

AUTOMATE PROGRAMABILE

9.1. Diagrame Ladder (programarea grafică cu simboluri)

Metoda de programare grafică cu simboluri de tip contact și bobină de releu (programarea folosind diagrame ladder) a apărut în cazul automatelor programabile datorită necesității de a exista un mod facil de programare care să permită realizarea de aplicații fără a fi nevoie de cunoștințe complexe de programare.

Scrierea unui program în limbajul orientat pe diagrame ladder presupune desenarea unei diagrame similare unei scheme electrice cu contacte. Elementele componente ale diagramei ladder modelează funcționarea elementelor unei scheme cu contacte.


Elementele de bază utilizate pentru scrierea unui program în limbajul Ladder sunt contactele, bobinele, temporizatoarele, numărătoarele și blocurile funcționale.

Contacte. Sunt cunoscute ca intrări ale automatelor programabile. Ca și notație cea mai utilizată este folosind litera I (I0.0., I0.1 ... I0.n)

La intrările automatelor programabile pot fi conectate dispozitive care au două stări de funcționare cum ar fi contactele auxiliare ale contactoarelor și releelor, contactele normal închis sau normal deschis ale butoanelor de comandă, detectoarelor de mărimi fizice, ieșirile digitale ale unor aparate de măsură, protecție sau comandă, etc.

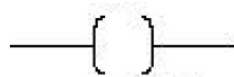
Ca și în lumea reală, contactele pot fi normal deschise sau normal închise în diagrama ladder ca și cele din figură:

 contact normal deschis

 contact normal închis

Prin aranjarea contactelor în serie sau în paralel pot fi realizate operații logice asupra stării semnalelor. Cele normal deschise sunt testate pentru valoarea „1” a semnalului iar cele normal închise pentru valoarea „0”. În al treilea caz sunt reprezentate contacte care nu sunt doar citite, asupra lor se execută și anumite modificări/reinițializări.

Bobine. Acestea sunt de fapt ieșirile dinspre automat spre proces. Ca și notație cea mai utilizată este folosind litera Q (Q0.0, Q0.1 ... Q0.n). Ca și în cazul bobinelor din schemele electrice, bobinele din programele utilizând diagrame ladder pot avea două stări: alimentate sau nealimentate. La aceste ieșiri pot fi conectate dispozitive care au două stări de funcționare cum ar fi bobinele contactoarelor sau releelor, elemente de semnalizare acustică sau luminoasă, intrările digitale ale unor aparate de măsură, protecție sau comandă, intrările digitale ale altor automate programabile sau sisteme de comandă etc.

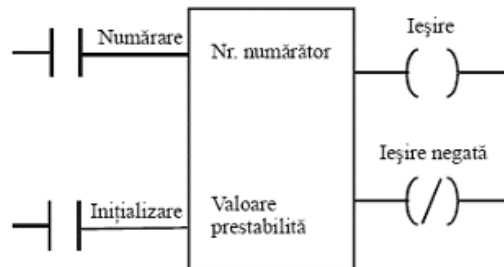
 bobină simplă

etichetă

Temporizatoarele sunt elemente de programare care modelează funcționarea releelor de timp și a contactelor temporizate. Ele sunt utilizate pentru a realiza acțiuni întârziate sau ce durează un anumit interval de timp.



Numărătoarele sunt elemente de programare care pot primi o serie de impulsuri care sunt analizate în cadrul programului Ladder pentru a detecta numărul de apariții ale unor evenimente cum ar fi: numărul de pași efectuați de un motor pas cu pas, numărul de conectări-deconectări ale unui aparat. Numărul acestor evenimente poate fi comparat cu anumite valori prestabilite și în funcție de rezultatul acestor comparații pot fi luate anumite decizii și date comenzile corespunzătoare.



Blocurile funcționale materializează funcții mai complexe menite să ușureze scrierea programelor în limbaj Ladder. Aceste blocuri modelează diverse categorii de funcții dintre care cele mai utilizate sunt următoarele: funcții de încărcare a unor constante numerice, funcții aritmetice, funcții logice pe 8 sau 16 biti, funcții de conversie a informației din diferite formate (funcții pentru actualizarea rapidă a intrărilor și ieșirilor. De obicei, formatul și modul de funcționare a blocurilor funcționale diferă de la un automat la altul, fiind specific fiecărui producător în parte.

Scrierea unui program în limbajul diagrame ladder poate fi realizată într-un mod simplu pornind de la schema electrică cu contacte.

Ceea ce trebuie să facă programatorul este să transpună schema respectivă folosind elementele de programare ale limbajului. Pentru acest lucru el va trebui să parcurgă următoarele etape:

- definirea listei dispozitivelor conectate la intrările și ieșirile AP;
- atribuirea unor identificatori de intrare și de ieșire acestor dispozitive;
- trasarea diagramei ladder;
- indicarea conexiunilor la automatele programabile a elementelor de comandă.

Exemple:

- scrierea unui program pentru comanda unui led cu ajutorul unui întrerupător:



- scrierea unui program pentru implementarea funcției SI:

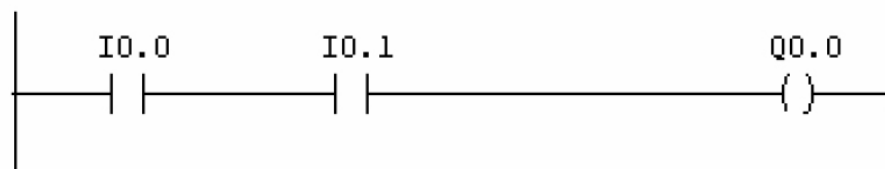


Tabela de adevăr atașată diagramei ladder de mai sus este:

I0.0	I0.1	I0.0 ȘI I0.1
0	0	0
0	1	0
1	0	0
1	1	1

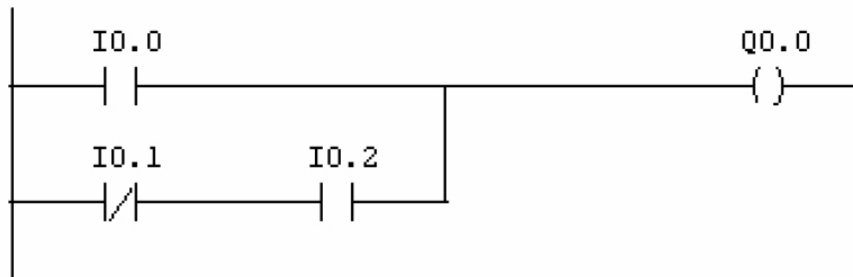
- scrierea unui program pentru implementarea funcției SAU:



Tabela de adevăr atașată diagramei ladder de mai sus este:

I0.0	I0.1	I0.0 SAU I0.1
0	0	0
0	1	1
1	0	1
1	1	1

- scrierea unui program pentru următoarea relație: $Q0.0 = I0.0 + \overline{I0.1} \cdot I0.2$



Aplicație: Controlul pornirii/opririi motorului electric

Să considerăm un sistem de control al pornirii și opririi unui motor. Comutatorul buton conectat la intrarea X₁ a automatului programabil este utilizat pentru pornirea motorului, și i se atribuie identificatorul de intrare I0.0.

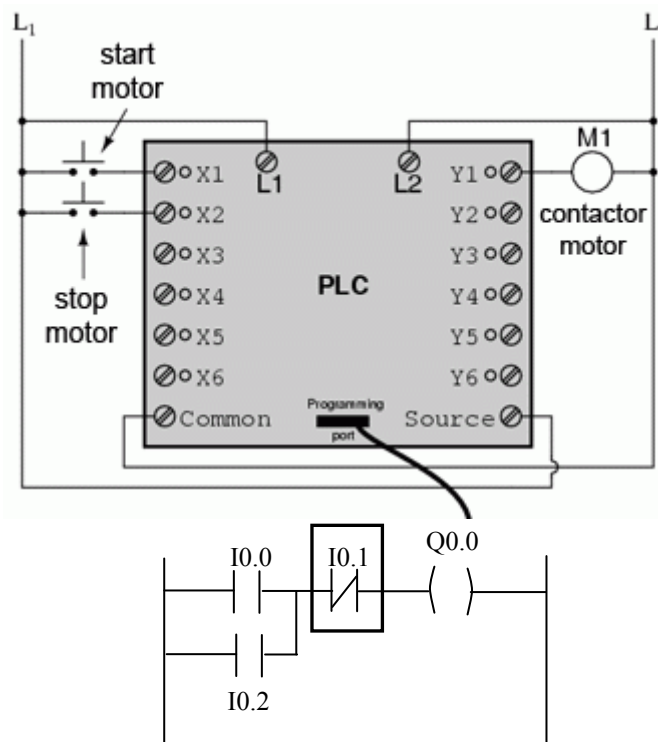


Fig. 9.1.

Comutatorul buton conectat la intrarea X_2 a automatului programabil este utilizat pentru oprirea motorului, și i se atribuie identificatorul de intrare I0.1.

Un contact adițional (virtual), adăugat în interiorul programului și denumit I0.2, utilizează bobina de ieșire ca și contact de reținere. Contactul motorului continuă să fie sub tensiune chiar și după ce butonul „start” este eliberat. Contactul I0.1 normal-închis este încadrat în dreptunghi semnifică faptul că este închis și conduce energie electrică.

La ieșirea Y1 a automatului programabil este conectată bobina contactorului motorului și i se atribuie identificatorul de ieșire Q0.0

<http://www.circuiteelectrice.ro/electronica-digitala/logica-ladder/automate-programabile-plc>

a. Pornirea motorului

Dacă apăsăm butonul de „start”, intrarea X_1 se va pune sub tensiune, închizând contactul I0.0 din program. Astfel se va aplica o tensiune pe bobina contactorului motorului. Contactul paralel I0.2 se va închide și el, iar circuitul va rămâne alimentat (figura 9.2).

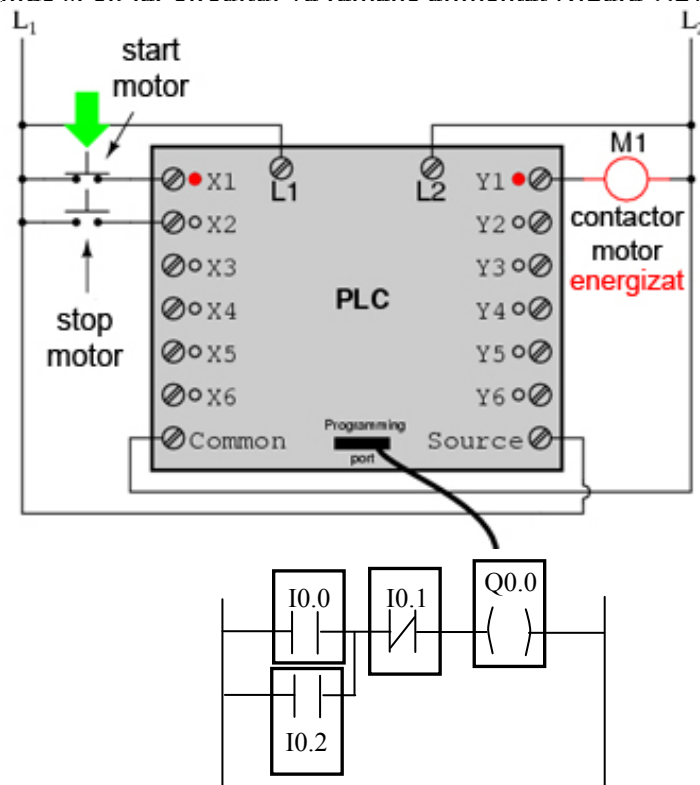


Fig. 9.2.

Acum, dacă eliberăm butonul de „start”, contactul I0.0 normal-deschis se va reîntoarce la poziția sa normală (deschis). Motorul va continua însă să funcționeze, deoarece contactul de reținere intern continuă să alimenteze bobina motorului (figura 9.3).

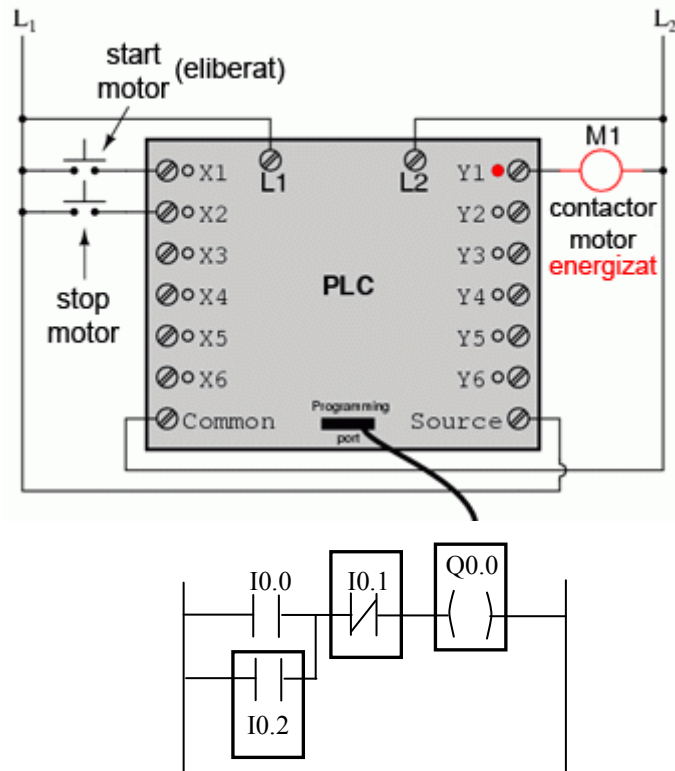


Fig. 9.3.

b. Oprirea motorului

Pentru a opri motorul, trebuie să apăsam pentru o durată scurtă butonul „stop”. Acesta va alimenta intrarea X₂ și va deschide contactul (virtual) normal-închis. Continuitatea circuitului înspre bobina Q0.0 va fi întreruptă (figura 9.4).

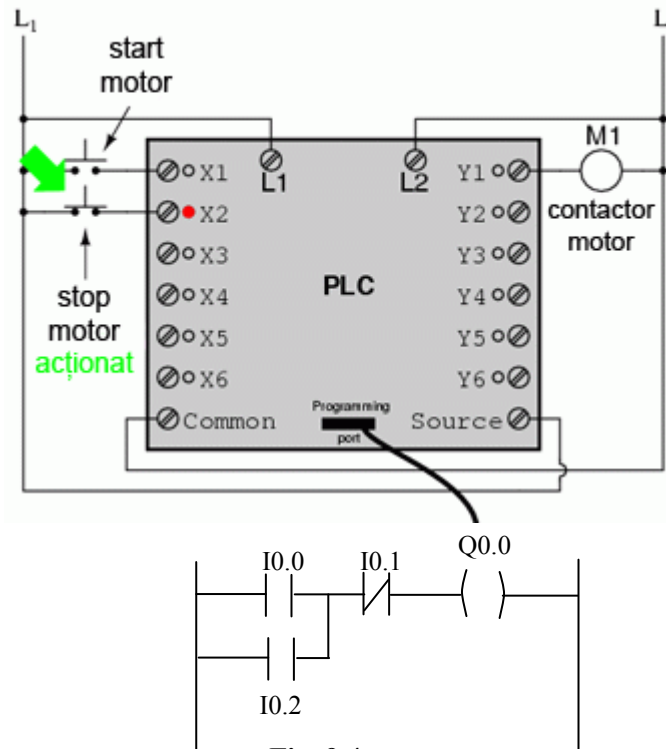


Fig. 9.4.

Când butonul de „stop” este eliberat, intrarea X₂ nu mai este sub tensiune. Contactul I0.1 se reîntoarce în poziția sa normală (închis).

Motorul nu va reporni însă până când comutatorul de „start” nu este acționat, datorită „pierderii” contactului de reținere I0.2.

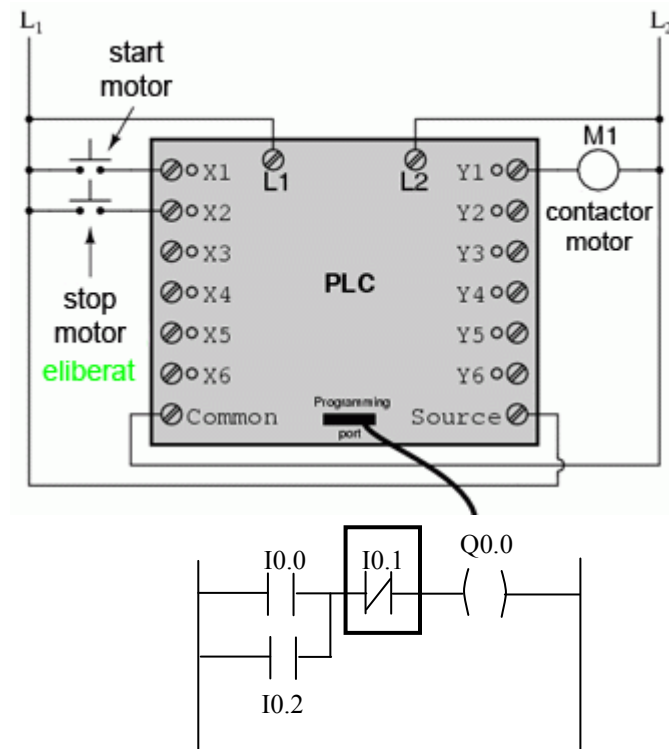


Fig. 9.5.

Întrebări:

1. Care sunt elementele componente ale Diagramei ladder?
2. Ce sunt contactele? Reprezentare grafică.
3. Ce sunt bobinele? Reprezentare grafică.
4. Ce sunt temporizatoarele? Reprezentare grafică.
5. Realizarea unui program utilizând limbajul cu diagrame ladder pentru implementarea unei funcții SI. Tabela de adevăr.
6. Realizarea unui program utilizând limbajul cu diagrame ladder pentru implementarea unei funcții SAU. Tabela de adevăr.

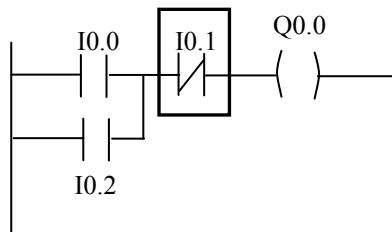
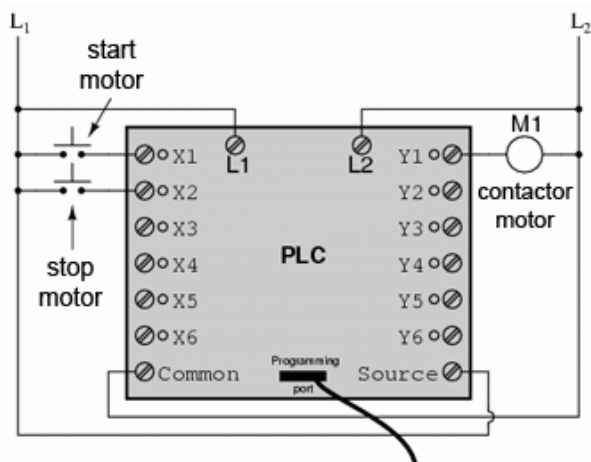


Fig. 9.1.

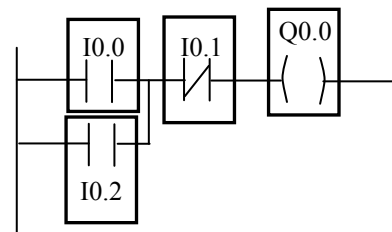
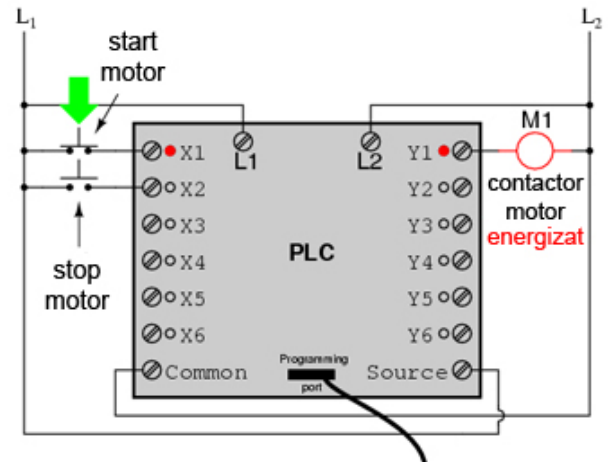


Fig. 9.2.

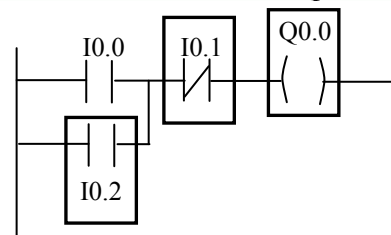
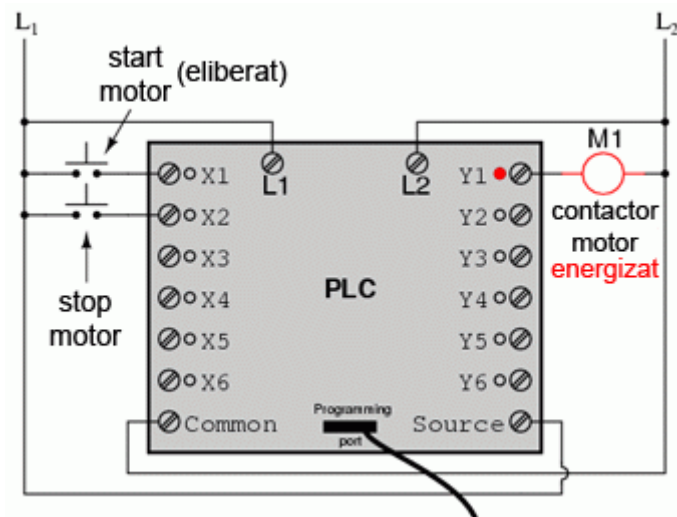


Fig. 9.3.

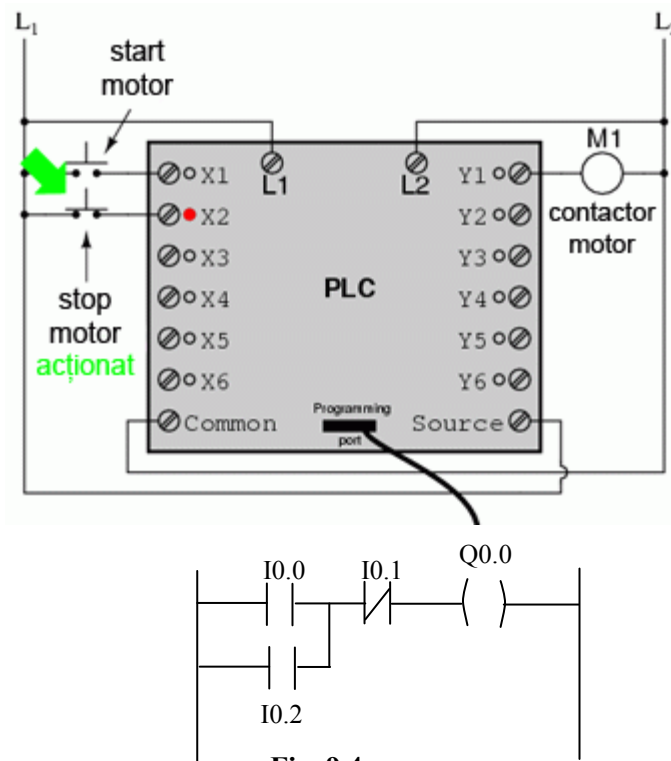


Fig. 9.4.

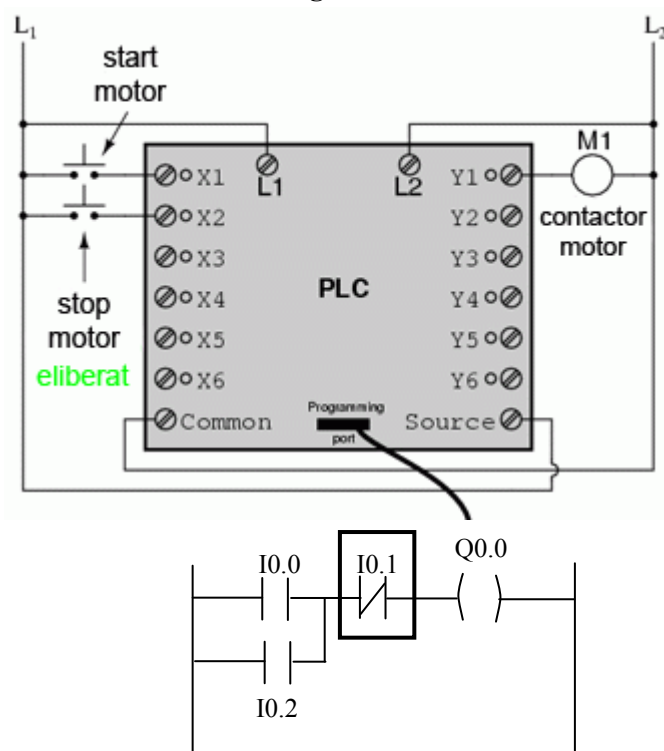


Fig. 9.5.