

**PROGRAMAREA CONTROLERULUI SPC200
PENTRU COMANDA UNUI SISTEM PNEUMATIC
DE POZIȚIONARE FESTO**

Scopul lucrării

Scopul acestei lucrări este înțelegerea modului de funcționare a sistemului pneumatic de poziționare Festo și însușirea unor noțiuni de programare a controlerului SPC200 pentru comanda celor două axe de translație și a efectorului final din componența sistemului pneumatic.

1. Generalități

Standul pneumatic Festo (figura 1) conține un calculator de proces (1) două axe pneumatice de translație, una orizontală (axa X – 3)) și una verticală (axa Y – 2), un gripper pneumatic (5) conectat în partea inferioară a axei Y, un controler tip SPC200 (4) alimentat de la o sursa de curent continuu (7) și un sistem de oprire de urgență (6).

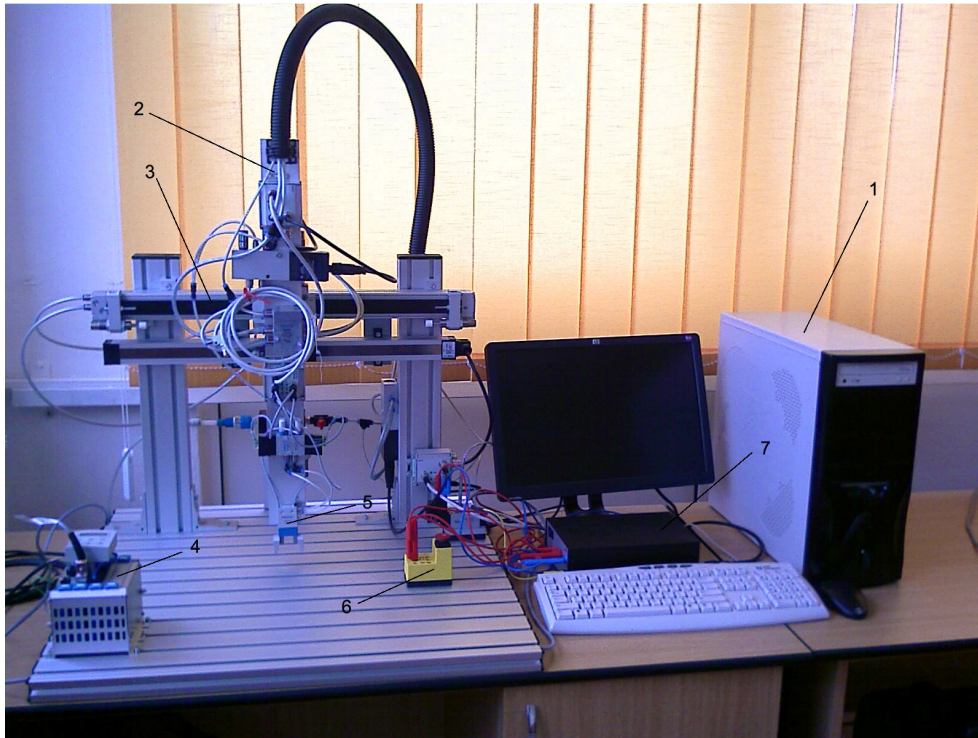


Figura 1. Standul pneumatic Festo

Programarea controlerului SPC200 se realizează cu ajutorul aplicației WinPISA 4.41. Aplicația permite crearea unor proiecte/programe noi, editarea unor programe existente, editarea listei cu poziții predefinite (*position list*), compilarea proiectelor/programelor, încărcarea și descărcarea programelor în/din memoria controlerului.

Programele pentru controlerul SPC200 se realizează în „*cod G*”, limbaj de programare folosit în programarea mașinilor unelte cu comandă numerică.

2. Crearea unui proiect nou

Un proiect nou se creează apelând funcția „*New Project*” din meniul „*File*”. Datele necesare pentru noul proiect sunt numele fișierului (.prj), titlul proiectului și descrierea proiectului (opțional).

Fereastra proiect (figura 2) asigură o ușoară gestionare a programelor pentru controler. Programele și lista cu poziții predefinite (figura 3) se găsesc în directorul „*Software*”. Controlerul permite utilizarea a 100 de programe (numerotate de la 0 la 99). Lista cu poziții

predefinite este unică pentru fiecare proiect și este creată automat împreună cu proiectul. Astfel, lista poate fi modificată dar nu poate fi recreată sau ștearsă.

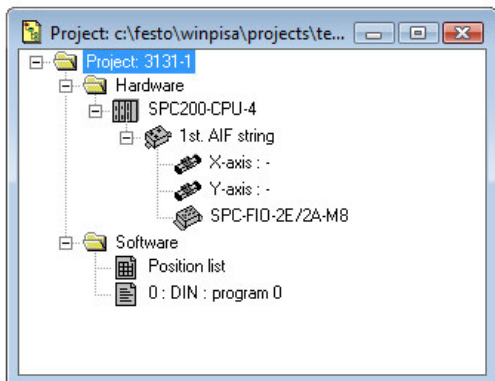


Figura 2. Fereastra Proiect

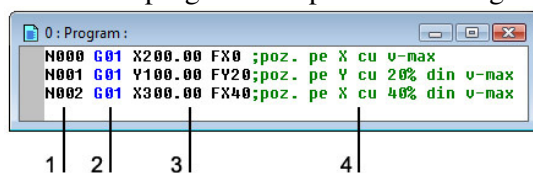
Index	Symbol	Axis X	Axis Y	Axis Z	Axis U
0	P0S0	-	-	-	-
1	P0S1	-	-	-	-
2	P0S2	-	-	-	-
3	P0S3	-	-	-	-
4	P0S4	-	-	-	-
5	P0S5	-	-	-	-
6	P0S6	-	-	-	-
7	P0S7	-	-	-	-
8	P0S8	-	-	-	-
9	P0S9	-	-	-	-
10	P0S10	-	-	-	-
11	P0S11	-	-	-	-

Figura 3. Lista cu poziții predefinite

Pașii pentru crearea unui program nou sunt următorii:

- Se selectează directorul „Software”;
- Din meniul „Edit” se apelează funcția „Insert object”;
- Se completează câmpurile „Title” și „Description” din fereastra nou deschisă;
- Se apelează butonul „Ok” pentru confirmare.

Structura unui program este prezentată în figura 4.



- 1 - Număr de înregistrare
- 2 - Comandă (cod G)
- 3 - Parametri comandă
- 4 - Comentarii

Figura 4. Structura unui program

Pentru editarea unui program nou sunt necesare caracteristicile hardware ale axelor și controlerului. Caracteristicile hardware se obțin astfel:

- Se pornește sursa de alimentare a controlerului (figura 1, (7));
- Din meniul „Online” se apelează funcția „Online mode”;
- Din meniul „Online->Upload” se apelează funcția „Hardware”.

Caracteristicile axelor de translație ale standului sunt prezentate în figura 5.

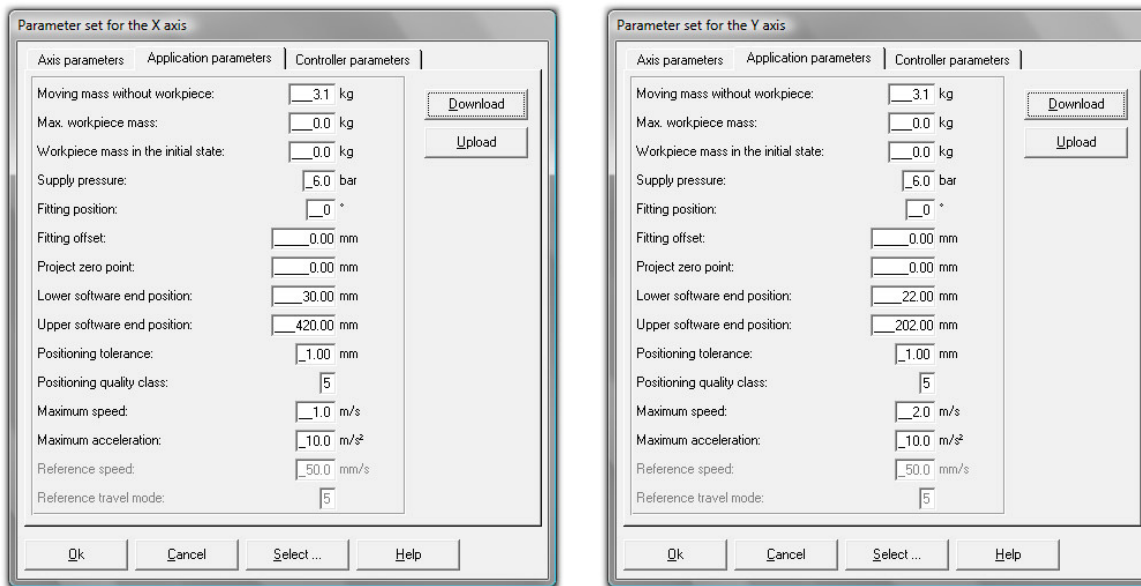


Figura 5. Caracteristicile axelor de translație

Un program pentru controlerul SPC200 este valid doar dacă fiecare linie are un număr de înregistrare. Atribuirea numărului de înregistrare se poate face manual sau automat. Controlerul permite încărcarea a maxim 100 de programe având în total maxim 2000 de linii de cod. Un program poate conține maxim 1000 de linii de cod. Numerotarea liniilor de cod se face începând de la 0 (N000) până la 999 (N999), fiecare linie având un număr de înregistrare unic. Pentru numerotare automată a liniilor se va apela funcția „*Start numbering*” din meniul „*Edit*”.

3. Comenzi pentru controlerul SPC200

G00

Sintaxă:

G00 [G90 | G91] [X | Y]<poziție>

unde <poziție>: [40-390] pentru axa X și [40-190] pentru axa Y

sau <poziție>: X@n Y@n (unde n – simbolul poziției din lista de poziții)

Descriere:

- poziționează efectorul final în coordonatele definite de <poziție>, cu viteză maximă, absolut sau relativ la poziția curentă (G90 sau G91)

Exemplu:

N000 G00 G91 X100	poziționează gripperul la 100mm în sens pozitiv pe axa X cu viteză maximă
-------------------	---

Notă:

În cazul în care nu se folosește una din opțiunile G90 sau G91 poziționarea va fi absolută.

G01

Sintaxă:

G01 [G90 | G91] [X | Y]<poziție> F<X|Y><viteza>

unde <poziție>: [40-390] pentru axa X și [40-190] pentru axa Y

sau <poziție>: X@n Y@n (unde n – simbolul poziției din lista de poziții)

<viteză>: reprezintă procent din viteza maximă [0..99] unde (0=100%,..., 99 = 99%)

Descriere:

- poziționează efectorul final în coordonatele definite de <poziție>, cu viteza definită de <viteză>, absolut sau relativ la poziția curentă (G90 sau G91)

Exemplu:

N000 G01 X100 FX20 Y150 FY30	poziționează gripperul pe axa X la 100mm cu viteza 20% din viteza maximă, pe axa Y la 150mm cu viteza 30% din viteza maximă
------------------------------	---

Notă:

În cazul în care nu se folosește una din opțiunile G90 sau G91 poziționarea va fi absolută.

G04

Sintaxă:

G04 < timp >

unde < timp >: timp de staționare (timp*10ms)

Descriere:

- oprește funcționarea sistemului timp de < timp >*10ms.

Exemplu:

N000 G04 250	Oprește sistemul timp de 2.5 secunde.
--------------	---------------------------------------

G08

Sintaxă:

G08 [X | Y]<acelerație>

unde <acelerație>: reprezintă procent din accelerația maximă [0..99] unde
(0=100%,..., 99 = 99%)

Descriere:

- setează accelerația axei X și/sau Y pentru apropierea de un punct de coordonate.

Exemplu:

N000 G08 X50	Axa X este accelerată cu 50% din accelerația maximă în apropierea punctului X 80mm.
N001 G01 X80 FX50	Accelerația este setată la 100% după poziționarea în punctul X 80mm.
N002 G08 X0	

G09

Sintaxă:

G09 [X | Y]<acelerație>

unde <acelerație>: reprezintă procent din accelerația maximă [0..99] unde
(0=100%,..., 99 = 99%)

Descriere:

- setează accelerația de frânare a axei X și/sau Y.

Exemplu:

N000 G09 X50	Accelerația de frânare este 50% din accelerația maximă înainte de atingerea punctului X 80mm.
N001 G01 X80 FX50	

G90

Sintaxă:

G90 [X | Y]<poziție>

unde <poziție>: [40-390] pentru axa X și [40-190] pentru axa Y

sau <poziție>: X@n Y@n (unde n – simbolul poziției din lista de poziții)

Descriere:

- poziționează efectorul final în *coordonatele absolute* (relative față de sistemul de referință 0) definite de <poziție>.

Exemplu:

N000 G00 G90 X200	poziționează gripperul pe axa X la 200mm față de 0, cu viteză maximă
-------------------	--

G91

Sintaxă:

G91 [X | Y]<poziție>

unde <poziție>: [40-390] pentru axa X și [40-190] pentru axa Y

sau <poziție>: X@n și/sau Y@n (unde n – simbolul poziției din lista de poziții)

Descriere:

- poziționează efectorul final în *coordonatele relative* față de poziția curentă definite de <poziție>, în sens pozitiv sau negativ.

Exemplu:

N000 G00 G91 X200	poziționează gripperul pe axa X la X+200mm față de punctul curent, cu viteză maximă; poziționează gripperul pe axa X la X-100mm față de punctul X+200, cu viteză maximă
N001 G00 G91 X-100	

M00

Sintaxă:

M00

Descriere:

Oprește execuția unui program.

Exemplu:

N000 G01 X100 FX50	poziționează gripperul pe axa X la 100mm față de 0, cu 50% din viteza maximă; oprește execuția programului
N001 M00	

M02

Sintaxă:

M02

Descriere:

Oprește execuția unui sub-program.

Exemplu:

N000 G01 X100 FX50	poziționează gripperul pe axa X la 100mm față de 0, cu 50% din viteza maximă; oprește execuția sub-programului
N001 M02	

M30

Sintaxă:

M30

Descriere:

Repetă un program.

Exemplu:

N000 G01 X100 FX50	poziționează gripperul pe axa X la 100mm față de 0, cu 50% din viteza maximă; repetă execuția programului
N001 M30	

L

Sintaxă:

L<n>

unde <n>: numărul programului din memoria controlerului;

Descriere:

Lansează în execuție un sub-program.

Exemplu:

N000 G01 X100 FX50	poziționează gripperul pe axa X la 100mm față de 0, cu 50% din viteza maximă; lansează în execuție sub-programul 1.
N001 L1	

#SQ

Sintaxă:

#SQ<n.n>

unde <n.n>: port de ieșire (de la 0.0 la 13.15);

Descriere:

Setează bitul unui port de ieșire.

Exemplu:

N000 #SQ0.0	Setează bitul portului 0.0 (semnal „1” logic).
-------------	--

#RQ

Sintaxă:

#RQ<n.n>

unde <n.n>: numărul portului (de la 0.0 la 13.15);

Descriere:

Resetează bitul unui port de ieșire.

Exemplu:

N000 #RQ0.0	Resetează bitul portului 0.0 (semnal „0” logic).
-------------	--

E05

Sintaxă:

E <număr de înregistrare>

Descriere:

Execută un salt necondiționat la linia <număr de înregistrare> din program.

Exemplu:

N010 G00 G91 X10	Poziționează axa X la 10mm în sens pozitiv față de
N012 E05 030	punctul curent;
...	Execută un salt necondiționat la linia 030;
N030 G00 G90 X100	Poziționează axa X la 100m față de 0.

#T

Sintaxă:

#T<I | Q><n.n> <număr de înregistrare>

unde I : port de intrare;

Q: port de ieșire;

<n.n>: numărul portului (de la 0.0 la 13.15);

Descriere:

Testează valoarea unui port (intrare sau ieșire). Dacă valoarea este „1” logic trece la linia <număr de înregistrare>. Dacă valoarea este „0” logic, execută linia următoare.

Exemplu:

N010 #TI0.0 12	Testează bitul portului de intrare 0.0 pentru
N011 E05 10	valoarea „1” logic. Trece la linia 012 dacă bitul
N012 G01 Y100 FY50	este 1. Re-execută linia 010 dacă bitul este „0”
	logic.

#TN

Sintaxă:

#TN<I | Q><n.n> <număr de înregistrare>

unde I : port de intrare;

Q: port de ieșire;

<n.n>: numărul portului (de la 0.0 la 13.15);

Descriere:

Testează valoarea unui port (intrare sau ieșire). Dacă valoarea este „0” logic trece la linia <număr de înregistrare>. Dacă valoarea este „1” logic, execută linia următoare.

Exemplu:

N010 #TNI0.0 12	Testează bitul portului de intrare 0.0 pentru
N011 E05 10	valoarea „0” logic. Trece la linia 012 dacă bitul
N012 G01 Y100 FY50	este 0. Re-execută linia 010 dacă bitul este „1”
	logic.

#LR

Sintaxă:

#LR<Registru> = <valoare>

unde <Registru>: numărul registrului (între 0 și 99);

<valoare>: valoare întreagă între -32768 și 32767

Descriere:

Salvează o valoare întreagă <valoare> în registrul <Registru> .

Exemplu:

N010 #LR0=0	Salvează valoarea „0” în registrul 0.
-------------	---------------------------------------

#AR

Sintaxă:

#AR<Registru> = <valoare>

unde <Registru>: numărul registrului (între 0 și 99);

<valoare>: valoare întreagă între -32768 și 32767

Descriere:

Incrementează valoarea salvată în registrul <Registru> cu valoarea întreagă <valoare>.

Exemplu:

N010 #LR0=0	Salvează valoarea „0” în registrul 0. Incrementează
N011 #AR0=1	cu „1” valoarea registrului „0”.

#TR

Sintaxă:

#TR<Registru> = <valoare> <număr de înregistrare>

unde <Registru>: numărul registrului (între 0 și 99);

<valoare>: valoare întreagă între -32768 și 32767

Descriere:

Testează valoarea salvată în registrul <Registru>. Dacă valoarea este egală <valoare>, execută un salt la linia <număr de înregistrare>. Dacă nu, execută linia următoare.

Exemplu:

N009 #LR0=0	Salvează valoarea „0” în registrul 0; Execută sub-
N010 L1	programul 1. Incrementează cu „1” valoarea
N011 #AR0=1	registrului „0”. Dacă valoarea registrului este 4,
N012 #TR0=4 14	execută linia 014. Dacă nu, execută un salt
N013 E010	necondiționat la linia 010.
N014 G01 X100 FX20	

4. Compilarea și execuția programelor

Programele din cadrul unui proiect se compilează cu ajutorul funcției „Project” din meniul „Compile”. După compilare, aplicația WinPISA va afișa într-o fereastră rezultatul compilării. În cazul în care există erori de compilare, acestea vor fi afișate indicându-se liniile din program în care apar erorile.

Pașii pentru salvarea unui proiect în memoria controlerului sunt următorii:

- Se salvează toate programele din cadrul proiectului;
- Se pornește sursa de alimentare a controlerului (figura 1, (7));
- Se activează modul „Online” cu ajutorul funcției „Online mode” din meniul „Online”;
- Se apelează funcția „Project” din meniul „Online->Download”.

Pentru lansarea în execuție a unui program se apelează funcția „Control axes” din meniul „Online” (în mod *online*). WinPISA va deschide o fereastră cu programele existente în memoria controlerului. Se va selecta programul dorit pentru lansare în execuție.

IMPORTANT !!!

Înainte de lansarea în execuție a oricărui program, se va pune în funcțiune standul, astfel:

- Se va porni compresorul și se va aștepta până ce manometrul acestuia indică presiunea de 6 [bar].
- Se va deschide **foarte încet** robinetul de siguranță (de culoare roșie) de pe filtrul de la alimentarea cu presiune a standului, prin rotire în sens orar până ce butonul robinetului se află în poziție orizontală (deschis). La deschiderea robinetului, axa Y se va deplasa automat în poziția maximă.
- Se va lansa în execuție programul dorit.
- În cazul constatării unei funcționări necorespunzătoare a standului, se va acționa butonul pentru oprire de siguranță (figura 1 (6)).

5. Exerciții propuse

1)

Să se realizeze un proiect care să conțină programul din figura 6.

```

2: Program : program 2
N000 G01 X100.00 FX10 Y150.00 FY10
N001 #SQ1.0
N002 #RQ1.1
N003 #TI1.0 5
N004 E05 3
N005 G01 Y100.00 FY10
N006 G01 X300.00 FX10
N007 G01 Y100.00 FY10
N008 #RQ1.0
N009 #SQ1.1
N010 #TI1.1 12
N011 E05 10
N012 G01 Y150.00 FY10
N013 M30
    
```

Figura 6. Program demonstrativ

- Să se identifice rolul fiecărei linii din program.
- Să se identifice secvențele de cod utilizate pentru deschiderea și închiderea gripperului.

2)

Să se realizeze un program care poziționează gripperul sistemului de poziționare în următoarele puncte:

	Poziție pe axa X [mm]	Poziție pe axa Y [mm]	Viteza axei X [m/s]	Viteza axei Y [m/s]
1	120	80	0.30	0.50
2	120	130	0.30	2.00
3	330	130	1.00	2.00
4	330	60	1.00	1.00

3)

Să se realizeze un program pentru o aplicație de paletizare știind că:

- piesele prelucrate sunt de două tipuri iar prelucrarea lor se face alternativ la fiecare 20 de secunde.
- magazia cu piese are coordonatele (x,y) = (360,50);

- paleții pentru cele două tipuri de piese au coordonatele $(x,y)=(70,60)$ respectiv $(x,y) = (100,60)$;
- în zona de deplasare există un obstacol dreptunghiular de lungime 150mm (de la $X=150$ mm până la $X=300$ mm) și înălțime 50mm (de la $Y=80$ mm până la $Y=130$ mm);
- viteza de deplasare nu poate fi mai mare de 1 m/s (pentru ambele axe);
- accelerația la oprire în punctele de deschidere și închidere a gripperului nu poate depăși 0.5 m/s^2 pe axa Y;
- numărul de piese prelucrate este de 6 (câte 3 din fiecare tip);
- proiectul trebuie să conțină un program principal care numără piesele prelucrate și două sub-programe apelate din programul principal (câte unul pentru fiecare palet);
- punctele prin care va trece efectorul final (inclusiv punctele de încărcare/descărcare a pieselor) trebuie definite și preluate din lista de poziții (figura 3).