

Lucrarea 5. Controloare logice programabile (PLC – Programmable Logic Controller)

1. Obiectivul lucrării

Lucrarea își propune studierea modului de implementare a unor aplicații de monitorizare și control cu ajutorul controloarelor logice programabile. Se studiază configurația hardware a unui astfel de dispozitiv și modul de conectare a unor elemente externe de tip senzori și elemente de execuție. Se prezintă modul de programare a unei aplicații utilizând un limbaj de programare specific.

2. Considerații teoretice

Controloarele logice programabile, denumite în jargonul curent PLC-uri, sunt dispozitive robuste și compacte special concepute pentru monitorizarea și controlul unor parametri de proces. Cu ajutorul lor pot fi implementate diferite scheme de automatizare de complexitate medie. Inițial ele au fost concepute pentru implementarea unor funcții de control binar: funcții logice combinatoriale și automate programabile (control secvențial). Ulterior s-au adăugat funcții suplimentare de reglaj continuu și adaptiv, funcții de comunicație și funcții de vizualizare și stocare a datelor culese. Astăzi, prin caracterul lor robust, fiabil și autonom, reprezintă cele mai utilizate componente inteligente de automatizare.

Din punct de vedere constructiv un dispozitiv PLC se compune din:

- microsistem de calcul, de obicei implementat cu ajutorul unui microcontrolor
- set de interfețe digitale și analogice, care conțin circuite de adaptare pentru semnale industriale
- modul de alimentare electrică
- opțional: interfețe de comunicație serială (ex: RS232, RS485) și în rețea (ex: CAN, Profibus)
- carcasă de protecție

În cadrul lucrării de față, ca și exemplu de dispozitiv PLC, se va studia dispozitivul **FX3U** și componentele sale auxiliare, produse de firma MITSUBISHI. În figura 1 se poate observa imaginea acestui dispozitiv, pe care s-au marcat zonele de conectare/cuplare a unor elemente externe sau de extensie.

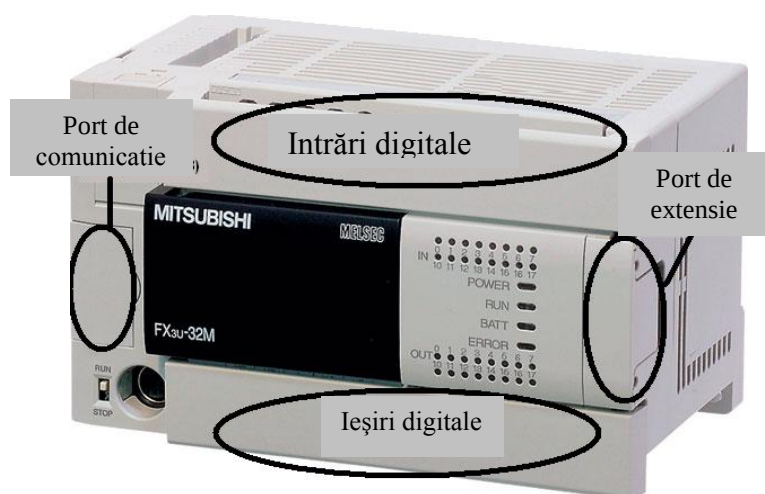


Figura 1. Exemplu de PLC – Hitsubishi FX3u

2.1 Interfețele digitale de intrare

Aceste interfețe permit conectarea unor semnale digitale generate de senzori, comutatoare sau întrerupătoare. Din punct de vedere al modului de conectare intrările digitale pot fi de două tipuri:

- drenă de curent (eng. Sink input), pentru senzori de tip NPN
- sursă de curent (eng. Source input), pentru senzori de tip PNP

În cazul unei intrări de tip drenă, în starea “cuplat/închis” a sensorului, curentul circulă dinspre intrare către senzor. Figura 2 prezintă schema de principiu a unui senzor de tip NPN conectat la intrarea de tip Sink a unui PLC.

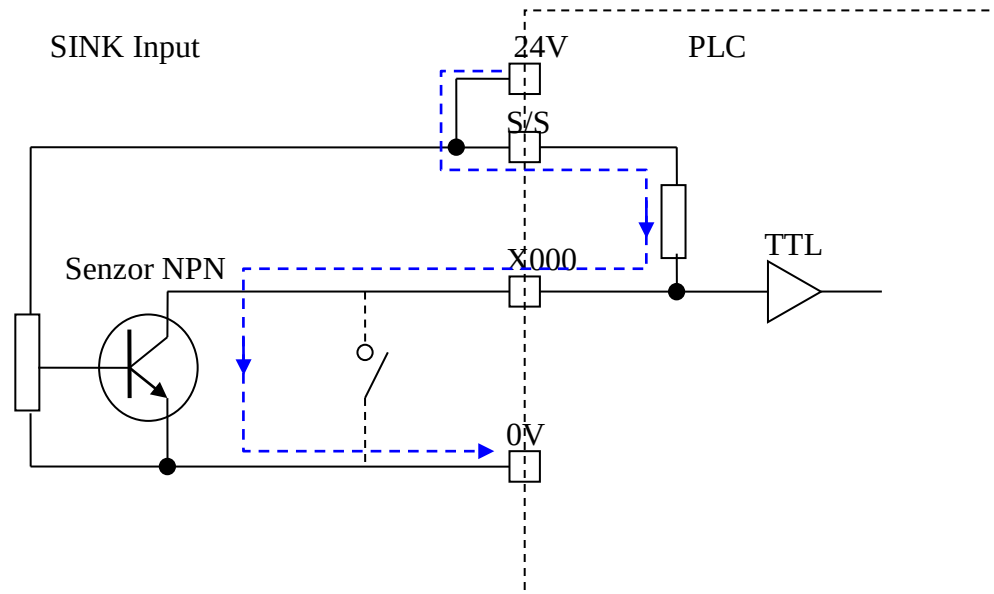


Figura 2 Schema de conectare a unui senzor NPN la o intrare de tip drenă

În cazul unei intrări de tip sursă, în starea “cuplat/închis” senzorul generează un curent ce intră în dispozitivul PLC. Figura 3 prezintă acest caz.

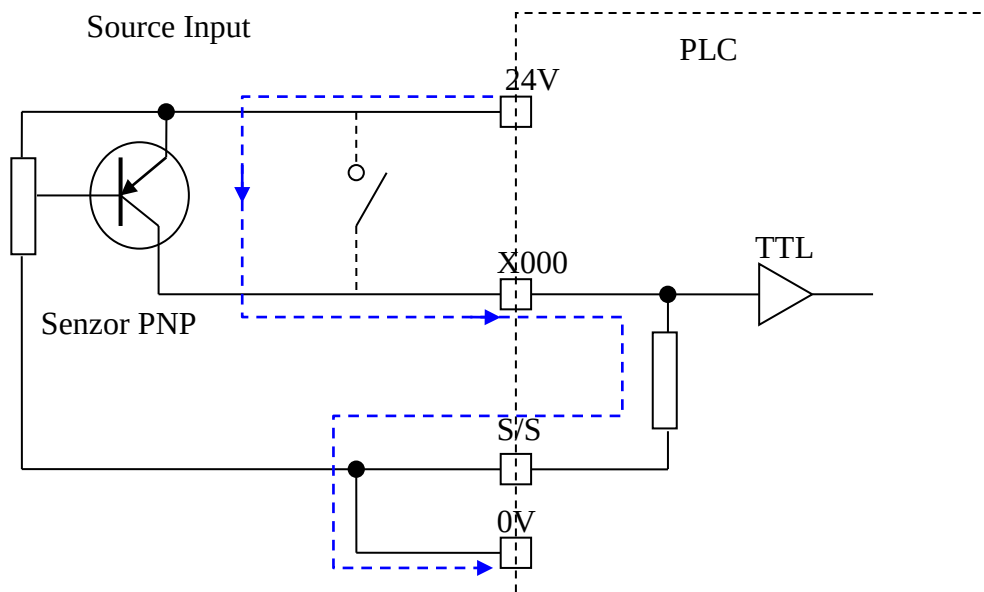


Figura 3. Schema de conectare a unui senzor PNP la o intrare de tip sursă

La PLC-ul FX3U intrările pot fi configurate în bloc ca sursă sau ca drenă. Toți senzorii trebuie să fie de același tip. Un simplu comutator poate fi atât senzor de tip sursă cât și senzor de tip drenă, deoarece curentul poate circula în ambele sensuri. La fel și unii senzori pot fi configurați să lucreze în unul din cele două moduri.

La un senzor activ, pe lângă cele două semnale care corespund terminalelor unui comutator, mai există un semnal suplimentar folosit pentru alimentarea senzorului. La cuplarea unui senzor la un PLC se va studia manualul de utilizare al senzorului și mai ales schema indicată pentru cuplarea acestuia la o intrare de tip sursă sau drenă.

În figura 4 s-a reprezentat schema de conectare a intrărilor dispozitivului PLC în cele două cazuri. Se observă că modul de cuplare a intrării S/S la 24V și respectiv la 0V determină configurația de drenă sau sursă a intrărilor. De asemenea se observă că în cazul unei intrări drenă un comutator simplu se leagă între 0V și intrare, iar în cazul unei intrări de tip sursă între 24V și intrare.

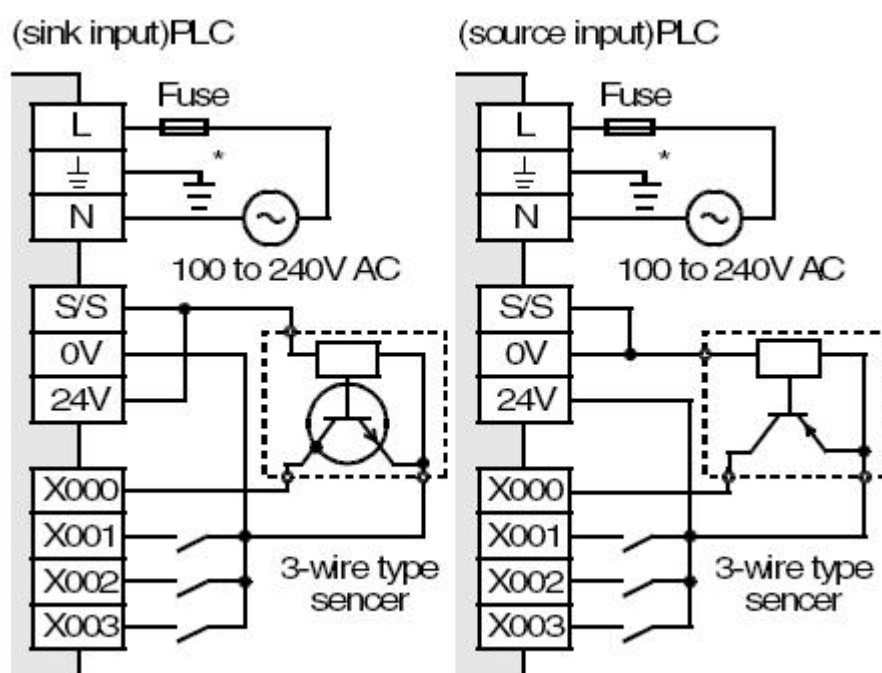


Figura 4. Schema de conectare a senzorilor la PLC

ATENȚIE: CUPLAREA GREȘITĂ A SENZORILOR SAU A COMUTATOARELOR POATE DUCE LA ARDEREA CIRCUITELOR DE INTRARE

O intrare se consideră 1 logic dacă tensiunea de intrare este mai mare de 16V și 0 logic dacă este mai mică de 8V. Pe frontul urcător al semnalului de intrare comutarea se face la 16V iar pe frontul coborâtor la 8V.

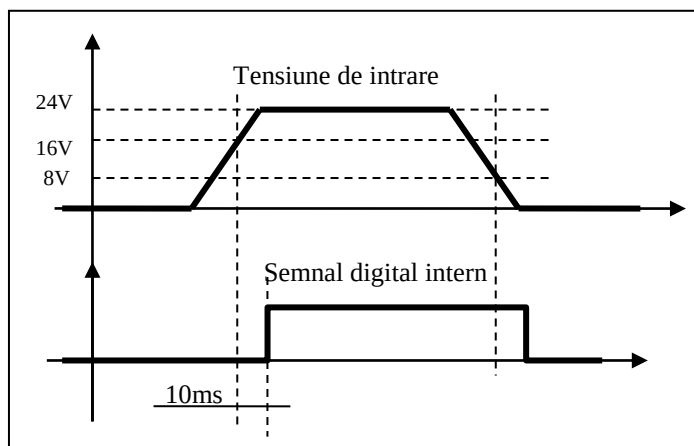


Figura 5. Diagrama de comutare a unei intrări

Din punct de vedere al programării intrările digitale sunt notate cu **X000-Xnnn**.

2.2 Interfețele digitale de ieșire

Cu ajutorul interfețelor de ieșire PLC-ul comandă diferite elemente de execuție digitale (bipoziționale). Ieșirile sunt de obicei izolate galvanic de partea de logică pentru a permite cuplarea unor tensiuni sau curenți de valoare mai mare și pentru a proteja partea de logică de eventuale tensiuni accidentale prea mari. În funcție de construcția PLC-ului ieșirile pot fi de tip contact de releu, tranzistor de putere sau tiristor. În cazul de față ieșirile sunt de tip contact de releu. Figura 6 prezintă modul de conectare a unui element de execuție sau sarcină (eng. Load) la o ieșire a PLC-ului.

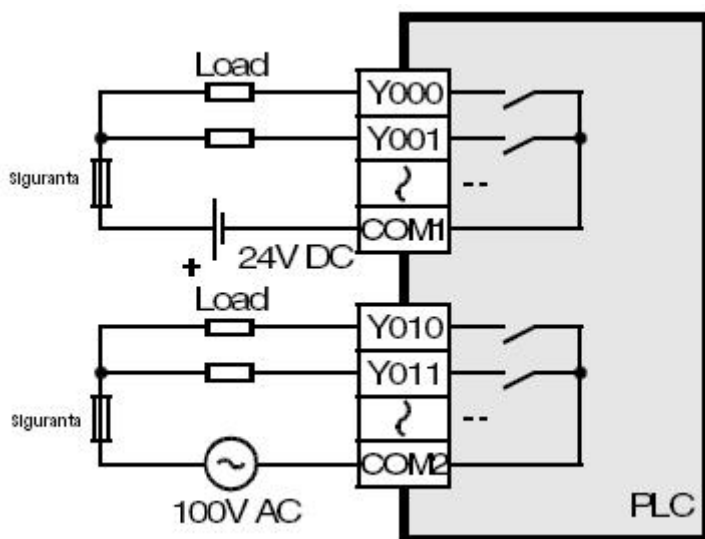


Figura 6. Modul de conectare a ieșirilor

Alimentarea elementului de execuție sau a sarcinii se va face de la o sursă externă de alimentare. Ieșirile se notează cu Y000-Ynnn. O ieșire are două contacte, din care unul se consideră ca fiind semnalul de ieșire, iar al doilea contact este punctul comun (de masă sau de alimentare – COM).

ATENȚIE. A NU SE UTILIZA IESIRILE AUTOMATULUI IN CIRCUITE UNDE SE VEHICULEAZA CURENTI MAI MARI DE 1-2A.

În momentul în care o ieșire este activă, pe panoul frontal al PLC-ului, LEDul corespunzător acelei ieșiri de se va aprinde. În mod similar un alt șir de LEDuri indică starea semnalelor de intrare.

2.3 Programarea PLC-ului

Standardul IEC 61131 stabilește 5 limbaje de programare utilizabile pentru programarea dispozitivelor de tip PLC:

- LD – Ladder Diagram – limbaj grafic de tip “schemă cu relee”
- FBD – Function Block Diagram – limbaj grafic de tip “flux de date” (cu blocuri funcționale interconectate)
- ST – Structured Text – limbaj de nivel înalt asemănător cu C sau Pascal
- IL – Instruction List – limbaj de nivel scăzut de tip limbaj de asamblare
- SFC – Sequential Function Chart – limbaj care permite exprimarea secvențelor de pași pentru un automat de stare

Pentru dezvoltarea de aplicații se pot folosi diferite medii de programare, precum ISAGRAPH sau DX-Developer, care permit: editarea, compilarea, descărcarea pe un PLC țintă și executia programului (în regim normal și în regim pas-cu-pas).

Pentru PLC-urile din familia Mitsubishi FX3u se va folosi mediul de programare Dx-Developer care pune la dispoziție 2 limbaje de programare: LD și SFC (bazat pe comenzi). În cadrul lucrării de față se folosește varianta LD.

Descrierea logicii programului se face similar cu modul de desenare a schemelor de automatizare cu relee. În schemă apar contacte de relee “--| |---”, conexiuni “-----” și ieșiri “—()---”. Suplimentar în schemele mai complexe pot să se utilizeze funcții predefinite. Principal, schema astfel construită se evaluează în timpul execuției programului ca și cum curentul circulă de la stânga la dreapta în mod paralel prin fiecare linie a schemei: acolo unde contactele pe o linie sunt închise, ieșirea este activă (contactul este închis). Practic schema se evaluează în trepte după cum se vede în figura 7.

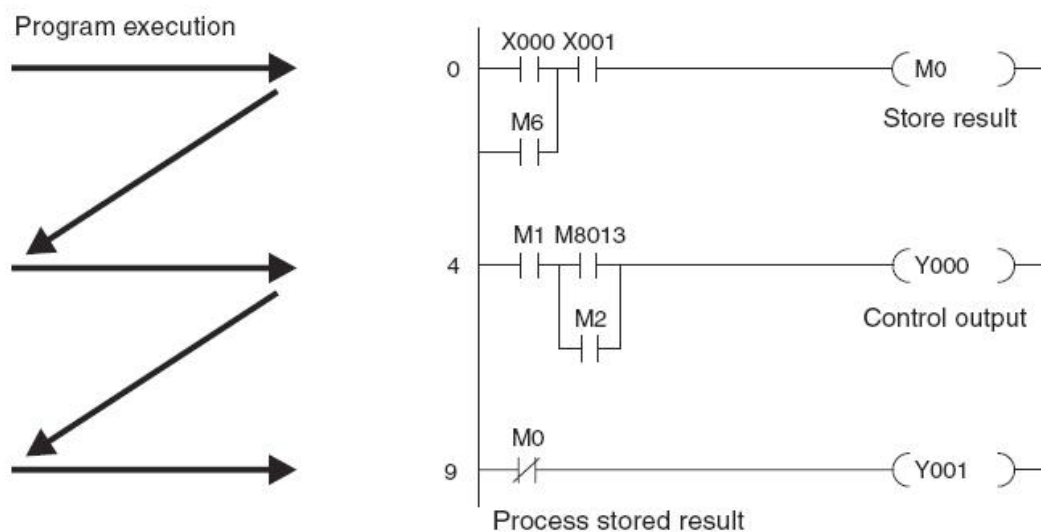


Figura 7 Executia programului se face de sus in jos si de la stanga la dreapta.

În cadrul liniilor de program se utilizează diferite tipuri de “dispozitive” incluse în structura PLC-ului, după cum urmează:

- **X – Intrări fizice** (ex: X000-X007 la PLC-ul utilizat) – intrări digitale de tip contact
- **Y – Iesiri fizice** (ex: Y000-Y007 la PLC-ul utilizat) – iesiri digitale de tip contact
- **M – Relee auxiliare** (ex: M000-...) – dispozitive bistabile de tip releu folosite pentru memorarea temporară a unor stări
- **S – Relee de stare** – folosite în limbajul STL pentru a indica un pas (o stare) din diagrama de stare a automatului
- **T – timer** – dispozitive de măsurare a timpului tip “timer”; sunt de 3 feluri: cu increment la 100ms, 10ms sau 1ms; timerele sunt pe 16 biți; pot fi cu reținere (păstrează valoarea până se resetează) sau fără reținere; ieșirea timerului este un contact care se închide în momentul în care se atinge valoarea prefixată a timerului; valoarea prefixată se specifică printr-o constantă (precedată de litera “K”, valoarea maximă 32767) sau un registru de date (precedat de litera “D”)
- **C – contoare** – sunt similare cu timerele, doar că incrementarea nu este controlată de timp ci de impulsuri de semnal
- **D – registre de date** – sunt registre de 16 biți care se folosesc pentru păstrarea unor parametri sau date de proces; se pot adresa la nivel de bit; de exemplu adresarea bitului 3 din registrul 0 se face astfel: D0.3
- **K, H, E** – constante numerice în zecimal, hexazecimal sau în forma exponențială

Simbolurile grafice cele mai des folosite sunt indicate în tabelul de mai jos:

X001 — —	Contact normal deschis; X001 este numele semnalului care controlează releul
M0 — / —	Contact normal închis; M0 este numele semnalului care controlează releul
-- ↑ --	Contact cu detecție de front urcător
-- ↓ --	Contact cu detecție de front coborător
—(Y001) —	Controlul bobinei unui releu; Y001 este controlul ieșirii
—[End]—	Apel comandă

Folosind aceste simboluri de bază se pot forma instrucțiuni mai complexe gen:

SI logic



SAU logic



Cele două linii implementează următoarele funcții logice:

$Y000 = X000 \text{ SI } X001 \text{ SI } X002$ și

Y000 = X000 SAU X001\ SAU X002

În continuare, pentru scrierea unui program destinat dispozitivului FX3u se folosește mediul de programare și depanare DX-Developer oferit de firma Mitsubishi. Mediul se instalează de pe CD-ul furnizat la achiziționarea dispozitivului PLC. Primul pas este crearea unui proiect nou prin secvența de comenzi *File -> New Project*. Se alege tipul dispozitivului pentru care se generează programul (PLC Type – FX3U(C)), tipul de limbaj folosit (în cazul nostru LD), numele proiectului și directorul unde se va salva proiectul (la sfârșit se apasă OK).

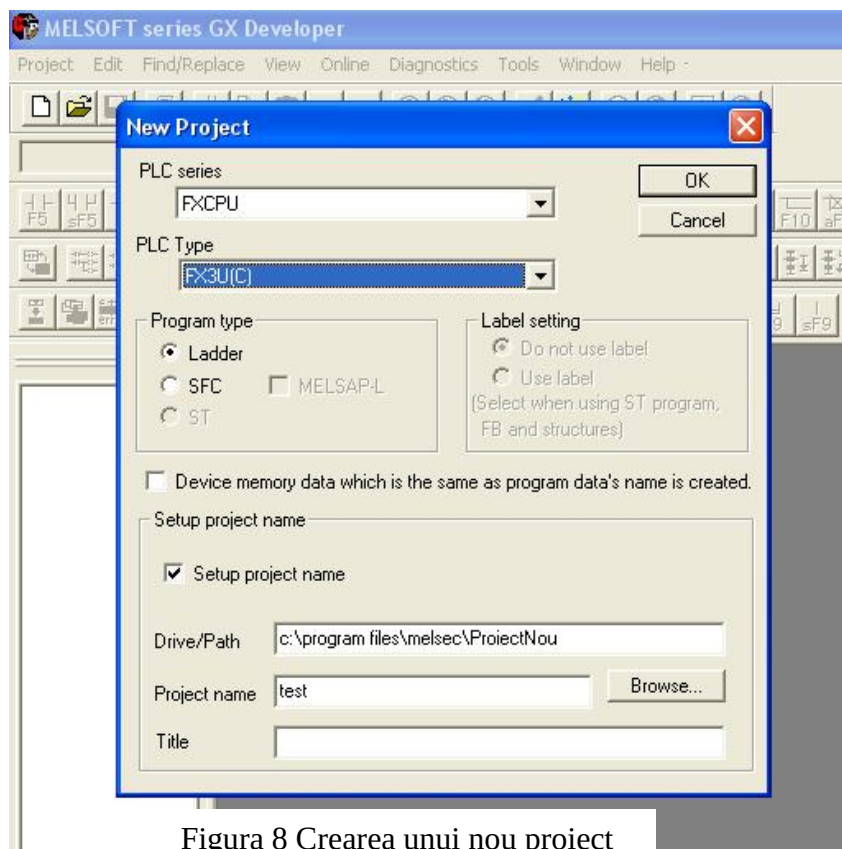


Figura 8 Crearea unui nou proiect

Editarea programului se poate face cu ajutorul simbolurilor grafice prezente în “toolbar”-ul mediului (Vezi figura de mai jos). Aceste simboluri reprezintă diferite elemente ale limbajului de programare. Simbolurile pot fi selectate și cu ajutorul tastelor funcționale F5-F10.

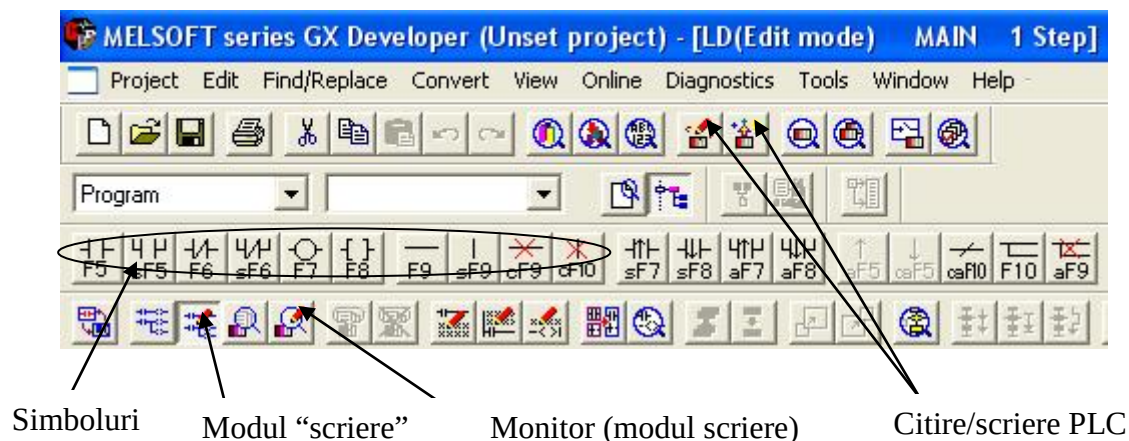


Figura 9 Meniul, icoanele și tastele funcționale ale mediului DX-Developer

Tasta F10 se va folosi pentru a crea ramificatii. Deasemenea se observa ca dupa inserarea unui simbol pe o linie nou introdusa, la apasarea Click dreapta vor aparea urmatoarele optiuni:

- **Insert line** – va insera o linie noua (numita rung) deasupra liniei curente
- **Delete line** – va sterge linia curenta
- **Insert row**- va insera o pozitie noua in fata simbolului curent
- **Delete row** – se va sterge simbolul curent

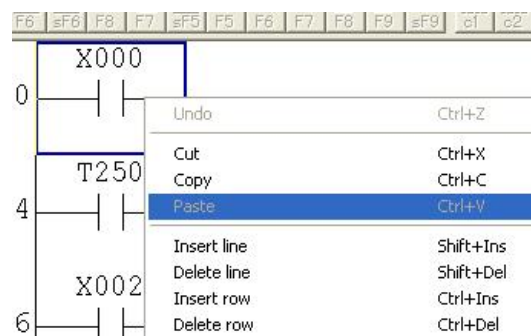


Figura 10. Opțiuni pentru linii

Pentru încărcarea și execuția programului trebuie să atașăm dispozitivului de bază PLC un modul de extensie pentru comunicație serială. Există două variante: un modul RS232 sau un modul de conversie USB-RS232.

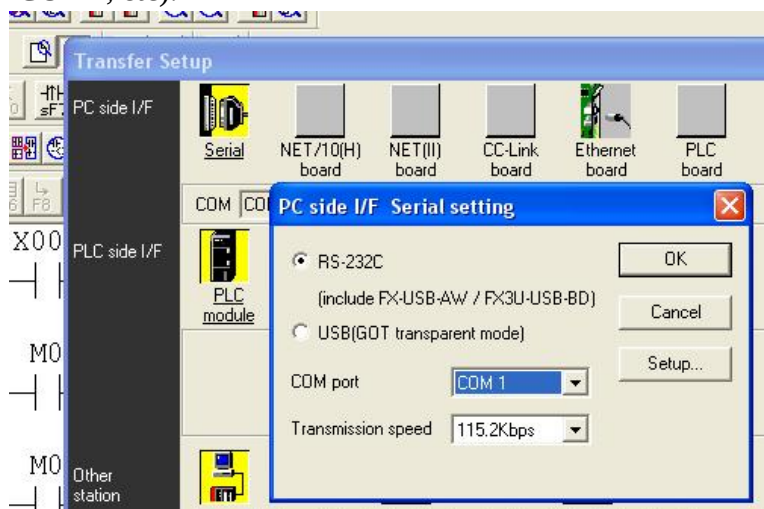


a).Modul USB

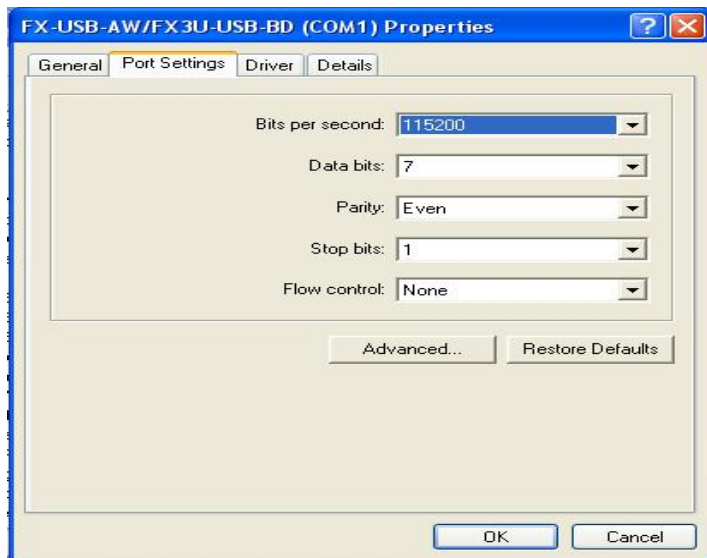


b).Modul RS232.

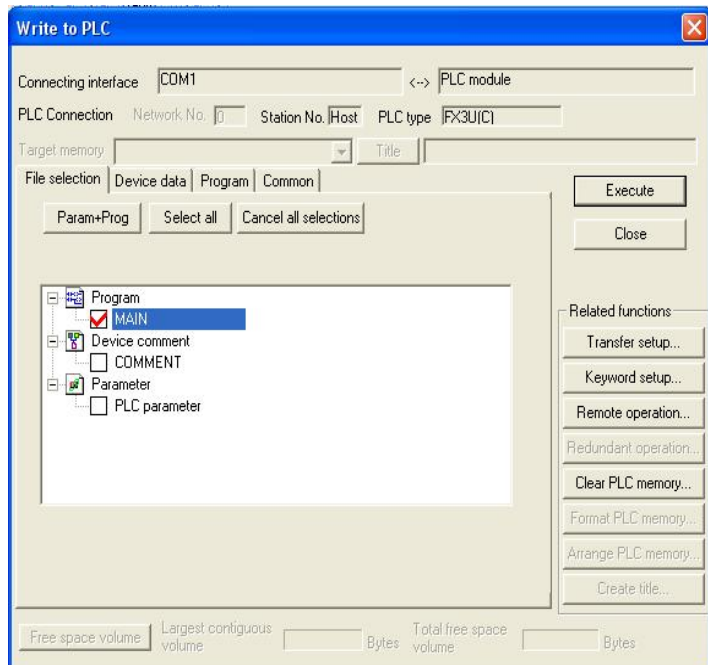
Pentru modulul a). se vor instala driverule existente in directorul curent cu numele KIT USB Bd. Setarile portului serial se fac in modul urmatoar. Se va intra in meniul *Online-> Transfer setup* si se va seta portul serial sa fie acelasi cu cel instalat pe statia de lucru (COM1, COM2, etc).



Deasemenea se va seta portul serial al statiei prin *Control Panel->System-> Device Manager -> Ports (Com and LPT)* cu urmatoarele setari:



Cand programul (LD) este complet se va converti într-un format transferabil prin secvența: *Convert -> Convert*; implicit se verifică și corectitudinea programului editat. După conversie programul este înscris în PLC prin selectarea icoanei: *Write PLC*. Se va bifa din fereastra aparuta numai programul dupa care se va apasa OK.



Acum sunteți pregătiți să începeți editarea primului program. În continuare urmează câteva exemple simple de programe:

1). Primul exemplu de program realizează un contact cu automenținere. (Mentineră în starea ON prin simpla apăsare și apoi eliberare a unui buton, sau activarea pentru un scurt moment a unui senzor). Această secvență va fi folosită des mai departe.



“Circuitul” funcționează astfel: la apăsarea unui buton (Start) conectat la intrarea X000 a PLC-ului ieșirea Y000 va fi activată (contactul Y000 închis). Se presupune că butonul (Stop) conectat la intrarea X002 nu este apăsat, ceea ce face ca prima linie a circuitului să fie închisă. În continuare, chiar și la eliberarea butonului Start ieșirea Y000 rămâne validă datorită ramurii paralele (funcție logică SAU) în care apare contactul Y000 închis; la apăsarea butonului Stop circuitul se întrerupe și ieșirea Y000 devine invalidă (contact deschis).

2). Următoarea aplicație va utiliza un timer pentru a obține un efect de temporizare. Timerele sunt de mai multe tipuri, diferind prin frecvența semnalului de ceas și prin modul de lucru (cu și fără reținere). Tabelul de mai jos indică adresele la care se află diferitele tipuri de timere.

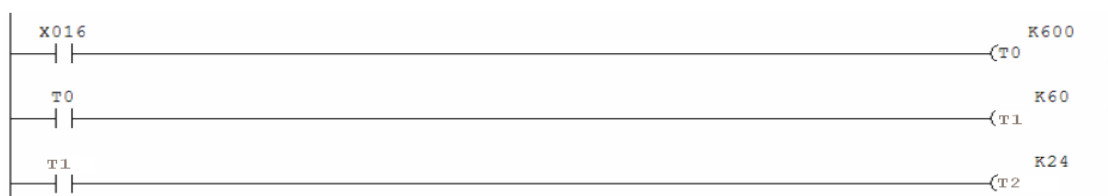
Baza de timp	Adresa timer în PLC-ul FX3u
100ms	0-199
10ms	200-245
1ms (cu reținere)	246-249
100ms (cu reținere)	250-255
1 ms	256-511



Conform tabelului, timerul folosit în schema de mai sus este unul cu o perioadă de 100ms între 2 incrementări successive. Atributul “cu reținere” denotă faptul că la inactivarea lui X000 timerul va păstra ultima valoarea până la care s-a incrementat, cât timp X000 era

activ. Acest tip de timere are nevoie si de comanda de **Reset** care se executa in cazul de fata prin activarea intrarii X002. Conform schemei ieșirea Y000 se va activa după 4s (40*100ms) de la apăsarea butonului X000.

Printr-un calcul simplu se observa ca durata maxima care poate fi obtinuta folosind un timer este de $32.767 \times 0.1\text{sec} / 60 = 54.36$ minute. Daca in schimb e nevoie de mai mult decat atat, tinand cont ca aplicatiile industriale pot avea nevoie de evenimente de verificare de exemplu odata la 12 sau 24 de ore, atunci se vor cascada mai multe timere obtinand astfel durate mai lungi. În schema de mai jos s-au cascadat 3 timere: primul va număra minute, al doilea ore, iar al treilea zile.



2.4. Senzori de proximitate

Pentru controlul intrarilor PLC-ului se pot folosi diferite tipuri de senzori digitali de proximitate. În funcție de principiul fizic folosit distingem mai multe tipuri de senzori de proximitate:

- capacitivi – determină prezența unui obiect în perimetrul senzorului prin variația cîmpului static creat în jurul senzorului; se detectează diferite tipuri de obiecte, indiferent de material (plastic, metal, sticlă, etc.); distanța de detecție este dependentă de dimensiunea obiectului, fiind în intervalul 0-2cm.
- inductivi – determină prezența unui obiect metalic în imediata apropiere a senzorului prin variația cîmpului magnetic creat de senzor; distanța de detecție este mult mai mică (0-0,3 cm) și se detectează numai obiecte fero-magnetice
- optici – detectează trecerea unui obiect prin fața senzorului optic prin opturarea sursei de lumină (ex: LED)
- mecanici – detectează un obiect care prin atingere cu senzorul închide un contact (microîntrerupătoare, limitatoare, etc.)

În lucrarea de față se folosesc senzori capacitivi (figura 11,a), inductivi (figura 11,b) și optici (figura 11,c) produși de firma Fotek



Figura 11. Senzori de proximitate.

2.5 Elemente de execuție.

Ca și elemente simple de execuție se pot folosi: electromagneți, electrovalve, servomotoare, motoare electrice de curent continuu sau de tip pas-cu-pas, dispozitive de avertizare sonoră (buzzer) și auditivă (LEDuri sau becuri), etc. Aceste elemente necesită o sursă de alimentare și o anumită schemă de acționare.

Pentru motoarele de curent continuu acționarea într-o singură direcție de rotație se poate face direct printr-o ieșire a PLC-ului. Ieșirea va juca rol de comutator într- o schemă în care mai apare motorul electric și sursa de alimentare. Cu scop de protecție, în paralel cu motorul se va conecta o diodă în sens invers cu direcția de curgere a curentului Pentru

acționarea motorului în două direcții de rotație este nevoie de un circuit de acționare în punte (denumit și circuit de tip H) și de 2 ieșiri ale PLC-ului (pornit/oprit și direcție). Electromagneții se pot folosi în mod similar cu acționarea într-o direcție.

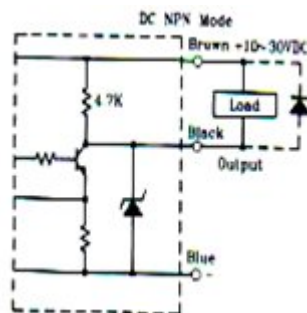
Pentru acționarea buzzerelor sau a LED-urilor se poate folosi sursa internă de 24V a PLC-ului. Limitarea curentului se va face prin rezistențe calibrate la curentul nominal prin aceste dispozitive (ex: 10mA în cazul LED-urilor).

3. Mersul lucrării

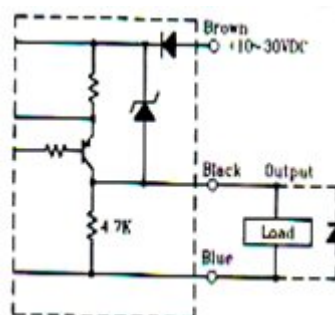
- 3.1 Se instalează mediul de programare DX-Developer și se testează diferitele facilități ale mediului
- 3.2 Se editează mai multe programe folosind diverse tipuri de simboluri grafice, cu diferite funcționalități. Se va verifica corectitudinea programului scris prin funcția de conversie
- 3.3 Sa se conecteze senzori de diferite tipuri la intrările PLC-ului si sa se controleze iesirile pe baza unei logici prestabilite. Sa se temporizeze activarea/dezactivarea iesirilor cu ajutorul timerelor.
- 3.4 Să se realizeze un program care controlează o bandă transportoare pe care se deplasează piese. În momentul în care un senzor detectează existența unei piese pe bandă, PLC-ul va activa comanda unui braț de împingere (ex: cu electromagnet) care va muta piesa pe o altă bandă transportoare; mișcarea va fi temporizată cu 2 secunde. Procesul începe la apăsarea unui buton simplu.
- 3.5 Sa se realizeze un program care simulează pornirea a 8 benzi transportoare legate la iesirile Y000 – Y007. Pentru o economie de energie, benzile fiind foarte lungi iar motoarele consumand multa energie electrica, se cere pornirea celor 8 benzi transportoare la un interval de 5 secunde între fiecare 2 consecutive. După ce toate benzile au pornit, se așteaptă un interval de timp de 10 secunde (în realitate ar fi un regim de 8 ore de lucru) înainte ca acestea să fie oprite pornind de la ultima cu aceeași pauză de 5 secunde între ele.

Anexa:

Schema de conectare a senzorilor de proximitate din seria Fotek CP



Senzor de tip NPN



Senzor de tip PNP