

Lucrarea 6. Programarea dispozitivelor de tip PLC

1. Obiectivul lucrării

În cadrul acestei lucrări se vor aprofunda instrucțiunile de programare a dispozitivelor de tip PLC. Se vor prezenta mai multe categorii de instrucțiuni ale limbajului LD, cum ar fi instrucțiuni de setare/resetare variabile, transfer de date, generare de impulsuri, operații aritmetice și logice, transmisie și recepție serială, conversitt de date, instrucțiuni de control al programului, etc.

2. Considerații teoretice

Limbajul LD oferă un set bogat de instrucțiuni, unele specifice pentru manipularea semnalelor (ex: setare/resetare, conversie analog/digitală, etc.), altele cu caracter mai jeneral, asemănătoare celor existente în alte limbaje de programare (instrucțiuni condiționate, de salt, etc.). O instructiune, sau mai corect o funcție se activează în cazul în care circuitul care o precede s-a închis. Grafic funcția apare între paranteze drepte (un dreptunghi în standardul limbajului LD) și conține numele unei funcții/instrucțiuni și parametrii aferenți.

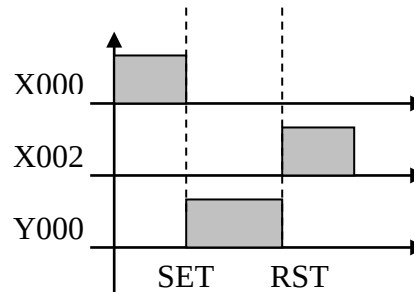
-----[IFUNCTIE par1, par2, ...]----

În continuare se prezintă succint cele mai uzuale instrucțiuni, din fiecare categorie în parte.

2.1. Instrucțiuni comune:

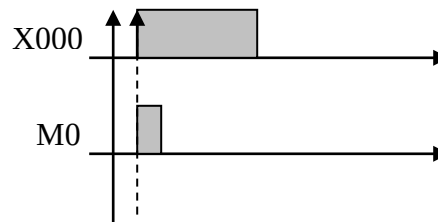
- Setare SET si resetare RST bit

```
      X000
|-----| |-----[ SET Y000 ]-|
|      X002      |
|-----| |-----[ RST Y000]-|
```

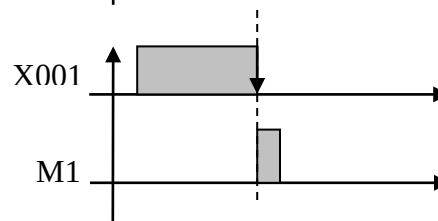


- Generare de impuls pe frontal urcator (PLS) si pe frontal coborator (PLF) al conditiei (se genereaza un impuls scurt care dureaza cat ciclul de parcurgere a diagramei)

```
      X000
|-----| |-----[ PLS M0 ]--|
```



```
      X001
|-----| |-----[ PLF M1 ]--|
```



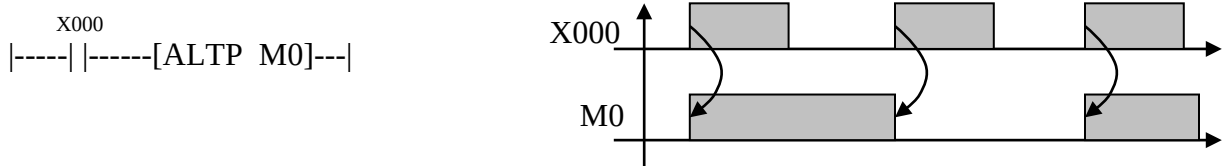
- o metoda echivalenta de actionare a iesirii pe front este utilizarea “releului” cu actionare pe front:

-----|↑|----- actionare pe front urcator

sau

-----|↑|----- actionare pe front coborator

- **Comutarea bitului intre 0 si 1 la fiecare impuls al conditiei (ALT)**



2.2. Instructiuni aplicative

- **Instructiuni de transfer:**

Transfer	16 biti	32 biti	flotant
1 la 1	MOV	DMOV	DEMOV
N la N	BMOV	-	-
1 la N	FMOV	DFMOV	-

X000
|-----| |-----[MOV D1 D2]-----| - muta D1 in D2

|-----| |-----[BMOV D1 D7 K3]-| - muta 3 cuvinte incepand de la D1 la destinatia D7,D8, D9

|-----| |-----[FMOV D1 D7 K3]--| - functie “fill” de umplere; muta data din D1 in 3 campuri incepand de la D7 (D7, d8, D9)

- **instructiuni de comparare**

Comparare	16 biti	32 biti	flotant
1 la 1	CMP	DCMP	DECMP
1 la interval	ZCP	DZCP	DEZCP

|-----| |-----[CMP D0 K10 M0]-----| - instructiunea seteaza M0 daca D0>K10
Seteaza M1 daca D0=K10
Seteaza M2 daca D0<K10

ZCP – comparare cu “zona”

|----| |-----[CMP K10 K20 D0 M0]-----| - instructiunea seteaza M0 daca $D0 > K10$
Seteaza M1 daca $K10 \leq D0 \leq K20$
Seteaza M2 daca $D0 < K10$

- **comparatii “inline”**

Comparatie	16 biti	32 biti
Mai mare	>	D>
egal	=	D=
Mai mic	<	D<
Mai mare sau egal	>=	D>=
Nu este egal	<>	D<>
Mai mic sau egal	<=	D<=

|--[= T10 K6]-----[Y000]-| - Y000 este cuplat daca conditia $T10 = K6$ este indeplinita

2.3. Instructiuni aritmetice

Operatia	16 biti	32 biti	flotant
Adunare	ADD	DADD	DEADD
Scadere	SUB	DSUB	DESUB
multiplicare	MUL	DMUL	DEMUL
Impartire	DIV	DDIV	DEDIV
Radacina patrata	SQR	DSQR	DESQR

X000
|----| |-----[ADD K20 D0 D5] - cand cond. X000 este indeplinita atunci
 $D10 = D0 + 20$

2.4. Instructiuni de transmitere/receptie: TO/FROM

- sunt instructiuni care opereaza cu SFM-uri (Special Function Modules) – module cu functii speciale atasabile la unitatea de baza PLC (ex: modul de conversie analogical, modul de transmisie seriala, etc.)
- in cadrul unui modul functional exista o memorie tampon (BFM Buffer Function Memory) in care sunt locatii cu functii predefinite; de exemplu la modulul analogic FX2N-4DA BFM #0 pastreaza “modul de iesire” iar BFM #1 pastreaza valoarea digitala pentru canalul 1.
- Manualul modulului special descrie functia fiecarei locatii din memoria tampon atasata

Instructiunea TO trimite o data de la CPU la SFM. Formatul:

-[TO SFM BFM SRC NUMAR]-

Exemplu:

|---| |-----[TO K2 K0 H1122 K1]-|

- scrie o data (constanta hexa 1122) la BFM #0 in a 3-a unitate SFM intalnita (K2); de exemplu daca este o unitate FX2N-4DA atunci se seteaza canalul 1 si 2 la iesire 0-20mA si canalul 3 si 4 la iesire 4-20mA

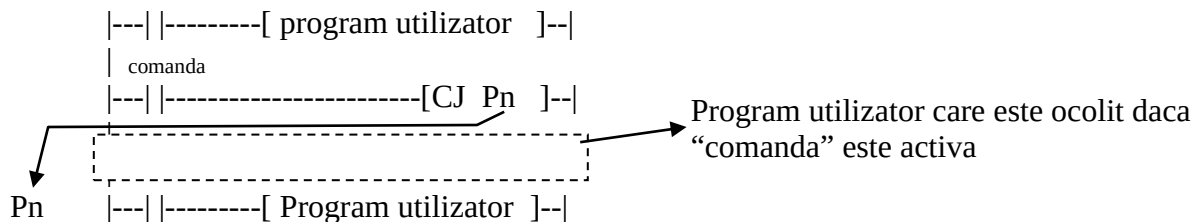
Instructiunea FROM muta o data din SFM (dintr-o locatie BFM) in memoria PLC-ului. Formatul este acelasi cu instructiunea TO.

-[FROM SFM BFM DEST NUMAR]-

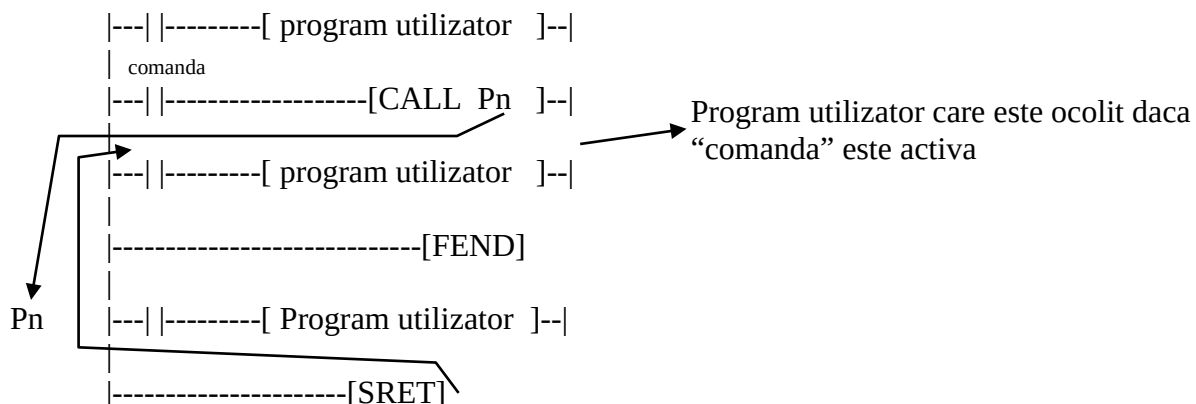
2.5. Instructiuni de control al programului

In aceasta categorie intra instructiunile de salt conditionat (**CJ**) apel de subrutina (**CALL**), revenire din subrutina (**SRET**) si sfarsitul programului principal (**FEND**).

Instructiunea CJ – salt conditionat



Instructiunile CALL si SRET – salt la rutina si revenirea din rutina



2.6. Instrucțiuni ale ceasului de timp-real

PLC-ul are un ceas de timp-real incorporate care se poate utiliza pentru a măsura timpul scurs sau pentru a temporiza anumite acțiuni în funcție de ora curentă.

Informațiile de timp se păstrează în registrele D8013-D8019 după cum urmează:

D8013 – secunde

D8014 – minute

D8015 – ora (format 24 ore)

D8016 – ziua

D8017 - luna

D8018 – anul (2 cifre)

D8019 – ziua din săptămână (0- duminică, 6- sâmbătă)

Instrucțiunea TRD (time read) citește cei 7 regiștri ai ceasului într-o zonă de memorie indicată

Instrucțiunea TWR (time write) scrie cei 7 regiștri ai ceasului dintr-o zonă de memorie.

Instrucțiunea TCMP (time compare) compară ora, minutul și secunda cu conținutul a 3 regiștre consecutive.

3. Mersul lucrării

3.1 Se vor scrie programe care încorporează instrucțiunile prezentate în lucrare

3.2 Să se scrie exemple de program care activează anumite elemente de acționare în funcție de timp (ex: achiziție periodică de date)

3.3 Să se scrie un program cu mai multe regimuri de lucru controlate cu ajutorul timpului și al instrucțiunilor de salt.

3.4 Să se analizeze asemănările și deosebirile dintre limbajul LD și un limbaj de programare obișnuit (ex: C); să se analizeze în mod special aspectele de execuție în regim concurent.