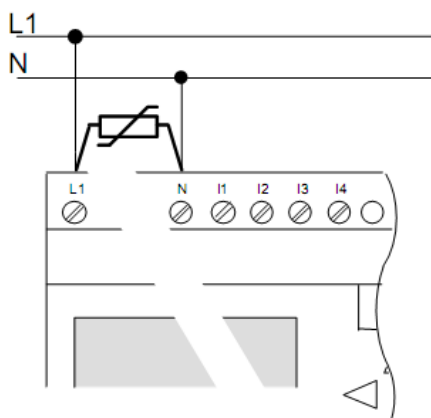


Fig. 2.2 – Alimentarea modului **Logo! 230RC**

Varistorul se montează pentru a elimina vârfurile de tensiune, dar poate fi omis din schemă, la fel ca și conectarea la pământare, acolo unde este cazul. În cazul intrărilor digitale, o tensiune la borne de mai puțin de 40 V și 0,03 mA va reprezenta 0 logic, pentru 1 logic fiind nevoie de o tensiune mai mare de 79 V și un curent mai mare de 0,08 mA. Modul de conectare a senzorilor la intrările digitale ale PLC-ului sunt ilustrate în figura de mai jos; pentru modelul **Logo! 230RC** conectarea celor 6 intrări se poate face și de pe aceeași fază.

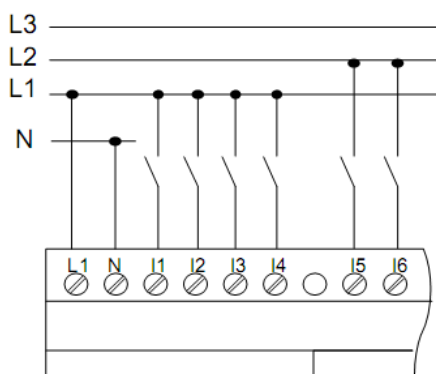


Fig. 2.3 - Modul de conectare a intrărilor digitale pentru **Logo! 230RC**

Modelul **Logo! 230RC** prezintă 4 ieșiri de tip releu (relay), putând comuta cu o întârziere de 50 ms curenți de maxim 10 A (c.c.), în curent alternativ puterea maximă depinzând de tipul sarcinii conectare la borne, în cazul unui bec cu incandescență ajungând la 1000 W (pentru aproximativ 25.000 cicluri pornire/oprire).

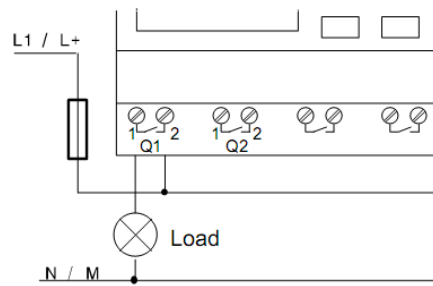


Fig. 2.4 – Modul de conectare a ieşirilor pentru **Logo! 230RC**

## Interfaţa grafică

Datorită faptului că **Logo!** nu este prevăzut cu un buton de ON/OFF, la conectarea la reţea PLC-ul va porni automat. Mesajul afişat pe display depinde de o serie de factori, cum ar fi prezenţa unui program înscris în memorie, conectarea unui cablu de date sau durata trecută de la ultima alimentare. În continuare se vor prezenta paşii ce trebuie făcuţi în cazul unui **Logo!** nou, aflat la prima punere în funcţiune. În acest caz, în momentul conectării la reţea, PLC-ul va afişa mesajul „No program”, indicând faptul că memoria internă este goală, nefiind înscris nici un program. Acesta poate fi înscris în memorie prin programare directă pe PLC sau prin transfer de pe un calculator prin intermediul unui cablu de date.

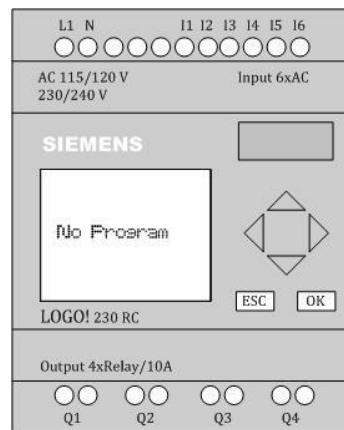


Fig. 2.5 – Interfaţa **Logo!** la alimentarea cu energie electrică

Apăsând în același timp tastele  $\triangleleft + \triangleright + \text{OK}$  se va accesa meniul principal al PLC-ului, care conține 3 opțiuni: Program, PC/Card și Start. Utilizând butoanele UP / DOWN se va alege opțiunea dorită, marcată printr-un semn „>” ce apare în partea din stânga a ecranului. Aici se va selecta opțiunea „Program” dacă se dorește editarea sau realizarea locală a unui program, „PC/Card” dacă se dorește conectarea la un calculator a PLC-ului pentru transferul de programe spre/dinspre calculator sau opțiunea „Start”, pentru intrarea **Logo!**-ului în modul RUN.

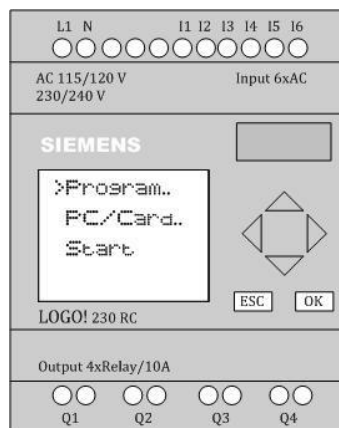


Fig. 2.6 – Meniul principal **Logo!**

Opțiunile din cadrul sub-meniului „PC/Card” vor fi prezentate ulterior, când va fi abordată conectarea PLC-ului la calculator. Prin accesarea opțiunii „Program” din meniul principal vor fi accesibile 3 noi opțiuni: „Edit prg”, „Clear prg” și „Set Clock”. Din cauză că o parte dintre funcțiile **Logo!** nu sunt disponibile dacă ceasul intern al PLC-ului nu este setat, primul pas este de a seta ora și data curentă. Folosind din nou tastele de direcție se vor seta, în ordine, ziua săptămânii, ora și minutul, precum și data curentă, în formatul lună.zi.an, după finalizare fiind apăsată tasta  $\text{OK}$ .

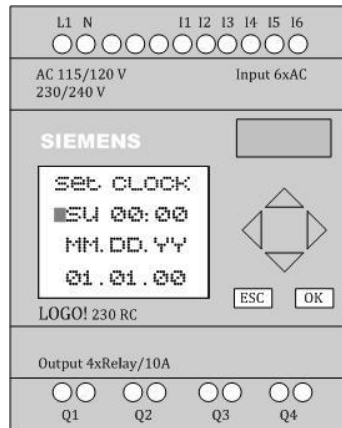


Fig. 2.7 – Setarea orei și a datei în **Logo!**

Submeniul „Clear program” va șterge programul înscris în memorie, fiind necesară o confirmare prealabilă din partea utilizatorului. Pentru realizarea unui program nou sau editarea unuia deja existent se alege „Edit program”, funcționarea editorului de programe fiind prezentată în paragraful următor.

## 2.2 Programarea Logo!: Scheme logice, diagrame bloc

Din punct de vedere istoric PLC-urile au apărut din nevoia simplificării schemelor electrice de comandă, care au devenit extrem de complexe, întreținerea, reparația și upgradarea acestora fiind foarte dificile. Din acest motiv, schemele logice inițiale erau asemănătoare cu schemele electrice, astfel încât trecerea spre programare să fie cât mai simplă. Au fost dezvoltate 3 moduri de programare: Ladder Diagram, Statement List și Block Diagram, acestea fiind prezentate în Figura 2.8.

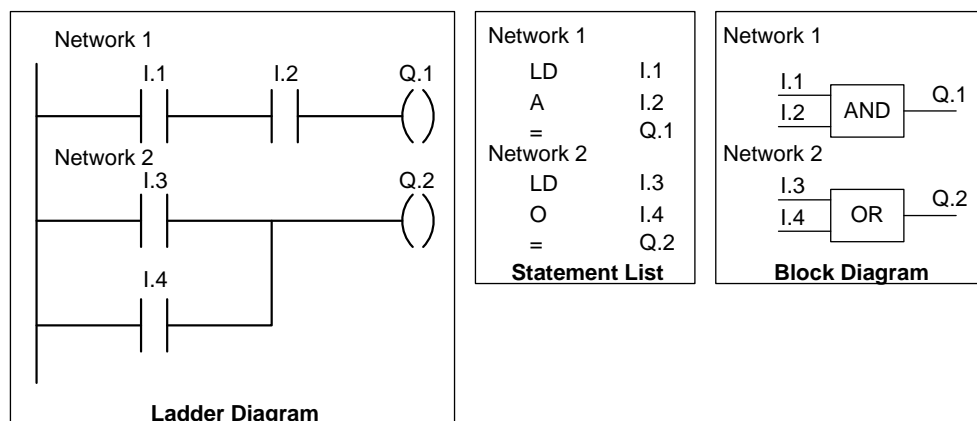


Fig. 2.8 – Metode de programare a PLC-urilor

Modul de programare Ladder seamănă cu schemele electrice clasice, considerându-se că linia verticală din stânga este alimentarea cu energie electrică iar ieșirile notate cu Q.1 și Q.2 sunt alimentate dacă o parte din intrările notate cu I.1-I.4 sunt închise. Acest tip de programare este specific automatelor programabile Siemens din familia S7, ce vor fi abordate în capitolele următoare. „Statement list” este un mod de programare mai avansat, funcțiile fiind prezentate sub formă de linii de program, similare limbajelor de programare clasice. Diagramele bloc sunt similare schemelor logice utilizate în programare, fiind modul în care se realizează programele în **Logo!**.

Fiecare bloc are desemnat un nume unic în cadrul diagramei bloc realizate. Elementele unui bloc sunt prezentate în figura de mai jos, în partea din stânga fiind reprezentate bornele de intrare, partea centrală arătând tipul funcției, iar partea din dreapta reprezintă ieșirea. În **Logo!** funcțiile sunt împărțite în două categorii, funcții de bază și speciale, conectorii blocului modificându-se în concordanță cu funcția aleasă. Intrările unui bloc pot fi legate direct la una dintre cele 6 intrări sau conectori, notate cu I1-I6, un registru de memorie, o altă funcție de bază sau specială sau poate fi lăsată liberă, fapt marcat printr-un „x”. Pe lângă aceste variante, ce se pot modifica în timpul rulării programului, mai pot fi folosite două valori interne, „hi” și „low”, reprezentând 1(ON), respectiv 0(OFF) logic. Ieșirile se vor conecta direct la o ieșire Q1-Q4 sau vor reprezenta intrări pentru alte blocuri.



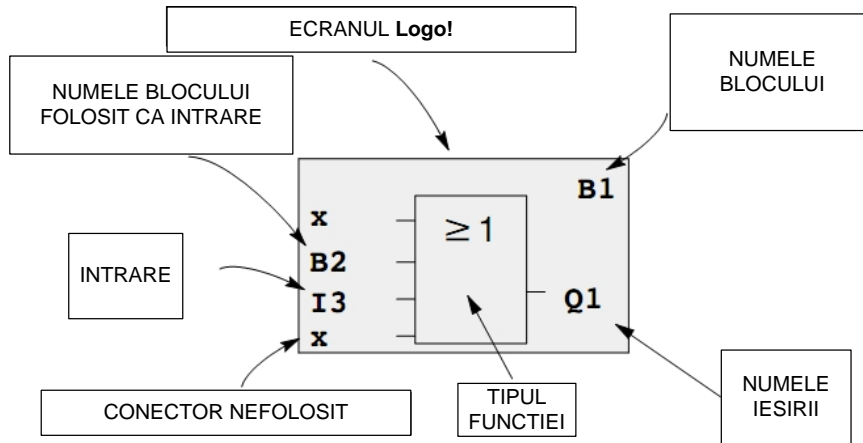


Fig. 2.9 –Elementele unui bloc de programare

Un lucru specific programării direct pe **Logo!** e reprezentat de faptul că programarea se face dinspre ieşiri spre intrări, deci înainte de realizare programului pe PLC este necesară o studiere atentă a problemei și găsirea unei soluții optime. Astfel, primul pas este alegerea ieşirii ce va corespunde ramurii de diagramă ce urmează a se realiza, având opțiunile Q1-Q4 și M1-M8, locații de memorie unde pot fi stocate informații, apelabile ulterior în cadrul diagramei. Următorul pas este alegerea tipului conectorului ce va fi cuplat la această ieşire, existând 3 opțiuni: „Co” – connector, „GF” – funcție de bază sau „SF” – funcție specială. În cadrul categoriei „Co” se poate alege una dintre acela 6 intrări I1-6, una dintre cele 4 ieşiri Q1-4, o locație de memorie M1-8, bitul „hi” sau „low” sau blocul poate rămâne neconectat prin alegerea „x”. Prin alegerea „GF” sau „SF” un bloc va apărea automat pe ecran, tipul lui depinzând de funcția aleasă. Cele două categorii vor fi tratate separat în cele ce urmează.

O diagramă bloc ce cuprinde 3 funcții de bază de tip „SAU” este prezentată în Figura 2.10. Blocul B1 are ca ieşire conectorul Q1, iar ca intrare sunt conectate ieşirile blocurilor B2 și B3, ce au ca intrări conectorii I1-I3, respectiv I4-I6. În acest mod a fost realizată o funcție „SAU” între 6 intrări, lucru ce nu poate fi realizat cu ajutorul unui singur bloc. Pe acest exemplu simplu se poate observa modul în care blocurile sunt numerotate de la dreapta la stânga, începând de la ieşire spre bornele de intrare.

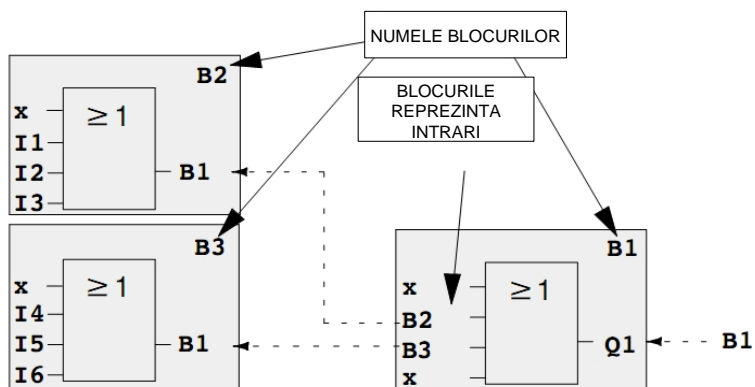

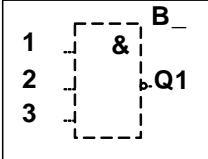
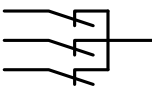
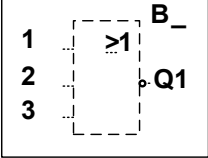
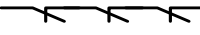
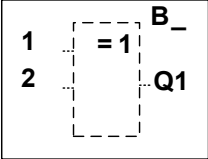
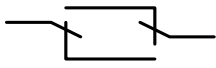
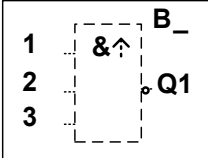
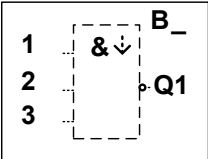


Fig. 2.10 – Conectarea blocurilor în cadrul diagramei

## 2.3 Funcții de bază

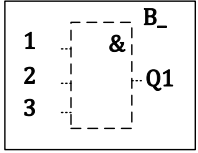
Funcțiile de bază se regăsesc în marea lor majoritate în circuitele electrice clasice, din acest motiv ele vor fi prezentate alături de echivalentul lor electric; **Logo!** oferă 8 astfel de funcții, prezentate alături de simbolul lor în tabelul următor.

Numele funcției	Blocul <b>Logo!</b>	Echivalentul în circuitele clasice
<b>ȘI (AND)</b>		Circuit serie cu contacte normal deschis (NO) 
<b>SAU (OR)</b>		Circuit paralel cu contacte normal deschis (NO) 
<b>NEGARE (NOT)</b>		Contact normal închis (NC) 

		
<b>ȘI NEGAT (NAND)</b>		Circuit paralel cu contacte normal închis (NC)  
<b>SAU NEGAT (NOR)</b>		Circuit serie cu contacte normal închis (NC)  
<b>SAU EXCLUSIV (XOR)</b>		Contact tip „cap de scară”  
<b>ȘI NEGAT CU EVALUAREA STĂRII ANTERIOARE</b>		
<b>SAU NEGAT CU EVALUAREA STĂRII ANTERIOARE</b>		

### Funcția „SI”

Funcția „SI” este similară unui circuit ce cuprinde mai multe contacte de tip normal deschis conectare în serie, deci va returna „1 logic” la ieșire în situația în care **TOATE** intrările sunt ON. Stările prin care poate trece funcția sunt sintetizate în următorul tabel de adevăr.

	I1	I2	I3	Q
	0	0	0	0
	0	0	1	0
	0	1	0	0
	0	1	1	0
	1	0	0	0
	1	0	1	0
	1	1	0	0
	1	1	1	1

În cazul că nu sunt utilizate toate cele trei intrări, una dintre ele este înlocuită cu „x” și se consideră că aceasta este în starea de ON.

### Funcția „SAU”

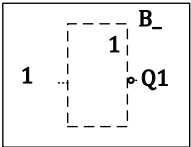
Funcția „SAU” este similară unui circuit electric ce are mai multe contacte legate în paralel, astfel încât ieșirea trece pe starea ON în momentul în care cel puțin una dintre intrări se află în starea de 1 logic.

1	}	≥1	}	B <sub>1</sub>
2				Q1
3				
	I1	I2	I3	Q
	0	0	0	0
	0	0	1	1
	0	1	0	1
	0	1	1	1
	1	0	0	1
	1	0	1	1
	1	1	0	1
	1	1	1	1

O intrare neutilizată, înlocuită cu simbolul „x” va avea starea 0 logic, neinfluențând starea ieșirii. Cele 8 cazuri posibile în cazul blocului SAU cu 3 intrări sunt sintetizate în tabelul de adevăr prezentat mai sus.

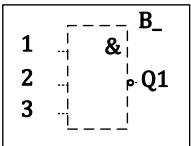
### Funcția „NEGARE”

Transformă semnalul digital de la intrare în opusul acestuia, fiind similar unui contact normal închis, care în momentul în care este acționat va trece în stare de 0 logic. Prin utilizarea acestui bloc în cadrul diagramelor nu mai e necesară utilizarea de contacte fizice normal închis. Funcția are o singură intrare, tabelul de adevăr fiind unul extrem de simplu.

	I1	Q
	0	1
	1	0

### Funcția „ȘI NEGAT ”

Funcția este similară unui circuit electric cu contacte normal închis legate în paralel, rezultând că ieșirea este în stare de 0 logic doar în momentul în care **TOATE** cele 3 intrări sunt în stare de 1 logic. Funcția poate fi înlocuită cu o funcție ȘI urmată de o funcție NEGARE, dar în situațiile în care programul are un număr mare de blocuri, aproape de limita memoriei **Logo!** este de preferat să se utilizeze funcția ȘI NEGAT. Tabelul de adevăr este opus cel al funcției ȘI.

	I1	I2	I3	Q
	0	0	0	1
	0	0	1	1
	0	1	0	1
	0	1	1	1
	1	0	0	1
	1	0	1	1
	1	1	0	1
	1	1	1	0

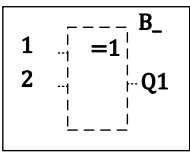
**Funcția „SAU NEGAT”**

Este similară unui circuit electric format din contacte de tip normal închis conectate în serie, deci ieșirea este pe 1 logic atunci când toate intrările sunt pe 0 logic, fapt ce poate fi urmărit în tabelul de adevăr prezentat mai jos. Ca și funcția anterioară, acest bloc îmbină funcția SAU și funcția NEGARE, pentru condensarea diagramei bloc.

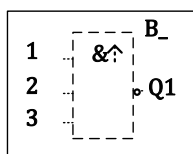
<div style="border: 1px solid black; padding: 5px; display: inline-block;"> <p>1 2 3</p> <div style="border: 1px dashed black; padding: 5px; display: inline-block;"> <p><math>\geq 1</math></p> </div> <p>B<sub>-</sub></p> <p>Q1</p> </div>	I1	I2	I3	Q
	0	0	0	1
	0	0	1	0
	0	1	0	0
	0	1	1	0
	1	0	0	0
	1	0	1	0
	1	1	0	0
	1	1	1	0

### Funcția „SAU EXCLUSIV”

Această funcție e similară unui circuit serie având două contactoare cu două poziții, ieșirea fiind ON doar dacă stările logice ale celor două intrări sunt diferite, întocmai ca și în cazul utilizării de întrerupătoare tip „cap de scară”. Dacă una dintre borne este înlocuită cu „x”, starea acesteia se consideră 0 logic. Funcționarea blocului este sintetizată în tabelul de adevăr de mai jos.

	I1	I2	Q
	0	0	0
	0	1	1
	1	0	1
	1	1	0

### Funcția „SI NEGAT CU EVALUAREA STĂRII ANTERIOARE”



Această funcție nu are corespondent în circuitele electrice clasice, fiind specifică mediului de programare utilizat de **Logo!**. Ieșirea acestui bloc este în stare de 1 logic dacă cel puțin una dintre intrările blocului este **OFF** și dacă în ciclul anterior toate intrările au fost pe **ON**. O intrare neutilizată se consideră în starea de 1 logic. Funcționarea blocului poate fi urmărită în figura de mai jos, unde se prezintă modificarea stării ieșirii Q în funcție de starea intrărilor. Se observă că dacă una dintre intrări comută între stările ON-OFF pentru o durată mai scurtă decât lățimea unui ciclu, ieșirea nu va fi influențată.



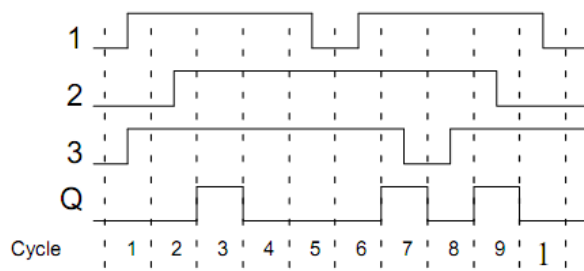
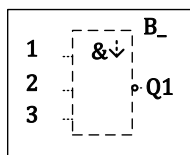


Fig. 2.11 – Modul de funcționare al funcției

**Funcția „SAU NEGAT CU EVALUAREA STĂRII ANTERIOARE”**



Ca și în cazul anterior, această funcție nu se regăsește în schemele electrice clasice. Diferența e dată de faptul că ieșirea comută pe starea de 1 logic dacă cel puțin una dintre intrări este OFF și în momentul anterior toate intrările au fost ON. O intrare înlocuită cu „x” va avea tot timpul starea de 1 logic. Datorită faptului că starea intrărilor este citită doar la finalul ciclurilor interne ale PLC-ului, un impuls mai lung de 1 ciclu poate declanșa comutarea în 1 logic a ieșirii pentru o durată de un ciclu, similar situației în care intrarea rămâne pe OFF puțin timp, dar prinde sfârșitul ciclului.

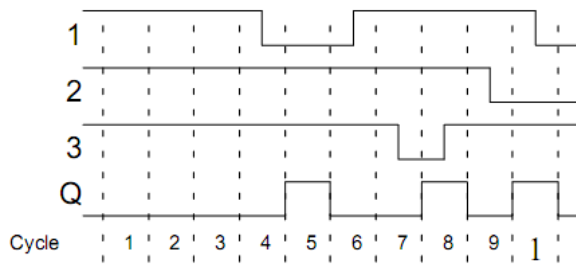


Fig. 2.12 – Modul de funcționare al funcției

### 2.3.1: Combinarea unei funcții „ȘI” cu o funcție „SAU”

Un exemplu simplu de program realizat în **Logo!** este combinarea a două funcții de bază, „SAU” și „ȘI”, conform schemei electrice prezentate în figura de mai jos. Sunt considerate 3 intrări, notate cu **I1-I3** și ieșirea **Q1**. Primele două intrări sunt conectate în serie, ce-a de-a treia fiind conectată în paralel după primele două intrări.

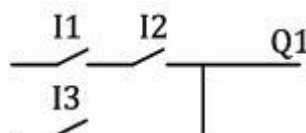


Fig. 2.13 – Circuit electric serie-parallel

Echivalentul acestei scheme electrice pentru **Logo!** este prezentat în figura de ai jos, fiind utilizate două blocuri, notate de la ieșire spre intrări, după cum s-a explicat mai sus. Blocul **B1** reprezintă funcția **SAU**, având ca intrări ieșirea din blocul **B2** și intrarea **I3** (cea de-a treia intrare în bloc nu este folosită) ieșirea blocului fiind conectată la ieșirea **Q1**. Blocul **B2** reprezintă o funcție **ȘI**, având ca intrări **I1** și **I2** (borna e treia nu este conectată) ieșirea din bloc fiind conectată la intrarea **1** a blocului **B1**.

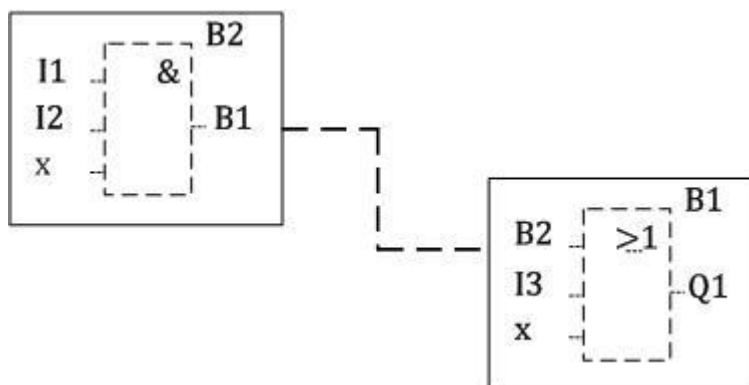


Fig. 2.14 – Conectarea unui bloc ȘI cu un bloc SAU

### Implementarea programului

În cele ce urmează se va prezenta modul de implementare al acestui program direct pe interfața oferită de **Logo!**. În momentul alimentării cu energie electrică a automatului programabil, în memoria acestuia poate să existe un program sau memoria poate să fie goală. Dacă ecranul afișează mesajul “No Program Press ESC”, înseamnă că în memoria internă a **Logo!** nu este înscris nici un program.



Fig. 2.15 –Ecranul **Logo!** dacă nu există un program în memorie

În acest caz prin apăsarea oricărei taste se va accesa meniul principal. În cazul în care automatul programabil este la prima alimentare se recomandă reglarea ceasului intern, care va fi afișat în timpul rulării programelor. În cazul în care există un program înscris în **Logo!**, display-ul va arăta starea intrărilor și a ieșirilor, precum și ora internă a automatului programabil. În acest caz, pentru accesarea meniul principal al PLC-ului este necesară utilizarea combinației de taste :  $\leftarrow + \rightarrow + \boxed{\text{OK}}$ , aleasă astfel pentru a se evita apăsarea accidentală în timpul rulării programelor.



Fig. 2.16 – Accesarea meniului principal **Logo!**

Meniu principal prezintă trei opțiuni: **Edit Prg**, unde se pot edita programele existente sau realiza programe noi, **Clear Prg**, de unde se pot șterge programele existente și **Set Clock**, unde se poate regla data și ora internă a **Logo!**. Pentru a evita complicațiile cauzate de eventualele programe existente în memoria automatului programabil, înainte de realizarea noului program se va goli memoria PLC-ului, prin alegerea opțiunii **Clear Prg**, fiind necesară confirmarea ștergerii prin alegerea **Yes** din submeniul apărut.

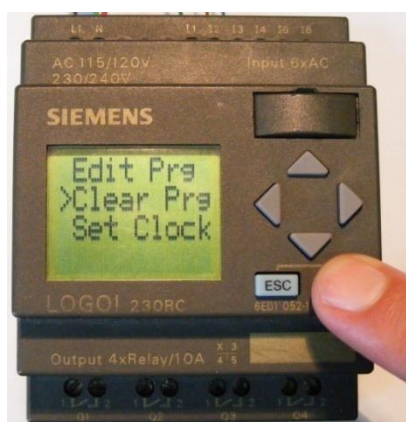


Fig. 2.17 – Ștergerea memoriei interne a **Logo!**

După resetarea automatului programabil se va trece la realizarea exemplului de program prezentat anterior, primul pas fiind alegerea opţiunii **Edit Prg** din meniul principal. La apăsarea tastei **OK** pe ecranul **Logo!** va fi afişat numele primei ieşiri, **Q1**. Prin apăsarea tastelor direcţionale UP / DOWN se pot alege oricare dintre cele 4 ieşiri şi 8 locaţii de memorie disponibile; pentru acest exemplu va fi folosită doar prima ieşire.



Fig. 2.18 – Alegerea ieşirii Q1 ca ieşire pentru program

După alegerea ieşirii dorite se poate trece la următoarea etapă de realizare a programului, acest lucru fiind realizat prin utilizarea tastei direcţionale  $\blacktriangleleft$ . În partea din stânga a ecranului se poate alege tipul blocului care se conectează la ieşirea **Q1**, existând 3 opţiuni disponibile: „Co” (Conector), GF (Funcţie de bază) sau **SF** (Funcţie specială). Prin utilizarea tastei DOWN se pot vizualiza cele 3 categorii amintite mai sus, în cazul nostru fiind necesară conectarea ieşirii **Q1** la o funcţie de bază, alegerea acesteia fiind realizată prin apăsarea tastei **OK**.



Fig. 2.19 – Alegerea unei funcții de bază

Imaginea afișată pe ecranul **Logo!** se va modifica, un prim bloc reprezentând o funcție de bază apărând pe ecran. În colțul din dreapta-sus va fi afișat numărul blocului curent, numărul ieșirii conectate la acesta fiind afișat în partea din dreapta sus. Tipul blocului este dat de simbolul prezentat în zona centrală a ecranului, simbolurile specifice pentru cele 8 funcții de bază disponibile în cazul **Logo! 230RC** fiind prezentate anterior. Prin utilizarea tastelor direcționale UP / DOWN se poate alege funcția dorită, putându-se observa că numărul de intrări în bloc se va modifica în funcție de specificul fiecărei funcții.



Fig. 2.20 – Selectarea funcției SAU

Pentru realizarea programului propus este necesară o funcție de tip **SAU**, pentru confirmarea utilizării acesteia fiind folosită tasta **OK**, apăsarea acestia mutând cursorul în partea din stânga a ecranului, unde se vor defini cele 3 intrări utilizate, de sus în jos. Conform diagramei prezentate anterior la intrarea 1 a funcției de tip **SAU** este necesară conectarea ieșirii dintr-o funcție de tip **ȘI**. Alegerea acestei funcții se face prin repetarea procedurii prezentate mai sus în cazul blocului conectat la ieșirea **Q1**. După alegerea din listă a simbolului **GF** reprezentând categoria funcțiilor de bază, la apăsarea tastei **OK** se va accesa un nou ecran, prezentat în Figura 2.21 a), conținând o nouă funcție de bază. În colțul din dreapta-sus a ecranului se observă că numărul funcției s-a schimbat, iar la ieșirea acestui bloc este conectat blocul cu numărul **B01**. Se va alege o funcție de tip **ȘI**, iar la intrările acesteia se vor conecta primele două intrări digitale ale automatului programabil. Pentru a realiza conectarea unei intrări la blocul curent se va alege opțiunea **Co**, la apăsarea tastei **OK** fiind disponibile cele 6 intrări, 4 ieșiri, 8 locații de memorie, biții **hi** și **lo**, precum și opțiunea **x**, însemnând că la acea bornă nu se conectează nimic. Pentru realizarea schemei propuse este necesar ca la intrările funcției de tip **ȘI** să se conecteze primele două intrări ale automatului programabil, **I1** și **I2**, cea de-a treia bornă a blocului **B02** fiind neutilizată.



a) selectarea funcției **ȘI**

b) Selectarea intrărilor **I1** și **I2**

Fig. 2.21 – Configurarea funcției **ȘI**

După finalizarea configurării acestui bloc se va reveni la blocul **B01**, pentru definirea ultimelor două intrări ale funcției **SAU**. Trecerea de la un bloc la altul se poate face utilizând tastele de direcție **LEFT/RIGHT**, prin această metodă putându-

se realiza și parcurgerea unui program existent pentru modificarea blocurilor sau adăugarea de blocuri noi. După adăugarea intrării **I3** la a doua intrare a blocului **B01** și dezactivarea celei de-a treia intrări programul este finalizat, putând fi rulat.



Fig. 2.22 – Confirmarea finalizării programului prin apăsarea tastei **OK**

Pentru pornirea programului este necesară revenirea la meniul principal, acest lucru putând fi realizat prin utilizarea tastei **ESC**, la o primă apăsare fiind accesate cele 3 opțiuni ale submeniului **Program**, o nouă apăsare afișând meniul principal. Trecerea **Logo!** în modul de lucru se face prin selectarea opțiunii **Start**. Din acest moment automatul programabil va rula programul realizat până în momentul întreruperii alimentării cu energie electrică sau la intrarea în modul meniu, prin metoda prezentată anterior. **Logo!** va citi toate intrările disponibile, va parcurge diagrama bloc existentă în memorie și va comuta ieșiriile în stare **ON** sau **OFF**, în funcție de logica utilizată, trecându-se la o nouă citire a intrărilor. În funcție de mărimea și complexitatea programului realizat, durata necesară relizării unui ciclu complet va varia.

Prezența unui semnal de tip 1 logic la una dintre intrări sau ieșiri este reprezentată, în timpul rulării programului, pe interfața **Logo!** prin modificarea modului de afișare a numărului corespunzător acelei borne. În Figura 2.23 se observă că cifra 2 din zona intrărilor este afișată în mod diferit, sugerând existența unui semnal pe intrarea cu **I2**, celelalte intrări fiind în stare 0 logic.





Fig. 2.23 – Ecranul **Logo!** în timpul rulării programului

Datorită programului realizat, intrarea **Q1** va rămâne în stare **OFF**, pentru ca ea să devină activă fiind necesar ca unul din următoarele (sau ambele) scenarii să se întâmple:

- **I1** și **I2** sunt active în același timp, determinând ieșirea din blocul **B02** să devină activă. În acest caz prima intrare în blocul **B01**, de tip **SAU** va fi activă, ieșirea din acest bloc și implicit ieșirea **Q3** fiind în stare **ON**.



Fig. 2.24 – Ecranul **Logo!** când **Q1** este activă datorită îndeplinirii condițiilor din funcția **ȘI**

- Intrarea **I3** este activă, determinând ca a doua intrare a blocului **B01** să fie activă. Fiind vorba de un bloc de tip **SAU**, indiferent de starea celorlalte două intrări, ieșirea funcției, conectată la ieșirea **Q1**, va fi în stare **ON**.



Fig. 2.25 – Ecranul **Logo!** când **Q1** datorită îndeplinirii condițiilor din funcția **SAU**

În cazul în care toate cele 3 intrări utilizate la realizarea acestui program sunt active, atât ieșirea din blocul **B02** cât și intrările în blocul **B01** vor fi active, însemnând că și ieșirea blocului **B01** și **Q1** vor fi **ON**.



Fig. 2.26 – Ecranul **Logo!** când toate intrările sunt active

### 2.3.2: Realizarea unui circuit cu automenţinere

În circuitele de comandă se folosesc foarte des schemele de alimentare cu automenţinere, acestea permiţând funcţionarea continuă a unui circuit fără ca butonul de pornire să rămână apăsat pe toată durata funcţionării. Un exemplu de astfel de circuit cu automenţinere e reprezentat de un contactor, a cărei bobină rămâne alimentată şi după întreruperea comenzii iniţiale, astfel încât circuitul aflat în aval este alimentat în continuare. **I1** reprezintă butonul de pornire, fiind un contact normal deschis, iar **I2** reprezintă butonul de oprire, fiind un contact normal închis. În paralel cu aceste două butoane se consideră ieşirea sistemului, **Q1**, care odată iniţializată cu ajutorul **I1** rămâne activă până la apăsarea **I2**.

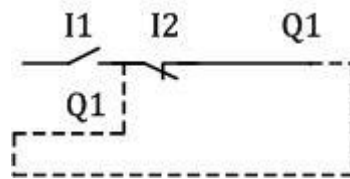


Fig. 2.27 –Circuit de comandă cu automenţinere

Transpunerea acestei scheme în **Logo!** se face cu ajutorul a 3 blocuri reprezentând funcţii de bază. Blocul **B1** este o funcţie **ŞI**, având conectate la bornele de intrare ieşirile blocurilor **B2** şi **B3**, iar ca ieşire conectorul **Q1**. Blocul **B2** este o funcţie **SAU** care are ca intrări conectorul **I1** şi ieşirea blocului **B1**, de fapt chiar ieşirea pe conectorul **Q1**. Blocul **B3** reprezintă o funcţie de negare pentru a transforma semnalul de pe intrarea **I2** dintr-unul normal deschis într-unul normal închis. Dacă un contact **NC** este disponibil la intrarea **I2**, acest bloc de negare nu mai este necesar.

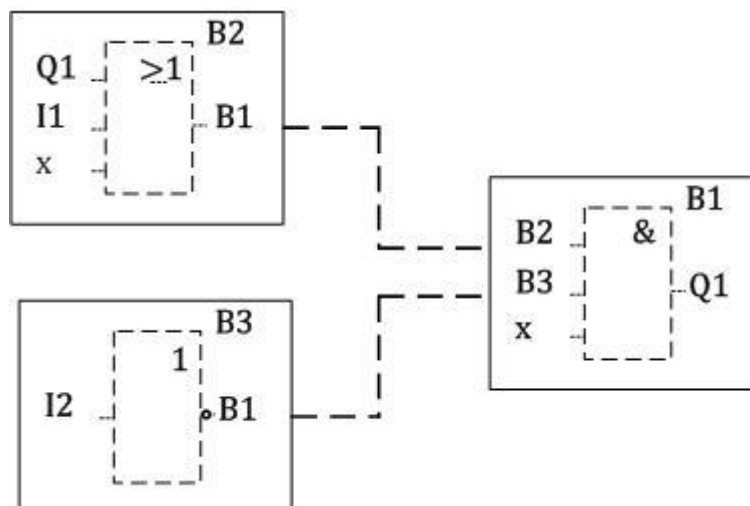


Fig. 2.28 – Schema logică pentru circuit cu automenținere

Funcționarea întregului circuit în momentul inițial este prezentată în cele ce urmează:

- la comutarea din 0 logic în 1 logic a intrării **I1** (momentul  $t1$ ) ieșirea blocului **B2** devine 1 logic, iar în cazul în care **I2** este 0 logic ieșirea din **B3** este tot 1 logic, astfel încât blocul **B1** va avea pe ieșirea **Q1** 1 logic;
- la rularea următoare a programului ( $t2$ ), chiar dacă **I1** a devenit între timp 0 logic, blocul **B2** va avea la ieșire 1 logic datorită conectării **Q1** la **B2**;
- Starea ieșirii **Q1** se păstrează până în momentul ( $t3$ ) în care intrarea **I2** devine 1 logic prin acționarea contactului NC, bucla fiind întreruptă datorită faptului că blocul **B1** va comuta pe 0 logic;
- starea **Q1** se păstrează chiar dacă starea **I2** redevine 0 logic;
- starea ieșirii **Q1** se va modifica doar în momentul în care starea **I1** devine 1 logic.

Modul în care funcționează acest program poate fi urmărit în tabelul de adevăr de mai jos, coloana timp reprezentând momentul de timp în care se apar modificări ale stării intrărilor, iar coloana **Q1\*** reprezintă valoarea de feedback folosită ca intrare pentru blocul **B2** din diagrama din Figura 2.28, adică valoarea ieșirii **Q1** din iterația anterioară.

Tabelul de adevăr al programului

timp	Q1*	I1		I2		Q1
t0	OFF	OFF	OFF	OFF	ON	OFF
t1	OFF	ON	ON	OFF	ON	ON
t2	ON	OFF	OFF	OFF	ON	ON
t3	ON	OFF	OFF	ON	OFF	OFF
t4	OFF	OFF	OFF	OFF	ON	OFF
t5	OFF	ON	ON	OFF	ON	ON

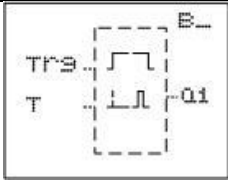
## 2.4 Funcții speciale

După ce a fost făcută o trecere în revistă a funcțiilor de bază utilizate în **Logo!**, precum și a modalității de conectare a intrărilor și a ieșirilor acestuia, vom trece la prezentarea funcțiilor speciale regăsite în cadrul acestei familii de PLC-uri.

Modelul **Logo! 230RC** oferă 21 de funcții speciale pentru realizarea de programe avansate. Acestea vor fi grupate în funcție de tipul lor, ordinea diferind de cea regăsită în meniul **Logo!**.

### Funcția „ON delay”

Funcția permite comutarea ieșirii cu o anumită întârziere față de semnalul de la intrare, timpul maxim de întârziere ce poate fi setat în cadrul blocului fiind de 99.95 secunde, 99:55 minute sau 99:55 ore, în funcție de unitatea de măsură a timpului selectată. Semnificația intrărilor este sintetizată în tabelul următor:

 <p>poziția 1 în meniul <b>Logo!</b></p>	Input Trg	Intrarea conectată la borna <b>Trg</b> declanșează timpul de delay
	Parametrul T	Parametrul <b>T</b> setează întârzierea cu care ieșirea comută din 0 în 1 logic
	Output Q	Ieșirea <b>Q</b> comută după expirarea timpului <b>T</b> , dacă <b>Trg</b> este în continuare activă

#### Diagrama de funcționare

Pentru o mai ușoară înțelegere a modului de funcționare a acestui bloc se va urmări diagrama de timp din figura următoare:

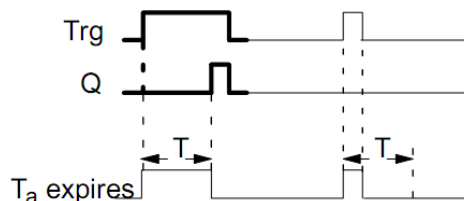


Fig. 2.28 – Diagrama de timp pentru funcția „ON delay”

În momentul când apare un semnal pe borna de trigger, se declanșează ceasul intern al PLC-ului; se face o comparare continuă cu valoarea setată a parametrului **T**, iar la expirarea timpului ieșirea **Q** comută de pe 0 logic pe 1 logic. În momentul în care semnalul de trigger dispare, ieșirea va reveni în starea inițială iar blocul așteaptă un nou semnal pe borna de trigger. Un caz aparte este atunci când durata semnalului de trigger este mai mică decât valoarea parametrului **T** setat, caz în care la dispariția semnalului **Trg** blocul revine la starea inițială fără ca ieșirea **Q** să fie modificată.

**Funcția „OFF delay”**

Funcția permite comutarea ieșirii de pe stare **ON** pe stare **OFF** cu o anumită întârziere față de dispariția semnalului de la intrare, timpul maxim de întârziere ce poate fi setat în cadrul blocului fiind similar celui prezentat la funcția anterioară. Semnificația intrărilor este sintetizată în tabelul următor:

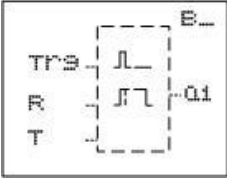
 <p>poziția 2 în meniul <b>Logo!</b></p>	Input Trg	Intrarea conectată la borna <b>Trg</b> declanșează timpul de delay la trecerea semnalului din 1 logic în 0 logic
	Input R	Intrarea <b>R</b> resetează ieșirea <b>Q</b> și timpul setat de parametrul <b>T</b>
	Parametrul T	Parametrul <b>T</b> setează întârzierea cu care ieșirea comută din 1 în 0 logic după dispariția semnalului de trigger
	Output Q	Ieșirea <b>Q</b> devine 1 logic odată cu semnalul de trigger și comută în 0 după expirarea timpului <b>T</b>

Diagrama de funcționare

Funcționarea blocului poate fi urmărită pe diagrama de timp din figura de mai jos:

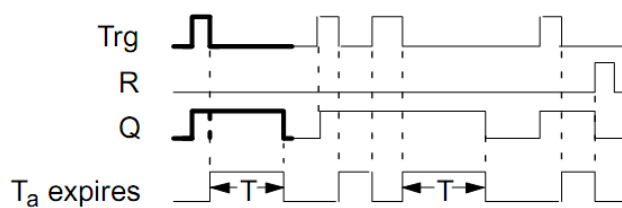


Fig. 2.29 – Diagrama de timp pentru funcția „OFF delay”

La trecerea din starea de 0 logic în 1 logic a semnalului de la borna **Trg**, ieșirea **Q** va comuta automat pe stare **ON**, iar la dispariția semnalului de trigger, se începe măsurarea timpului de delay setat cu ajutorul parametrului **T**. La expirarea acestui interval de timp, ieșirea **Q** revine la starea inițială **OFF**. În cazul în care după dispariția semnalului de trigger apare o nouă trecere din stare **ON** în stare **OFF** la borna **Trg** înainte ca timpul de delay să expire, măsurarea timpului este resetată, ieșirea rămânând în stare **ON** până la expirarea acestuia. Intrarea **R** permite trecerea forțată a ieșirii în stare **OFF** înainte de expirarea timpului **T** setat, fiind necesar un nou semnal pe **Trg** pentru ca ieșirea să redevină **ON**.



**Funcția „ON/OFF delay”**

Funcția permite setarea de timpi de întârziere a semnalului pe ieșire, atât la trecerea semnalului de trigger din 0 în 1 logic, cât și la dispariția semnalului la borna **Trg**.

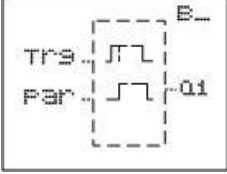
 <p>poziția 14 în meniul <b>Logo!</b></p>	Input Trg	O trecere din 0 logic în 1 logic pe borna <b>Trg</b> declanșează timpul <b>TH</b> setat în <b>Par</b> , iar o trecere din 1 logic în 0 logic declanșează timpul <b>TL</b> setat
	Parametrul Par	În <b>Par</b> se pot seta 2 timpi:  <b>TH</b> este timpul după care ieșirea este comutată pe <b>ON</b> după apariția semnalului de trigger,  <b>TL</b> este timpul după care ieșirea este comutată pe <b>OFF</b> după dispariția semnalului de trigger
	Output Q	Ieșirea Q devine 1 logic după expirarea <b>TH</b> , în condițiile în care mai există semnal pe Trg; ieșirea devine 0 logic după expirarea <b>TL</b> , dacă semnalul de pe Trg nu a redevenit <b>ON</b>

Diagrama de funcționare

Funcționarea blocului poate fi urmărită pe diagrama de timp din figura de mai jos:

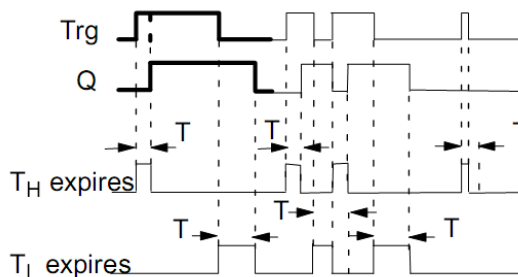


Fig. 2.30 – Diagrama de timp pentru funcția „ON/OFF delay”

La fiecare trecere din stare 0 în 1 logic a semnalului **Trg** începe măsurarea timpului setat în parametrul **TH**, iar la expirarea acestuia ieșirea **Q** comută pe starea **ON**, în cazul în care semnalul de trigger mai este prezent. Timpul **TL** este măsurat începând cu momentul trecerii semnalului **Trg** din 1 în 0 logic, ieșirea **Q** comutând pe **OFF** la expirarea timpului setat în **TL**. În cazul în care înainte de expirarea **TL** apare un nou semnal pe **Trg**, ieșirea **Q** va trece în stare **ON**.

Funcția „Retentive ON delay”

La apariția unui semnal pe borna **Trg** se declanșează ieșirea **Q** cu o întârziere setată în parametrul **T**. Ieșirea **Q** revine la starea inițială după apariția unui semnal pe borna **R**.

	Input Trg	Intrarea conectată la borna <b>Trg</b> declanșează timpul de delay la trecerea semnalului din 1 logic în 0 logic
--	-----------	--

poziția 7 în meniul <b>Logo!</b>	Input R	Intrarea <b>R</b> resetează ieșirea <b>Q</b> la 0 logic
	Parametrul T	Parametrul <b>T</b> setează întârzierea cu care ieșirea comută din 0 în 1 logic după apariția semnalului de trigger
	Output Q	Ieșirea <b>Q</b> devine 1 logic la expirarea timpului <b>T</b>

Diagrama de funcționare

Funcționarea blocului poate fi urmărită pe diagrama de timp din figura următoare:

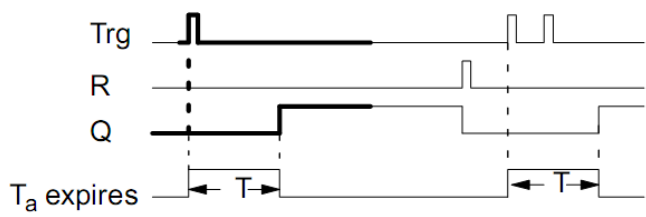
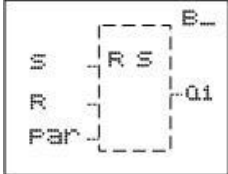


Fig. 2.31 – Diagrama de timp pentru funcția „Retentive ON delay”

Diferența majoră față de blocurile prezentate anterior este dată de faptul că semnalul de trigger nu trebuie să rămână activ, fiind suficient un impuls pentru inițializarea timpului de delay. După expirarea timpului setat în cadrul parametrului **T** ieșirea **Q** comută pe stare **ON**, rămânând activă până la apariția unui semnal la intrarea **R**. În cazul în care înainte de expirarea timpului de delay apare un nou semnal la borna **Trg** cronometrul intern nu este resetat. Un caz aparte e reprezentat de cazul în care după declanșarea cronometrului apare un semnal pe borna **R** înainte de expirarea timpului setat, caz în care ieșirea **Q** nu va deveni activă până la apariția unui nou semnal de trigger și expirarea timpului **T**.

### Funcția „Latching relay”

La apariția unui semnal pe borna **S** se comută ieșirea **Q** în stare 1 logic, iar la apariția unui semnal pe borna **R** se comută ieșirea **Q** în stare 0 logic. Parametrul **Par** determină starea ieșirii **Q** în situația unei întreruperi a alimentării cu energie electrică, starea putând fi memorată astfel încât să revină la starea anterioară sau putând fi păstrată în starea 0 logic. Un lucru important este faptul că dacă atât **S** cât și **R** sunt în stare **ON**, ieșirea va fi **OFF**.

 <p>Poziția 5 în meniul <b>Logo!</b></p>	Input S	Intrarea <b>S</b> în stare <b>ON</b> determină ieșirea <b>Q</b> să fie pe stare <b>ON</b>
	Input R	Intrarea <b>R</b> resetează ieșirea <b>Q</b>
	Parametrul Par	Parametrul <b>Par</b> determină activarea memorării stării ieșirii <b>Q</b> în cazul întreruperii alimentării PLC-ului
	Output Q	Ieșirea <b>Q</b> devine 1 logic la apariția unui semnal la intrarea <b>S</b> și devine 0 logic la apariția unui semnal la intrarea <b>R</b>

### Diagrama de funcționare

Funcționarea blocului poate fi urmărită pe diagrama de timp din figura următoare:

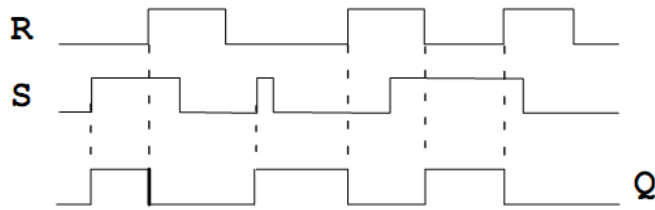
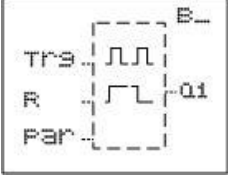


Fig. 2.32 – Diagrama de timp pentru funcția „Latching relay”

Funcționarea blocului este simplă, intrarea **S** setează starea **ON** pentru ieșirea **Q**, iar intrarea **R** resetează ieșirea **Q** în starea **OFF**. Intrarea **R** are prioritate față de **S**, adică: chiar dacă apare un semnal de SET, ieșirea nu va reacționa în cazul în care intrarea de RESET este activă. O caracteristică deosebită a acestui bloc este că prin setarea parametrului **Par** pe stare **ON**, în cazul întreruperii alimentării cu energie electrică a PLC-ului, la realimentare starea ieșirii va fi cea setată înainte de întreruperea alimentării.

### Funcția „Pulse relay”

Un semnal scurt apărut la borna **Trg** va determina comutarea stării ieșirii **Q**. Intrarea **R** determină trecerea ieșirii în starea **OFF**.

 <p>Poziția 3 în meniul <b>Logo!</b></p>	Input Trg	Intrarea <b>Trg</b> comută ieșirea <b>Q</b> în stare <b>ON</b> sau <b>OFF</b>
	Input R	Intrarea <b>R</b> resetează ieșirea <b>Q</b>
	Parametrul Par	Parametrul <b>Par</b> determină activarea memorării stării ieșirii <b>Q</b> în cazul întreruperii alimentării PLC-ului

	Output Q	Ieșirea <b>Q</b> devine 1 logic la apariția unui semnal la intrarea <b>Trg</b> și devine 0 logic la apariția unui nou semnal la intrarea <b>Trg</b>
--	----------	---

Diagrama de funcționare

Funcționarea blocului poate fi urmărită pe diagrama de timp din figura de mai jos:

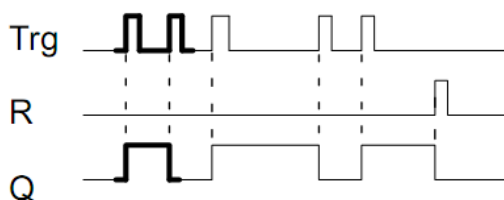


Fig. 2.33 – Diagrama de timp pentru funcția „Pulse relay”

Comutarea semnalului de trigger din 0 în 1 logic determină schimbarea stării ieșirii **Q**, iar intrarea **R** resetează ieșirea **Q** în starea **OFF**. Ca și în cazul blocului anterior, prin setarea parametrului **Par** pe stare **ON**, în cazul dispariției alimentării cu energie electrică a PLC-ului, la realimentare starea ieșirii va fii cea setată ultima dată.

**Funcția „Wiping relay”**

La apariția unui semnal pe borna de **Trg** ieșirea **Q** va trece în stare **ON** pentru o durată de timp setată în parametrul **T**.

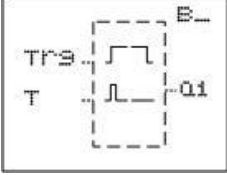
 <p>Poziția 9 în meniul <b>Logo!</b></p>	Input Trg	Intrarea <b>Trg</b> comută ieșirea <b>Q</b> în stare <b>ON</b> și inițializează cronometrul
	Input T	Setează timpul după care ieșirea <b>Q</b> revine în starea inițială
	Output Q	Ieșirea <b>Q</b> devine 1 logic la apariția unui semnal la intrarea <b>Trg</b> și devine 0 logic la expirarea timpului setat în parametrul <b>T</b>

Diagrama de funcționare

Funcționarea blocului poate fi urmărită pe diagrama de timp din figura de mai jos:

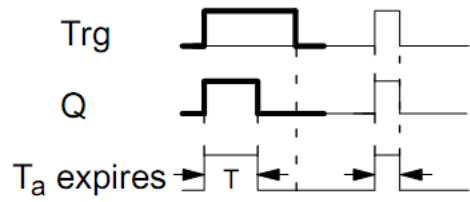
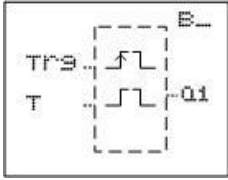


Fig. 2.34 – Diagrama de timp pentru funcția „Wiping relay”

Comutarea semnalului de trigger din 0 în 1 logic determină schimbarea stării ieșirii **Q** în **ON**, iar după expirarea timpului setat în cadrul parametrului **T** ieșirea **Q** revine în starea **OFF**. În cazul în care durata semnalului de trigger este mai mică decât timpul setat **T**, la dispariția semnalului de la borna **Trg** ieșirea va reveni la starea inițială.

### Funcția „Edge-triggered Wiping relay”

La apariția unui impuls pe borna de **Trg** ieșirea **Q** va trece în stare **ON** pentru o durată de timp setată în parametrul **T**.

 <p>Poziția 18 în meniul <b>Logo!</b></p>	Input Trg	Intrarea <b>Trg</b> comută ieșirea <b>Q</b> în stare <b>ON</b> și inițializează cronometrul
	Input T	Setează timpul după care ieșirea <b>Q</b> revine în starea inițială
	Output Q	Ieșirea <b>Q</b> devine 1 logic la apariția unui semnal la intrarea <b>Trg</b> și devine 0 logic la expirarea timpului setat în parametrul <b>T</b>

### Diagrama de funcționare

Funcționarea blocului poate fi urmărită pe diagrama de timp din următoarea figură:

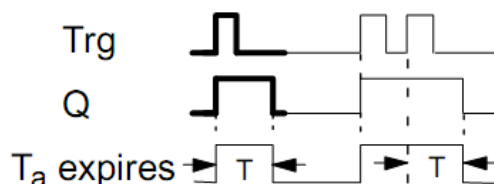


Fig. 2.35 – Diagrama de timp pentru funcția „Edge-triggered Wiping relay”

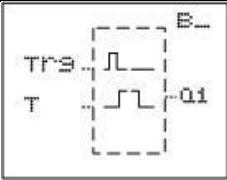
Comutarea semnalului de trigger din 0 în 1 logic determină schimbarea stării ieșirii **Q**, iar după expirarea timpului setat în cadrul parametrului **T** ieșirea **Q** revine în starea **OFF**. Spre deosebire de funcția prezentată anterior semnalul de trigger nu



trebuie menţinut activ, fiind suficient un impuls pentru declanşarea cronometrului şi comutarea ieşirii pe starea **ON**. Timpul **T** este resetat dacă la bornele intrării **Trg** apare un nou impuls, fără ca ieşirea **Q** să îşi schimbe starea.

### Funcția „Stairway Light switch”

Semnalul de intrare de la borna **Trg** determină comutarea ieşirii **Q** pe stare **ON** şi inițializarea cronometrului. La expirarea timpului setat în parametrul **T** ieşirea va trece în starea **OFF**. Cu 15 secunde (sau altă valoare, în funcție de baza de timp aleasă) înainte de expirarea timpului o avertizare este generată.

 <p>Poziția 16 în meniul <b>Logo!</b></p>	Input Trg	Intrarea <b>Trg</b> comută ieşirea <b>Q</b> în stare <b>ON</b> şi inițializează cronometrul
	Input T	Setează timpul după care ieşirea <b>Q</b> comută pe starea <b>OFF</b> ; implicit timpul este setat în minute.
	Output Q	Ieşirea <b>Q</b> devine 1 logic la apariția unui semnal la intrarea <b>Trg</b> şi devine 0 logic la expirarea timpului setat în parametrul <b>T</b>

### Diagrama de funcționare

Funcționarea blocului poate fi urmărită pe diagrama de timp din figura de mai jos:

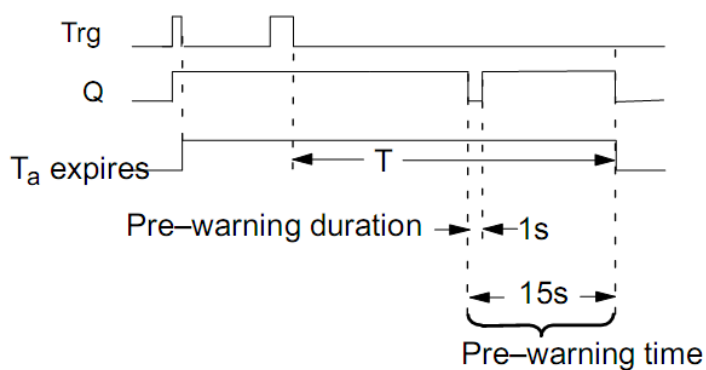


Fig. 2.36 – Diagrama de timp pentru funcția „Stairway light switch”

Apariția unui impuls pe borna de intrare **Trg** determină comutarea ieșirii **Q** în stare **ON**, iar disipația acestui impuls duce la inițializarea cronometrului. În momentul în care timpul atinge valoarea setată în parametrul **T** ieșirea va comuta pe stare **OFF**. Dacă înainte de expirarea timpului **T** apare un nou impuls pe borna de trigger timpul este resetat. Cu 15 secunde înainte de expirarea timpului setat ieșirea va comuta pentru 1 secundă pe stare **OFF**, revenind ulterior pe stare **ON**. Durata acestui impuls și momentul când apare poate fi modificat în funcție de baza de timp setată:

Baza de timp	Momentul când apare avertizarea	Durata avertizării
Secunde	750 msec	50 msec
Minute	15 sec	1 sec
Ore	15 min	1 min

**Funcția „Multifunctional (confort) switch”**

Blocul poate executa două funcții: întrerupător de tip „pulse switch” cu funcție de **OFF** delay sau întrerupător clasic.

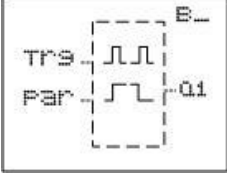
 <p>Poziția 17 în meniul <b>Logo!</b></p>	Input Trg	Intrarea <b>Trg</b> comută ieșirea <b>Q</b> între stările <b>ON</b> și <b>OFF</b> în funcție de parametrii setați
	Input Par	<p>- <b>TH</b> care determină întârzierea cu care ieșirea <b>Q</b> este comutată pe stare <b>OFF</b> la trecerea din 1 în 0 logic a semnalului de trigger</p> <p>- <b>TL</b> durata cât triggerul trebuie să devină activ astfel ca ieșirea să rămână activă până la un nou semnal de trigger</p>
	Output Q	Ieșirea <b>Q</b> este activată de semnalul de trigger și în funcție de durata acestui semnal va devenii <b>OFF</b> după expirarea <b>TH</b> sau la un nou impuls pe borna <b>Trg</b>

Diagrama de funcționare

Funcționarea blocului poate fi urmărită pe diagrama de timp din figura următoare:

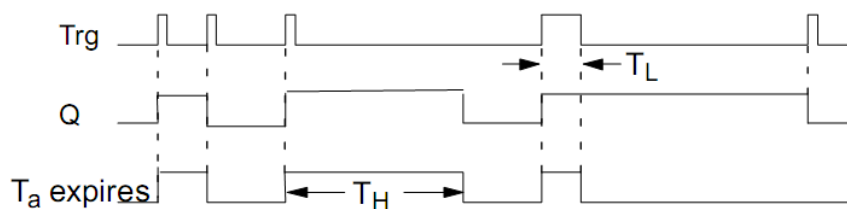
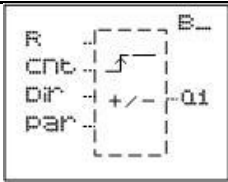


Fig. 2.37 – Diagrama de timp pentru funcția „Multifunctional switch”

La apariția unui semnal pe intrarea **Trg** ieșirea **Q** va deveni **ON** iar la următorul impuls, la trecerea din 1 în 0 logic, se inițializează timerul. La expirarea **TH** ieșirea **Q** comută pe starea **OFF**. În cazul în care **Q** este **OFF** și durata semnalului de trigger este mai mare decât parametrul **TL** setat ieșirea **Q** devine **ON** și rămâne în această stare până la apariția unui nou semnal pe intrarea **Trg**.

### Funcția „UP/DOWN Counter”

Blocul execută o funcție de numărare a semnalelor de intrare primite la borna **Cnt**. În cadrul blocului poate fi setată direcția de numărare precum și valoarea la care ieșirea să devină **ON**.

 <p>Poziția 10 în meniul <b>Logo!</b></p>	Input R	Intrarea <b>R</b> resetează valoarea contorului la zero
	Input Cnt	Trecerile din 0 în 1 logic sunt considerate ca intrări pentru blocul de numărare, trecerile din 1 în 0 logic nefiind numărate. Frecvența maximă de măsurare este 5 Hz
	Input Dir	În cazul în care valoarea acestei

		intrări este 0 logic contorul va fi incrementat la fiecare semnal pe borna <b>Cnt</b> , iar dacă <b>Dir</b> este 1 logic valoarea contorului scade la fiecare semnal <b>Cnt</b>
	Parametrul Par	În cadrul parametrului <b>Par</b> se poate seta limita contorului (maxim 999999), depășirea valorii determinând comutarea stării ieșirii <b>Q</b> . În același timp se poate activa și funcția de memorare în cazul întreruperii alimentării
	Output Q	Ieșirea <b>Q</b> este activată de depășirea în sens pozitiv a valorii setate în cadrul parametrului <b>Par</b> ; în cazul în care ieșirea e <b>ON</b> și valoarea contorului scade sub valoarea setată în <b>Par</b> ieșirea va deveni <b>OFF</b>

Diagrama de funcționare

Funcționarea blocului poate fi urmărită pe diagrama de timp din figură:

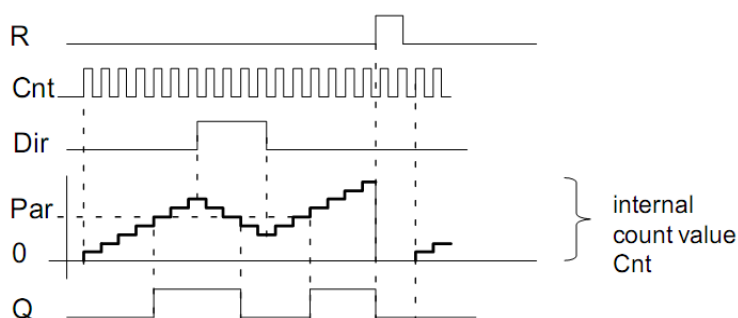
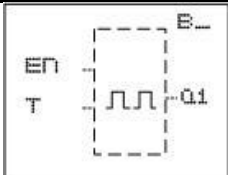


Fig. 2.38 – Diagrama de timp pentru funcția „UP/Down counter”

Fiecare trecere din 0 în 1 logic de pe intrarea **Cnt** reprezintă o modificare a contorului, sensul în care se face modificarea fiind dat de valoare intrării **Dir**: 0 logic determină creștere a contorului, 1 logic o descreștere a valorii contorului. O valoare a contorului mai mare sau egală decât valoarea setată în parametrul **Par** determină trecerea ieșirii **Q** în stare **ON**, iar valori ale contorului mai mici decât valoarea **Par** setează starea **OFF** pentru ieșirea **Q**. În cadrul **Par** se poate seta și starea parametrului **Rem** care activează memorarea stării sistemului la alimentarea survenită după o întrerupere a tensiunii de alimentare.

### Funcția „Symmetric Clock Generator”

Blocul permite realizarea unei unde simetrice la ieșire, lățimea impulsurilor putând fi setată.

	Input En	Intrarea <b>En</b> activează / dezactivează funcția de generare a semnalului simetric
	Parametrul T	Parametrul <b>T</b> determină durata cât semnalul de la ieșire este în stare <b>ON</b> sau <b>OFF</b> .

Poziția 6 în meniul **Logo!**

	Output Q	Ieşirea <b>Q</b> este comutată între starea <b>ON</b> şi <b>OFF</b> durata impulsurilor fiind dată de valoarea parametrului <b>T</b> .
--	----------	--

Diagrama de funcţionare

Funcţionarea blocului poate fi urmărită pe diagrama de timp din figura de mai jos:

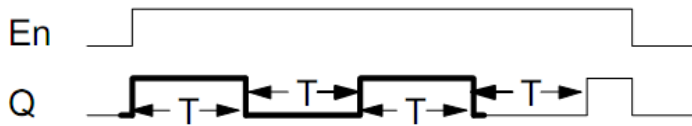


Fig. 2.39 – Diagrama de timp pentru funcţia „Symmetric clock generator”

Trecea intrării **En** din 0 în 1 logic determină comutarea ieşirii **Q** în stare **ON**, pentru durata setată în parametrul **T**. După expirarea acestui timp ieşirea va comuta pe starea **OFF**, tot pentru durata **T**, ciclul repetându-se până la trecerea intrării **En** în stare 0 logic.

Funcţia „Asynchronous Pulse Generator”

Blocul permite realizarea unei unde nesimetrice la ieşire, prin setarea independentă a lăţimii impulsurilor şi a perioadei de pauză.

	Input En	Intrarea <b>En</b> activează / dezactivează funcţia de generare a semnalului simetric
	Input Inv	Această intrare permite inversarea

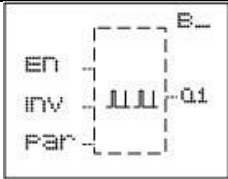
		perioadelor active și de pauză ale semnalului de ieșire
Poziția 12 în meniul <b>Logo!</b>	Parametrul Par	Prin alegerea <b>TH</b> și <b>TL</b> se setează lățimea impulsului și a pauzei
	Output Q	Ieșirea <b>Q</b> este comutată între starea <b>ON</b> și <b>OFF</b> durata impulsurilor fiind dată de valoarea parametrilor <b>TH</b> și <b>TL</b> .

Diagrama de funcționare

Funcționarea blocului poate fi urmărită pe diagrama de timp din figura următoare:

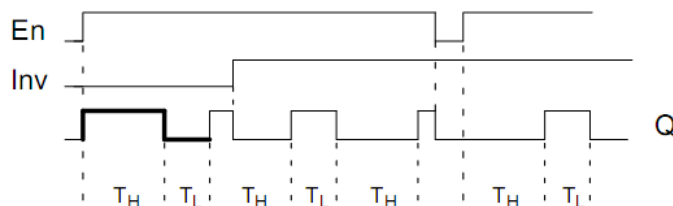


Fig. 2.40 – Diagrama de timp pentru funcția „Asynchronous pulse generator”

Trecerea intrării **En** din 0 în 1 logic determină comutarea ieșirii **Q** în stare **ON**, pentru durata setată în parametrul **TH**, urmată de o trecere în stare **OFF** pe durata setată în parametrul **TL**. În cazul în care intrarea **Inv** devine 1 logic, starea ieșirii **Q** se modifică, comportarea fiind similară cu cea obținută în urma introducerii unei negări. **TH** va determina astfel timpul de pauză, iar **TL** va determina lățimea impulsului pe ieșirea **Q**.



### Funcția „Random Generator”

Utilizând această funcție ieșirea este comutată în stare ON sau OFF cu o întârziere aleatorie.

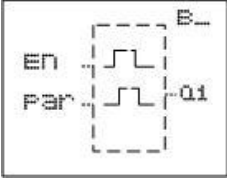
 <p>Poziția 15 în meniul <b>Logo!</b></p>	Input En	Trecerea din stare 0 în stare 1 logic a intrării <b>En</b> determină activarea întârzierii la pornire a funcției, iar trecere din stare 1 în 0 logic activează întârzierea la oprire a funcției
	Parametrul Par	Timpul aleatoriu de întârziere la pornire este între 0 și <b>TH</b> , iar timpul aleatoriu de întârziere la oprire este între 0 și <b>TL</b>
	Output Q	Ieșirea <b>Q</b> este comutată în starea <b>ON</b> după expirarea timpului < <b>TH</b> dacă intrarea <b>En</b> este încă activă, trecerea în stare <b>OFF</b> fiind realizată după expirarea < <b>TL</b> dacă intrarea <b>En</b> nu a devenit activă între timp.

Diagrama de funcționare

Funcționarea blocului poate fi urmărită pe diagrama de timp din figura 2.41:

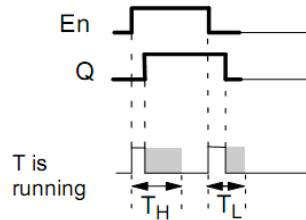
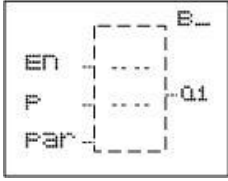


Fig. 2.41 – Diagrama de timp pentru funcția „**Random generator**”

La trecea intrării **En** din 0 în 1 logic se declanșează timpul de întârziere la pornire, având o valoare aleatorie între 0 și **TH**. Dacă în acest interval intrarea **En** nu redevine 0 ieșirea va fi comutată în stare **ON**. În cazul în care ieșirea este **ON** și intrarea **En** trece din 1 în 0 logic se declanșează timpul de întârziere la oprire, având o valoare aleatorie între 0 și **TL**. Dacă în acest interval nu apare un nou semnal pe intrare, la expirarea timpului ieșirea va comuta în stare **OFF**.

Funcția „Message Texts”

Funcția permite afișarea pe ecranul **Logo!** a unui mesaj în timpul rulării unui program.

 <p>Poziția 19 în meniul <b>Logo!</b></p>	Input En	Trecerea din stare 0 în stare 1 logic a intrării <b>En</b> determină afișarea mesajului pe ecranul <b>Logo!</b>
	Parametrul P	P reprezintă prioritatea mesajului
	Parametrul Par	În cadrul acestui parametru se

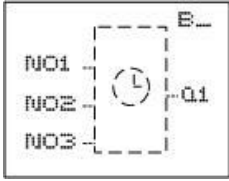
		introduce textul ce va fi afişat
	Output Q	Ieşirea <b>Q</b> este activă cât timp mesajul este afişat

### Descrierea funcţionării

La trecerea intrării **En** din stare 0 în 1 logic mesajul este afişat pe ecranul **Logo!**. În cazul în care sunt utilizate mai multe funcţii de acest tip, mesajul afişat pe ecran este cel care are prioritatea cea mai mare (între 0 şi 9). Dacă se doreşte vizualizarea celorlalte mesaje se utilizează tastele UP/DOWN. **Logo!** permite utilizarea a maxim 5 mesaje text simultan, unele variante utilizând parametrul **Ack** care, în cazul în care este activat, face ca mesajul să rămână activ şi după dispariţia semnalului **En**, până la apăsarea butonului **OK** de către utilizator, confirmându-se astfel citirea mesajului.

### Funcţia „Weekly Timer Switch”

Starea **ON** sau **OFF** a ieşirii este controlată în funcţie de ora şi ziua săptămânii.

 <p>Poziţia 4 în meniul <b>Logo!</b></p>	Parametrul No1, No2, No3	Funcţia prezintă posibilitatea de a seta starea ieşirii cu ajutorul a 3 canale pentru setarea zilelor şi a orelor de funcţionare
	Output Q	Ieşirea <b>Q</b> este comutată în stare <b>ON</b> când oricare dintre cele 3 canale este activat.

### Descrierea funcționării

În cadrul funcției pot fi setate momentele de pornire / oprire pentru fiecare din cele 3 canale disponibile. În momentul în care unul dintre canale devine activ ieșirea **Q** trece pe stare 1 logic, în situația în care un alt canal nu era deja activ. Trecerea ieșirii din stare 1 în stare 0 logic se face când toate canalele sunt inactice. În cazul în care momentele de pornire / oprire ale canalelor se suprapun și apar conflicte, prioritate o are starea canalului cu număr de ordine mai mare (3 față de 2 sau 1, 2 față de 1).

Configurarea unui canal se face într-o fereastră de dialog similară celei din figura de mai jos. Primul rând prezintă datele de identificare ale blocului curent și ale canalului pentru care se setează parametrii. Zilele săptămânii ce vor fi active pentru acest canal sunt alese pe rândul al doilea, zilele care nu sunt active fiind înlocuite de simbolul „-”. Ora de pornire și oprire sunt selectate pe rândurile 3, respectiv 4, iar în cazul în care nu se selectează o oră de pornire sau oprire se utilizează simbolurile „--:--”.

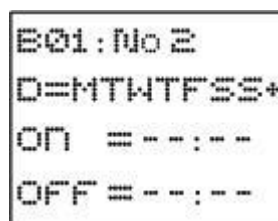
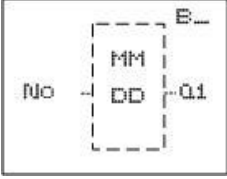


Fig. 2.42 – Setarea parametrilor în cadrul funcției „**Weekly Timer Switch**”

### Funcția „Yearly Timer Switch”

Starea **ON** sau **OFF** a ieșirii este controlată în funcție de data internă a automatului programabil.

	Parametrul No	În cadrul parametrului <b>No</b> se poate seta data de pornire / oprire
	Output Q	Ieșirea <b>Q</b> este comutată în starea <b>ON</b>

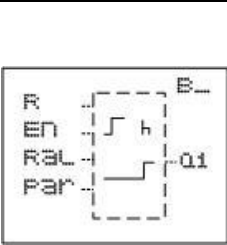
 <p>Poziția 13 în meniul <b>Logo!</b></p>		<p>la ora 00:00 la data setată în parametrul <b>No</b> este activat și trece în stare <b>OFF</b> la ora 00:00 setată în rubrica Off a blocului.</p>
--	--	---

### Descrierea funcționării

În cadrul funcției se poate seta data la care ieșirea **Q** să treacă în stare **ON** respectiv **OFF**. În cadrul parametrului **No** se va seta luna și ziua pentru pornirea și oprirea ieșirii.

### Funcția „Operating Hours Meter”

O funcție de contorizare a timpului în momentul în care intrarea este în stare 1 logic. Ieșirea devine **ON** la expirarea limitei de timp setate în cadrul blocului.

 <p>Poziția 8 în meniul <b>Logo!</b></p>	<p>Input R</p>	<p>Măsurarea timpului se execută cât timp <b>R</b> este în stare 0 logic.</p>
	<p>Input En</p>	<p><b>En</b> este intrarea care va fi monitorizată. <b>Logo!</b> va măsura timpul cât această intrare este setată</p>
	<p>Input Ral</p>	<p>Măsurarea timpului se execută cât timp <b>Ral</b> nu este în stare 1 logic. Această intrare are semnificația de Reset all, trecând ieșirea <b>Q</b> în stare <b>OFF</b>, timpul măsurat</p>

		fiind 0, iar timpul limită este reinițializat
	Parametrul Par	Timpul <b>MI</b> reprezintă intervalul de mentenanță setat, în ore, putând lua valoarea maximă de 9999 ore
	Output Q	Ieșirea devine <b>ON</b> când timpul atinge valoarea <b>MI</b> setată în parametrul <b>Par</b>

**MI** – valoarea setată pentru contor

**MN** – timpul rămas

**OT** – timpul trecut de la ultima valoare 1 logică a intrării **Ral**

Diagrama de funcționare

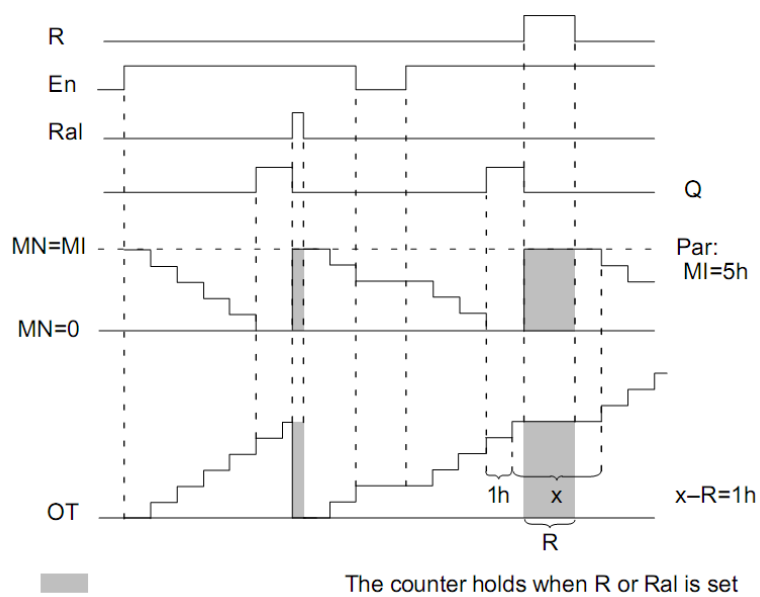
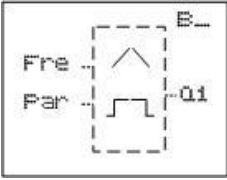


Fig. 2.43 – Diagrama de timp pentru funcția „**Operating Hours Meter**”

Blocul monitorizează orele cât timp intrarea **En** este activă, determinând timpul trecut și valoare rămasă până la atingerea valorii setate în **MN**. Ieșirea **Q** devine 1 logic în momentul în care valoarea **MN** ajunge la 0. Intrarea **R** este utilizată pentru resetarea stării ieșirii **Q** și a contorului de timp rămas, dar contorul **OT** continuă înregistrarea duratei de funcționare. Intrarea **Ral** resetează în plus și contorul intern **OT**. Unele modele **Logo!** permit afișarea valorii momentane a **MN** și **OT** în timpul rulării programului

### Funcția „Threshold trigger”

Starea **ON** sau **OFF** a ieșirii este controlată în funcție de frecvența semnalului conectat la borna **Fre**.

 <p>Poziția 11 în meniul <b>Logo!</b></p>	Parametrul Fre	<p>La intrarea <b>Fre</b> se conectează semnalul a cărui impulsuri trebuie monitorizate.</p> <p>Unele modele <b>Logo!</b> au intrări care permit realizare de numărări la frecvențe ridicate, de până la 1kHz</p>
	Parametrul Par: SW ↑, SW ↓ , G_T	<p>SW ↑ limita de comutare <b>ON</b></p> <p>SW ↓ limita de comutare <b>OFF</b></p> <p>G_T intervalul de timp cât se face măsurarea intrării <b>Fre</b></p>
	Output Q	Ieșirea <b>Q</b> este comutată în starea <b>ON</b>

		sau <b>OFF</b> în funcție de valorile SW ↑ și SW ↓.
--	--	---

### Diagrama de funcționare

Frecvența semnalului de intrare de la borna **Fre** este monitorizat pentru perioada de timp setată prin **G\_T**. Starea ieșirii **Q** este dată de rezultatul comparării semnalului de intrare cu parametrii SW ↑ și SW ↓. În cazul în care **G\_T** este 1 secundă, rezultatul măsurării este dat în Hz.

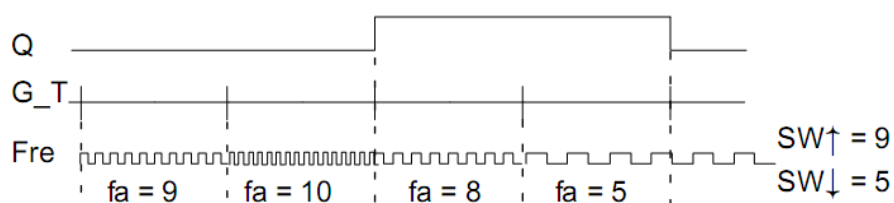


Fig. 2.44 – Diagrama de timp pentru funcția „Threshold trigger”


## 2.5 Logo! Soft Comfort

Pentru a crește eficiența și confortul utilizatorilor **Logo!**, Siemens oferă programul **LOGO! Soft Comfort** care oferă posibilitatea realizării programării automatelor programabile utilizând o interfață grafică și transferul schemelor bloc în memoria acestora.

### 2.5.1 Transferul de date între Logo! și PC

Transferul de date către și dinspre automatul programabil se poate face prin mai multe metode, cele mai uzuale fiind conectarea de cabluri speciale (pe portul USB sau RS232 al calculatorului), conectarea la rețele Ethernet sau utilizarea de memorii portabile. Modelul **Logo!230RC** oferă doar posibilitatea conectării printr-un cablu furnizat de către Siemens la portul RS232 sau USB al calculatorului.



În momentul conectării acestui cablu la automatul programabil pe ecranul acestuia va apărea un mesaj care avertizează utilizatorul că **Logo!** este conectat la calculator și este pregătit pentru transferul de date. În cazul nu se trece automat în acest mod de lucru se poate selecta manual din meniu opțiunea „PC/CARD”, respectiv „PC ↔ ”.

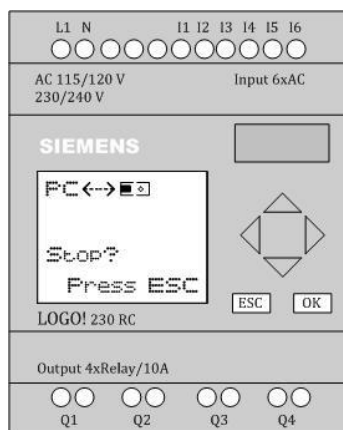


Fig. 2.45 – Interfața **Logo!** la conectarea cablului de date

### Interfața **Logo! Soft Comfort**

Interfața programului Logo! Soft Comfort este prezentată în figura de mai jos, principalele elemente fiind: 1 – Bara de meniu, 2 – Bara de unelte, 3 – Fereastra pentru programare, 4 – Fereastra de informare, 5 – Bara de status, 6 – Funcții de bază și speciale, 7 – Uneltele pentru programare.

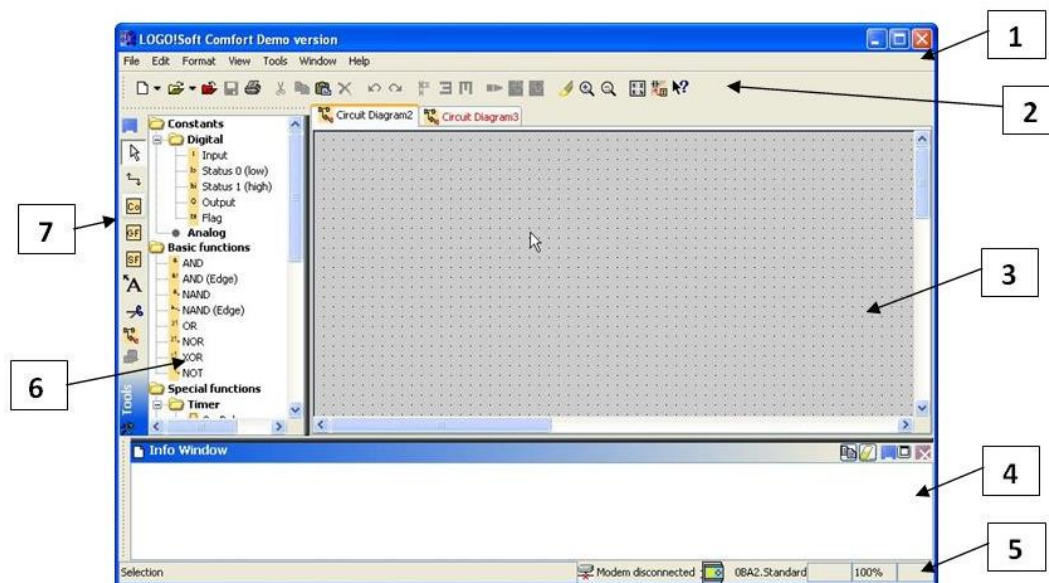


Fig. 2.46 – Interfața Logo!Soft Comfort

### 1 Bara de meniu

Bara de meniu conține elementele uzuale regăsite în majoritatea programelor: File, Edit, Format, View, Tools, Window și Help. Meniurile și submeniurile permit manipularea fișierelor și a blocurilor, conectarea la automatul programabil, customizarea interfeței grafice sau accesarea documentației puse la dispoziție de producător.

### 2 Bara de unelte

Bara de unelte oferă acces rapid la cele mai uzuale comenzi disponibile în Logo! Soft Comfort: New File, Open File, Close File, Save File și Print. În același timp se pot accesa funcțiile de decupare, copiere, lipire și ștergere (Cut, Copy, Paste și Delete) a elementelor diagramelor bloc, precum și comenzile Undo și Redo, ce permit anularea ultimelor comenzi, respectiv refacerea comenzilor anulate deja. Pe lângă aceste comenzi, regăsite în majoritatea programelor disponibile pe piață, în bara de unelte a Logo! Soft Comfort se regăsesc 3 butoane ce permit alinierea automată, respectiv pe verticală și orizontală a elementelor diagramei bloc, lucru

extrem de util pentru realizarea unor programe ușor de urmărit. Următorul grup de butoane realizează interacțiunea cu automatul programabil conectat la calculator, putându-se comuta **Logo!** în modul de rulare a programului sau stand-by, precum și transferul de programe între PC și PLC (download), respectiv PLC și PC (upload)



Fig. 2.47 – Bara de unelte

În continuare bara de unelte conține butoane pentru identificarea liniilor ce sunt conectate la un bloc selectat, zoom in și zoom out sau setarea formatului paginii de lucru și a numărului de pagini. Logo! Soft Comfort permite comutarea între diagramă bloc și diagrama Ladder, în funcție de preferințele utilizatorului, precum și posibilitatea accesării rapide a meniului de ajutor (Help) prin utilizarea funcției de ajutor contextual.

### 3 Fereastra pentru programare

Reprezintă zona în care se realizează diagrama bloc, existând posibilitatea inserării rapide de blocuri prin accesarea unui sub-meniu prin realizarea unui ”click dreapta” pe această zonă. Sub-meniu oferă acces rapid la principalele funcții și categorii de blocuri disponibile, precum și posibilitatea căutării rapide a blocurilor în diagrama bloc realizată.

### 4 Fereastra de informare

Este localizată sub fereastra pentru programare, fiind zona unde se afișează mesajele de eroare apărute, automatele programabile conectate la calculator, ora și data afișării mesajului și numele programului pentru care se afișează mesajul, în cazul în care sunt deschise mai multe programe în același timp. În momentul rulării unei simulări, resursele și **Logo!**-ul utilizat sunt analizate, numele acestora și eventuale erori fiind afișate.

### 5 Bara de status

Bara de status este localizată în partea de jos a ferestrei programului, fiind împărțită în 5 secțiuni, oferind informații rapide utilizatorului referitor la unele activă în acel moment, modelul automatului **Logo!** conectat, starea programului, nivelul zoom-ului și numărul paginii pentru diagrame bloc mari



Fig. 2.48 – Bara de status

### 6 Paleta de funcții de bază și speciale

În această zonă a interfeței Logo! Soft Comfort sunt regăsite bibliotecile cu intrările analogice și digitale utilizabile, funcțiile de bază și cele speciale, grupate pe categorii.

### 7 Unelte pentru programare

Conține icoane pentru accesul rapid al celor mai uzuale unelte necesare creării, editării și testării diagramei bloc realizate. În funcție de icoana selectată cursorul și operațiile efectuate de mouse se modifică corespunzător.



Fig. 2.49 – Bara de unelte

Elementele din această bară de unelte se regăsesc și în meniul accesibil direct pe fereastra de programare, prima icoană trecând mouse-ul în modul de selectare a diferitelor elemente ale diagramei bloc realizate. Pentru realizarea de conexiuni între blocurile diagramei se va utiliza unealta de conectare, cea de-a doua icoană din bara de unelte. În continuare se regăsesc icoane ce permit inserarea de conectori, funcții de bază și speciale, precum și de etichete sau comentarii în

diagrama bloc. În cazul diagramelor bloc care conţin un număr mare de elemente, pentru evitarea încărcării excesive a programului există posibilitatea întreruperii conexiunilor între două blocuri prin utilizarea unelei „Cut / Join”, conexiunea fiind înlocuită de 2 etichete care conţin numele blocurilor corespondente, inclusiv numărul paginii unde se regăseşte blocul, numărul blocului şi portul unde este realizată conexiunea. Ultimele icoane realizează trecerea în modul de simulare a programului realizat, respectiv testarea online în cazul în care un PLC este conectat la calculator.

La activarea modului de simulare a diagramei bloc realizate, pe interfaţa Logo! Soft Comfort apare o nouă bară, conţinând elementele necesare realizării rulării programului. În funcţie de numărul de blocuri de intrare şi ieşire utilizate în cadrul diagramei bloc pe bara de simulare vor apărea întrerupătoare şi becuri care reprezintă intrările, respectiv ieşirile, utilizatorul putând modifica starea intrărilor pentru a verifica funcţionarea programului. În acelaşi timp se poate simula întreruperea alimentării cu energie a automatului programabil şi se poate porni, opri şi pune pe pauză programul. Pentru faza de depanare a diagramei bloc, foarte utilă este funcţia de rulare a programului pentru un număr limitat de ori, putând fi selectate cicluri, secunde, minute sau ore de rulare.



Fig. 2.50 – Interfaţa pentru simulare

În timpul rulării programului culoarea cu care sunt reprezentate conexiunile între blocurile diagramei se va modifica în funcţie de prezenţa sau absenţa unui semnal, permiţând astfel urmărirea modului de funcţionare a programului.

## Exemple de programe realizate în Logo!Soft Comfort

### 2.5.2 Automatizarea unui sistem de irigat al unei sere

Se va folosi **Logo!** pentru realizarea unui sistem de irigat într-o seră în care sunt plantate trei categorii de plante, fiecare necesitând o cantitate diferită de apă. Prima categorie de plante sunt plante acvatice care se află într-un rezervor al cărui nivel trebuie păstrat între anumite limite. Plantele din a doua categorie sunt irigate doar atunci când nivelul umidității din sol scade sub o anumită limită, iar cele din ultima categorie valoare trebuie udate în fiecare dimineață și seară câte 3 minute. Va exista un întrerupător general care să permită oprirea întregului sistem de irigat atunci când se dorește acest lucru.

Pentru realizarea practică a acestei aplicații se va întocmi o listă cu echipamentele necesare și se va determina modul de realizare a programării cu ajutorul funcțiilor de bază și speciale disponibile în cadrul **Logo!**.

Componente utilizate:

Componenta	Conectarea la <b>Logo!</b>
Contact cu plutitor pentru nivelul maxim al apei (contact NO)	I1
Contact cu plutitor pentru nivelul minim al apei (contact NO)	I2
Senzor de umiditate (contact NO)	I3
Contact pentru pornirea/oprirea controlului automat (contact NO)	I4
Electro-valvă pentru irigarea plantelor de tipul 1	Q1
Electro-valvă pentru irigarea plantelor de tipul 2	Q2
Electro-valvă pentru irigarea plantelor de tipul 3	Q3
<b>Logo! 230 RC</b>	

Strategii de programare:

Plante de tipul 1: Nivelul de apă în bazin va fi păstrat între limitele impuse de contactele conectate la intrările I1 și I2.

Plante de tipul 2: plantele voi fi irigate pentru 2 minute atunci când nivelul de umiditate din sol va scădea sub pragul setat și senzorul de umiditate va emite un impuls.

Plante de tipul 3: cu ajutorul unui bloc „Weekly timer” se va seta pornirea zilnică a sistemului de irigații în fiecare dimineață între 6:00 și 6:03 și seara între 8:00 și 8:03.

Avantaje ale utilizării **Logo!**:

Durata de irigare poate fi modificată independent pentru fiecare tip de plantă, iar pe blocurile de intrare și ieșire rămase libere se poate realiza automatizarea sistemului de iluminat și ventilație.

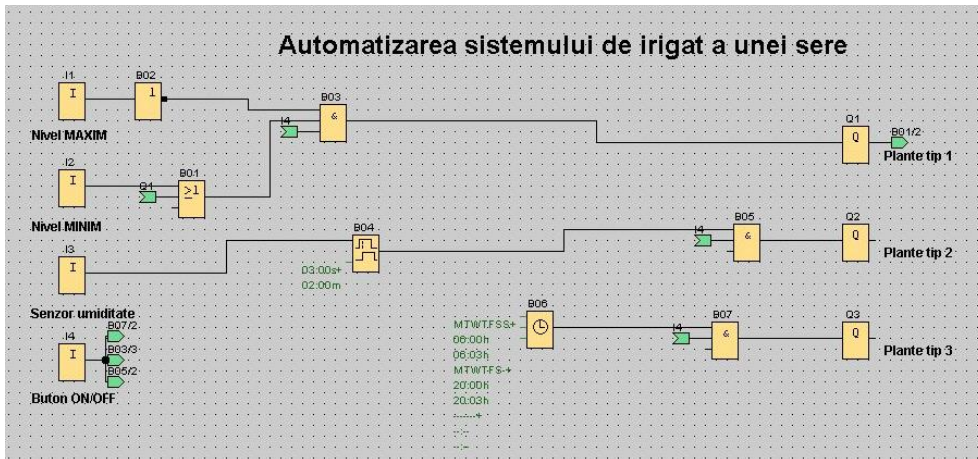


Fig. 2.51 – Diagrama bloc

Pentru realizarea diagramei bloc prezentate mai sus se vor parcurge o serie de etape, prezentate în cele ce urmează. După pornirea programului **Logo! Soft Comfort** se va deschide un fișier nou, existând posibilitatea de a alege realizarea noului program sub formă de „Block diagram” sau „Ladder diagram”. Prin alegerea primei opțiuni se va realiza programul sub formă de diagramă bloc, specifică programării pe **Logo!**.



Fig. 2.52 – Deschiderea unui program nou

Logo! Soft Comfort va deschide automat o fereastră unde se pot seta proprietățile documentului proaspăt realizat, fiind disponibile câmpuri pentru numele celui care realizează diagrama bloc, numele proiectului și al instalației pentru care se realizează programul, numele clientului, numărul diagramei, persoana care a verificat programul, compania și numărul versiunii curente a programului. Aceste informații sunt extrem de utile în cadrul unei companii unde numărul de proiecte realizate este mare, permițând gestionarea eficientă a fișierelor. Totuși, în cazul programului realizat în acest exemplu nu este necesară completarea tuturor câmpurilor, putându-se trece rapid la următoarea etapă.



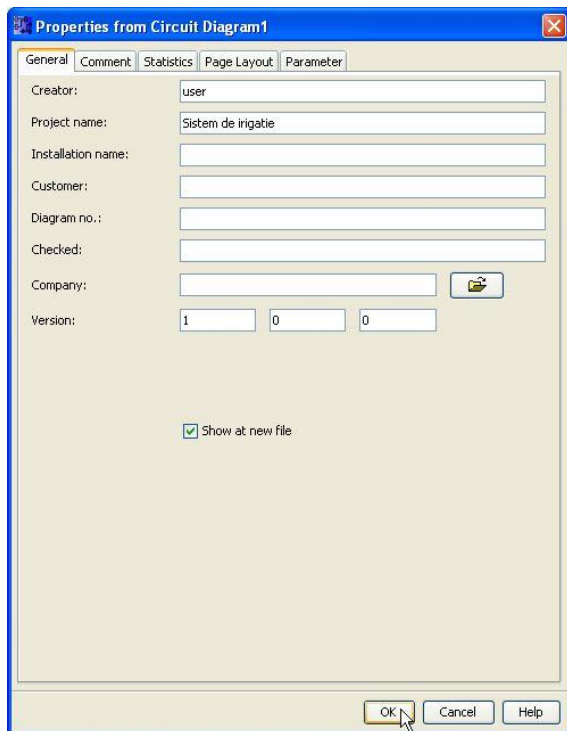


Fig. 2.53 – Proprietățile unui program nou

Pe baza listei de componente și a strategiei de programare realizate anterior se poate trece la realizarea diagramei bloc. Vor fi folosite 4 din cele 6 intrări digitale ale **Logo!**, primele două fiind conectate la senzorii de nivel necesari automatizării pentru primul tip de plante considerat, a treia intrare fiind conectată la senzorul de umiditate, iar a patra intrare fiind păstrată pentru comutatorul ON/OFF general. Pentru inserarea în cadrul diagramei a blocurilor de intrare se alege din cadrul toolbar-ului blocul cu conectori, în partea de jos a interfeței de programare fiind disponibile toate tipurile de blocuri de intrare/ieșire specifice **Logo!**. Alegând opțiunea Input, la fiecare click în zona de realizare a diagramei bloc se va insera câte un bloc de intrare, pornind de la intrarea cu numărul cel mai mic neutilizată, numerotarea făcându-se incremental, mergând până la intrarea cu numărul cel mai mare disponibilă, 6 în cazul **Logo! 230RC**. Pentru realizarea programului propus se vor insera 4 blocuri de intrare, dispuse în partea din stânga a ecranului.

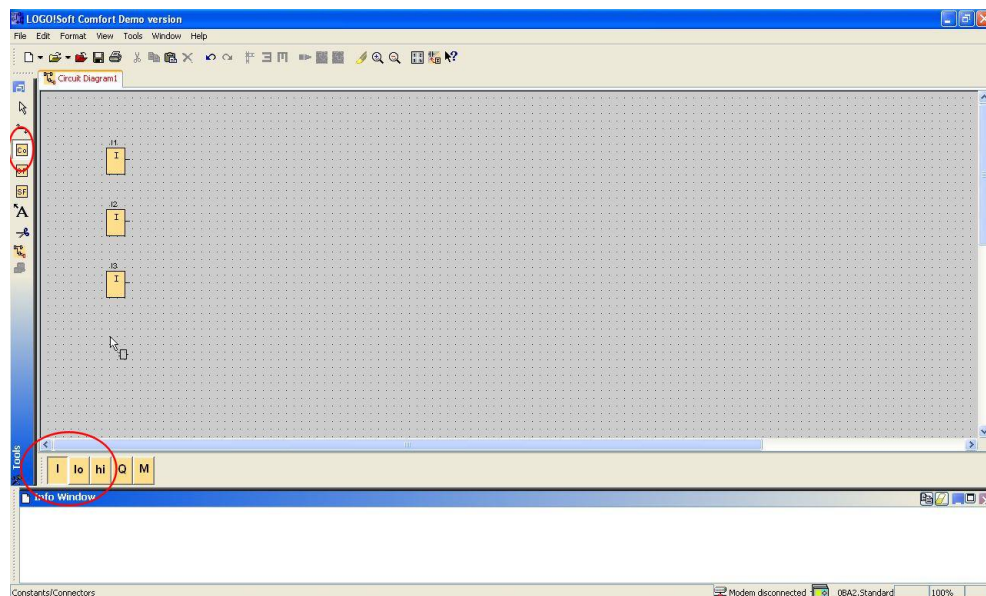


Fig. 2.54 – Modul de inserare a blocurilor de intrare

În mod similar se vor introduce și cele 3 blocuri de ieșire necesare, alegându-se blocul de Output din cadrul conectorilor disponibili. Cele 3 blocuri de ieșire se vor dispune în partea din dreapta a zonei de lucru, lăsându-se suficient spațiu între blocurile de intrare și cele de ieșire pentru a se putea introduce toate blocurile necesare realizării întregului program. Pentru a urmării mai ușor funcționarea diagramei bloc în cadrul programului se pot introduce blocuri de comentariu, utile pentru etichetarea intrărilor și a ieșirilor sau pentru cosmetizarea programului.

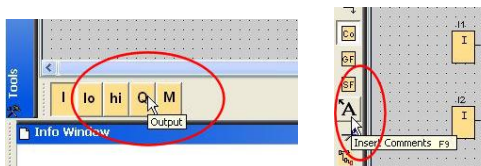


Fig. 2.55 – Modul de inserare a blocurilor de ieșire și a comentariilor

Diagrama bloc poate fi împărțită în 3 ramuri, pentru fiecare ieșire în parte, singura conexiune între ele fiind utilizarea comună a butonului de pornire/oprire conectat la intrarea I4. Pentru utilizatorii neexperimentați se recomandă realizarea în prima fază a unei scheme logice pentru determinarea modalității corecte de abordare a problemei, doar după finalizarea acesteia trecându-se la implementarea efectivă în cadrul **Logo! Soft Comfort** cu ajutorul blocurilor disponibile. Pentru realizarea diagramei bloc pentru automatizarea irigației primei categorii de plante se vor utiliza doar funcții de bază, disponibile prin apăsarea taste **F7** sau prin activarea din toolbar a icoanei **GF**. Cele 8 funcții de bază specifice **Logo!**, prezentate anterior, devin disponibile sub formă de icoane în partea de jos a zonei de programare.

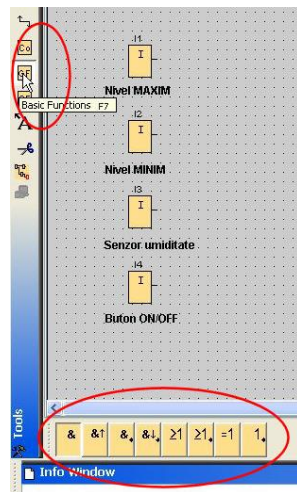


Fig. 2.56 – Inserarea unei funcții de bază

Intrarea **I2** se va conecta la prima intrare a unui bloc de tip **SAU**, pentru acest lucru utilizându-se unealta de conectare, disponibilă prin apăsarea tastei **F5** sau prin selectarea icoanei corespunzătoare din toolbar. Pentru conectarea a două blocuri se va realiza click pe intrarea sau ieșirea unui bloc și, păstrând butonul mouse-ului apăsat, se va muta cursorul pe intrarea/ieșirea blocului corespondent. La eliberarea butonului de la mouse se va realiza automat conexiunea între cele două blocuri, căutându-se calea cea mai scurtă între blocuri, fiind utilizate doar linii orizontale și verticale. Pentru transformarea contactului cu plutitor pentru nivelul maxim din bazin conectat la intrarea **I1**, dintr-un contact **NO** într-un contact **NC** se va utiliza un bloc de negare, disponibil tot în cadrul funcțiilor de bază.

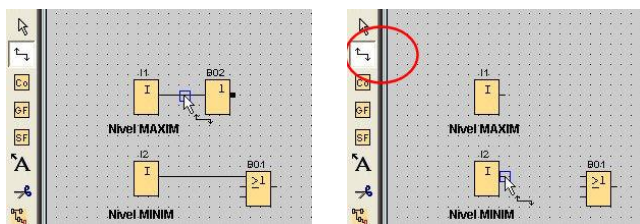


Fig. 2.57 – Conectarea blocurilor

Pentru că în cadrul cerințelor generale de realizare a automatizării s-a prevăzut un buton general (conectat la intrarea **I4**) de pornire/oprire a întregului sistem, este necesară introducerea unui bloc de tip **ȘI** unde se vor conecta ieșirile din blocurile **B02**, **B01** și **I4**. După cum se poate observa în imaginea alăturată, conexiunea realizată între **I4** și intrarea **3** a **B03** ocupă o zonă importantă a diagramei bloc, împiedicând dezvoltarea ulterioară a programului. Pentru a „aerisi” schema se va utiliza unealta „Cut/Join Connections” disponibilă în paleta de unelte sau prin apăsarea tastei **F11**. La realizarea unui click pe conexiunea dintre cele două blocuri aceasta se va întrerupe, în locul ei apărând două elemente noi, în apropierea bornelor blocurilor între care se face conexiunea. Fiecare dintre aceste elemente va conține o etichetă arătând numele blocului corespunzător și al intrării unde se realizează conexiunea. În acest mod zona centrală a diagramei bloc rămâne disponibilă pentru funcțiile ce vor fi utilizate pentru celelalte ramuri ale programului.

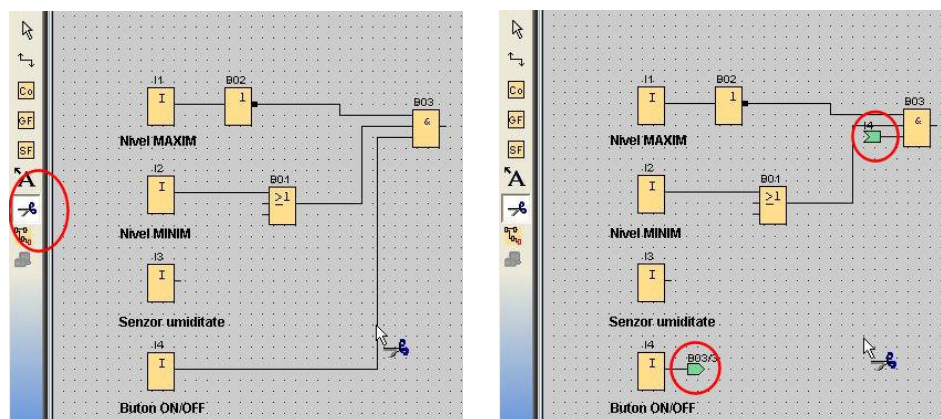


Fig. 2.58 – Modalitatea de întrerupere a unui fir de conexiune

Pentru ca sistemul să funcționeze automat și după ce impulsul dat de senzorul de minim dispare, este necesară realizarea unei bucle de automenținere. Pentru aceasta se va utiliza ca intrare în blocul **B01** de tip **SAU** ieșirea **Q1**. Prin acest lucru se asigură funcționarea pompei conectate la ieșirea **Q1** în mod continuu până la întreruperea

semnalului dat de butonul general **ON/OFF** sau la trecerea din stare 0 logic în stare 1 logic a senzorului de nivel conectat la intrarea **I1**. În mod similar celor prezentate anterior, pentru a simplifica diagrama bloc, conexiunea între **B01** și ieșirea **Q1** se întrerupe.

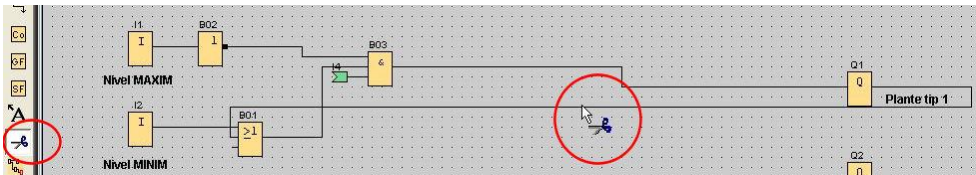


Fig. 2.59 – Realizare feed-back-ului pentru starea ieșii **Q1**

Pentru realizarea automatizării pentru a doua categorie de plante, care necesită păstrarea unui nivel minim al umidității în sol se va utiliza o funcție specială și o funcție de bază. Pentru inserarea unei funcții speciale se va selecta din toolbar icoana „Special functions” sau se va apăsa tasta **F8**, toate funcțiile din această categorie devenind disponibile în partea de jos a ecranului.

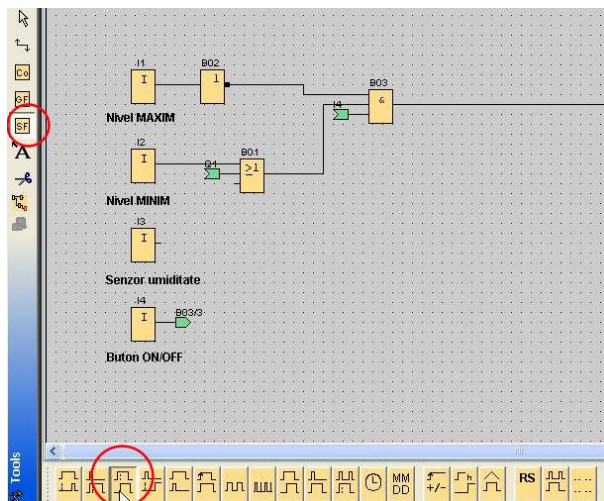


Fig. 2.60 – Inserarea unei funcții speciale

Se va utiliza o funcție „**On/Off Delay**” care permite introducerea unei întârzieri atât la apariția unui semnal de trigger cât și la dispariția acestuia. Prin setarea corespunzătoare a timpilor de întârziere la pornire și oprire se asigură

cuplarea pompei de irigare doar în situația când semnalul de la senzor de umiditate se păstrează activ pentru câteva secunde, în același timp setându-se și durata de funcționare a pompei. Blocul permite setarea timpului în secunde, minute sau ore, în funcție de necesitățile utilizatorului.

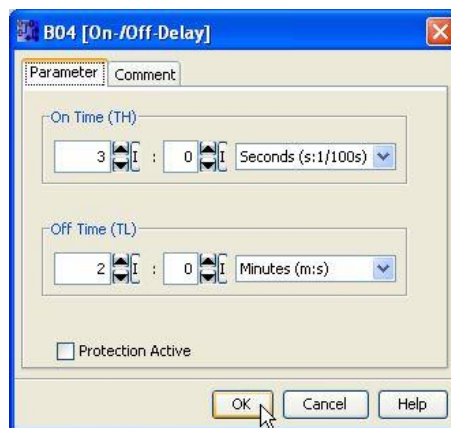


Fig. 2.61 – Parametrizarea funcției ON/OFF Delay

Diagrama bloc este simplă, pe lângă conexiunea realizată între intrarea **I3** și ieșirea **Q2** prin intermediul blocului de tip „On/Off Delay” utilizându-se un bloc de tip **SI** pentru introducerea în schema de comandă a condiției de funcționare dată de butonul de pornire/oprire conectat la intrarea **I4**. Din nou, pentru simplificarea diagramei conexiunea între intrarea **I4** și blocul **B05** a fost întreruptă.

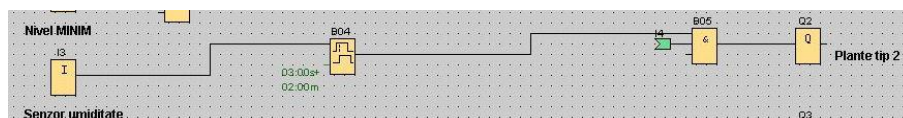


Fig. 2.62 – Finalizarea programului pentru tipul 2 de plante

Ultima parte a programului, pentru automatizarea irigării celei de-a treia categorii de plante, se realizează utilizând un bloc de tip **Weekly Timer** și un bloc **SI** pentru activarea automatizării cu ajutorul butonului conectat la intrarea **I4**.

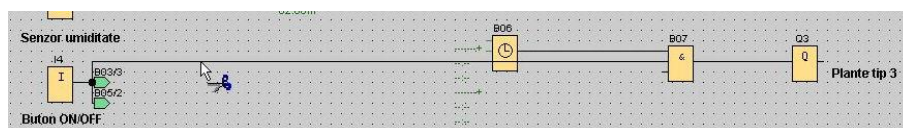


Fig. 2.63 – Programul pentru tipul 3 de plante



Blocul **Weekly Timmer** permite realizarea concomitentă a programării pe 3 canale diferite, pe fiecare dintre acestea putându-se alege ziua săptămânii în care ieşirea să devină activă, precum şi intervalul orar de funcţionare. Pentru fiecare dintre cele 3 canale pot fi setate două intervale orare precum şi zilele de funcţionare dorite, rezultând o multitudine de combinaţii posibile. În cazul în care intervalele orare de pe cele 3 canale se suprapun, prioritate va avea canalul cu numărul de ordine cel mai mare, după cum s-a arătat anterior.

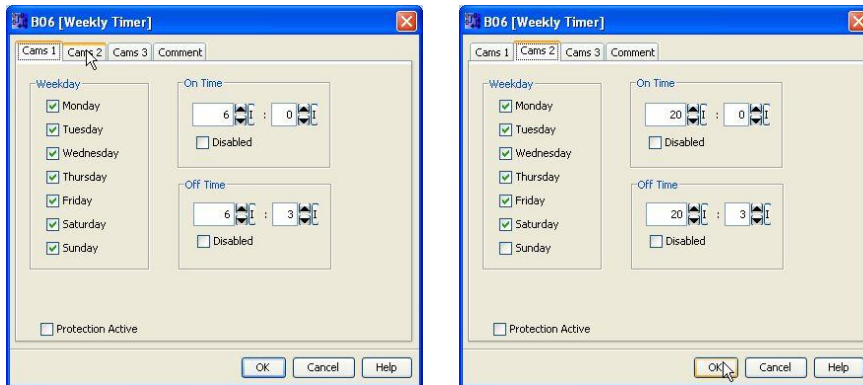


Fig. 2.64 – Parametrizarea primelor 2 canale ale funcţiei Weekly Timer

După finalizarea diagramei bloc pentru automatizarea sistemului de irigaţii, înainte de încărcarea programului pe **Logo!** şi testarea acestuia „in site”, programul **Logo! Soft Comfort** oferă posibilitatea simulării funcţionării programului fără a fi necesară conectarea unui automat programabil sau a diferiţilor senzori la calculatorul utilizat. Trecerea în modul de simulare se face alegând „Simulation” din bara de unelte sau apăsând tasta **F3**.

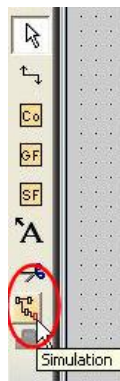


Fig. 2.65 – Trecerea în modul de simulare al funcţionării programului

În modul de simulare zona de sub diagrama bloc va conține icoane pentru toate elementele de intrare/ieșire ce sunt incluse în programul realizat, precum și icoane care permit simularea întreruperii alimentării cu energie electrică, trecerea automatului programabil în modurile de rulare, stop sau pauză, rularea programului pentru un număr finit de cicluri, precum și ora internă a **Logo!**-ului simulat. Prin trecerea în stare de 1 logic a intrării **I4** se realizează un click pe icoana corespunzătoare acestei intrări din diagrama bloc sau din zona cu uneltele pentru simulare. Prezența unui semnal prin conexiunile diagramei este marcată prin schimbarea culorii de reprezentare a acestei conexiuni, după cum se observă și în figura următoare.

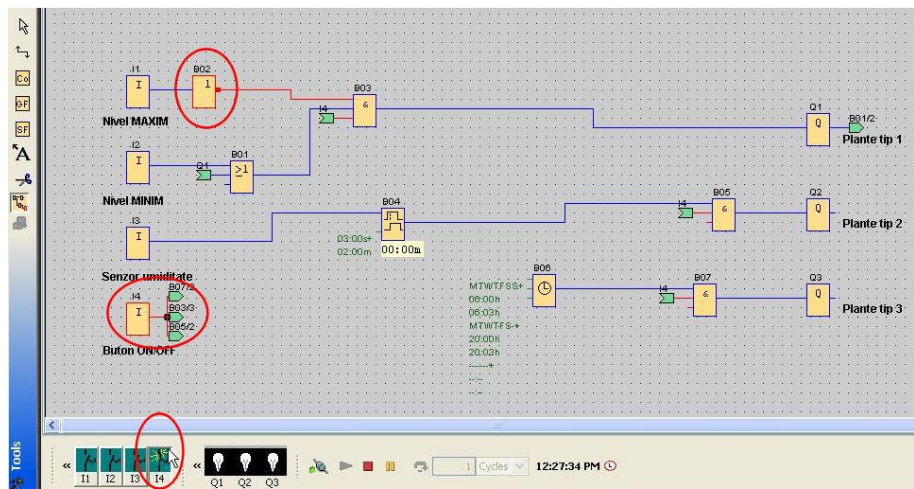


Fig. 2.66 – Interfața programului în modul de simulare

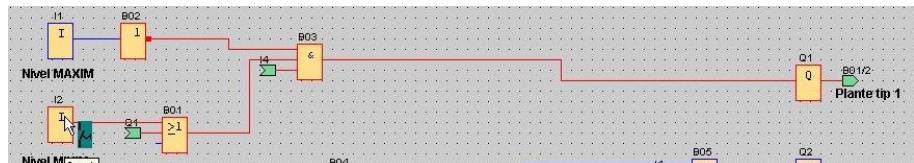


Fig. 2.67 – Modul de prezentare al ieșirii Q1 în stare ON

Pentru simularea funcționării primei ramuri a programului realizat se păstrează intrarea **I1**, corespunzătoare senzorului de nivel maxim, în stare 0 logic, iar butonul de ON/OFF general, conectat la intrarea **I4** este în stare **ON**. Când intrarea **I2**, unde este conectat senzorul de nivel minim, trece în stare **ON**, funcția **SAU** reprezentată de blocul **B01** va avea la ieșire 1 logic, iar funcția **ȘI** din blocul **B03**, având cele 3 intrări active, va determina trecerea ieșirii **Q1** în stare **ON**. Acest



lucru este vizibil atât în cadrul diagramei bloc cât și în zona cu uneltele de simulare, ieșirea **Q1** fiind reprezentată de un bec aprins. La dispariția semnalului dat de senzorul de nivel minim, pompa conectată la ieșirea **Q1** va continua să funcționeze datorită buclei de automenținere realizată în cadrul diagramei bloc. În momentul în care lichidul va atinge nivelul maxim dorit, lucru simulat prin trecerea intrării **I1** în stare **ON**, ieșirea blocului **B03** va deveni inactivă, întrerupând astfel alimentarea ieșirii **Q1** și a buclei de automenținere.

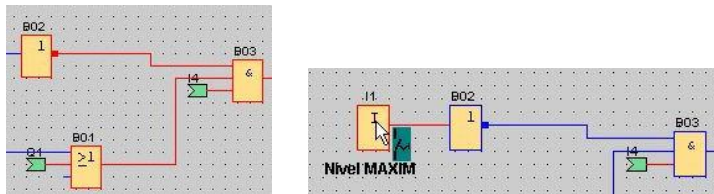


Fig. 2.68 – Simularea funcționării ramurii 1 a programului

Funcționarea celei de-a doua părți a diagramei bloc este mai simplă, conținând doar două funcții, prima dintre ele fiind de tip „**On/Off Delay**”. La apariția unui semnal pe intrarea **I3**, corespunzătoare unui senzor de umiditate, se va declanșa timer-ul pentru măsurarea timpului de întârziere la pornire a blocului. Timpul trecut de la apariția acestui semnal va fi afișat sub blocul **B04**, iar ieșirea blocului va rămâne inactivă. Orice dispariție a semnalului la intrarea blocului va reseta timpul măsurat, care va crește din nou la apariția unui nou semnal. La expirarea timpului setat în cadrul parametrilor funcției, ieșirea blocului va deveni activă, iar în cazul în care butonul ON/OFF general este activ ieșirea **Q2** va trece în stare **ON**. Irigarea plantelor se va realiza atât timp cât senzorul de umiditate este activ, iar la dispariția semnalului provenit de la acesta se activează întârzierea la oprire din cadrul blocului **B04**.



Fig. 2.69 – Simularea funcționării ramurii 2 a programului

În cazul celui de-al treilea tip de plante, ieșirea **Q3** devine activă în anumite intervale de timp, setate în cadrul blocului **Weakly Timmer**. În exemplul prezentat se dorește irigarea acestei categorii de plante dimineața, între orele 6:00 și 6:03 și după-masa, între 8:00 și 8:03. La trecerea în acest interval orar ieșirea blocului comută în stare **ON**, în cazul în care butonul general de pornire/oprire este **ON** ieșirea **Q3** devenind și ea activă.



Fig. 2.70 – Simularea funcționării ramurii 3 a programului

După ce diagrama bloc a fost testată și eventualele erori au fost corectate, programul se poate transfera în memoria automatului programabil, iar la trecerea în modul de lucru a acestuia, la apariția de semnale de la senzorii și butoanele conectate la intrările **Logo!** ieșirile vor acționa corespunzător.

### 2.5.3 Automatizarea unui sistem de distrugere a microbilor cu gaz

Se va folosi **Logo!** pentru realizarea unui sistem de distrugere a microbilor în cadrul unei crescătorii de păsări. Se utilizează o încăpere în care se pompează un gaz produs cu ajutorul unui dispozitiv electric de încălzit. Pentru distrugerea microbilor este necesar ca acest gaz să fie păstrat în încăpere o perioadă definită de timp, după care acest gaz este evacuat cu ajutorul unui ventilator.

Componente utilizate:

Componenta	Conectarea la <b>Logo!</b>
Buton pornire/oprire (contact NO)	I1
Buton pornire/oprire ventilator evacuare (contact NO)	I2
Generator de gaz	Q1
Ventilator de evacuare	Q2
Lampă de avertizare	Q3
<b>Logo! 230 RC</b>	

Strategii de programare:

Pentru realizarea etapelor enumerate mai sus se va utiliza un buton care să inițializeze procesul de realizare a gazului. Sistemul de producere a gazului este oprit după 10 minute, după care, în funcție de dimensiunile camerei, gazul este evacuat

după o perioadă de timp considerată suficientă pentru distrugerea microbilor. Evacuarea gazului se face cu ajutorul unui ventilator, iar desfășurarea procesului este semnalizată cu ajutorul unei lămpi. Procesul poate fi oprit în orice moment prin apăsarea pentru 3 secunde a butonului de pornire, iar ventilatorul de evacuare poate fi pornit în orice moment, indiferent de starea sistemului, cu ajutorul unui buton.

Avantaje ale utilizării **Logo!**:

În funcție de dimensiunile încăperii utilizate se poate seta durata de păstrare a gazului în încăpere, permițând astfel utilizarea unui program în mai multe locații, cu modificări minime. Sistemul poate fi oprit în orice moment prin apăsarea pentru 3 secunde a butonului de pornire, reducând numărul de componente necesare realizării practice a sistemului.

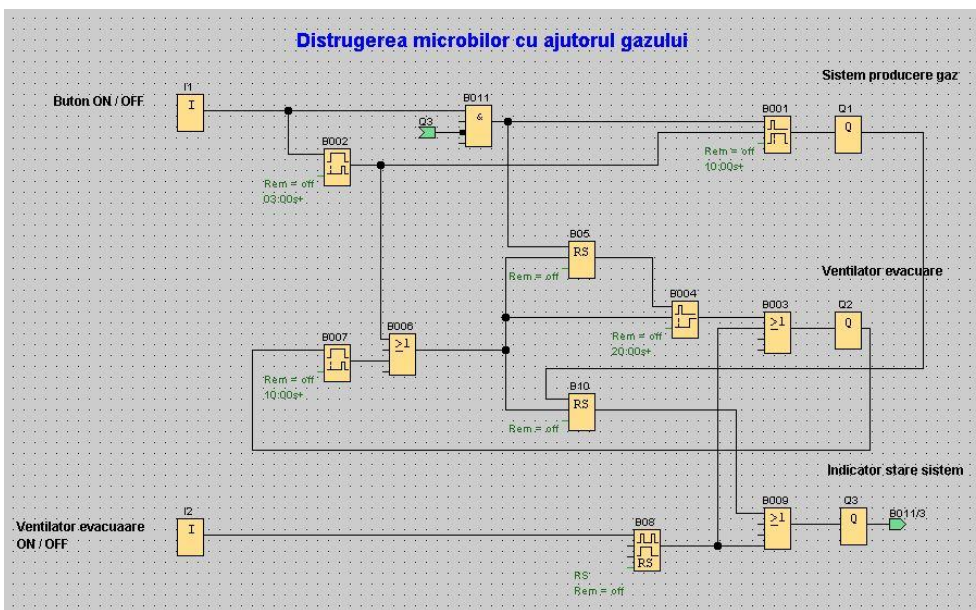


Fig. 2.71 – Diagrama bloc pentru sistemul propus

## 3. Automatul programabil Siemens S7-1200

### 3.1 Introducere

Clasa de automate programabile **SIMATIC S7** este încadrată de către producătorul **Siemens** în categoria controlerelor modulare, clasă superioară celei reprezentate de modulele logice **Logo!**. Gama de automate programabile oferită de producătorul german este completată de controlerele embedded și de controlerele virtuale, realizate utilizând programe specializate ce rulează pe calculatoare industriale. În această categorie au fost disponibile inițial 3: **S7 – 200**, **S7 – 300** și **S7 – 400**, cărora li s-au adăugat începând cu 2012 modelele **S7 – 1200** și **S7 – 1500**. Caracteristicile acestora variază, pornind de la performanțe similare celor oferite de **Logo!**, ajungându-se la sisteme ce permit controlul și automatizarea proceselor industriale de producție, unde volumul de informații vehiculat este extrem de mare.

Principalele diferențe între cele 5 categorii de automate programabile enumerate anterior sunt date de numărul și tipurile de porturi de intrare / ieșire disponibile, funcțiile logice ce pot fi utilizate, capacitatea memoriei interne, posibilitățile de conectare sau viteza de lucru. Toate aceste modele permit creșterea numărului de intrări / ieșiri sau adăugarea de funcții suplimentare prin conectarea de module de extensie, numărul maxim de module ce pot fi folosite și tipul acestora depinzând de timpul PLC-ului. Dacă utilizarea **S7 -200**, **S7 – 1200** și **S7 – 1500** se rezumă în general la aplicații reduse ca dimensiune, de tipul ascensoarelor, spălătoriilor auto, părți din procese industriale, modelul **S7 – 300** permite automatizarea integrală a proceselor industriale, în special în domeniul producerii de echipamente pentru industria auto și al ambalării de produse. Modelul de top al gamei, **SIMATIC S7 – 400**, este ideal pentru procese unde volum de date vehiculat este foarte mare, PLC-ul oferind viteze ridicate de procesare a datelor și timpi de răspuns reduși, aspecte vitale în cadrul proceselor de producție.

Atenția noastră se va îndrepta spre modelul de bază al acestei familii de automate programabile, **S7 – 1200**, fiind puse în evidență avantajele oferite de acesta în comparație cu **Logo!**, micro PLC-ul prezentat anterior. Modul de programare și majoritatea funcțiilor vor fi regăsite și în modelele superioare ale gamei, astfel trecerea spre o structură mai complexă fiind facilă.

## 3.2 Prezentare generală

Automatul programabil **S7-1200** oferă un design compact combinat cu capacități ridicate de lucru în timp real, viteză de lucru ridicată și componente hardware și software ușor de utilizat. Aceste caracteristici, combinate cu numeroasele posibilități de extindere datorită structurii sale modulare, impunându-l de-a lungul timpului drept soluția ideală pentru automatizări de mărime mică și medie.

### Caracteristici tehnice

Principalele elemente prezente pe partea frontală a unui automat programabil din familia **S7-1200** sunt prezentate în figura de mai jos, putându-se remarca lipsa unui afișaj sau a tastelor ce permiteau programarea directă în cazul **Logo!**. În schimb se pot observa o serie de LED-uri ce oferă informații despre starea intrărilor, a ieșirilor și a PLC-ului în general, conectori pentru intrări, ieșiri, module suplimentare, extensii și comunicarea cu alte dispozitive.



Fig. 3.1 –Automat programabil **Siemens S7 – 200**

Clasa **S7-1200** conține 5 modele: CPU 1211C, CPU 1212C, CPU 1214C, CPU 1215C și CPU 1217C, pentru fiecare dintre aceste modele existând mai multe variante disponibile, diferențiate prin tipul intrărilor și al ieșirilor și numărul acestora, dimensiunile geometrice, capacitatea memoriei sau alte performanțe. Tabelul următor prezintă principalele caracteristici ale acestora, inclusiv variantele existente pentru fiecare model.

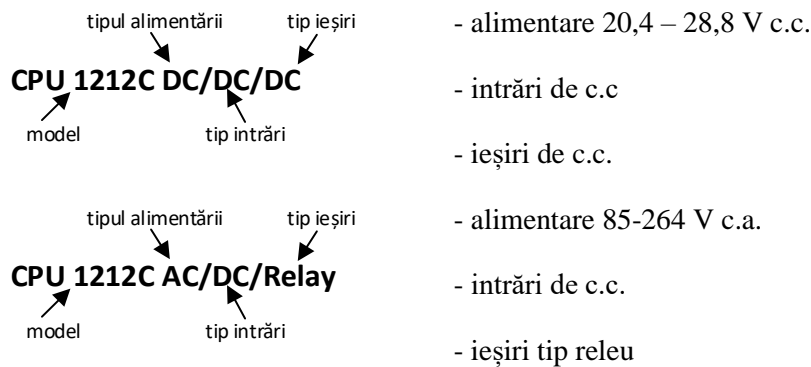
Caracteristicile automatelor programabile din familia **S7-1200**

Caracteristică	CPU 1212C	CPU 1212C	CPU 1214C	CPU 1215C	CPU 1217C
Dimensiuni (mm)	90 x 100 x 75	90 x 100 x 75	110 x 100 x 75	130 x 100 x 75	150 x 100 x 75
Memorie de lucru	50 Kb	75 Kb	100 Kb	125 Kb	150 Kb
Memorie internă	1Mb	2Mb	4Mb	4Mb	4Mb
I/O digitale	6 In/4 Out	8 In/6 Out	14 In/10 Out	14 In/10 Out	24 In/16 Out
I/O analogice	2 In	2 In	2 In	2 In/2 Out	2 In/2 Out
Module de extensie (SM)	-	2	8	8	8
Module de extensie (SB, BB, CB)*	1	1	1	1	1
Module de comunicație (CM)	3	3	3	3	3
Frecvență comutație pentru ieșiri	4 x 100 kHz	4 x 100 kHz 2 x 20 kHz	4 x 100 kHz 6 x 20 kHz	4 x 100 kHz 6 x 20 kHz	6 x 100 kHz 4 x 1 MHz
Contoare de mare viteză	6 x 100 kHz	6 x 100 kHz 2 x 30 kHz	6 x 100 kHz 8 x 30 kHz	6 x 100 kHz 8 x 30 kHz	2 x 100 kHz 2 x 30 kHz 4 x 1 MHz
Păstrarea datelor	20 zile sau 12 zile la 40°C				

Notă:

- \* SB – Modul extensie tip Signal Board
- BB – Modul extensie tip Battery Board
- CB – Modul extensie tip Communication Board

Ca și în cazul **Logo!** alimentarea se face de la o sursă de tensiune alternativă 85-264 V cu 47-63 Hz sau de tensiune continuă 20,4-28,8 V, în funcție de tipul PLC-ului. Informații despre tipul alimentării și al intrărilor / ieșirilor este prezent într-o formă condensată în denumirea inscripționată pe carcasa automatului programabil:



Montarea automatelor programabile din clasa **S7-1200** se poate face pe șine de 35 de mm, specifice tablourilor electrice, cu ajutorul sistemului de prindere cu cleme accesibil în partea inferioară a modului. Se recomandă păstrare unei zone libere de 25mm deasupra și sub carcasa automatului programabil pentru a permite accesul facil la bornele de intrare / ieșire. În acest context trebuie amintit că unele modele prezintă conectori detașabili pentru intrări / ieșiri, depanarea automatului programabil fiind astfel mult ușurată.

Partea frontală a unui modelului S7-1200 CPU 1211C, AC/DC/RLY este prezentată în Figura 3.2, principalele elemente fiind puse în evidență și descrise în cele ce urmează:

1. Conector alimentare cu tensiune alternativă, ieșire de la sursa internă de 24Vcc și bornele de intrări digitale în cc;
2. Conector pentru intrările analogice;
3. Slot pentru conectarea memoriei de tip SD;
4. LED-uri status intrări digitale;

5. Informații referitoare la modelul de automat programabil (CPU, tip intrări și, ieșiri);
6. LED-uri status ieșiri digitale;
7. Conector ieșiri digitale. În cazul acestui model doar 5 din cele 8 conectori sunt utilizați, la modelele superioare toți cei 8 fiind activi;
8. Conector pentru cablul de rețea LAN;
9. LED-uri status conexiune TCP/IP;
10. LED-uri status PLC: RUN/STOP, ERROR și MENT.

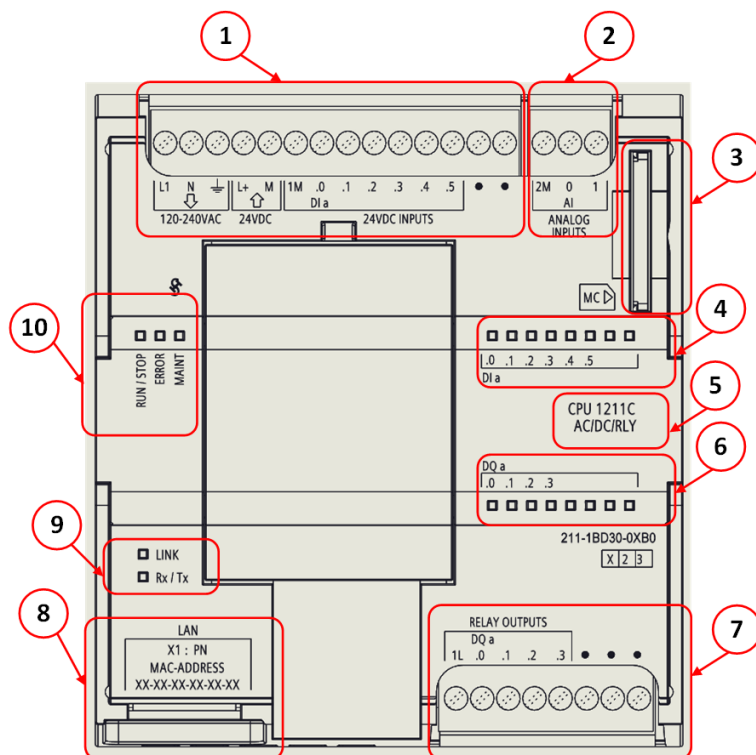


Fig. 3.2 – Aspectul părții frontale a S7-1200 CPU1211C



## Module de extensie

În situațiile în care numărul de intrări/ieşiri sau performanțele modelului ales nu acoperă nevoile aplicației vizate, familiar **S7-1200** oferă un număr foarte mare de module de extensie, sintetizate în cele ce urmează în funcție de tipul lor:

- **SB – Signal Boards**

Aceste plăci se montează pe partea frontală a modului central și permit conectarea suplimentară de semnale la modulul central, fiind la rândul lor de mai multe tipuri:

- Intrări digitale – 4 DI pentru 24Vcc sau 5Vcc
- Ieşiri digitale – 4 DO pentru 24Vcc sau 5Vcc
- Intrări/ieşiri digitale - 2 DI și 2 DO pentru 24Vcc sau 5Vcc
- Intrări analogice – 1 AI, 1 RTD sau 1AI + 1 TC (tip J sau K)
- Ieşiri analogice – 1 AO

- **CB – Communication boards**

Permit extinderea posibilităților de conectare a unității centrale prin mufă de tip RS485

- **BB – Battery boards**

Acest modul permite păstrarea informațiilor în memoria automatului programabil pentru o perioadă mai lungă de timp.

- **DM – Digital Modules**

Această categorie reprezintă module care se conectează pe partea laterală a modului central, permițând o creștere cu un număr mai mare de intrări / ieşiri comparativ cu plăcile enumerate anterior.

- Intrări digitale – 8DI 24Vcc, 16DI 24Vcc
- Ieşiri digitale – 8DO Relay, 16DO Relay, 8DO 24Vcc, 16DO 24Vcc, 16DO 24V sink

- Intrări/Ieșiri digitale – 8DI / 8DO, 16DI / 16DO, 8DI AC/ 8DO Relay, 16DI / 16DO Relay, 16DI / 16DO sink, etc.
- Module tip fail-safe – 16 F-DI 24Vcc, 4 F-DO 24Vcc, 2 F-DO Relay

- **AM – Analog Modules**

Permit extinderea unității centrale cu module analogice de intrare sau ieșire sau cu module mixte de intrare / ieșire:

- Intrări analogice – 4AI, 8 AI, 4AI 16 bit, RTD 4AI, RTD 8AI, TC 4AI, TC 8AI, Energy Meter 480V AC
- Ieșiri analogice – 2AO, 4AO
- Intrări / Ieșiri analogice – 4AI/2AO

- **SM – Simulation Modules**

Aceste module se conectează direct la intrările modulului central și permit schimbarea stării acestora, fără a fi necesară realizarea unui montaj suplimentar sau conectarea de senzori sau comutatoare, permițând testarea funcționării programului înainte de introducerea automatului programabil în sistemul pentru care a fost pregătit.

- Intrări digitale – 8 Inputs, 14 Inputs
- Intrări analogice – 2 Potentiometer

- **CM – Communication Modules**

Aceste module permit conectarea unității centrale utilizând alte protocoale de comunicații:

- RS232, RS422/485, 4xIO Link Master, CP 1242-5, etc.

- **Technology Modules**

Permit realizare de operațiuni specifice, de cântărire sau măsurare a vibrațiilor:

- WP231, WP 241, WP251 Weighing Electronic
- SM1281 – Condition Monitoring

- **Power Supplies**

În această categorie intră sursa de tensiune PM 1207/1AC/24Vcc/2,5 A

- **Accessories**

Accesoriile disponibile pentru automatele programabile din familia S7-1200 includ carduri de memorie cu capacitate între 4Mb și 32 Gb, Manuale de utilizare, Terminale cu număr variabil de conectori, module pentru comunicații pentru telefonie mobilă sau fixă, cabluri, repeater-e, etc.



Placă de extensie



Modul de extensie

Fig. 3.3 –Extensii pentru modelele S7-1200

Figura 3.3 prezintă un exemplu de placă de extensie care se montează pe partea centrală a PLC-ului, precum și un modul de extensie intrări/ieșiri, montat în partea laterală a controllerului.

Pentru a ușura testarea programelor realizate în TIA Portal și încărcate pe automatul programabil, firma Siemens oferă un modul de simulare a semnalelor digitale de intrare, prezentat în Figura 3.4. Acesta constă în 8 contacte de tip ON/OFF care se conectează la cele 8 intrări ale PLC-ului, precum și un conector la bornele căruia se regăsește o tensiune de 24Vcc provenită de sursa internă a automatului programabil.

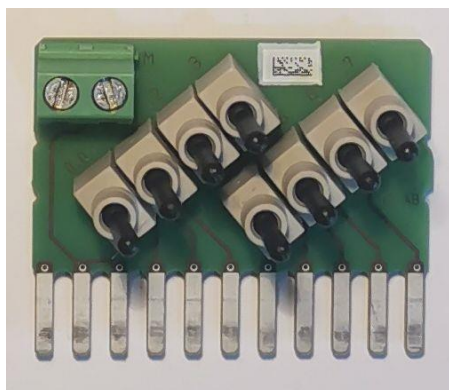


Fig. 3.4 – Extensie pentru simularea semnalelor de intrare

Zona centrală a modelului **S7-1200** conține două rânduri de LED-uri care indică starea logică a intrărilor și a ieșirilor. Astfel, prezența unui semnal pe prima bornă de intrare, echivalând cu 1 logic, va fi semnalată prin aprinderea primului LED din rândul superior. Modul de lucru în care se află automatul programabil este indicat prin cele 3 LED-uri montate în partea din stânga, fiind astfel ușor de determinat dacă PLC-ul este în modul RUN (verde) sau STOP (galben) sau în timpul procesului de pornire, când culoarea LED-ului de status alternează între verde și galben. LED-ul din mijloc este cel de eroare, putând indica o problemă hard sau la nivelul unității centrale, a cardului de memorie sau de configurare. LED-ul din dreapta este cel de mentenanță (MAINT) este util când se citește un card de memorie, oferind informații despre accesarea informațiilor de pe card.

Numerotarea intrărilor și a ieșirilor este diferită de cea utilizată în cazul **Logo!** fiind folosite pentru identificarea unei borne o literă urmată de două cifre,

despărţite de un punct. Etichetele aferente porturilor I/O sunt prezente atât în zona celor două şiruri de LED-uri ce indică starea lor logică, cât şi lângă bornele unde sunt legate intrările şi ieşirile, zone acoperite cu două capace. Sistemul de notare oferă informaţii referitoare la tipul portului (**I** pentru intrări şi **Q** pentru ieşiri), prima cifră indicând byte-ul aferent adresei portului, iar ce-a de-a doua cifră indicând bitul adresei. Tabelul de mai jos prezintă modul de interpretare al notaţiilor folosite în cazul unui **S7-1200 CPU 1214C**:

Sistemul de indexare al porturilor I/O

I0.0	Intrarea 1	I1.0	Intrarea 9	Q0.0	Ieşirea 1	Q1.0	Ieşirea 9
I0.1	Intrarea 2	I1.1	Intrarea 10	Q0.1	Ieşirea 2	Q1.1	Ieşirea 10
I0.2	Intrarea 3	I1.2	Intrarea 11	Q0.2	Ieşirea 3		
I0.3	Intrarea 4	I1.3	Intrarea 12	Q0.3	Ieşirea 4		
I0.4	Intrarea 5	I1.4	Intrarea 13	Q0.4	Ieşirea 5		
I0.5	Intrarea 6	I1.5	Intrarea 14	Q0.5	Ieşirea 6		
I0.6	Intrarea 7			Q0.6	Ieşirea 7		
I0.7	Intrarea 8			Q0.7	Ieşirea 8		

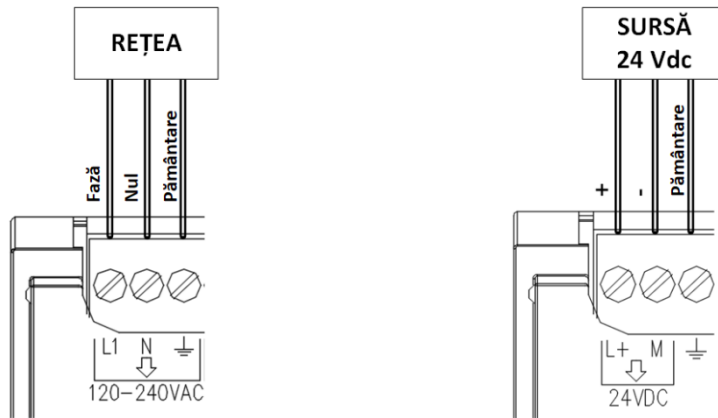
În cazul în care acestui PLC i se va ataşa unul sau mai multe module de extensie, numerotarea acestora se va face în funcţie de tipul şi numărului porturilor de intrare/ieşire disponibile, prima cifră a simbolului aferent noilor porturi indicând modulul de extensie. Câteva exemple de module de extensie şi modul în care porturile acestora se vor numerota sunt prezentate în tabelul de mai jos, considerându-se că acestea se utilizează cu modulul **S7-200 CPU 1214C** prezentat mai sus.

Sistemul de indexare al porturilor I/O modulelor de extensie

Modulul 0		Modulul 1	Modulul 2		Modulul 3	Modulul 4	
4 In / 4 Out		8 In	4 An In / 4 An Out		8 Out	4 An In / 1 An Out	
I2.0	Q2.0	I3.0	AIW4	AQW 4	Q3.0	AIW1 2	AQW 8
I2.1	Q2.1	I3.1	AIW6	AQW 6	Q3.1	AIW1 4	AQW 10
I2.2	Q2.2	I3.2	AIW8		Q3.2	AIW1 6	
I2.3	Q2.3	I3.3	AIW10		Q3.3	AIW1 8	
I2.4	Q2.4	I3.4			Q3.4		
I2.5	Q2.5	I3.5			Q3.5		
I2.6	Q2.6	I3.6			Q3.6		
I2.7	Q2.7	I3.7			Q3.7		

**Punerea în funcțiune (comisionare)**

În funcție de modelul de automat programabil ales va fi necesară alimentarea acestuia de la rețeaua de curent alternativ (Figura 3.5 a) sau de la o sursă de tensiune continuă, cum se poate observa în Figura 3.5 b). Modul de conectare al alimentării este prezentat, pentru cele două cazuri, în figura de mai jos:

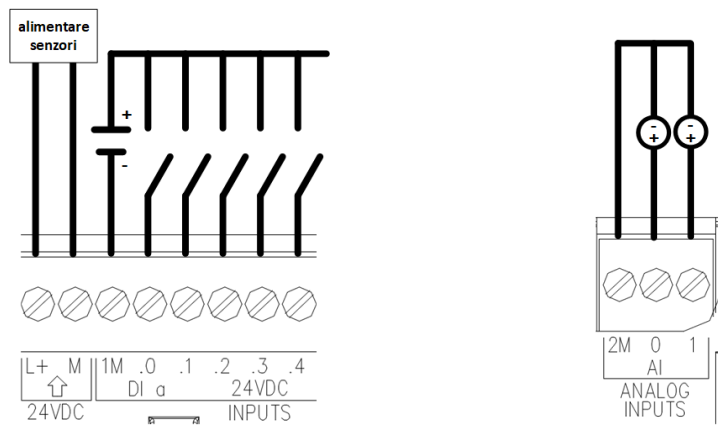


a) CPU 1212C, AC/DC/Relay

b) CPU 1212C, DC/DC/DC

Fig. 3.5 – Alimentarea PLC S7-1200

Indiferent de modul de alimentare al automatului programabil, intrările sunt în curent continuu (C.C. sau D.C.). Pentru conectarea acestora este necesară o sursă externă de 24 Vcc care se conectează ca în Figura 3.6. PLC-ul oferă și o sursă internă de tensiune de continuă de 24 Vcc, care poate fi utilizată atât pentru conectarea intrărilor cât și pentru alimentarea senzorilor utilizați ca sursă de semnal pentru intrări. În cazul intrărilor analogice prezente la toate modelele din clasa **S7-1200**, conectarea senzorilor se face în mod similar, fiind disponibilă o bornă ”-” dedicată, notată cu 2M, precum în Figura 3.6 b).



a) conectarea intrărilor digitale

b) conectarea intrărilor analogice

Fig. 3.6 – Conectarea bornelor de intrare digitale și analogice

În cazul modelelor cu ieșiri de tip releu conectarea ieșirilor se face simplu, ieșirile automatului programabil funcționând ca niște întrerupătoare în circuitul de alimentare (cu curent continuu sau alternativ) al actuatorilor controlate. În cazul modelelor cu ieșiri de tip DC (cc) acestea includ un circuit intern adecvat sarcinilor de tip inductiv. În ambele situații, valoarea maximă a curentului ce poate fi vehiculat de ieșirile automatului programabil depinde de tipul sarcinii conectate.

### **Conectarea cu alte dispozitive**

În cazul echipamentelor de generație mai veche conectarea cu alte echipamente se făcea utilizând protocoale de comunicație dedicate, cum ar fi PROFIBUS (Process Field Bus). Astfel se realizau conexiuni de tip **PPI** (Point to Point Interface pentru conexiuni tip Master-Slave) sau **MPI** (Multi-Point Interface – pentru comunicare bidirecțională între mai multe echipamente). Pentru echipamentele din familia S7-1200, modul nativ de realizarea a conexiunilor cu alte dispozitive este prin intermediul portului integrat **TCP/IP**, dar pot fi utilizate module sau plăci de extensie, prezentate anterior, pentru realizarea de conexiuni utilizând alte protocoale de comunicații.

### **Funcționarea ciclică a automatelor programabile**

Automatele programabile funcționează ciclic, prin repetarea a 3 pași: citirea intrărilor, executarea programului și scrierea ieșirilor. Cu toate acestea, există diferențe subtile în modul în care diverși producători sau diferite generații de automate programabile realizează acest ciclu. În cazul familie S7-1200, funcționarea include două stări: STOP și RUN.

Cât timp PLC-ul este în modul STOP acesta va gestiona doar partea de comunicații cu alte echipamente și va realiza o auto-diagnosticare, fără a executa programele încărcate de utilizator. La trecerea în modul RUN apare o etapă tranzitorie, denumită STARTUP, în care se execută o serie de operațiuni:

1. se copiază starea intrărilor fizice în zona de memorie **I**, dedicată intrărilor;



2. inițializează zona de memorie **Q** dedicată ieșirilor, fie cu valoarea 0, cu ultima valoare pe care ieșirea respectivă a avut-o sau cu valori presetate;
3. inițializează zona de memorie (fără **păstrare**) **M** și blocurile de date la valorile inițiale și activează evenimentele pe bază de oră/dată sau bazate pe **întreruperi**;
4. activează scrierea valorilor din zona de memorie **Q** în ieșirile fizice;

\* în paralel cu aceste 4 procese sunt identificate procesele care generează întreruperi și le încarcă în buffer, astfel încât acestea vor fi tratate, în ordinea priorității, după intrarea în modul RUN. Tranziția între aceste procese este reprezentată în partea din stânga a Figurii 3.7.

În timpul etapei de STARTUP nu se monitorizează durata ciclului, se configurează intrările/ieșirile care vor funcționa ca HSC (numărător de mare viteză), generator semnal PWM sau pentru comunicații PPI, fără ca acestea să genereze valori.

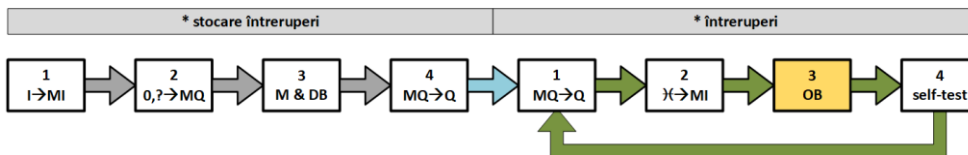


Fig. 3.7 – Funcționarea ciclică a unui PLC

După încheierea procesului de STARTUP, automatul programabil trece în modul RUN, unde se execută în mod ciclic următorii pași, prezentați în partea dreaptă a Figurii 3.7:

1. actualizarea valorilor ieșirilor fizice cu valorile din zona de memorie **Q**;
2. copierea stării intrărilor fizice în zona de memorie **I**;
3. executarea programului inclus în blocurile operaționale **OB**;
4. realizarea unei auto-diagnosticări;

\* în paralel se execută întreruperi cauzate de procese sau de comunicații, în orice moment al ciclului.

Sistemul garantează că toți acești pași se execută într-o perioadă mai scurtă decât durata maximă a unui ciclu (maximum cycle time), în caz contrar fiind generat un mesaj de eroare.

### 3.3. Siemens TIA Portal

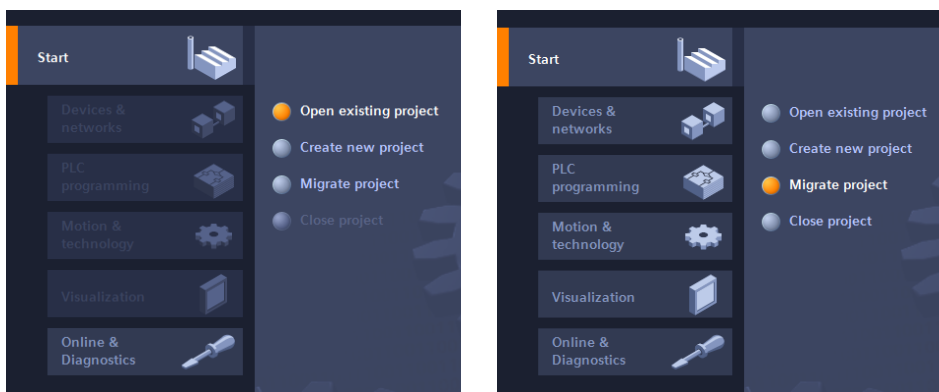
Program TIA Portal, prescurtare de la Totally Integrated Automation Portal, a fost dezvoltat de către compania Siemens în anul 2010 cu scopul de a oferi utilizatorilor o platformă complexă cu ajutorul căreia să se poată realiza conexiunea cu o serie de echipamente ale acestui producător. TIA Portal este considerat continuatorul mai multor programe software: Step 7, utilizat pentru programarea automatelor programabile, WinCC, utilizat pentru dezvoltarea de interfețe om-mașină (HMI) sau ProTool, utilizat anterior pentru configurarea panourilor pentru operatori sau a interfețelor grafice. Prin combinarea diverselor funcționalități ale acestor programe TIA Portal oferă o soluția mai eficientă pentru programarea, configurarea și punerea în funcțiune a echipamentelor Siemens.

Prima versiune poartă codul v11, fără a reprezenta continuarea numărării programelor pe care le-a înlocuit, ci mai degrabă pentru a semnifica începerea unei noi ere. În anii ce au urmat au fost lansate, cu o frecvență biennială, v17 fiind lansat în 2021. TIA Portal oferă o interfață user-friendly și un mediu unificat de programare, permițând inginerilor să dezvolte, testeze și să pună în funcțiune proiecte de automatizare industrială, oferind următoarele facilități și componente:

1. Programarea automatelor programabile din familiile S7-1200, S7-1500, and S7-300 și S7-400. Programarea poate fi realizată prin LAD (ladder logic)– logică tip ladder, FBD (function block diagram) – diagramă bloc, SCL – (structured control language) – programare text și programare grafică, utilizând Graph-uri;
2. Dezvoltarea intuitivă a interfețelor grafice pentru utilizator (HMI), oferind uneltele necesare proiectării și realizării de elemente grafice pentru vizualizarea, configurare de panouri de informare și alarme pentru utilizatori;
3. Programarea și parametrizarea invertoarelor electronice pentru sisteme de acționare electrică cu ajutorul Startdrive;
4. Controlul sistemelor de control al mișcării cu ajutorul SCOUT TIA;
5. TIA Portal include funcții de siguranță, cum ar fi oprirea de urgență sau realizarea de comunicații în condiții de siguranță;
6. Creșterea eficienței prin facilități cum ar fi reutilizarea codului, utilizarea de librării pentru configurarea echipamentelor sau configurarea automată a rețelelor;

7. Diagnosticarea la nivel de sistem prin integrarea de funcții care permit monitorizarea în timp real a echipamentelor, identificarea și diagnosticarea defectelor sau analiza performanțelor sistemelor de automatizări industriale;
8. Simularea și punerea în funcțiune virtuală, funcții ce permit utilizatorilor să testeze și să valideze programele și interfețele HMI create înainte ca acestea să fie puse în funcțiune, reducând astfel riscul apariției de erori și crescând performanțele sistemelor;
9. Integrarea cu alte sisteme, programe sau unelte Siemens, cum ar fi Siemens Industrial Edge, MindSphere (IoT platform) sau SIMATIC Energy Suite.

TIA Portal are două interfețe, numite Project View și Portal View, care permit utilizatorului să gestioneze proiectele de automatizare dezvoltate. În "Project View" pot fi deschise proiecte existente, crea proiecte noi sau pot fi importate proiecte realizate în programele ce au fost înlocuite de TIA Portal: Step 7, WinCC sau Distributed Safety.



a) fără proiect deschis

b) după deschiderea unui proiect

Fig. 3.8 – Interfața TIA Portal

După deschiderea unui proiect devin active o serie de opțiuni (Figura 3.8 b), care permit utilizatorului să adauge, configureze, programeze și depaneze echipamentele necesare în cadrul proiectului, conținând categoriile prezentate în cele ce urmează:

- Devices & Networks – permite adăugarea de echipamente, vizualizarea celor existente în proiect, precum și realizarea conexiunilor între aceste echipamente și vizualizarea rețelelor de comunicații existente;

- PLC Programming – permite vizualizarea blocurilor de programe (OB – Organization blocks), realizarea de programe noi sau conexiunile existente în diferite programe realizate în cadrul proiectului;
- Motion & Technology – permite configurarea de secvențe de control al mișcării pe mai multe axe, configurarea de reglatoare PID sau configurarea echipamentelor din familia Sinamics;
- Visualization – secțiune ce permite configurarea echipamentelor de tip HMI, fiind posibilă programarea modului în care sunt afișate informațiile pe ecranul acestor dispozitive;
- Online & Diagnostics – permite vizualizarea stării în care se află echipamentele existente în proiect, precum și accesarea zone de manuale și Help.

Pe lângă modul de vizualizare "Portal View", care oferă unelte de lucru orientate pe activități/sarcini, TIA Portal permite lucrul și în modul "Project View", în cadrul căruia se poate realiza programarea efectivă a echipamentelor. Trecerea dintr-un mod de vizualizare se poate face manual, alegând modul dorit din partea din stânga-jos a ecranului, utilizând combinația de taste <ALT + F7> sau în mod automat atunci când se accesează anumite opțiuni din "Portal View".

### **Adăugarea unui echipament**

După realizarea unui proiect nou este necesară adăugarea unui echipament, acțiune care poate fi realizată prin mai multe variante, cea mai simplă fiind direct din Portal View, din zona Devices & Networks>Add new device. Din următorul ecran se poate alege tipul de echipament care se va utiliza în cadrul proiectului: Controller, HMI sau PC systems; în prima categorie regăsim automate programabile din familiile Simatic S7-1200, -1500, -300 și -400, Drive Controller, ET200 sau Device proxy. Categoria HMI conține elementele de interfață grafică, grupate după clasa lor: SIMATIC Basic Panel, Comfort Panel și Mobile Panel, fiecare dintre acestea fiind disponibile în variante având diagonala ecranului cu dimensiuni variind între 3" și 22". Acestea li se adaugă clasa HMI SIPLUS, cu variantele SIPLUS Basic Panel și Comfort Panel. În cadrul categoriei PC systems regăsim variante de calculatoare personale, calculatoare industriale, controller-e embedded, SINUMERIK și altele. Pentru exemplificare, vom adăuga un PLC din familia S7-1200, putând fi ales modelul automatului utilizat, iar în situația în care nu se cunoaște modelul poate fi ales o variantă generică, precum în Figura 3.9.

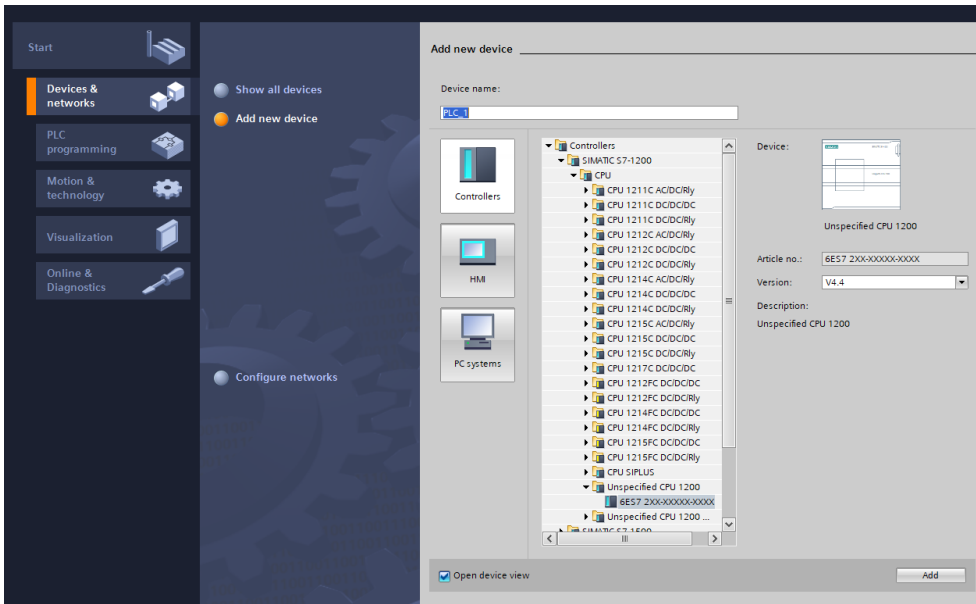


Fig. 3.9 – Adăugarea în proiect a unui PLC din familia S7-1200

TIA Portal trece automat în modul ”Project View”, unde se poate personaliza proiectul activ, după cum se arăta în paginile următoare.

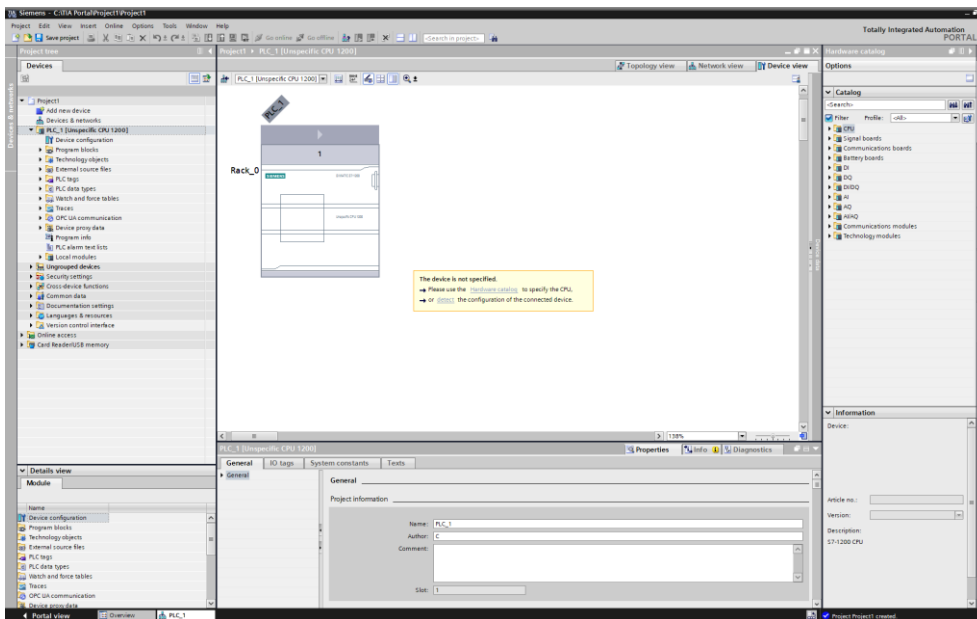


Fig. 3.10 – Interfața TIA Portal în ”Project View”

### Realizarea conexiunii PC – PLC

Un prim pas este realizarea conexiunii între calculator și automatul programabil, conectat în prealabil în aceeași sub-rețea. În cazul în care nu a fost specificat modelul PLC-ului, acest pas poate fi realizat acum cu ajutorul opțiunii ”**detect**”. În Figura 3.11 este prezentată fereastra de dialog unde se va selecta tipul conexiunii (în acest caz PN/IE) și placa de rețea utilizată. După realizarea unei căutări în rețeaua în care se află calculatorul se va afișa o listă cu toate automatele programabile care sunt detectate, opțiunea ”**Flash LED**” făcând ca cele 3 LED-uri de pe interfața PLC-ului să pâlpâie rapid de 6 ori.

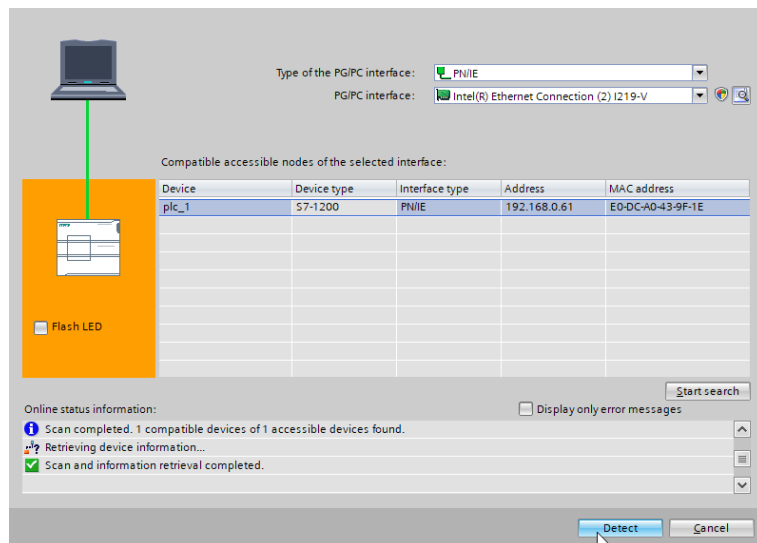


Fig. 3.11 – Realizarea conexiunii PC – PLC

După apăsarea butonului ”**Detect**” modelul generic afișat în zona ”Device View” a programului va fi înlocuit cu modelul identificat, în cazul nostru un S7-1200 CPU 1212C, tip AC/DC/Rly. În Figura 3.12 se poate observa reprezentarea grafică a automatului programabil, fiind vizibile LED-urile de status, portul de rețea, precum și LED-urile care arată starea celor 8 intrări și 6 ieșiri analogice. Prin realizarea unui click pe diferite zone ale automatului programabil se vor modifica informațiile prezentate în fereastra ”General” din partea de jos a interfeței grafice: pentru portul de rețea se vor afișa informații despre adresa ethernet și opțiuni despre modul în care se realizează transferul de date, iar un click pe o zonă neutră a PLC-ului va oferi acces la toate opțiunile modelului selectat: comunicații, porturi digitale și analogice, numărătoare de mare viteză, generare semnale tip PWM, timpul maxim permis pentru un ciclu, etc.

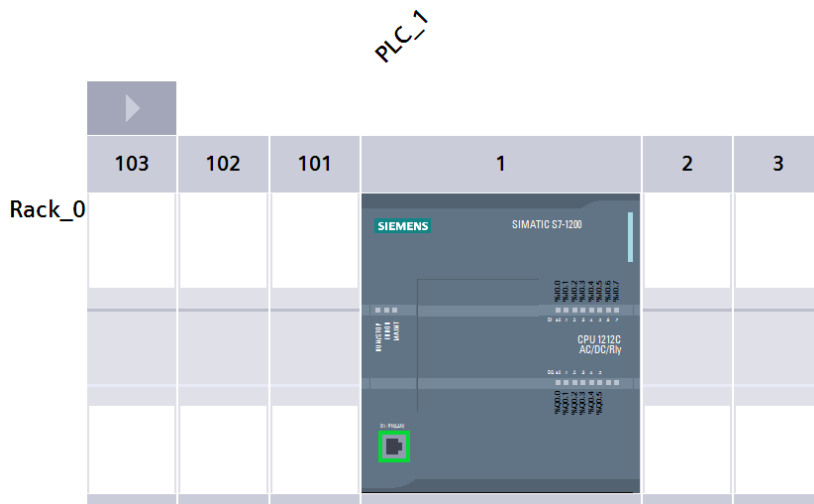


Fig. 3.12 – Prezentarea grafică în TIA Portal a PLC-ului utilizat

Pentru verificarea conexiunii între calculator și automatul programabil se va încerca comutarea stării PLC-ului între STOP și RUN, prin utilizarea butoanelor dedicate din meniul din partea superioară a programului. LED-ul de status "RUN /STOP" își va schimba culoarea în verde pentru modul RUN și în galben pentru modul STOP, tranziția spre modul RUN fiind marcată prin o serie de comutări rapide între cele două culori.



Fig. 3.13 – Modurile de lucru STOP și RUN

## Prezentarea Interfeței TIA Portal

”Work Area” (zona de lucru) a TIA Portal se va adapta în funcție de elementele selectate de către utilizator din structura proiectului (project tree), unde pot fi adăugate echipamente noi, pot fi configurate conexiunile între echipamentele existente în proiect, pot fi vizualizate și parametrizate aceste echipamente, pot fi identificate echipamente care nu au fost adăugate deja în structura proiectului, se pot accesa informații despre utilizarea încrucișată a resurselor din proiect, documentația realizată, porturi de comunicații sau carduri de memorie disponibile. Figura 3.14 prezintă modul în care această zonă se prezintă în cazul proiectului nou realizat anterior, pe măsură ce complexitatea proiectului crește fiind disponibile informații despre echipamentele și programele realizate.

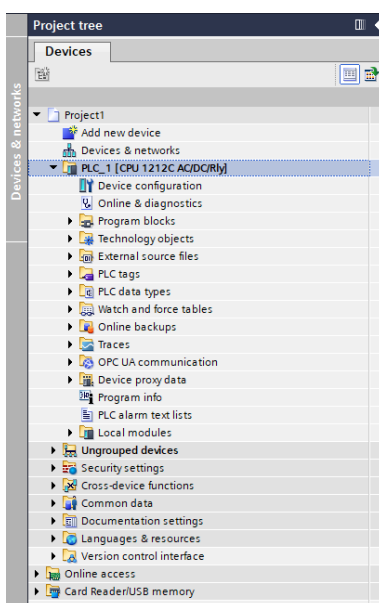


Fig. 3.14 – Structura proiectului în TIA Portal

Aspectul zonei de lucru se modifică în timpul utilizării TIA Portal, putând include mai multe zone (prin împărțire pe orizontală sau verticală) pentru situațiile în care utilizatorul dorește să urmărească în același timp informațiile disponibile în structura programului, interfața echipamentului, harta conexiunilor între echipamentele din proiect, tabelul cu etichetele utilizate, etc.



- **Prezentare simboluri grafice uzuale, codificarea adreselor**

Programarea automatelor programabile Siemens permite, în general, utilizarea a 3 metode de realizarea a programelor: Statement List (STL), Function Block (FBD) și Ladder Diagram (LAD). Fiecare dintre aceste moduri de programare prezintă unele avantaje comparativ cu celelalte două moduri, utilizatorul făcând alegerea în funcție de experiența pe care o are în domeniul programării. Totuși, pentru familia S7-1200 TIA Portal permite realizarea de programe doar sub formă de Ladder Diagram, motiv pentru care ne vom concentra în continuare asupra LAD și vor fi amintite câteva detalii despre celelalte două moduri de programare.

#### Editorul STL (Statement List)

Acest mod de programare este similar limbajelor de programare clasice, dar mai apropiat ca formă cu codul mașină. Fiind bazat pe linii de text, permite realizarea unor structuri inaccesibile în modurile de programare grafică, motiv pentru care e considerat modul nativ de programare al familiei **S7**. Constrângerile de natură grafică ce apar în realizarea diagramelor în FBS sau LAD fac ca nu orice program realizat în STL să poată fi convertit în alt mod de programare, reciproca nefiind întotdeauna valabilă. Similar altor limbaje de programare, instrucțiunile sunt executate pe rând, liniile de program fiind citite de sus în jos, după parcurgerea ultimei linii revenindu-se la începutul programului, dacă nu există instrucțiuni care să împiedice acest lucru.

Principalele caracteristici ale acestui mod de programare sunt sintetizate în cele ce urmează:

- STL este recomandat pentru programatorii cu experiență;
- unele probleme pot fi rezolvate mult mai ușor în STL comparativ cu celelalte moduri de programare;
- STL poate fi utilizat doar cu setul de instrucțiuni SIMATIC
- unele programe realizate în STL pot fi convertite întotdeauna în FBD sau LAD datorită constrângerilor de natură grafică. În schimb, programele realizate în cele două moduri de programare grafică pot fi convertite în STL.

#### Editorul FBD (Function Block Diagram)

Este asemănător celui utilizat pentru programarea **Logo!**, prezentată detaliat în capitolul anterior, fiind similar schemelor logice. În acest caz se consideră că energia „curge” de la stânga la dreapta, în funcție de starea funcțiilor logice, reprezentate prin blocuri, starea de 0 sau 1 logic fiind regăsită la elementele finale

ale schemei. Elementele ce trebuie considerate când se alege modul de programare FBD sunt:

- schemele bloc sunt ușor de parcurs pentru a înțelege funcționarea unui program;
- FBD permite utilizarea setului de instrucțiuni SIMATIC sau cel standardizat de IEC 1131-3;
- diagramele bloc sunt convertite ușor în STL, dacă e cazul.

### Editorul LAD (Ladder Diagram)

Reprezintă un mod de programare grafică asemănător cu montajele electrice clasice, partea din stânga a schemei fiind considerată alimentarea cu energie electrică iar partea de logică fiind dispusă sub forma unor ramuri orizontale, făcând întreaga structură să semene cu o scară (ladder). În funcție de starea blocurilor de pe ramurile diagramei terminalele din partea dreaptă a schemei vor fi alimentate cu energie electrică.

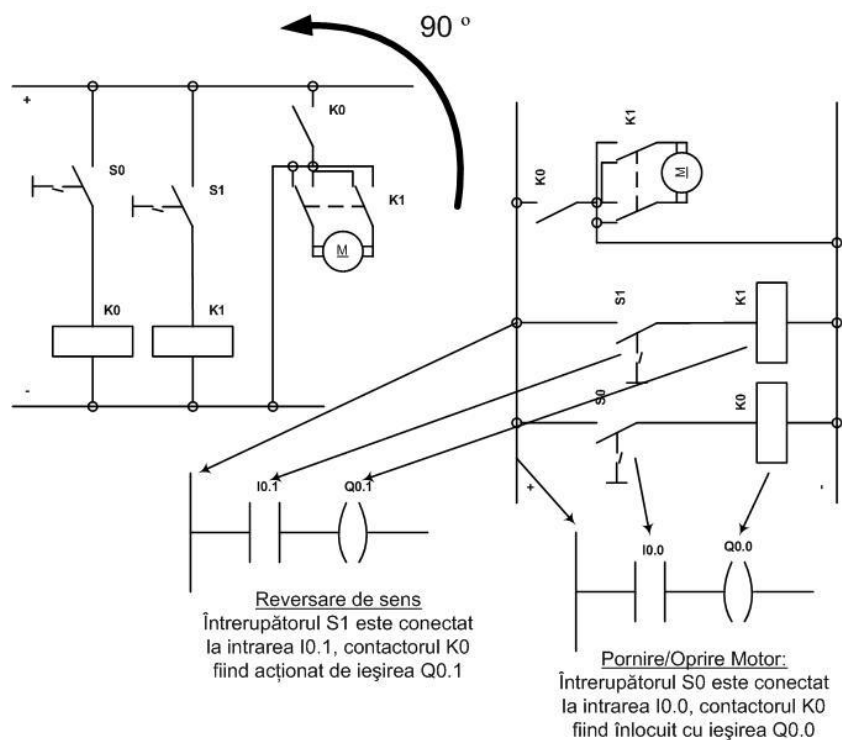


Fig. 3.15 – Conversia unei scheme electrice clasice în LAD

În cadrul acestor diagrame vor fi întâlnite 3 simboluri grafice: contacte, bobine și blocuri, detaliate în paragraful următor.

Principalele caracteristici ale editorului LAD sunt:

- ușor de folosit pentru utilizatori neinițiați;
- reprezentarea grafică este ușor de înțeles;
- poate utiliza atât setul de instrucțiuni SIMATIC cât și cel standardizat de IEC 1131-3;
- pentru o mai ușoară înțelegere a programelor realizate în STL acestea pot fi convertite în LAD.

Elemente grafice care sunt utilizate în cadrul diagramelor LAD pot fi încadrate în 3 categorii: contacte, bobine și blocuri. Acestea includ atât porturile de intrare/ieșire ale PLC-ului, cât și locații de memorie, funcții logice sau funcții avansate.

- **Contacte**

Un prim aspect ce trebuie amintit este faptul că acest simbol grafic este utilizat pentru a reprezenta porturi de intrare ale PLC-ului cât și biți de status utilizați în cadrul programului. Sunt utilizate două simboluri, reprezentând un contact normal deschis, respectiv unul normal închis. Contactul NO este închis (adică permite semnalului să treacă mai departe) atunci când intrarea sau bitul de status asociat sunt în stare 1 logic, iar contactul NC se deschide atunci când intrarea sau bitul de status care îl controlează sunt 1 logic.



Contact NO (normal deschis)



Contact NC (normal închis)

Fig. 3.16 – Tipuri de contacte utilizate în diagramele LAD

- **Bobine**

Acest simbol grafic sugerează bobina unui contact, care în momentul în care este alimentată cu energie electrică va închide un circuit. În mod similar, când starea logică a ramurii de circuit unde se află amplasată o bobină devine 1, ieșirea fizică a

automatului programabil sau blocul de memorie controlat de bobină își vor schimba starea logică în 1.



Fig. 3.17 – Reprezentarea grafică a ieșirilor din diagramele LAD

- **Blocuri**

Sub această reprezentare grafică Step7-Micro/WIN oferă un număr mare de funcții, începând cu blocuri de temporizare (timer-e), numărătoare, blocuri de control al programului, operatori logici, etc.

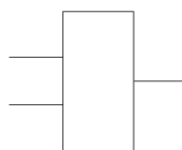


Fig. 3.18 – Aspectul general al unui bloc utilizat în diagramele LAD

Acestor simboluri grafice, ce definesc operația ce urmează a fi efectuată, li se va atașa întotdeauna în cadrul programelor și un operand, ce atașează operația unei locații de memorie sau unei intrări fizice. Structura unui astfel de simbol este prezentată în figura de mai jos, alături de structura unui operand.

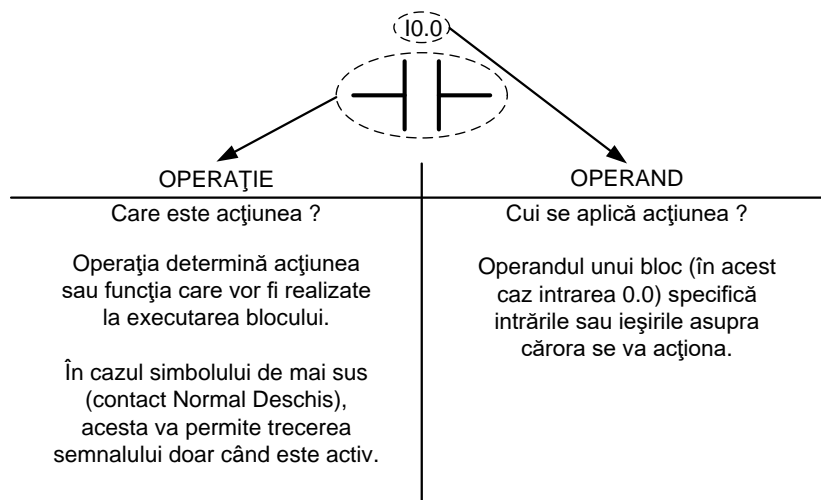


Fig. 3.19 – Aspectul general al elementelor grafice utilizate

Un operand conţine două elemente: un identificator şi un parametru. Identificatorul conţine o literă care specifică zona căreia se va adresa operandul (I pentru intrări, Q pentru ieşiri, C pentru contoare, T pentru timer-e, etc. ), iar parametrul reprezintă adresa operandului (un bit şi un byte de adresă).

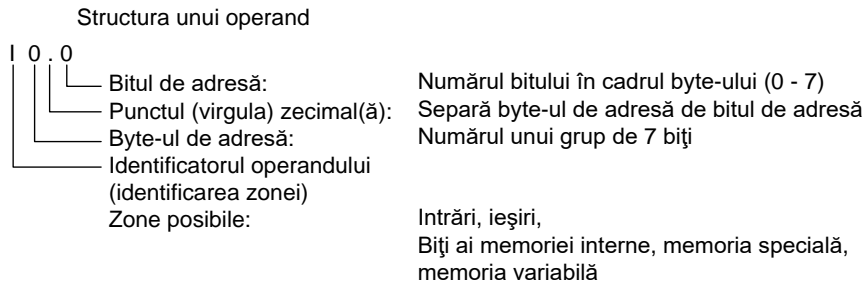


Fig. 3.20 – Structura unui operand

### 3.4 Instrucţiuni uzuale utilizate pentru programarea tip LAD

Instrucţiunile care pot fi utilizate în cadrul diagramelor LAD sunt grupate în funcţie de specificul lor în 3 categorii: Basic Instructions, Extended Instructions şi Technology. Fiecare dintre acestea conţin mai multe clase de funcţii, cele mai uzuale fiind prezentate în cele ce urmează. Pe lângă aceste directoare de funcţii, ramura de instrucţiuni mai conţine folder-ul „Favorites”, unde pot fi adăugate instrucţiunile cele mai utilizate, pentru a evita căutarea acestora în arborele de instrucţiuni. Librăriile pot fi pre-definite (furnizate de Siemens contra cost) sau create de către utilizator pentru a se adapta propriilor necesităţi. Librăriile pot conţine subrutine pre-configurate pentru lucrul cu diverse echipamente, rutine de întreruperi ce uşurează lucrul cu drivere de control sau instrucţiuni pentru comunicaţii. În plus, ramura de instrucţiuni conţine şi o zonă pentru apelarea de subrutine.

## Instrucțiunile de bază

Instrucțiunile de bază pot fi regăsite în partea din dreapta a ferestrei TIA Portal, conținând 10 clase de instrucțiuni și zona "General", care include elemente generale de editare a diagramei LAD.

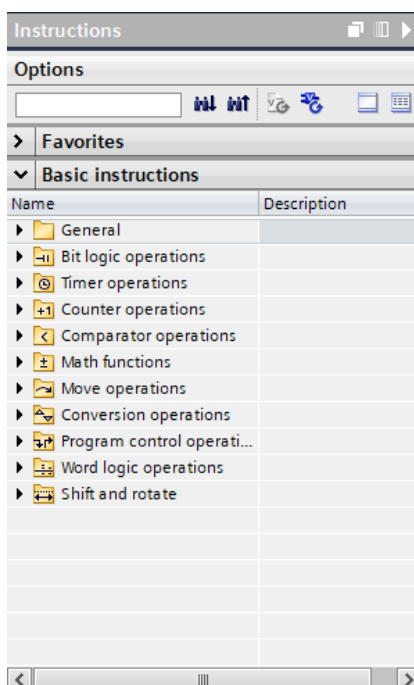


Fig. 3.21 – Instrucțiuni de bază

### Categoria "General"

În cadrul acestei categorii sunt incluse elemente care permit editarea diagramelor LAD: inserarea unei rețele noi, a unui bloc nou sau deschiderea și închiderea unei ramuri în paralel în cadrul rețelei existente. Aceste operațiuni pot fi executate și cu ajutorul comenzilor rapide (de exemplu Shift + F8 pentru deschidere ramură nouă) sau prin apăsarea simbolurilor (dacă acestea au fost salvate în prealabil în categoria "Favorite") disponibile deasupra zonei de editare.

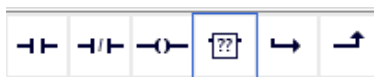


Fig. 3.22 – Comenzi rapide pentru editarea diagramei



AND NOT și OR NOT, similar transferului în memorie a valorii logice negate a bitului asociat cu contactul NC. Reprezentarea în FBD este similară cu cea a contactului NO, blocurile funcțiilor AND și OR putând avea până la 32 de intrări.

În această categorie intră și funcția de negare **NOT**, simbolizată în cadrul diagramelor LAD cu  $\neg$ , realizând schimbarea stării logice a semnalului prezent la intrarea în bloc.

În continuare regăsim două funcții specifice, instrucțiunile de detectare a tranzițiilor pozitive și negative. Prima dintre acestea determină trecerile din 0 în 1 logic ale semnalului de la intrarea în funcție și oferă la ieșirea asociată un impuls de durată unui ciclu intern al automatului programabil. În mod similar, funcția de detectare a tranzițiilor negative va determina o trecere din 1 logic în 0 logic a stării semnalului de intrare și va realiza trecerea în stare 1 logic a semnalului de ieșire din bloc. În cadrul diagramelor LAD cele două funcții sunt reprezentate prin  $\uparrow^P$  și  $\downarrow^N$ , iar în cadrul programelor realizate în STL vom întâlni **EU** (Edge Up), respectiv **ED** (Edge Down). Un fapt ce trebuie luat în considerare la utilizarea acestor instrucțiuni este comportamentul lor la alimentarea automatului programabil: dat fiind faptul că ambele funcții necesită determinarea schimbării stării semnalului de la intrare, la primul ciclu de funcționare al PLC-ului starea ieșirilor nu va fi schimbată. Începând cu al doilea ciclu de funcționare instrucțiunile vor putea modifica starea ieșirilor.

### Bobine

Un prim tip de instrucțiuni din această categorie este „Output”, ce permite asocierea unui bit cu starea logică a unei ramuri de circuit. Acestei ieșiri, reprezentate cu simbolul  $\overset{\text{bit}}{\neg}$ , i se poate atașa, prin intermediul bitului, o ieșiri fizică a PLC-ului, un numărător, un temporizator sau o locație de memorie. În această categorie este disponibilă și varianta negată a bobinei de ieșire, numită ”Negate Assignment”, precum și variantele ”Set output” și ”Reset output” care realizează trecerea în stare 1 logic (respectiv 0 logic) a operandului specificat. Spre deosebire de utilizarea ieșirii convenționale, la dispariția semnalului de activare, starea rămâne setate pe 1 logic (respectiv 0 logic) până în momentul în care aceasta este schimbată în mod activ de o instrucțiune complementară. Un alt set de instrucțiuni disponibile este perechea ”Set\_BF” – ”Reset\_BF”, care permite setarea unui număr de operanzi cu adrese consecutive (BF – Bit Field).



## Blocuri

În această categorie regăsim instrucţiunile cu bistabil Set și Reset denumite Set/Reset flip-flop, respectiv Reset/Set flip-flop. Prima instrucţiune reprezintă un releu în cadrul căruia instrucţiunea de Reset este dominantă, adică în situaţia în care atât semnalul de Set (S) cât și cel de Reset (R1) sunt în stare 1 logic ieşirea (OUT) va fi în stare 0 logic. În cazul instrucţiunii Reset/Set flip-flop, dacă ambele intrări, Set (S) și Reset (R1) sunt 1 logic, ieşirea (OUT) va fi 1 logic. Bitul asociat acestor funcţii reprezintă parametrul care va fi setat sau resetat. În cadrul programării de tip LAD instrucţiunile vor fi reprezentate prin blocurile prezentate mai jos.

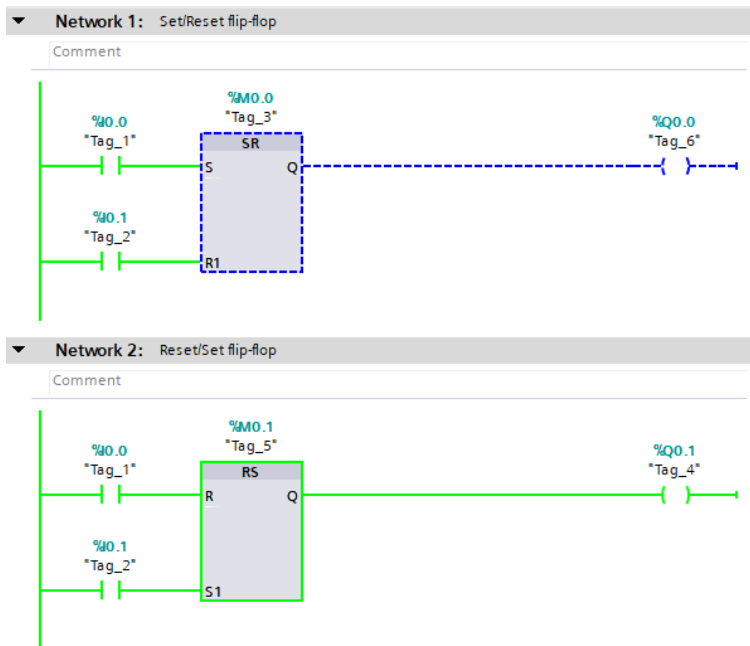


Fig. 3.24 – Instrucţiunile Set/Reset flip-flop și Reset/Set flip-flop

## Exemple

Pentru o mai bună înţelegere a instrucţiunilor prezentate, în continuare vor fi prezentate câteva programe de o complexitate redusă, urmărind doar familiarizarea cu mediul de lucru TIA Portal.

- **Ex BL-1**

În cadrul ”project tree” se regăsește și zona ”Program Block”, unde se pot gestiona programele care vor rula pe automatul programabil. Aceste programe pot fi de mai multe tipuri: OB – Organization blocks, FC – Functions, FB – Function Blocks, DB – Global Data Blocks sau Array of Data Blocks. După inițializare automatul programabil va rula în mod continuu programul principal (OB), celelalte tipuri de programe putând fi accesate din OB în cazul apariției unui eveniment declanșator (apelare subrutine sau funcții, semnale primite pe porturile de comunicații, etc.). Pentru exemplele prezentate în cele ce urmează toate instrucțiunile vor fi incluse în programul principal.

În acest exemplu se va realiza o funcție de tip **SI**, fiind necesară utilizarea unei singure rețele în cadrul diagramei LAD. Pentru a ușura munca de depanare a programelor, TIA Portal oferă posibilitatea introducerii de comentarii atât la nivelul programului cât și la nivelul fiecărei rețele în parte, precum și utilizarea de nume pentru rețelele utilizate, facilități importante în cazul realizării de programe de o complexitate ridicată, unde numărul de rețele utilizate este mare.

Pentru realizarea unui program se vor insera pe rând elementele necesare în rețeaua selectată în prealabil. În cadrul acestui exercițiu se vor folosi un contact normal deschis (NO), un contact normal închis (NC) și o bobină (OUT), legate în serie. Pentru inserarea contactului NO pot fi utilizate mai multe metode: alegerea blocului din zona ”Bit logic operations” și mutarea lui în diagrama prin ”Drag-and-Drop” sau simplu prin dublu-click, selectarea blocului din zona de instrucțiuni favorite sau prin utilizarea combinației de taste ”Shift + F2”. În mod similar se vor insera în program și celelalte instrucțiuni necesare.

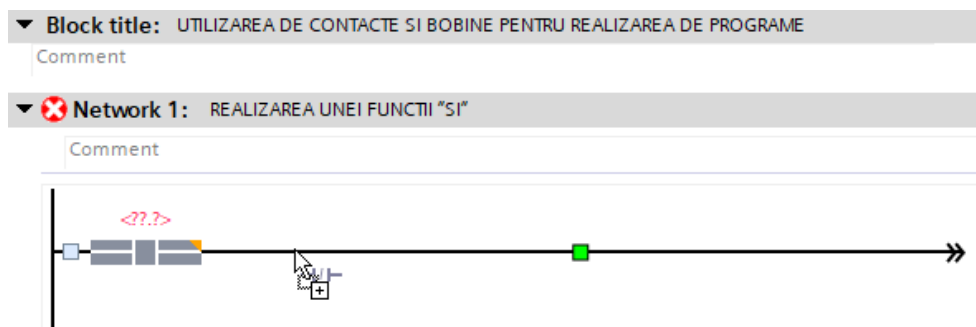


Fig. 3.25 – Inserarea unui contact NC

În Figura 3.26 se poate observa că blocurile inserate în prima rețea a diagramei LAD nu sunt configurate, eroare fiind semnalată și de simbolul din dreptul numelui acestei rețele. Pentru a putea rula programul este necesară asocierea unor operanzi pentru fiecare dintre cele 3 elemente; pentru acest prim exemplu vor fi folosite primele două intrări digitale ale automatului existent în proiect, **I0.0** și **I0.1**, respectiv prima ieșire fizică, **Q0.0**, dar pot fi folosite și locații din memoria PLC-ului.

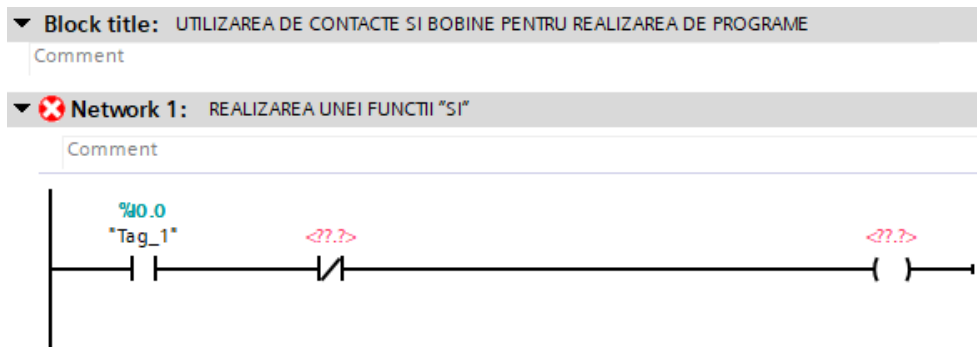


Fig. 3.26 – Asocierea porturilor fizice elementelor programului

După ce a fost finalizat acest pas este necesară descărcarea programului de pe calculator pe PLC, prin alegerea opțiunii "Download to device" din meniul Online sau din bara de unelte sau prin utilizarea combinației de taste "Ctrl + L". La primul transfer de date poate să fie necesară reidentificarea automatului programabil, similar cu procesul prezentat anterior la trecerea PLC-ului în modul RUN / STOP. După finalizarea transferului automatul programabil se comută în modul RUN și programul începe să ruleze. TIA Portal permite realizarea unei conexiuni în timp real între PC și PLC, prin alegerea opțiunii "Go online" (Ctrl + K), iar prin activarea monitorizării programului (Ctrl + T sau opțiunea "Monitoring ON/OFF") se poate observa în TIA Portal starea pe care o au ieșirile PLC-ului. În cazul în care conexiunea este reușită vor apărea simboluri verzi în dreptul elementelor active din proiect în zona de "Project Tree". Figura 3.27 a) prezintă interfața programului când intrarea **I0.0** este inactivă, drept urmare funcția **ȘI** returnează 0 logic la ieșirea **Q0.0**. Se poate observa că simbolul intrării **I0.1** este activ, fiind vorba despre un contact tip NC. În Figura 3.27 b) este prezentată grafica TIA Portal în situația în care **I0.0** este comutat, deci devine **ON**, drept urmare și **Q0.0** devine **ON**. Fiind vorba despre o funcție **ȘI**, dacă intrarea **I0.1** devine activă, prin asocierea acesteia cu un contact NC, semnalul se va întrerupe, **Q0.0** devenind **FALSE**.

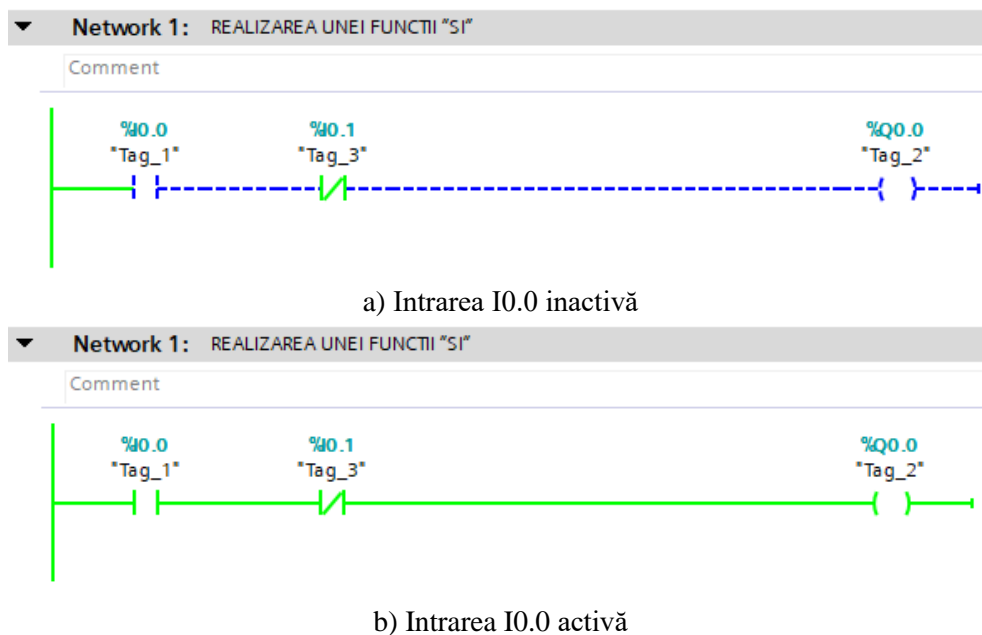


Fig. 3.27 – Monitorizarea funcției ȘI în TIA Portal

TIA Portal oferă și posibilitatea încărcării pe PC a programului existent în memoria automatului programabil, acest lucru fiind realizat cu ajutorul butonului "Upload from device"; se recomandă salvarea prealabilă a programului deja deschis, deoarece programul descărcat de pe PLC va fi scris peste acesta.

- **Ex BL-2**

În cadrul acestui exemplu se va realiza o buclă de automenținere similară celei realizate în capitolul anterior cu ajutorul **Logo!**, program ce implică utilizarea unor elemente noi față de programul anterior. În mod similar realizării funcției ȘI în cadrul exemplului BL-1 se vor introduce comentarii pentru program și rețele, precum și nume pentru rețele. Pentru realizarea acestei funcții vom folosi aceleași elemente ca și în exemplul anterior: un contact NO, un contact NC și o bobină.

Circuitul cu buclă de automenținere reprezintă de fapt o combinație între o funcție „ȘI” și o funcție „SAU”, fiind necesară realizarea unei diagrame LAD care să conțină două ramuri în paralel. După realizarea părții superioare a circuitului sau păstrarea blocurilor din exemplul anterior, este necesară inserarea laturii inferioare a rețelei. Pentru a realiza acest lucru este se poate utiliza comanda "Open Branch" din categoria Basic Instructions > General, din zona de comenzi Favorite sau utilizând

combinația de taste "Shift + F8". Pe noua ramură se inserează un contact NO care va fi atașat intrării **Q0.0**. Pentru a realiza funcția logică **SAU** este necesară închiderea ramurii prin conectarea acesteia la ramura principală, între **I0.0** și **I0.1**, ceea ce se poate realiza cu comanda "Close Branch" (Shift + F9) sau cu ajutorul mouse-ului.

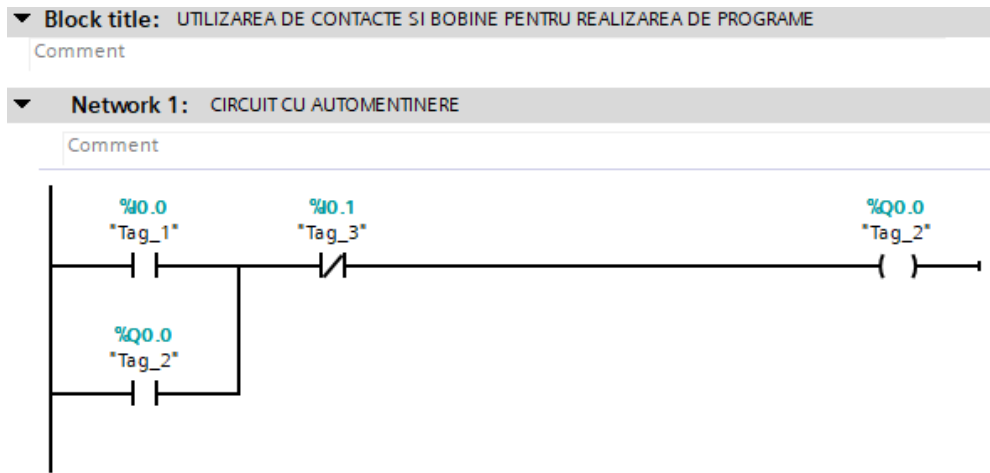


Fig. 3.28 – Implementarea unui circuit cu automenținere

Programul este finalizat, pentru transferul acestuia în memoria S7-1200 fiind repetată procedura prezentată anterior, nefiind necesară setarea conexiunii și căutarea automatului programabil dacă acest lucru a fost realizat în prealabil la exemplul anterior. Funcționarea programului poate fi monitorizată cu ajutorul TIA Portal, în Figura 3.29 fiind pusă în evidență utilizarea ieșirii **Q0.0** ca input în program pentru a obține funcționarea sistemului și după dispariția semnalului de pe **I0.0**.

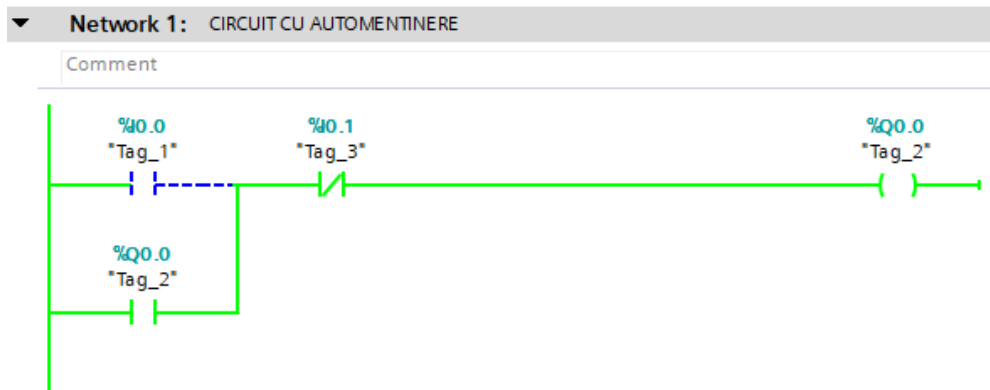


Fig. 3.29 – Funcționarea circuitului cu automenținere

- **Ex BL-3**

Pentru familiarizarea cu funcțiile ”Set\_BF” și ”Reset\_BF” se va realiza în continuare o ramură de program ce permite modificarea simultană a stării unui număr de porturi de ieșire ale automatului programabil. Pentru simplitate, se va înlocui programul anterior prin eliminarea întregii rețele sau prin ștergerea tuturor blocurilor folosite anterior. Pentru realizarea programului se vor folosi 3 contacte tip **NO** și două bobine. După realizarea ramurii superioare a programului, între contactele **NO** atașate intrărilor **I0.0** și **I0.1** se va realiza, cu ajutorul ”Open Branch”, cea de-a doua ramură, în paralel cu prima. Cele două bobine inserate anterior vor fi modificate cu ajutorul listei disponibile la realizarea unui click pe simbolul bobinei, lista din care se vor alege funcțiile ”Set\_BF” și ”Reset\_BF”. Parametrii instrucțiunilor vor primi valoarea 6, respectiv 2, ceea ce va determina ca în situația în care ramura superioară a programului este activă 6 porturi de ieșire, începând cu **Q0.0** vor deveni **ON**, iar când circuitul de pe ramura inferioară va fi închis, 2 porturi de ieșire, **Q0.1** și **Q0.2** vor deveni **OFF**. După încărcarea programului în memoria PLC-ului și trecerea acestuia în modul RUN se poate observa că dacă **I0.0** și **I0.1** sunt **ON** ieșirile **Q0.0 – Q0.5** vor deveni active, iar la trecerea **I0.2** în stare 1 logic ieșirile **Q0.1** și **Q0.2** vor deveni inactive, celelalte 4 ieșiri rămânând active până la întreruperea rulării programului. În situația în care toate intrările sunt active se observă că funcția Reset\_BF are prioritate, deci ieșirile **Q0.1** și **Q0.2** vor deveni inactive.

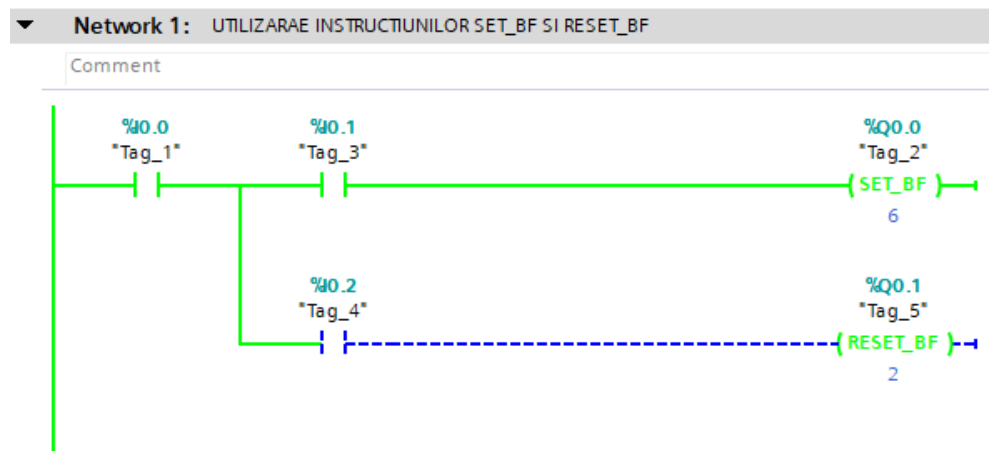


Fig. 3.30 – Implementarea funcțiilor Set\_BF și Reset\_BF

### **Categoria „Timer Operations”**

Acest grup conţine 10 instrucţiuni care permit lucrul cu ceasul intern al automatului programabil:

- **TP – Generate Pulse:** Permite generarea unui impuls cu o durată prestabilită, setată cu ajutorul parametrului **PT**, a semnalului de ieşire. La apariţia unui front pozitiv pe borna **IN** a blocului se va comuta ieşirea **Q** în stare **ON** şi se va începe măsurarea timpului trecut. La expirarea timpului **PT** ieşirea va deveni **OFF**, indiferent de starea intrării **IN** sau de eventuala apariţie a unui alt front pozitiv la această bornă. Valoarea parametrului **PT** se introduce implicit în milisecunde, dar valoarea poate fi introdusă şi direct în secunde dacă valoarea numerică este urmată de caracterul ”s”, în minute dacă se adaugă caracterul ”m”, ore cu ajutorul caracterului ”h”, zile dacă se utilizează caracterul ”d”; timpul scurs de la activarea blocului poate fi accesat la borna **ET**;
- **TON: Generate on-delay:** Funcţia permite introducerea unei întârzieri, egală cu valoarea setată în parametrul **PT**, la comutarea ieşirii în stare **ON** după apariţia unui front pozitiv la borna **IN** a blocului. La dispariţia semnalului de pe **IN** ieşirea va comuta în starea **OFF**;
- **TOF: Generate off-delay:** Blocul funcţionează similar cu funcţia anterioară, dar în acest caz ieşirea **Q** comută în stare **ON** instantaneu la detectarea unei tranziţii pozitive pe borna **IN** şi va comuta în stare **OFF** cu o întârziere dată de **PT** la apariţia unei tranziţii negative pe borna **IN**;
- **TONR: Time accumulator:** Funcţionează similar cu blocul **TON** prezentat anterior, dar, spre deosebire de acesta, la dispariţia semnalului de pe borna **IN** starea ieşirii **Q** rămâne **ON** până la detectarea unei tranziţii pozitive la borna **R**;
- **---( TP )---** **Start pulse timer:** instrucţiunea funcţionează similar cu blocul **TP**, un exemplu de implementare fiind prezentat în Figura 3.31. Detectarea unei tranziţii pozitive pe borna de intrare a instrucţiunii **TP** va determina trecerea ieşirii în stare **ON** şi cronometrarea timpului şi compararea acestuia cu valoarea setată în cadrul funcţiei; la depăşirea acestei valori starea instrucţiunii devine **OFF**. Cea de-a doua reţea a programului accesează locaţia de memorie asociată funcţiei **TP** şi scrie starea acesteia în ieşirea fizică **Q0.0**;

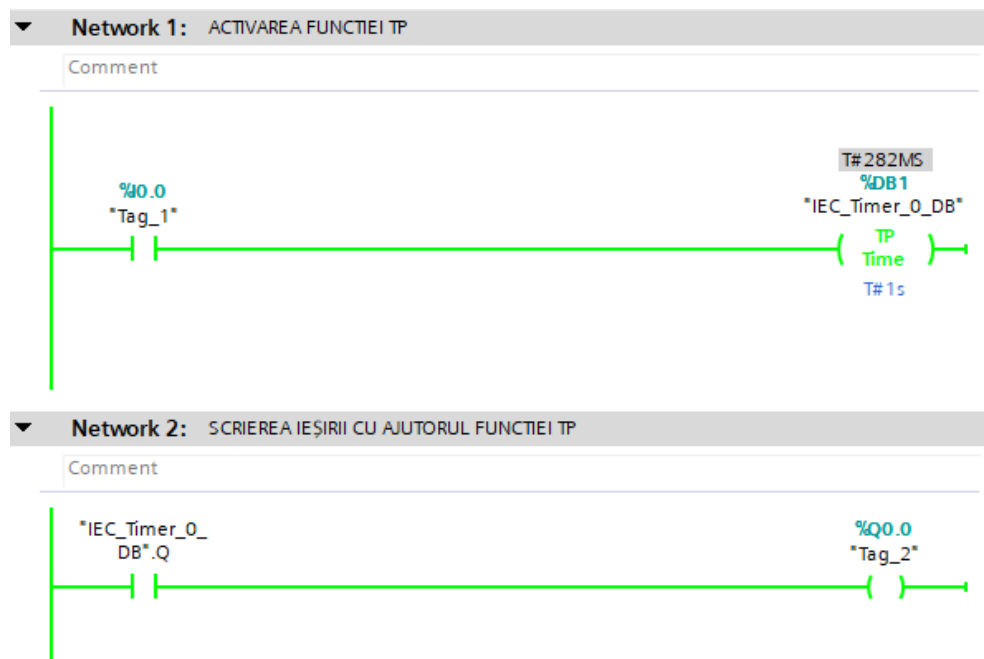


Fig. 3.31 – Implementarea instrucțiunii TP

- ---( TON )---: **Start on-delay timer**: funcționează similar cu instrucțiunea TON prezentată anterior, iar implementarea se face similar cu instrucțiunea (TP);
- ---( TOF )---: **Start off-delay timer**: funcționează similar cu instrucțiunea TOF prezentată anterior, iar implementarea se face similar cu instrucțiunea (TP);
- ---( TONR )---: **Time accumulator**: funcționează similar cu instrucțiunea TONR prezentată anterior, iar implementarea se face similar cu instrucțiunea (TP);
- ---( RT )---: **Reset timer**: permite resetarea valorii timpului măsurat de una dintre funcțiile prezentate anterior;
- ---( PT )---: **Load time duration**: această funcție permite scrierea unei valori ca parametru de timp pentru una dintre funcțiile prezentate anterior, permițând realizarea unor programe dinamice.



### **Categoria „Counter operations”**

Instrucţiunile de numărare sunt utilizate frecvent în automatizările industriale, motiv pentru care TIA Portal conţine 3 funcţii de tip counter: **CTU** (Count Up), **CTD** (Count Down) şi **CTUD** (Count Up/Down).

Numărătorul incremental **CTU** (Count Up Instruction) va creşte cu o unitate valoarea curentă a contorului la fiecare trecere din 0 în 1 logic a semnalului conectat la borna **CU**, iar la atingerea valorii impuse cu ajutorul intrării **PV** (Preset Value) bitul asociat numărătorului va deveni 1 logic. Dacă semnalul la borna **R** (Reset) devine 1 logic, valoarea numărătorului se resetează şi rămâne 0 până la dispariţia semnalului de pe borna **R**. La atingerea valorii maxime 32.767 numărătorul nu va mai fi incrementat.

Cea de-a doua instrucţiune din categoria „Counter operations” este **CTD** (Count Down), care funcţionează similar numărătorului incremental, realizând însă o scădere a valorii curente a contorului la fiecare tranziţie pozitivă a semnalului conectat la borna **CD**. Valoare iniţială a numărătorului este dată de parametrul **PV** (Preset Value) iar la atingerea valorii 0 bitul asociat numărătorului **CTD** va deveni 1 logic. La apariţia unui semnal la borna **LD** (Load) valoarea curentă a contorului va fi resetată la valoarea iniţială **PV**. După atingerea valorii minime, 0, semnalul de la borna **CD** nu influenţează starea instrucţiunii **CTD**.

**CTUD** (Count Up/Down) combină cele două funcţii prezentate anterior. La detecţia unei tranziţii pozitive a semnalului de pe borna **CU** a instrucţiunii valoarea **CTDU** va creşte cu 1, iar la apariţia unei treceri din 0 în 1 logic a semnalului conectat la borna **CD** contorul va scădea cu 1. Starea de 1 logic a semnalului conectat la borna **R** va reseta valoarea numărătorului la 0, stare ce nu va fi modificată de **CU** sau **CD** atât timp cât **R** este activă. La atingerea valorii impuse de parametrul **PV** bitul asociat cu instrucţiunea **CTUD** va deveni 1 şi va rămâne în această stare cât timp valoarea curentă a contorului este mai mare decât valoarea **PV**. Spre deosebire de primele două instrucţiuni din categoria „Counter”, parametrul **PV** poate avea valori negative, iar la atingerea valorii maxime 32.767, la apariţia unui nou semnal la borna **CU** contorul va trece la valoarea minimă -32.768, fiind valabilă şi situaţia inversă, cu atingerea pragului minim.

### Exemple

- **Ex COUNT-1**

În acest prim exemplu se va urmări funcționarea unei instrucțiuni Count Up (CTU), care necesită conectarea a două semnale la bornele **CU** și **R**, reprezentând semnalul care va realiza incrementarea numărătorului cu o unitate la fiecare apariție a unei tranziții pozitive, respectiv semnalul care va realiza resetarea contorului la valoarea inițială, 0. La aceste borne se vor conecta intrările **I0.0** și **I0.1** ale automatului programabil, iar valoarea la care ieșirea contorului va deveni activă se setează la borna **PV**; în acest caz a fost aleasă valoarea 10. La atingerea valorii setate în parametrul **PV** ieșirea **Q** devine 1 logic și va rămâne în această stare până la întreruperea alimentării sau la apariția unui semnal la borna **R**, ce va aduce valoarea curentă a numărătorului la 0. Pentru a vizualiza starea numărătorului, la borna **Q** va fi conectată ieșirea **Q0.0** a automatului programabil. În acest fel, la depășirea valorii **PV** ieșirea **Q0.0** devenind activă. În mod similar se va realiza conectarea unui contor de tip Count Down (CTD), motiv pentru care această instrucțiune nu va fi abordată separat.

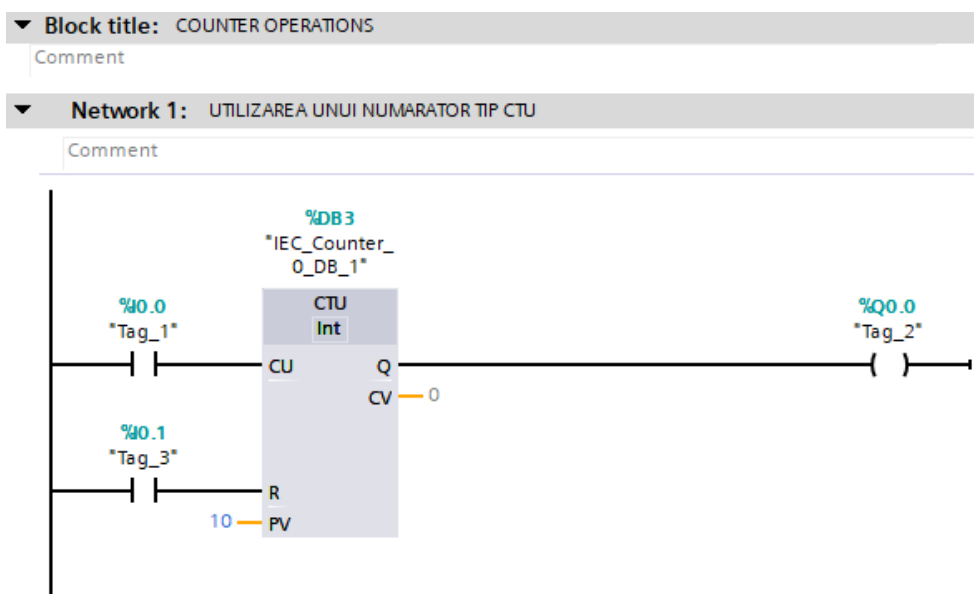


Fig. 3.32 – Utilizarea instrucțiunii Count up

- **Ex COUNT-2**

O instrucţiune care include funcţiile numărătorului incremental şi ale celui decremental este Count Up/Down (**CTUD**). Instrucţiunea monitorizează comutările 0 – 1 logic pe bornele **CU** şi **CD**, valoarea numărătorului fiind incrementată, respectiv decrementată, în funcţie de acestea; aceste borne au fost conectate la intrările **I0.0**, respectiv **I0.1**. Borna de reset este conectată la intrarea **I0.2**, iar pragul contorului a fost setat la 10 la borna **PV**. La detectarea unei tranziţii pozitive la borna **LD** a instrucţiunii **CTUD**, unde a fost conectată intrarea **I0.3**, valoarea contorului va fi actualizată automat cu valoarea setată în parametrul **PV**. La borna **QU** a instrucţiunii **CTUD** a fost conectată ieşirea **Q0.0**, ieşirea **QD** nefiind conectată. Pentru a vizualizarea stării acesteia a fost realizată încă o reţea în cadrul programului unde se accesează parametrul "**IEC\_Counter\_0\_DB\_1**".**QD** la care este conectată ieşirea **Q0.1**.

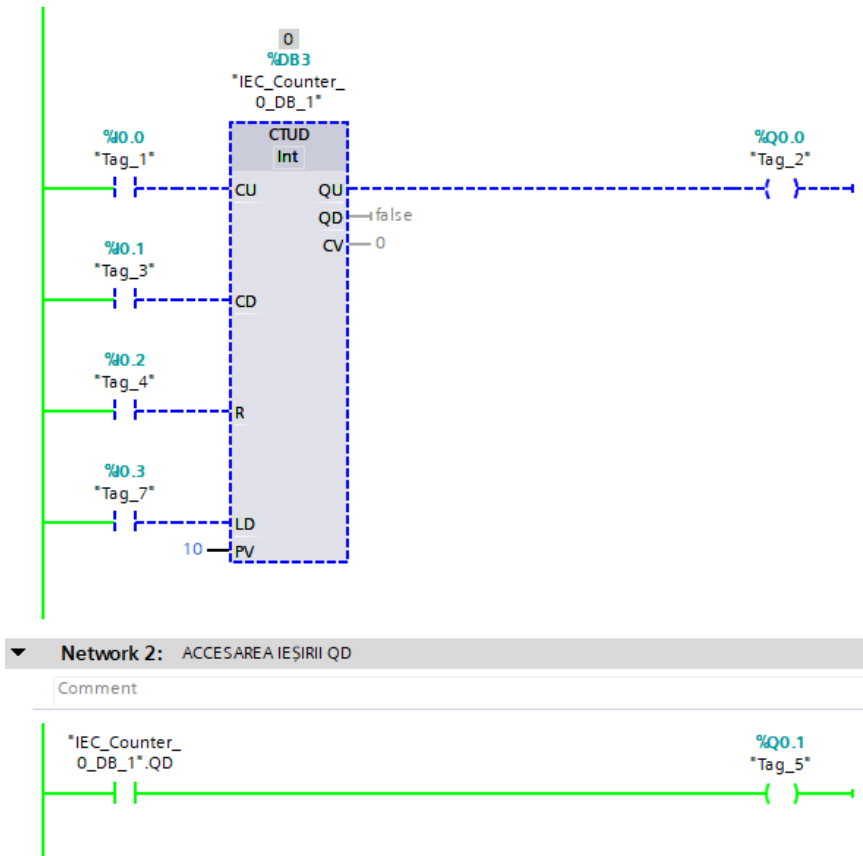


Fig. 3.33 – Utilizarea instrucţiunii CTUD

Funcționarea instrucțiunii **CTUD** poate fi urmărită în Figura 3.34, ieșirea **QD** fiind activă la pornirea programului, pentru că valoarea contorului **CV** este 0. După apariția primului semnal pe intrarea **CU** ieșirea **QD** devine **FALSE**, rămânând în această stare până când valoarea contorului redevine  $\leq 0$ . La fiecare front pozitiv al semnalului de pe intrarea **CU** valoare contorului crește, iar când valoarea **CV** este mai mare sau egală decât valoarea setată la borna **PV** ieșirea **QU** devine **ON**. În cazul apariției de semnale la intrarea **CD** contorul va fi decrementat, iar când **CV** devine mai mic decât **CV** ieșirea **QU** devine **OFF**.

La apariția unui semnal pe intrarea **R** valoarea **CV** devine 0, deci **QU** devine **OFF** și **QD** devine **ON** și rămâne în această stare și când **CV** va avea valori negative. Când este detectată o tranziție pozitivă pe intrarea **LD** contorul **CV** va lua valoarea parametrului **PV**, deci **QU** devine **ON** și **QD** devine **OFF**.

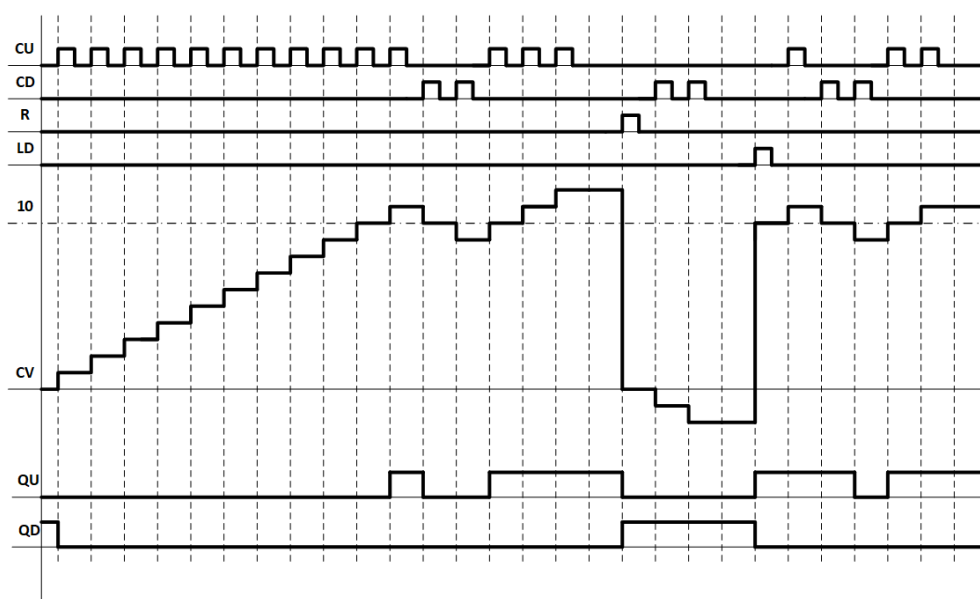


Fig. 3.34 – Funcționarea instrucțiunii CTUD

### **Categoria „Comparator operations”**

În cadrul acestei categorii se regăsesc instrucțiuni care permit compararea valorilor operanzilor nominalizați. Instrucțiunile acoperă funcțiile de comparare uzuale din limbajele de programare clasice: egal, diferit (not equal), mai mare sau egal, mai mic sau egal, mai mare, mai mic, valoare în interval, valoare în afara

intervalului, precum și două funcții care verifică dacă valoarea de la borne este un număr cu virgulă mobilă sau nu. Instrucțiunile din această categorie pot să manipuleze o varietate mare de tipuri de date: șiruri de Bit, numere întregi, numere cu virgulă mobilă, șiruri de caractere, timer-e, format dată și oră, matrice, STRUCT, VARIANT, ANY și PLC data type.

### Exemple

- **Ex Compare & Timer**

În cadrul acestui exemplu se va realiza un temporizator cu funcție de reset automat, iar valoarea timpului trecut de la declanșarea măsurării, va fi utilizată pentru a exemplifica funcțiile de comparare. În acest sens se va utiliza o instrucțiune de tip „On-Delay”, cu valoarea timpului de întârziere setată în parametrul **PT** la 1200 ms. Instrucțiunea este activată de valoarea stocată în locația de memorie **M0.0**. Pentru că inițial valoarea înscrisă în **M0.0** este 0 logic această valoare este atașată unui contact de tip **NC**, la prima scanare a programului timer-ul fiind activat.

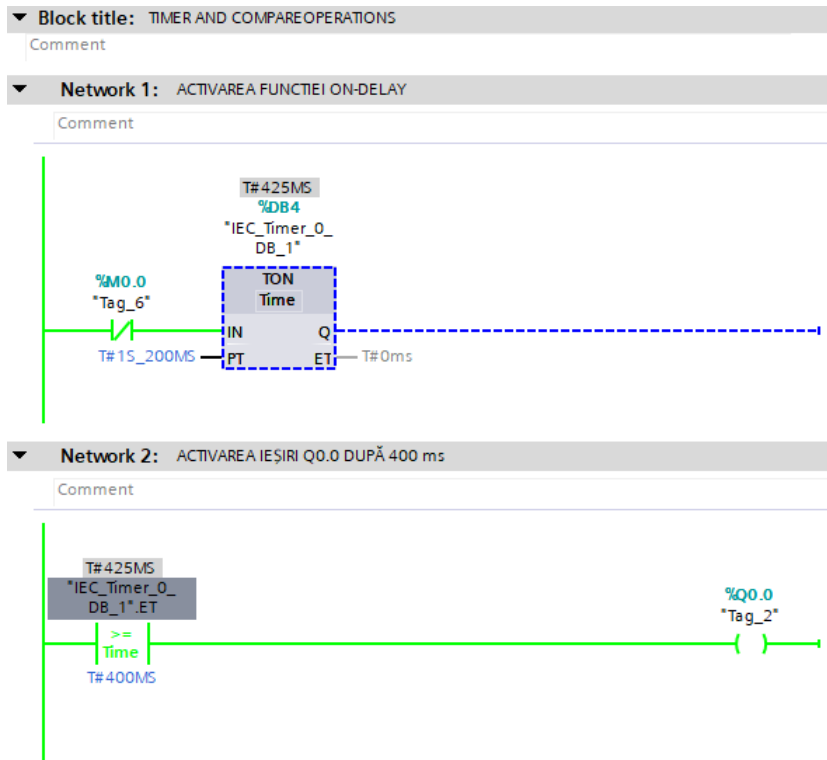


Fig. 3.35 – Activarea temporizatorului și a primei ieșiri

În a doua rețea a programului va realiza activarea unei ieșiri pentru o parte din durata ciclului realizat cu ajutorul instrucțiunii **TON**. În acest sens se va utiliza un bloc de comparare de tip „Greater Than Or Equal Integer”, care va compara valoarea curentă preluată din **"IEC\_Timer\_0\_DB\_1".ET** cu o valoare de 400 ms. Dacă rezultatul comparării este adevărat instrucțiunea va returna 1 logic, lucru transmis ieșirii **Q0.0** atașate acestei ramuri, care va fi activă pentru restul ciclului setat, adică pentru 0,8 secunde. În rețeaua 3 au fost inserate două instrucțiuni de comparare, „Greater than” și „Less than”, permițând identificare intervalului între 400 și 800 ms din fiecare ciclu de funcționare și activarea ieșirii **Q0.1** în acest interval.

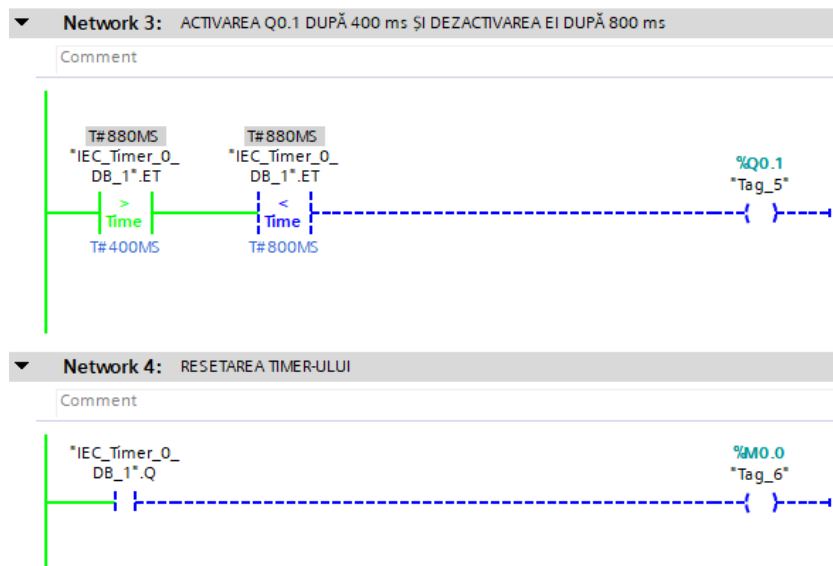


Fig. 3.36 – Resetarea ciclului temporizatorului

Pentru ca programul să realizeze un timer cu auto-resetare, așa cum s-a propus, este necesară reinițializarea timer-ului. Acest lucru se realizează în cadrul rețelei 4 a programului, ce conține un contact **NO** ieșirii **Q** a instrucțiunii **"IEC\_Timer\_0\_DB\_1".Q** și o bobină atașată memoriei **M0.0**. La atingerea pragului setat în cadrul funcției **TON** în prima ramură a programului contactul **NO** va trece în stare 1 logic, determinând și bitul asociat **M0.0** să devină 1 logic. Cum valoarea negată a acestui bit de memorie este cea care declanșează timer-ul din prima rețea a programului, blocul **TON** va deveni inactiv pentru durata unui ciclu intern al PLC-ului, determinând astfel resetarea întregului sistem.

### **Categoria „Math operations”**

În cadrul acestei categorii regăsim instrucţiuni care permit efectuarea de operaţiuni matematice de bază: adunare (**ADD**), scădere (**SUB**), înmulţire (**MUL**), împărţire (**DIV**), restul împărţirii (**MOD**), negare (**NEG**), incrementarea valorii la următoare valoare (**INC**), decrementare (**DEC**), valoare absolută (**ABS**), ridicare la pătrat (**SQR**), radical (**SQRT**), logaritm natural (**LN**), exponenţială (**EXP**), sinus (**SIN**), cosinus (**COS**), tangentă (**TAN**), şi alte tipuri de funcţii trigonometrice. În funcţie de modelul automatului programabil utilizat, o parte dintre aceste funcţii nu vor fi disponibile. Aceste instrucţiuni pot manipula numere reale, în mod uzual fiind necesare două valori la intrările **IN1** şi **IN2** asupra cărora se va aplica operaţiunea matematică aleasă, rezultatul fiind scris în locaţia de memorie definită de borna **OUT**, atât timp cât borna **EN** este activă.

### **Categoria „Move operations”**

Pentru manipularea informaţiilor stocate în diverse locaţii de memorie sunt utilizate instrucţiunile din categoria „Move operations”. Prima instrucţiuni este **Move** aceasta permiţând manipularea de date din registrul de memoria specificat la intrarea **IN** în locaţia specificată de **OUT**, fără ca aceste date să fie modificate. Instrucţiunea **Move\_BLK** permite manipularea de informaţii de formate diferite, preluând o parte din informaţia regăsită la memorie începând cu locaţia specificată de **IN** şi o mută în memoria începând cu adresa specificată de **OUT**. Lungimea blocului de date ce este manipulat este specificată de parametrul **COUNT**, care va defini numărul de bytes, cuvinte sau cuvinte duble care vor fi mutate.

Printre instrucţiunile din această categorie regăsim şi instrucţiunea **SWAP** permite accesarea unei locaţii de memorie specificată la intrarea **IN** şi înlocuirea **MSB** (most significant byte) cu **LSB** (least significant byte) din cadrul cuvântului găsit la adresa respectivă.

### **Categoria „Conversion operations”**

O primă instrucţiune este **Convert** permite conversia numerică a între diferite formate de date. Astfel vom regăsi la intrarea şi ieşirea acestor blocuri

formatele de date: șiruri de bit, numere întregi, numere cu virgulă mobilă, CHAR, WCHAR, BCD16 și BCD32.

În cadrul acestei categorii se regăsesc și instrucțiunile **ROUND**, ce va converti un număr real într-un număr întreg dublu, rotunjind la cel mai apropiat număr întreg, **CEIL** și **FLOOR**, care realizează rotunjirea în sus sau în jos spre următorul număr întreg și **TRUNC**, care transformă un număr real într-un număr întreg dublu, eliminând cifrele de după virgula zecimală. Ultimele două instrucțiuni sunt **SCALE\_X**, care realizează scalarea valorii de la intrare într-un interval definit la bornele **MIN** și **MAX**, după formula  $OUT = [VALUE * (MAX - MIN)] + MIN$  și instrucțiunea **NORM\_X**, care funcționează după formula  $OUT = (VALUE - MIN) / (MAX - MIN)$ .

### **Categoria „Program control operations”**

În cazul realizării unor programe complexe sunt utile instrucțiunile care permit controlul programelor prin realizarea de bucle repetabile sau de salturi condiționate. În cadrul acestei categorii regăsim instrucțiunile **JMP** și **JMPN** care întrerup funcționarea liniară a programului și realizează saltul la o anumită rețea din cadrul programului, pe baza etichetelor folosite pentru identificarea rețelilor. Diferența între cele două instrucțiuni este faptul că **JMP** se activează în prezența unui semnal **ON** la intrare, iar **JMPN** devine activă când semnalul de la intrare este **OFF**.

În cadrul categoriei mai regăsim instrucțiunile **LABEL**, **JMP\_LIST** care permite realizarea unei liste cu destinații ale salturilor din program și condițiile pentru fiecare dintre ele, **SWITCH** sau **RET**, fiecare dintre ele oferind diferite de opțiuni pentru rularea neliniară a programelor.

### **Categoria „Word logic operations”**

Instrucțiunile logice **ȘI**, **SAU** și **SAU EXCLUSIV** sunt incluse în această categorie, permițând realizarea de funcții logice între semnalele conectate la bornele **IN1** și **IN2**, rezultatul fiind disponibil la borna **OUT**. În funcție de tipul semnalelor de intrare se poate alege ca instrucțiunea să opereze cu bytes (**Byte**), cuvinte (**Word**) sau cuvinte duble (**DWord**). Pe lângă aceste instrucțiuni este disponibilă și instrucțiunea **INVERT**, instrucțiune care permite realizarea complementului



semnalului de la intrare și încărcarea acestuia la locația de memorie specificată la borna **OUT**. În plus regăsim instrucțiunile **DECO** și **ENCO**, prima dintre ele permițând crearea la borna **OUT** a unei date de tip **DWord** conținând elemente 0 și valoarea 1 la poziția indicată la borna **IN**; în mod similar, instrucțiunea **ENCO** identifică poziția în informația de tip **DWord** a bitului cel mai puțin semnificativ (LSB) și scrie această valoare la borna **OUT**.

O altă instrucțiune disponibilă este Select (**SEL**), care permite transmiterea la borna **OUT** a valorii de la borna **IN0** sau **IN1**, în funcție de starea intrării **G**: pentru **G** cu valoare 0 logic se va transmite la **OUT** valoarea de la borna **IN0**, iar pentru **G** egal cu 1 logic se va transmite valoarea de la borna **IN1**. O versiune mai complexă a acestei funcții este instrucțiune Multiplex (**MUX**), care permite conectarea unui număr de până la 32 de intrări. În funcție de valoarea parametrului **K** la ieșirea **OUT** a instrucțiunii va fi transmisă informația de la intrarea **IN\_K**. Dacă valoarea parametrului **K** este mai mare decât numărul de intrări disponibile, la ieșirea **OUT** va fi disponibilă valoarea conectată la intrarea **ELSE**. Instrucțiunea Demultiplex (**DEMUX**) funcționează în sens invers, permițând extragerea elementului cu numărul **K** din șirul de valori disponibil la intrarea **IN**.

### **Categoria „Shift and Rotate”**

Instrucțiunile din această categorie permit manipularea datelor aflate la locația de memorie de la portul **IN**. Shift Right (**SHR**) și Shift Left (**SHL**) deplasează spre dreapta sau spre stânga cu **N** poziții informația aflată în memorie la locația specificată de **IN**, înlocuind cu 0 biții eliminați. În funcție de tipul informație manipulate **N** poate avea valori maxime de 8 (pentru byte), 16 (pentru word) și 32 (pentru double word). Instrucțiunile Rotate Right (**ROR**) și Rotate Left (**ROL**) realizează rotirea valorii de intrare în funcție de numărul specificat la **N**, valoare obținută fiind scrisă în **OUT**.

## Instrucțiuni speciale

Pe lângă instrucțiunile de bază prezentate în paginile anterioare, TIA Portal pune la dispoziția utilizatorilor și categoria "Extended Instructions" care conține elemente necesare pentru realizarea unor programe de complexitate mai ridicată.

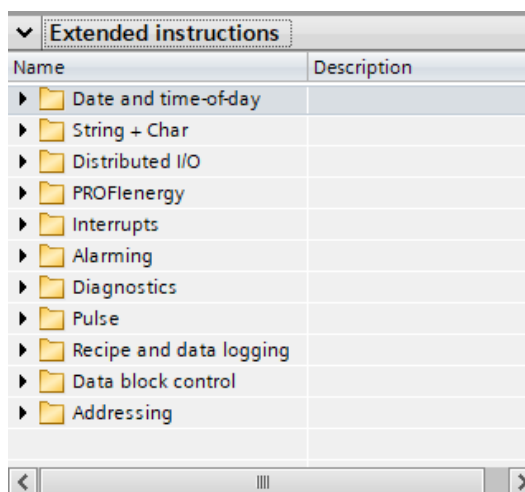


Fig. 3.37 – Instrucțiuni speciale

În cadrul acestei categorii regăsim următoarele tipuri de instrucțiuni:

- **Date and time-of-day** – instrucțiunile din această categorie permit manipularea datelor în format "dată", fiind posibile conversia, citirea, scrierea acestor date (ora locală, zone de fus orar, contoare de utilizare) sau operațiunile între două sau mai multe valori;
- **String + Char** – permit manipularea și conversia informațiilor sub formă de șiruri de caractere, inclusiv determinarea dimensiunilor unui șir, extragerea sau inserarea de elemente, concatenarea a două șiruri, căutarea în interiorul unui șir, etc.;
- **Process image** – disponibile doar pentru modele superioare din familia de automate programabile S7;
- **Distributed I/O** – utilizate pentru citirea și scrierea de date sau alarme utilizând protocoale de comunicații (Profinet, Profibus, AS-i)
- **PROFIenergy** – permite controlul echipamentelor pentru care se vizează realizarea unui management al energiei consumate, fiind disponibile instrucțiuni de pornire, oprire, realizare de liste de priorități ale echipamentelor, etc.

- **Module parameter assignment** – disponibile doar pentru modele superioare din familia de automate programabile S7;
- **Interrupts** – conține instrucțiuni care permit atașarea unui bloc de instrucțiuni (OB) unor evenimente de tip ”întrerupere”, cauzate de evenimente de comunicații, evenimente ciclice în timpul funcționării automatului programabil, evenimente cauzate de oră/zi, timp de întârziere sau evenimente asincrone;
- **Alarms** – pentru automatele programabile din clasa S7-1200 este disponibilă doar instrucțiunea de generare de mesaje pentru utilizatori în cazul apariției unor evenimente și încărcarea acestor mesaje în bufferul de diagnosticare;
- **Diagnostics** – conține instrucțiuni utilizate în diagnosticarea funcționării automatului programabil, putând fi citite informații generate în timpul utilizării în diferite momente ale ciclului de funcționare;
- **Pulse** – include două instrucțiuni: **CTRL\_PWM** și **CTRL\_PTD**, care permit realizarea se impulsuri. Prima instrucțiune permite realizarea de semnale cu lățime variabilă a impulsurilor (tip PWM), păstrând durata ciclului constantă și modificând doar factorul de umplere (duty cycle), iar funcția **CTRL\_PTD** permite generarea unui semnal cu o frecvență prestabilită;
- **Recipes and data logging** – conține instrucțiuni pentru importarea și exportarea de ”rețete” în format .csv de pe și spre cardurile de memorie disponibile, precum și instrucțiuni care permit utilizatorului să salveze valori de proces în jurnale salvate pe cardurile de memorie sau să citească și să manipuleze aceste jurnale;
- **Data block functions** – include instrucțiuni utile pentru crearea, scrierea, citirea, ștergerea locurilor de date sau editarea informațiilor despre aceste blocuri;
- **Addressing** – conține instrucțiuni care permit obținerea de informații despre componentele hardware conectate în cadrul proiectului.
  - **Tags, zoom pe PLC, drag&drop tags**
  - **Exemple programe**
  - **Va urma**

### 3.5 Exemple de programe

Realizarea de programe în modul Ladder oferă avantajul major al ușurinței transformării circuitelor electrice clasice în programe compatibile cu automatele programabile. Un mod simplu de a realiza această transformare este de a roti schema electrică în sens trigonometric cu 90 de grade astfel încât linia de alimentare cu energie electrică să devină orizontală, într-o poziție similară liniei comune din programele Ladder. Elementele circuitului electric (întrerupătoare, contactoare, bobine) vor fi înlocuite cu corespondentele lor disponibile în librăriile TIA Portal. Un exemplu de transformare a unui circuit de alimentare și reversare de sens pentru un motor electric a fost prezentat anterior, în Figura 3.15.

#### **Exemplul 1 – nivel lichid în bazin**

Programele prezentate până în acest moment au folosit în mod exclusiv operanzi pentru a defini contactele, bobinele sau blocurile din cadrul diagramelor. În cazul programelor ce conțin un număr mare de rețele este dificil de urmărit corespondența operanzilor cu intrările / ieșirile reale. Pentru a preîntâmpina această problemă softul TIA Portal oferă posibilitatea asocierii de nume în format text (TAG) elementelor din diagramele realizate.

Pentru exemplificarea utilizării notării simbolice în cadrul programelor se consideră o problemă clasică de automatizare: păstrarea nivelului de lichid într-un bazin între anumite limite. Se vor folosi în acest sens doi senzori de nivel pentru determinarea nivelului maxim și minim a lichidului în bazin, un buton de pornire a sistemului automat și un buton de avarie, care să poată întrerupe funcționarea întregului sistem. Este necesară și o pompă de apă care va alimenta bazinul atunci când nivelul lichidului scade sub minim.

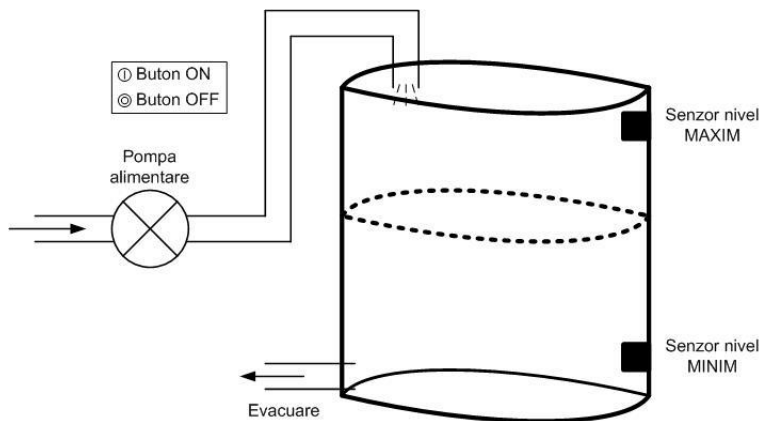


Fig. 3.38 – Schema sistemului ce urmează a fi automatizat

Pentru realizarea exemplului sunt considerate intrările și ieșirile din tabelul următor.

Element fizic	Conexiune la PLC	Tip bloc utilizat
<b>Buton pornire</b>	I0.0	Contact tip NO
<b>Buton oprire</b>	I0.1	Contact tip NC
<b>Senzor nivel minim</b>	I0.2	Contact tip NO
<b>Senzor nivel maxim</b>	I0.3	Contact tip NC
<b>Pompă alimentare</b>	Q0.0	Bobină ieșire

Cele 4 semnale de la senzori și butoanele de control vor fi conectate la 4 intrări ale automatului programabil, alimentarea pompei fiind comandată de ieșirea PLC-ului. Programul este simplu, conținând o singură rețea, după cum se poate observa în Figura 3.39.

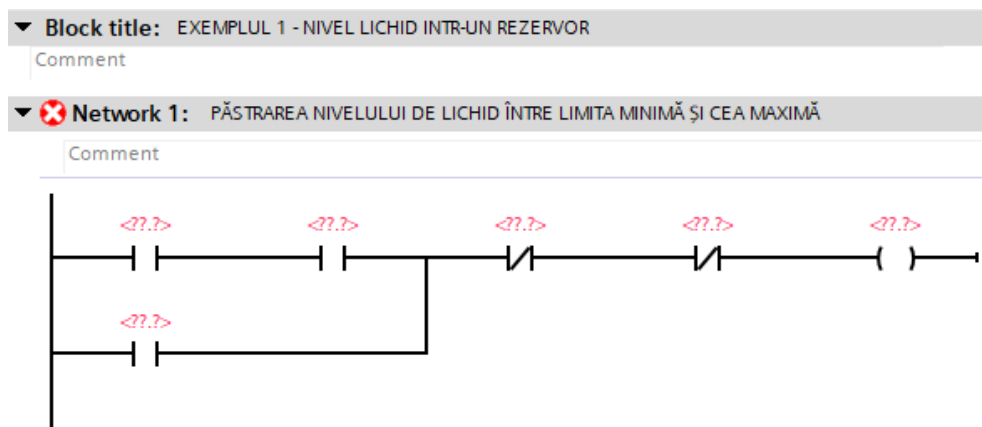


Fig. 3.39 – Realizarea programului în TIA Portal

Urmărirea funcționării programului este dificilă, chiar dacă sunt folosite doar 4 intrări și o ieșire, în cazul programelor mai complexe acest lucru devenind mult mai greu. În acest caz se poate utiliza notarea simbolică. Un prim pas este crearea simbolurilor și atașarea acestora operanzilor din diagrama LAD. Tabelul de simboluri poate fi accesat din "Project tree">PLC tags. Din această zonă se va accesa tabelul implicit (Default tag table), care nu va include nici un element. Se vor realiza cele 5 elemente necesare în programul considerat, folosind etichetele din tabelul anterior. Pentru ultimul element inserat, "Pompă alimentare" este necesară modificarea tipului de operand, din intrare **I** în ieșire **Q**, precum și modificarea numărului bit-ului din 4 în 0.

Default tag table								
	Name	Data type	Address	Retain	Acces...	Writa...	Visibl...	Comment
1	Buton pornire	Bool	%I.0	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
2	Buton oprire	Bool	%I.1	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
3	Senzor nivel minim	Bool	%I.2	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
4	Senzor nivel maxim	Bool	%I.3	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
5	Pompă alimentare	Bool	%I.4	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
6	<Add new>							

Operand identifier: Q

Operand type: Q

Address: M

Bit number: 0

Fig. 3.40 – Tabelul cu etichetele utilizate în cadrul programului

Asocierea elementelor din diagrama LAD cu etichetele definite anterior poate fi făcută prin mai multe metode, cea mai simplă fiind selectarea etichetei corespunzătoare direct din diagramă, cum este prezentat și în Figura 3.41

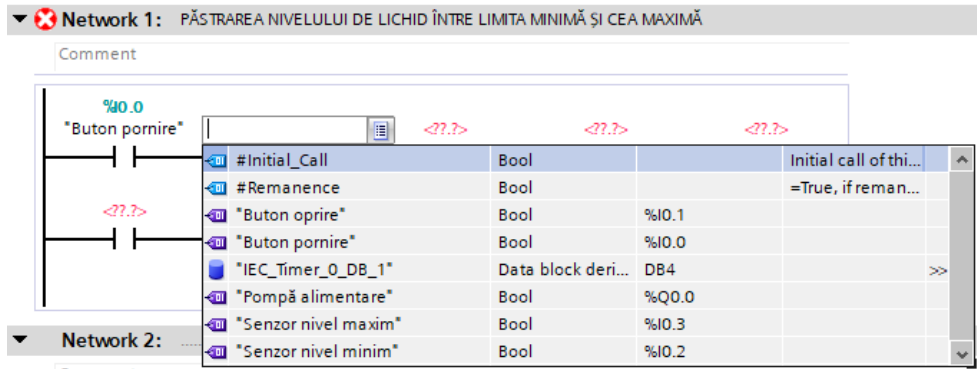


Fig. 3.41 – Asocierea etichetelor direct în digrama LAD

O altă abordare implică împărțirea zonei de lucru în două ferestre folosind opțiunile din meniul "Window" și plasare pe una dintre zonele de lucru a imaginii automatului programabil (prin realizarea unui dublu click pe zona "Device configuration" din "Project tree"), respectiv a diagramei LAD în cea de-a doua fereastră. Dacă nivelul de zoom se mărește suficient, pot fi văzute etichetele atașate intrărilor fizice ale automatului programabil.

Asocierea acestora cu elementele din diagrama LAD poate fi realizată prin Drag&Drop pe contactul sau bobina corespunzătoare. În Figura 3.42 este exemplificată asocierea ieșirii **Q0.0** "Pompă alimentare" cu elementul "Assignment" din diagrama LAD. În mod similar pot fi asociate etichetele cu elementele din program prin preluarea etichetele din "PLC tags" și realizarea Drag&Drop pe contactul sau bobina corespunzătoare.

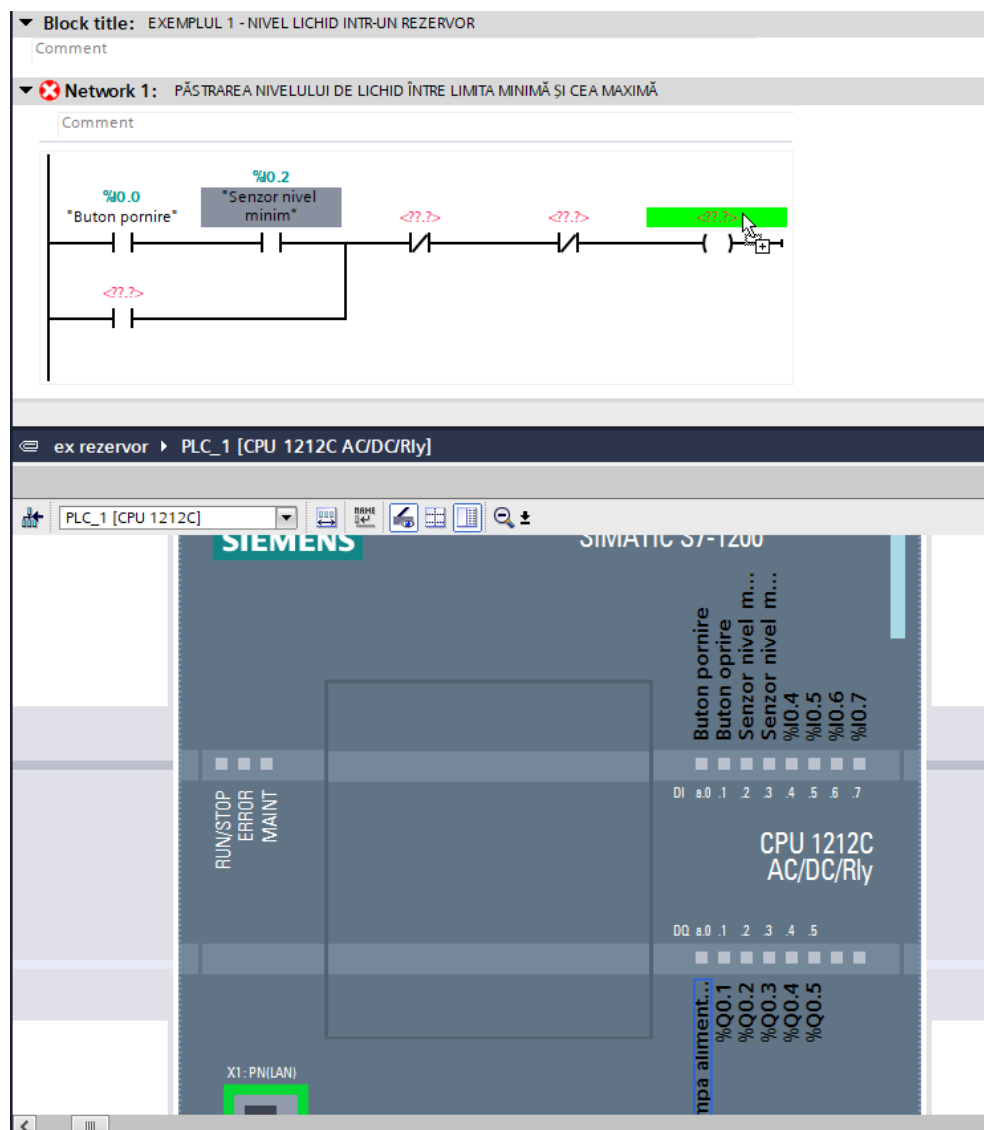


Fig. 3.42 – Asocierea porturilor cu elementele din diagrama LAD

După finalizarea programului este necesară copierea acestuia în memoria automatului programabil, utilizând opțiunea "Download to device". Pentru a putea urmări funcționarea programului se trece în modul de lucru "Online" și se activează opțiunea "Monitor" din meniul TIA Portal.



Pentru exemplul anterior, în cazul în care cele 4 intrări sunt în stare **OFF**, interfaţa programului va arăta ca în imaginea de mai jos, elementele ”Buton oprire” şi ”Senzor nivel maxim” fiind active datorită utilizării de contacte tip NC.

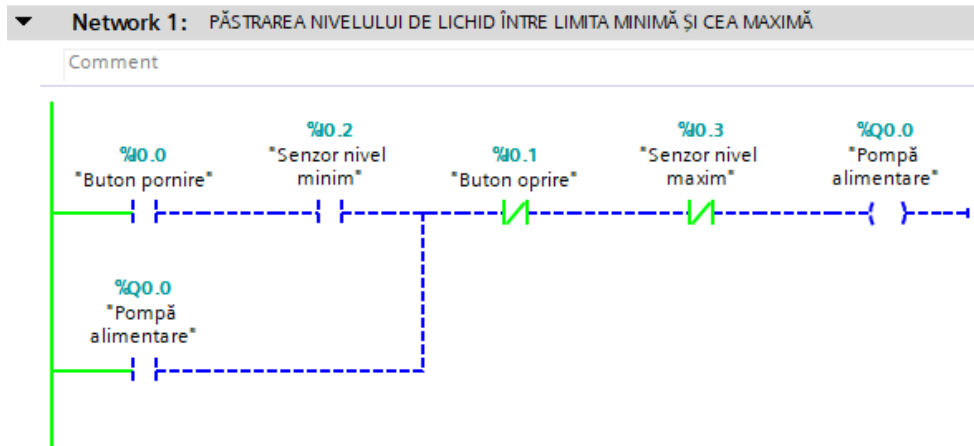


Fig. 3.43 – Monitorizarea programului – fără intrări active

Pentru ca ieşirea ”Pompă alimentare” să devină activă este necesar ca şi primele 2 elemente ale reţelei să aibă starea 1 logic. În momentul în care butonul de pornire devine 1 logic, la scăderea nivelului lichidului sub nivelul minim, ieşirea PLC-ului, conectată la pompa de alimentare, devine 1 logic.

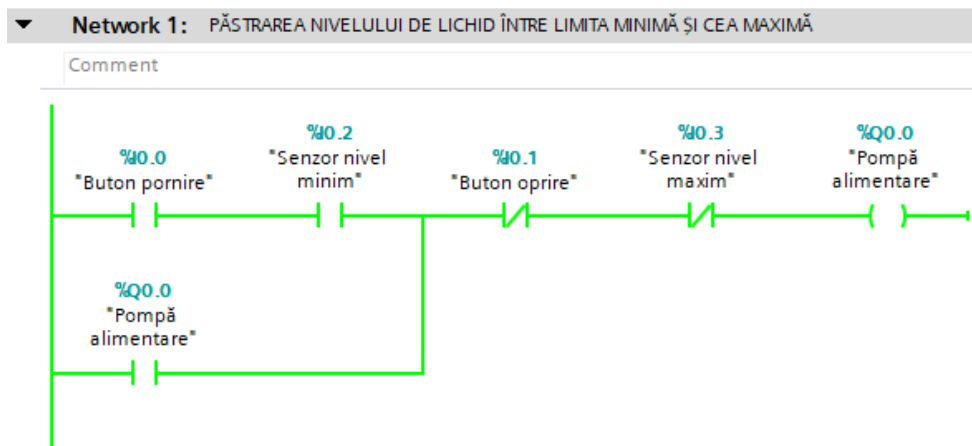


Fig. 3.44 – Starea intrărilor pentru activarea ieşirii

Odată cu creșterea nivelului de lichid din bazin, blocul MIN va deveni 0 logic, dar pompa va continua să funcționeze, datorită modului în care este realizat programul, până când se va atinge nivelul maxim dorit sau va fi acționat butonul de oprire. Pompa rămâne activă și în cazul în care butonul de oprire devine inactiv.

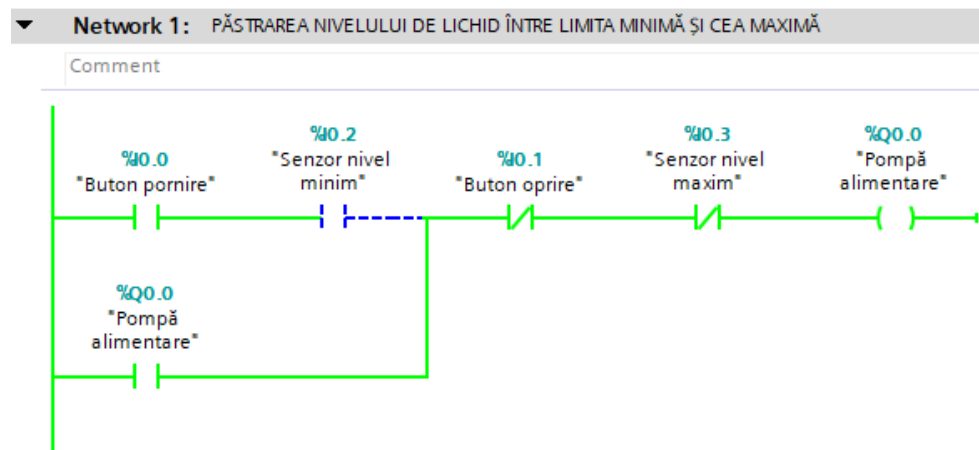


Fig. 3.45 – Funcționarea programului după ce s-a depășit nivelul minim

În automatizări se pot utiliza butoane de pornire / oprire cu revenire sau cu blocare, atașate unor contacte normal închis sau normal deschis. Alegerea combinației potrivite se face în funcție de o serie de factori, în continuare fiind prezentate câteva situații care pot fi întâlnite des în practică. În cazul utilizării unui buton de pornire cu blocare în cazul exemplului de mai sus, dacă se acționează butonul de oprire de urgență, iar buton de pornire a rămas blocat, la eliberarea butonului de oprire întregul sistem repornește, lucru care trebuie evitat. În consecință se va folosi un buton cu revenire pentru pornire și unul cu blocare pentru oprire, în acest caz sistemul neputând fi repornit accidental atât timp cât butonul de oprire este apăsat. Totuși, după operarea acestei modificări se observă că programul realizat nu mai funcționează în toate cazurile posibile: dacă butonul de pornire este apăsat cât timp nivelul de lichid este desupra nivelului minim (deci senzorul nivel minim este inactiv) pompa nu va funcționa. După ce butonul de oprire redevine în starea **OFF**, la apariția unui semnal de la senzorul nivel minim sistemul rămâne inactiv. În consecință programul trebuie regândit, o modificare posibilă fiind prezentată în cele ce urmează.

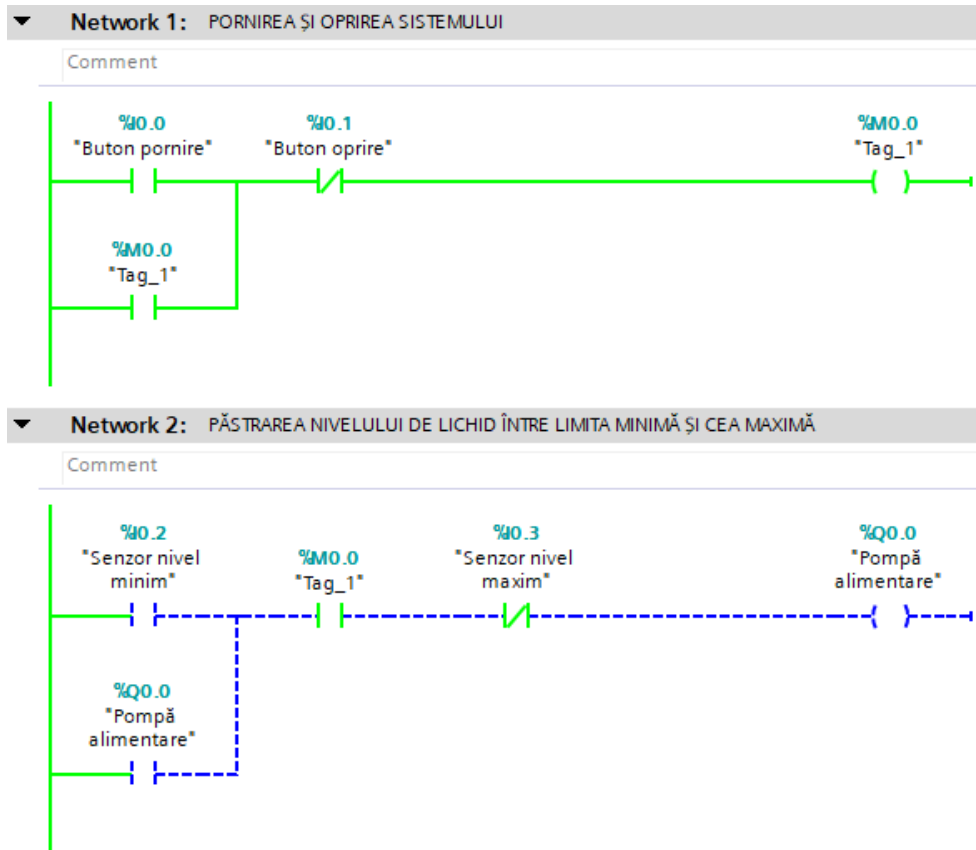


Fig. 3.46 – Implementarea programului cu ajutorul a două rețele

Prin introducerea unei noi rețele cu o buclă de automenținere care să conțină logica de pornire/oprire a sistemului și trecerea rezultatului într-o locație de memorie se rezolvă problema butonului de pornire prezentată anterior. Locația de memorie este apelată în a doua rețea a programului, cât timp **M0.0** este activă programul realizând lucrul dorit, păstrarea nivelului de lichid în intervalul dorit.

Un alt aspect ce trebuie considerat este utilizarea în anumite situații de contacte tip **NC** în locul celor **NO**. În exemplul de mai sus, butonul de oprire este atașat unui contact **NO**, a cărui valoare este negată în programul realizat. În acest caz, elementul de program trece în stare 0 logic doar în situația în care butonul **OFF** este activ. Dacă firele ce transmit semnalul de la buton la automatul programabil se întrerup, programul nu mai poate fi oprit, fiind imposibilă închiderea circuitului pe acea intrare. Dacă în schimb se va folosi un contact **NC** atașat unui contact Normally Open TIA Portal, o întrerupere a firelor determină trecerea automată în stare de 0

logic a blocului și oprirea programului. Prin urmare, se recomandă utilizarea de contacte NC acolo unde eventualele defecte ale circuitului electric pot introduce erori ce pot duce la deteriorarea componentelor sistemului sau chiar la accidente.

### **Exemplul 2 – Detectarea tranziției pozitive a unui semnal**

În acest exemplu se va prezenta realizarea și modul de funcționare al unui întrerupător acționat de impulsuri, ocazie cu care se va reaminti modul de funcționare al instrucțiunilor de detectare al tranzițiilor semnalului. Scopul exemplului este de a realiza un sistem care să schimbe starea ieșirii, conectată la o lampă, la fiecare închidere a unui contact conectat la intrarea automatului programabil, indiferent de durata acestui impuls. Programul trebuie ca la fiecare trecere a semnalului de intrare din stare 0 logic în stare 1 logic să determine starea actuală a ieșirii și să realizeze schimbarea acestei stări. Trecea intrării în stare OFF nu trebuie să influențeze starea ieșirii.

Pentru determinarea tranziției semnalului de intrare din stare OFF în stare ON se va folosi instrucțiunea „Scan operand for positive transition”, disponibilă în categoria ”Bit logic operations”, instrucțiune care va genera un impuls cu durata unui ciclu intern a automatului programabil la fiecare detectare a unei tranziții pozitive a semnalului de intrare (închidere a unui contact); în mod similar, instrucțiunea „Scan operand for negative transition” detectează tranzițiile negative ale semnalului. Modul de funcționare al instrucțiunii Positive Transition poate fi urmărit pe diagrama de timp prezentată mai jos.

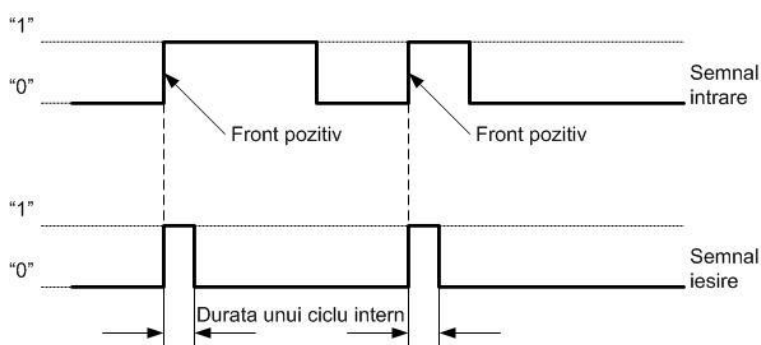


Fig. 3.47 – Modul de funcționare al instrucțiunii ”Scan operand for positive transition”

Programul va conţine două reţele, prima dintre ele conţinând instrucţiunile necesare detectării modificării stării intrării şi stabilirii stării ieşirii. Impulsurile ce apar la trecerea intrării din stare **OFF** în stare **ON** sunt conectate în serie cu contacte **NC** şi **NO** care determină starea actuală a ieşirii, rezultatul acestor funcţii modificând starea locaţiei de memorie **M0.1**, fiind utilizate instrucţiunile **Set output** şi **Reset output**. A doua reţea a programului atribuie ieşirii **Q0.0** starea locaţiei de memorie scrise în prima reţea.

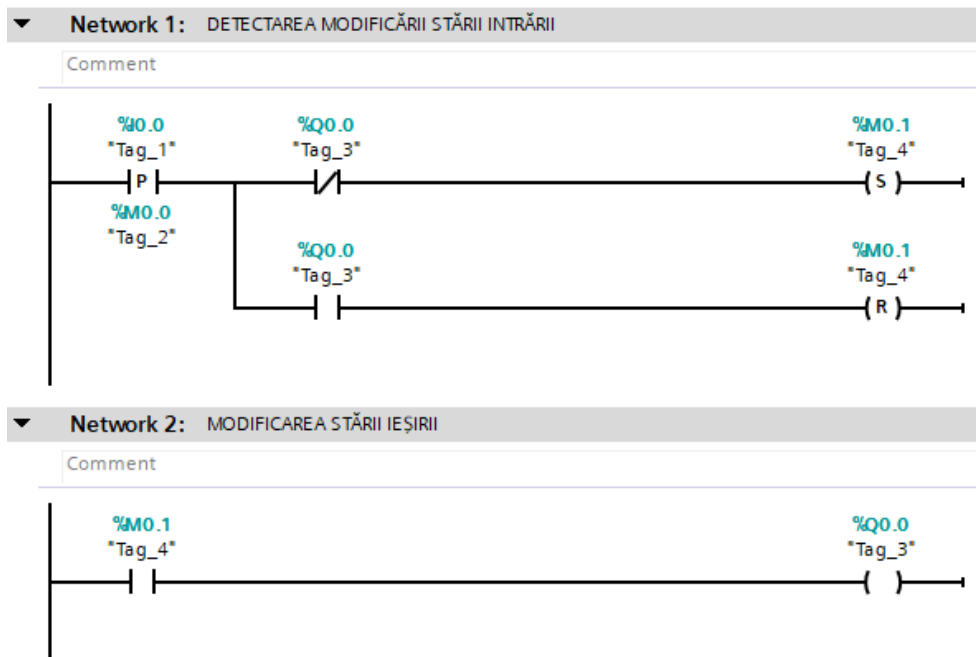


Fig. 3.48 – Programul implementat în TIA Portal

Funcționarea programului este prezentată în cele ce urmează:

Pasul 1: La prima rulare a programului starea ieșirii **Q0.0** este **OFF**, în consecință contactul **NC** atașat **Q0.0** din prima ramură a programului va fi în stare 1 logic, restul elementelor fiind în stare 0 logic.

Pasul 2: La apariția unui semnal pe intrarea **I0.0** instrucțiunea "Pozitive Transition" va genera un impuls de durata unui ciclu interna a automatului programabil, ce va determina activarea instrucțiunii **Set** (atașată **M0.1**) în stare 1 logic. În același timp instrucțiunea **Reset** devine inactivă, permițând astfel modificarea stării bitului de memorie în 1 logic. În consecință și starea ieșirii devine **ON**, iar în la următoarea parcurgere a programului starea contactelor **NO** și **NC** atașate **Q0.0** se va schimba.

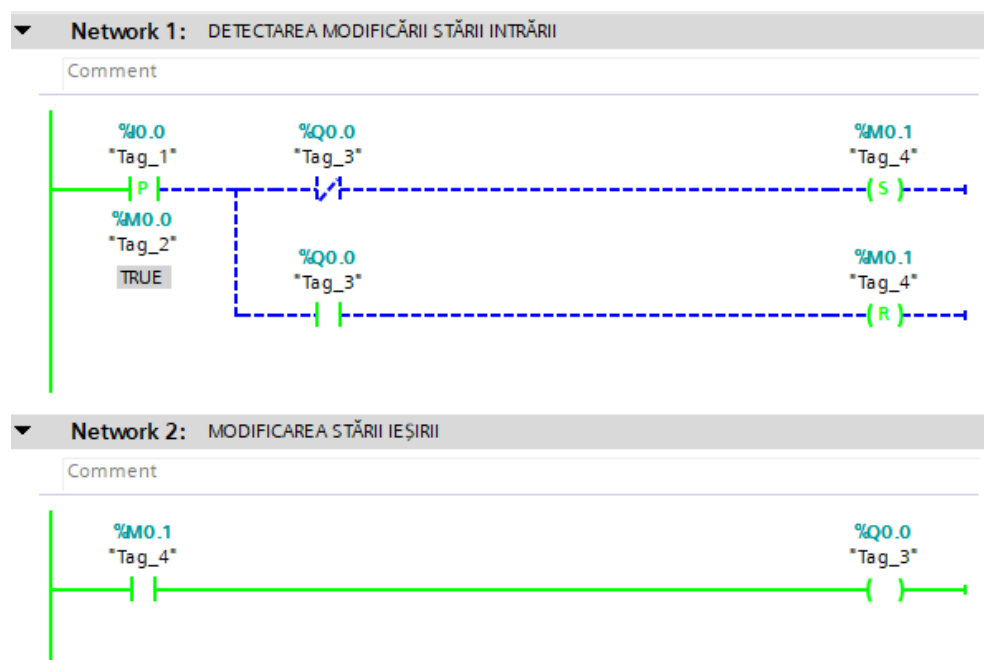


Fig. 3.49 – Activarea ieșirii la apariția unei tranziții pozitive

Dacă nu s-ar utiliza bitul de memorie **M0.1** pentru modificarea stării ieșirii, ci s-ar conecta în prima rețea a programului direct ieșirea **Q0.0** starea acesteia ar fi modificată în aceeași parcurgere a programului de 2 ori, fiind setată de prima ramură și resetată de cea de-a doua, rezultatul dorit nefiind obținut în acest caz. Altfel spus,

utilizarea **M0.1** previne suprascrierea valorii **Q0.0**. Starea ieşirii nu se modifică nici la dispariţia semnalului de intrare.

Pasul 3: Apariţia unui nou semnal la intrarea **I0.0** este generat un nou impuls care va activa instrucţiunea **Reset**, **M0.1** devenind **OFF**, la fel ca şi ieşirea **Q0.0**. În consecinţă, în următorul ciclu se va întâlni o stare a programului identică cu cea din primul pas. Starea ieşirii va fii întotdeauna **ON** după un număr impar de acţionării ale contactului conectat la intrare şi **OFF** după un număr par de impulsuri.

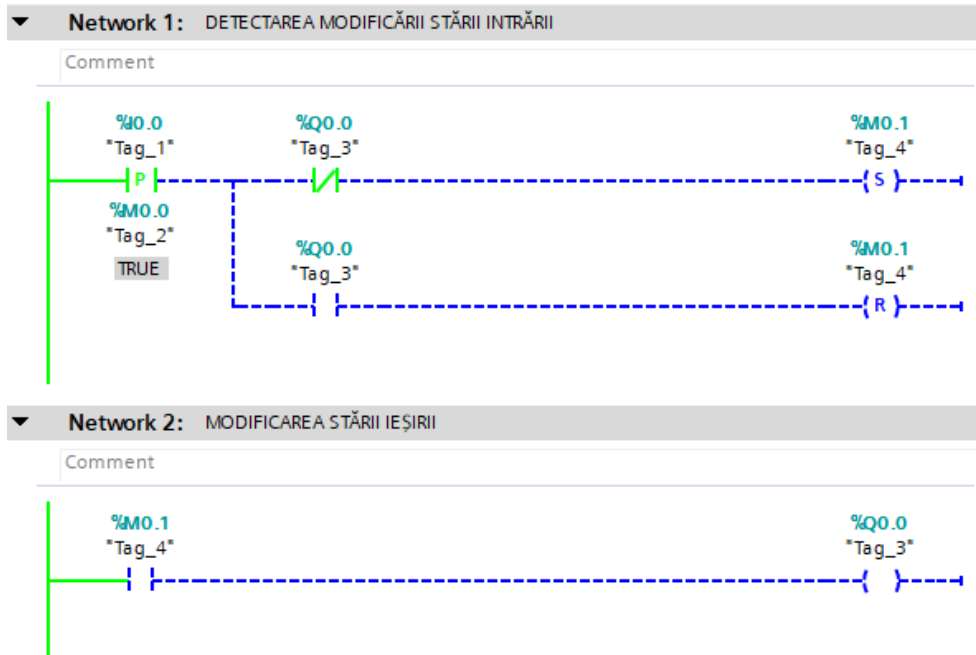


Fig. 3.50 – Dezactivarea ieşirii la apariţia unei noi tranziţii pozitive

O facilitate a TIA Portal foarte utilă în faza de testare a programului sau la depanarea acestuia este posibilitatea de modificare a stării elementelor programului prin comandă software. În meniul Online sau în meniul obţinut prin Right Click pe instrucţiunile din diagrama LAD se poate alege opţiunea ”Modify”, care permite comutarea blocului respectiv în stare 0 logic sau 1 logic.

### Exemplul 3 – Realizarea unui sistem automatizat de găurire

În acest exemplu se va prezenta modul în care se realizează controlul secvențial al unui proces de găurire. Procesul de găurire trebuie să includă următorii pași: la acționarea **S1** burghiul se învârtă în sens orar, iar după 3 secunde burghiul va înainta înspre piesa ce va fi găurită. Când se atinge poziția finală se va primi un semnal de la senzorul conectat la **I0.3**, înaintarea burghiului fiind întreruptă; revenirea la poziția inițială se face datorită unui arc. În timpul revenirii burghiului este necesar ca acesta să se învârtă în sens trigonometric. Când se ajunge la poziția inițială și senzorul de poziție conectat la **I0.4** este activat, burghiul se mai învârtă timp de 1 secundă înainte ca acesta să se oprească total. În orice moment al funcționării procesul poate fi întrerupt prin acționarea butonului **STOP**, conectat la **I0.0**. O schemă a procesului este prezentată în figura de mai jos.

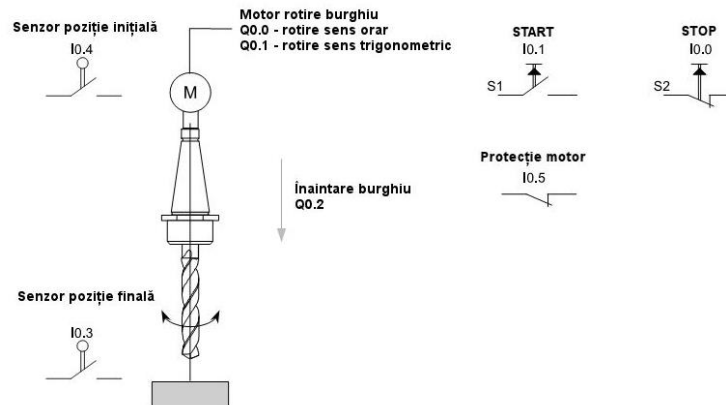


Fig. 3.51 – Schema procesului de găurire automată

Automatizarea aplicației prezentate anterior se bazează pe divizarea întregului proces în etape cât mai mici, care în general se execută secvențial, etape care sunt în general denumite pași. De cele mai multe ori un pas trebuie executat integral înainte ca următorul pas să înceapă, fiind impuse o serie de condiții de tranziție de la un pas la altul. Se vor utiliza „flag-uri” pentru acești pași, un pas fiind activ când flag-ul asociat lui este 1 logic. Principalele caracteristici ale pașilor sunt sintetizate în figura de mai jos.



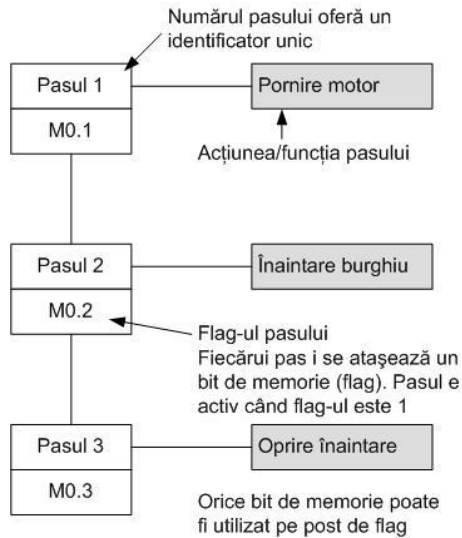


Fig. 3.52 – Utilizarea ”flag-urilor” pentru controlul secvențial al programului

Fiecare pas este activat în momentul în care sunt îndeplinite anumite condiții, în general derivate din stările mașinii. Acestea pot include semnale de la senzori de poziție, butoane, nivele de temperatura sau temporizatoare. În general una dintre condiții este legată de starea pasului anterior, iar în momentul activării flag-ului pasului curent flag-ul atașat pasului anterior fiind resetat.

Acest gen de programe conține în general două părți distincte, una care realizează tranziția de la un pas la altul pe baza condițiilor impuse și o parte care va realiza activarea corespunzătoare a ieșirilor. Prin această separare în cadrul programului, modificarea etapelor în care o ieșire este activă se face ușor, adăugând doar flag-ul corespunzător noilor etape.

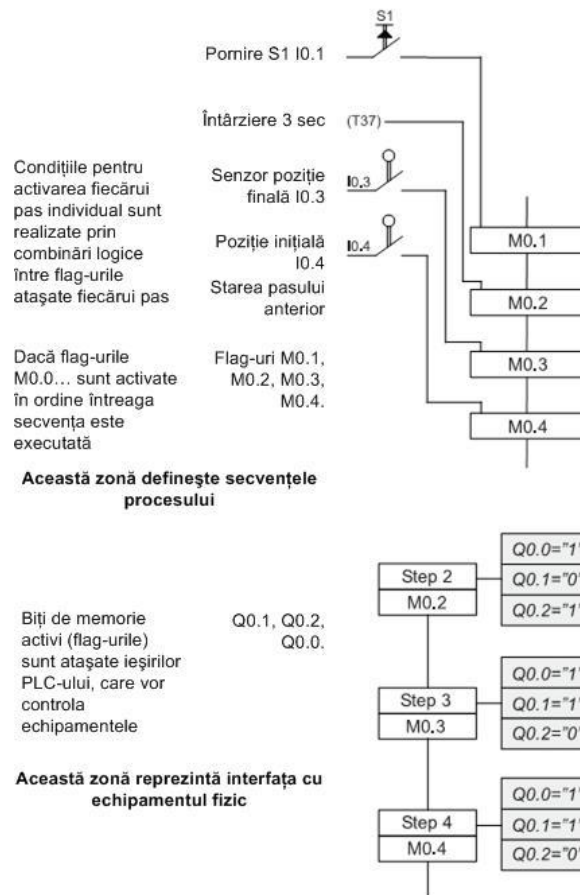


Fig. 3.53 – Schema de funcționare a procesului

Tranziția de la un pas la altul se face prin setarea și resetarea flag-urilor asociate pașilor. Ieșirile vor fi activate în funcție de starea biților de memorie atașați flag-urilor.

În general, la lucrul cu programe secvențiale, se recomandă ca un singur pas să fie executat la un moment dat, adică un singur flag să fie activ. Dacă procesul tehnologic care este automatizat necesită ca două sau mai multe acțiuni să fie executate simultan se vor realiza structuri secvențiale paralele, ce vor acționa independent.

Se recomandă ca primul pas din cadrul programelor secvențiale să nu implice activarea de ieșiri (conectare la actuatori) pentru a evita eventualele erori.

Butonul STOP trebuie să realizeze resetarea tuturor flag-urilor din program astfel încât toate acţiunile să fie întrerupte.

Pentru a nu fi nevoie ca butonul **START** să fie apăsat pe toată durata funcţionării sistemului a fost realizat un sistem cu automenţinere, similar cu cel prezentat anterior. Valoare rezultată în urma logicii de pornire este stocată într-o locaţie de memorie etichetată "Automenţinere".

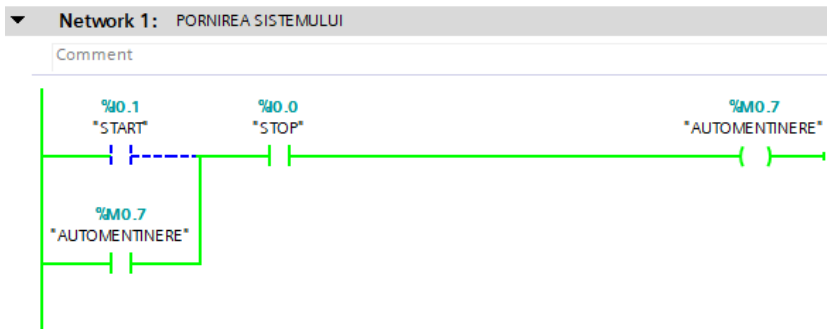


Fig. 3.54 – Circuitul de pornire/oprire al automatizării

### Activarea pasului 1

Flag-ul pasului 1 este setat când programul se află în starea iniţială (**M0.0** e 1 logic) **ŞI** intrarea variabila "Automenţinere" este activă. În acelaşi timp, flag-ul pasului 1, **M0.0**, este resetat.

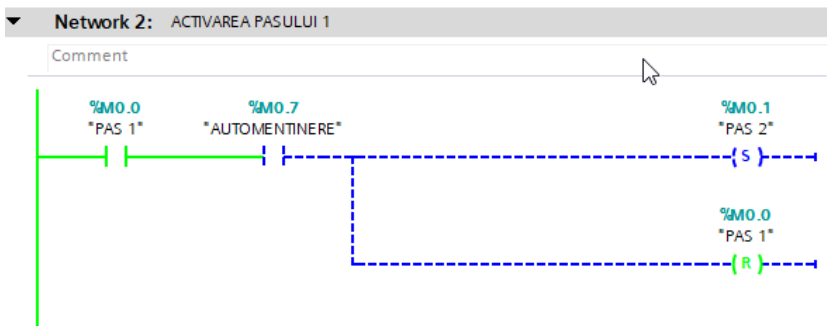


Fig. 3.55– Iniţializarea pasului 1

### Activarea pasului 2

Flag-ul **M0.2** este activat dacă programul se află la pasul 1 (adică **M0.1** este activ) ȘI timer-ul **IEC\_Timer\_0\_DB\_0** a devenit 1 logic (timpul setat în timer a expirat). **M0.1** este resetată. În acest pas programul stă 3 secunde, date de timpul setat în blocul **TON**, perioadă în care burghiul mașinii de găurit se rotește în gol pentru a atinge viteza nominală înainte începerii prelucrării.

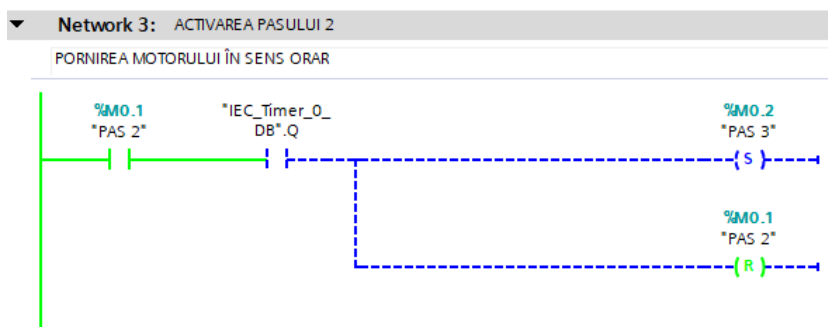


Fig. 3.56 – Inițializarea pasului 2

### Activarea pasului 3

Flag-ul **M0.3** este activat dacă programul se află la pasul 2 (adică **M0.2** este 1 logic) ȘI intrarea **I0.3**, atașată senzorului de cursă maximă, devine activă. **M0.2** este resetată. Acest pas reprezintă porțiunea din program în care se face găurirea materialului, burghiul înaintând în material până când se atinge poziția finală.

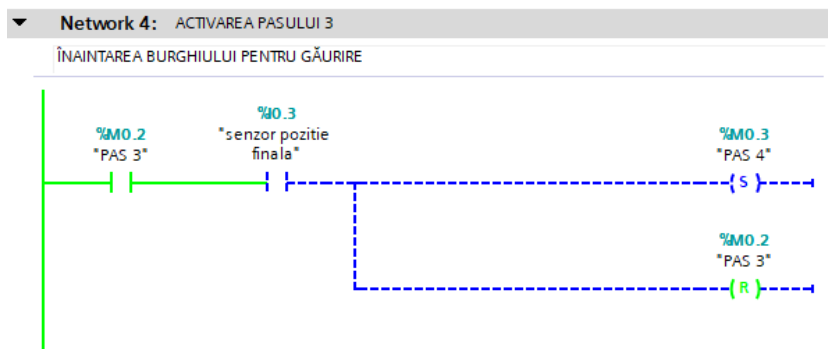


Fig. 3.57 – Inițializarea pasului 3

### Activarea pasului 4

**M0.4**, flag-ul pasului 4, primeşte valoarea 1 logic dacă programul se află la pasul 3 **ŞI** intrarea **I0.4**, conectată senzorului de poziţie iniţială, este activă. În acelaşi timp **M0.3** este resetată. Această secvenţă corespunde situaţiei în care burghiul se retrage din material sub acţiunea resortului cu care este echipat utilajul, iar burghiul se va roti în sens invers.

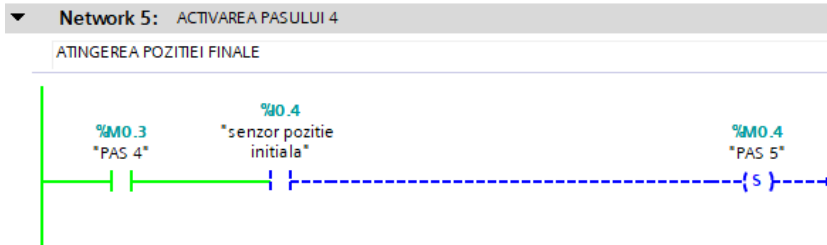


Fig. 3.58 – Inițializarea pasului 4

### Activarea pasului 5

Flag-ul **M0.5** este setat dacă programul se află în pasul 4 (**M0.4** este 1 logic) **ŞI** timer-ul **IEC\_Timer\_0\_DB\_1** este activ. **M0.4** este resetată. Starea aceasta a programului corespunde situaţiei în care burghiul a revenit în poziţia iniţială şi a trecut 1 secundă, timp în care se consideră că piesa găurită a fost înlocuită cu următoarea.

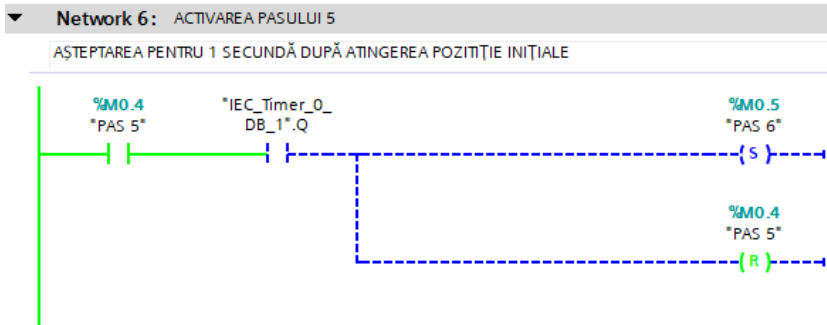


Fig. 3.59 – Inițializarea pasului 5

### Activarea pasului 0

Dacă flag-ul M0.5 este activ (timpul setat în timer-ul **IEC\_Timer\_0\_DB\_1** a expirat), pasul 0 (pasul de inițializare) este activat. Flag-urile pașilor 1 – 5 sunt resetate în acest punct, pentru a se realiza repornirea programului. Dacă se dorește introducerea unei condiții înainte de repornirea sistemului (un interval de timp care să permită înlocuirea piesei cu una nouă, contacte de siguranță, etc.) acestea pot fi introduse în rețea în paralel cu contactul **NC** atașat **M0.5**

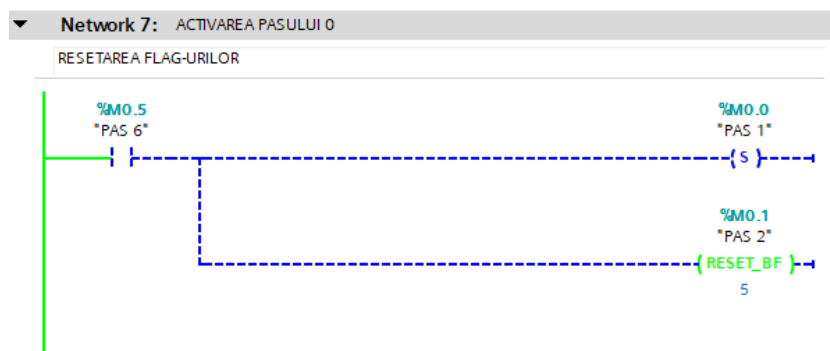


Fig. 3.60 – Inițializarea pasului 0

### Activarea timer-ului IEC\_Timer\_0\_DB\_0

Dacă pasului 1 este activ (flag-ul M0.1 este 1 logic) timer-ul **IEC\_Timer\_0\_DB\_0** este pornit.

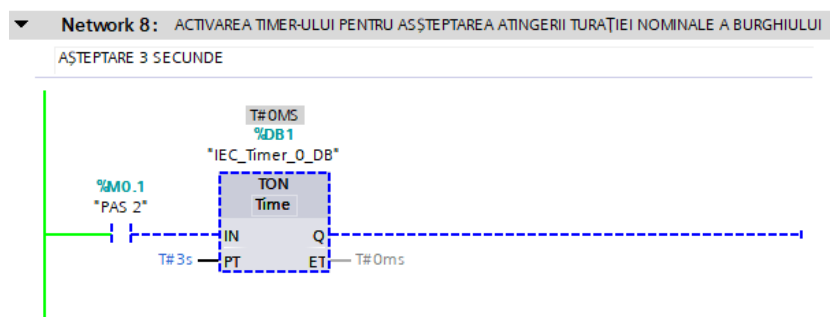


Fig. 3.61 – Activarea temporizării pentru pornirea burghiului

### Activarea timer-ului IEC\_Timer\_0\_DB\_1

Dacă pasului 4 este activ (flag-ul M0.1 este 1 logic) timer-ul IEC\_Timer\_0\_DB\_1 este pornit.

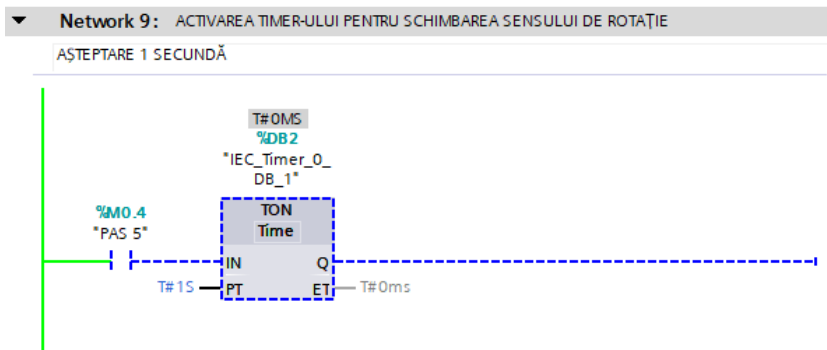


Fig. 3.62 – Activarea temporizării pentru repornirea ciclului

### Setarea condițiilor de oprire de urgență sau protecție

Flag-ul **M0.0**, atașat pasului 0, este activat dacă:

- 1) butonul STOP este activat, **I0.0** devine 0 logic

**SAU**

- 2) protecția motorului s-a activat, adică **I0.5** a devenit 0 logic.

Flag-urile pașilor 1-5 sunt resetate

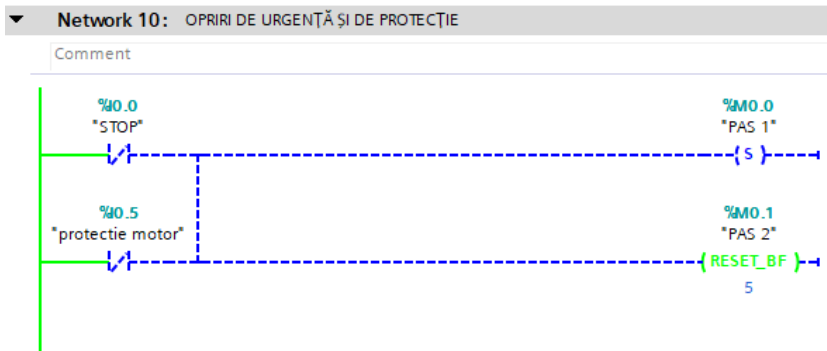


Fig. 3.63 – Revenirea la pasul 1 în condiții de avarie

### Activarea ieșirii Q0.0

Ieșirea **Q0.0** este activă pe durata pașilor 2 și 3, adică atunci când flag-urile **M0.1** sau **M0.2**, ceea ce corespunde la stările în care burghiul trebuie să se rotească în sens orar.

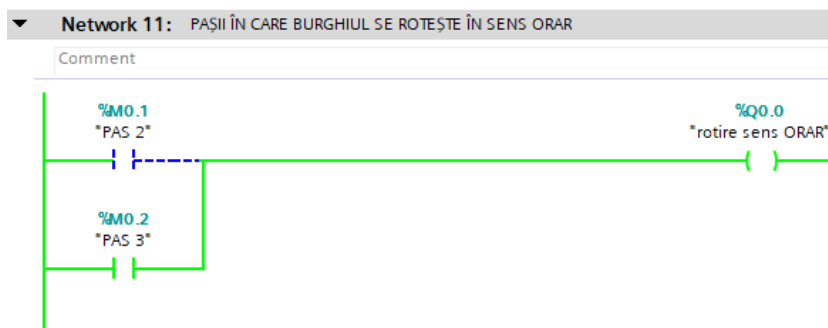


Fig. 3.64 – Definirea pașilor în care burghiul se rotește în sens orar

### Activarea ieșirii Q0.1

Ieșirea **Q0.1**, ce determină rotirea burghiului în sens trigonometric, este activă în pașii 4 și 5, atunci când **M0.3** sau **M0.4** sunt 1 logic.

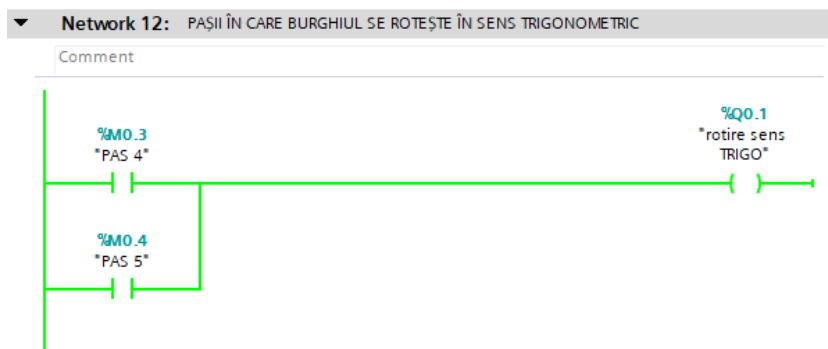


Fig. 3.65 – Definirea pașilor în care burghiul se rotește în sens trigonometric



### Activarea ieşirii Q0.2

Dacă bitul de memorie **M0.2** este activ ieşirea **Q0.2** va deveni activă, determinând înaintarea burghiului.

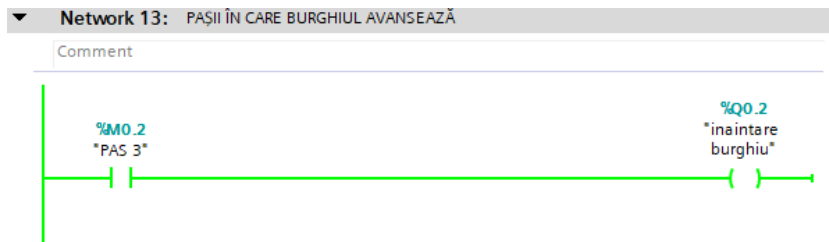


Fig. 3.66 – Definirea paşilor în care burghiul înaintează

În configurația prezentată mai sus, programul secvențial va parcurge toți pașii definiți, ieșirile fiind activate în funcție de starea flag-urilor. O parcurgere completă a programului implică pornirea burghiului la apăsarea butonului **START**, învârtirea acestuia în gol timp de 3 secunde, înaintarea spre piesă până la atingerea poziției finale, schimbarea sensului de rotație a burghiului în timpul revenirii acestuia la poziția inițială și oprirea rotirii burghiului odată atinsă această poziție. După parcurgerea întregului ciclu se revine la pasul 0, întregul proces să fiind repornit.

Dacă se dorește ca sistemul să aștepte o confirmare din partea utilizatorului pentru repornirea ciclului, programul poate fi modificat prin eliminarea primei rețele, unde este realizată automenținerea și înlocuirea în rețeaua numărul 2 a blocului "Automenținere" cu blocurile **START** și **STOP** conectate în serie.

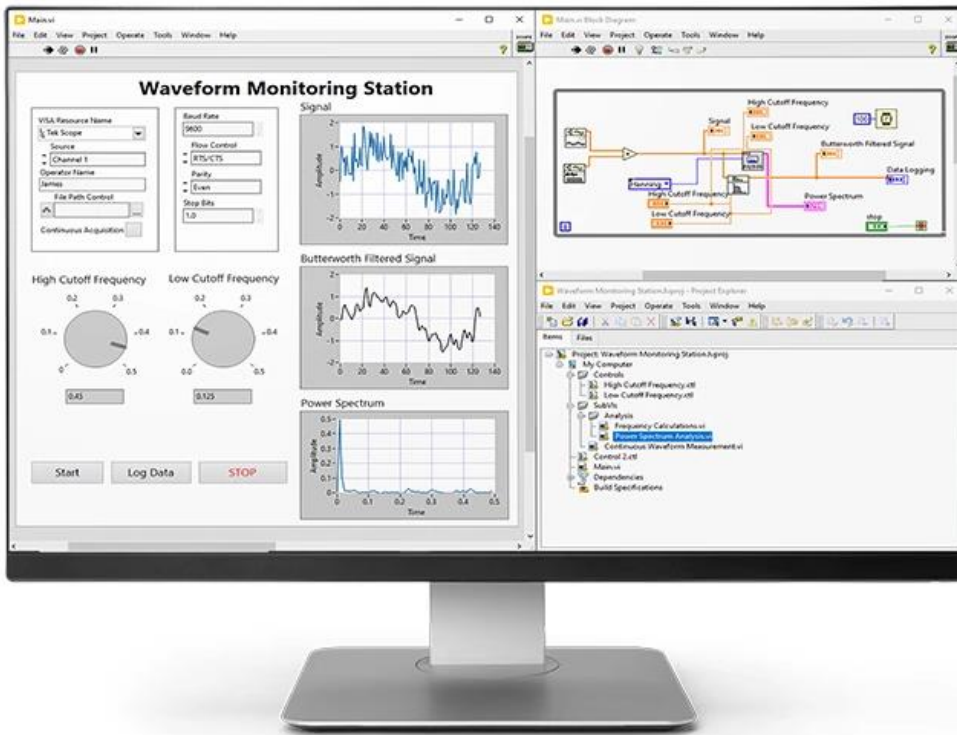
## 4. Concluzii și perspective

Această carte reprezintă o introducere în domeniul utilizării automatelor programabile pentru aplicații de automatizări industriale, fiind abordate două modele de bază din game de produse a companiei Siemens: Logo! și S7-1200. Pentru ambele echipamente au fost prezentate funcțiile de bază utilizate în automatizări simple, bazate pe valori primite de la senzori digitali și unde se dorește controlul de tip ON/OFF al actuatorilor conectate la ieșirile digitale ale PLC-ului. Lumea automatizărilor industriale implică procese mult mai complexe, unde valorile de intrare/ieșire sunt reprezentate de valori analogice sau necesită un control al ieșirilor, de cele mai multe ori și cu un feedback pentru controlul în buclă închisă. Acest tip de automatizări va fi abordat în cel de-al doilea volum am acestei cărți, avându-se în vedere prezentarea de unelte software ce pot fi folosite în învățarea experiențială, atât de utilă în dezvoltarea de aptitudini pentru absolvenții de inginerie. În cele ce urmează se va face o scurtă trecere în revistă a subiectelor ce vor fi incluse.

O unealtă foarte utilă în dezvoltarea competențelor pentru realizarea de automatizări industriale e reprezentat de programul LabVIEW (LV) dezvoltat de compania americană National Instruments (NI), care oferă un mediu de programare grafic, având un număr impresionant de librării de funcții care acoperă toate domeniile ingineriești. Programele dezvoltate în LabVIEW prezintă două elemente: o interfață grafică pentru utilizator, unde pot fi afișate sub formă de butoane, potențiometre, indicatoare, grafice, zone text, etc. toate elemente necesare pentru interacțiunea cu programul și partea de diagramă de blocuri, unde se realizează programul propriu-zis prin conectarea de blocuri de funcții cu elementele de intrare/ieșire de pe interfață. Un avantaj al utilizării LV este dat de faptul că National Instruments este în același timp și unul dintre liderii mondiali în dezvoltarea de echipamente de achiziții de date, ceea ce permite inserarea facilă în program de valori măsurate de la senzori și transmiterea valorilor de ieșire către module de control ale actuatorilor.

Complexitatea programelor poate varia de la simple calcule matematice sau realizarea de mașini de stare la sisteme complexe de achiziție de date multi-canal, analiză în timp real, prelucrarea de date, stocare locală sau pe cloud, etc. Gama de echipamente de achiziție și control este la rândul ei foarte variată, pornind de la echipamente destinate uzului didactic (myRIO), echipamente de prototipare și uz industrial (cRIO sau sbRIO) sau calculatoare industriale pentru control și testare în sisteme industriale complexe (PXI). Gama RIO (Reconfigurable Input/Output) oferă

echipamente cu o plajă mare de opțiuni în ceea ce privește puterea de calcul, capacitatea de stocare, numărul și tipul intrărilor/ieșirilor, modelul myRIO fiind ideal pentru uz educațional, oferind, pe lângă numărul generos de I/O digitale și analogice, posibilitate de conexiune pe cablu sau wireless, accelerometru pe 3 axe integrat, cip FPGA, etc.



Interfața LabVIEW [sursă: ni.com]

Prin realizarea de conexiuni prin protocoale de comunicații industriale ecosistemul NI LabVIEW poate fi extins la orice alt echipament disponibil pe piață, precum și cu programe de simulare echipate cu astfel de protocoale. Un astfel de protocol este OPC UA - Open Platform Communications Unified Architecture, care este un standard pentru transferul de date în condiții de securitate utilizat cu precădere în domeniul industrial. Acest protocol, dezvoltat și gestionat de OPC Foundation, permite transferul de date între echipamente dezvoltate de diverși producători, utilizarea lui nefiind limitată de platforma utilizată. OPC UA permite astfel transferul bi-direcțional de informații între automate programabile Siemens și LabVIEW, parte de control de proces putând rula local, pe PLC-uri, iar partea de monitorizare și control fiind rulată pe calculatoare industriale sau PC-uri, cu datele

stocate local sau în cloud. OPC UA specifică modul în care sunt informațiile modelate și comunicate într-un sistem ca IIoT: servicii abstracte care pot fi implementate în API-urile specifice limbii și mapate în diferite stive de comunicare, un model tip "AddressSpace" care oferă un mod standard pentru servere să reprezinte datele într-o manieră "object oriented" către clienți. În arhitectura OPC UA sunt definiți doi parteneri care vor interacționa: clienții și serverele.

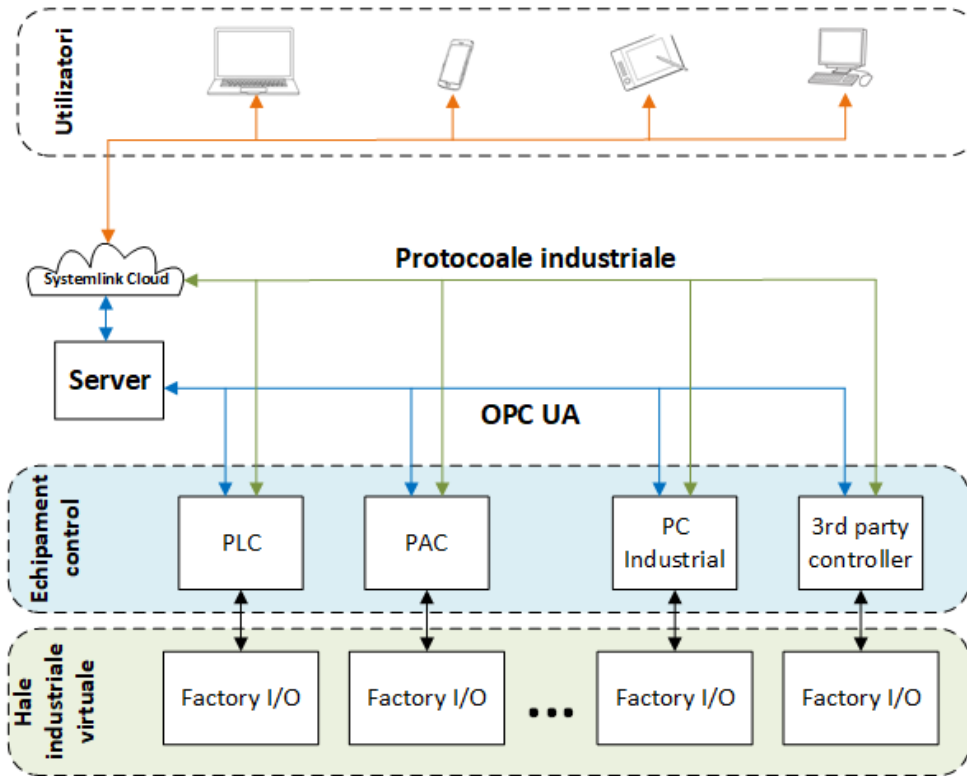
O altă unealtă foarte utilă în dezvoltarea de aptitudini în domeniul automatizărilor industriale este programul Factory I/O dezvoltat de către compania RealGames, program de simulare 3D care include o librărie vastă de echipamente specifice mediului industrial. Aceste echipamente pot fi utilizate pentru realizarea de sisteme industriale de prelucrare, transport, depozitare, avertizare, etc. cu grad de complexitate variind de la simple benzi transportoare până la procese industriale similare cu cel din imaginea de mai jos. Programul include drive de comunicații pentru produsele liderilor din domeniul automatizărilor industriale Allen-Bradley și Siemens, programele încărcate pe PLC-uri putând fi testate în condiții similare celor reale. În același timp sunt integrate și protocoalele de comunicații OPC UA și Modbus TCP/IP, care pot fi utilizate pentru conectarea Factory I/O la alte echipamente sau la calculatoare pe care rulează LabVIEW pentru testarea programelor de control realizate în aceste medii de programare.



Interfață Factory I/O [sursa factoryio.com]

Nu în ultimul rând, dezvoltarea de programe de monitorizare și control implică transferul, stocarea și prelucrarea de date local și pe cloud. O soluție ușor de

implementat pentru familiarizarea cu acest mode de lucru este reprezentată de mediul de administrare a sistemelor și datelor SystemLink dezvoltat de către National Instruments. Acesta permite accelerarea procesului de testare și reducerea timpului de introducere pe piață a noilor produse, facilitând transferul de date între echipamentele de testare și inginerii implicați în monitorizarea noilor produse, oferind o soluție scalabilă și rapidă de transfer pe cloud a informațiilor, securitatea datelor prin acces pe bază de autentificare și posibilitatea accesării informațiilor sub formă grafică de pe orice platformă.



Sistem complex de monitorizare și control

## 5. Bibliografie

1. United Nations – ESA, "World Urbanization Prospects The 2014 Revision", disponibil online la: [UN-ESA  
https://esa.un.org/unpd/wup/Publications/Files/WUP2014-Highlights.pdf](https://esa.un.org/unpd/wup/Publications/Files/WUP2014-Highlights.pdf)
2. The World Bank – "Urban Development – Context", disponibil online la <https://www.worldbank.org/en/topic/urbandevelopment/overview#1>
3. Hannah Ritchie and Max Roser (2019) - "Age Structure". Published online at OurWorldInData.org. Retrieved from: 'https://ourworldindata.org/age-structure' [Online Resource]
4. United Nations, Department of Economic and Social Affairs Population Division, World Population Prospects 2022, disponibil online la: <https://population.un.org/wpp/Download/Standard/MostUsed/>
5. Chunguang Bai, Patrick Dallasega, Guido Orzes, Joseph Sarkis, "Industry 4.0 technologies assessment: A sustainability perspective", International Journal of Production Economics, Volume 229, 2020, ISSN 0925-5273, <https://doi.org/10.1016/j.ijpe.2020.107776>
6. John Paliotta - Software Quality and the Industrial Internet of Things: Why it Matters NOW, Embedded System Engineering, July 6th, 2015, <http://eecatalog.com/IoT/2015/07/06/software-quality-and-the-industrial-internet-of-things-why-it-matters-now/>
7. Samuel Greengard – The Internet of Things, MIT Press, ISBN: 9780262527736, 232 pp., March 2015
8. Cisco – Internet of Everything, <http://ioeassessment.cisco.com/learn/ioe-faq>
9. Adrian McEwenm Hakim Cassimally – Designing the Internet of Things, J Wiley and Sons, ISBN: 978-1-118-43062-0, 336 pages, November 2013
10. Siemens Logo! Manual
11. Siemens S7 – 1200 Programmable controller – System Manual
12. Siemens S7 – 1200 Functional Safety Manual
13. Arshdeep Bahga, Vijay Madiseti – "Internet of Things: A Hands-On Approach", VPT, 2014 ISBN 0996025510, 9780996025515
14. Mohanty, A.R.. – "Machinery Condition Monitoring: Principles and Practices", CRC Press, 2014
15. Marțis, C., Hedeșiu H. – „Sisteme electromecanice”, Editura Mediamira, Cluj-Napoca, 2007, ISBN 978-973-713-168-3
16. Hedeșiu H. – „Sisteme ierarhizate de control secvențial”, Editura Mediamira, Cluj-Napoca, 2003, ISBN 973-9357-43-1