

**Az AutoSim 200 szoftver SFC (Sequential Function Chart)  
(sorrendi funkció ábra), más néven Grafcet (gráf)  
programozása**

**Maczik Mihály András**

**2020.**

Az SFC programozási nyelvről .....	1
Programkészítés.....	2
A programozás indítása .....	2
Új mappa létrehozása .....	2
Assistant használata.....	2
Az „Assistant (asszisztens)” ablak beállítási lehetőségei .....	3
A Grafcet típusú programozás lehetőségei.....	4
Asszisztens nélküli szerkesztés .....	5
Szimbólum táblázathasználata .....	6
Indítás .....	6
Szimbólum tulajdonságai (Symbol prpoerties) .....	6
Név (Name) .....	6
Változó megadása (Associated variable).....	6
Magyarázat (Associated comments).....	7
A szimbólum táblázat és szerkesztése .....	7
Operátorok.....	7
Az SFC program készítése .....	8
Szimuláció .....	9
Példák SFC programozáshoz.....	10
Egyszerű be-kikapcsolás .....	10
Öntartás .....	10
Flip-flop.....	11
Logikai függvények használata .....	11
Időkésleltetés .....	11
Feltételes akció .....	12
Számláló (Counter).....	12
Példa számlálóra.....	13
Késve bekapcsoló .....	13
Futófény .....	14
Háromfényes közlekedési jelzőlámpa .....	15
Két munkahenger mozgása 3 ciklusig.....	16
A bekötés.....	16
A szimbólumtábla.....	16
A program .....	17
Eseményvezérelt lefutó vezérlésbelső ismétlődéssel .....	18
A munkahengerek és a szelepek bekötése.....	18
A PLC elektromos bekötése .....	18
A szimbólumtábla.....	19
Az SFC program.....	19
Belső ismétlődés, inverz mozgás, belső várakozás, egyidejűség .....	20
Összetett feladat .....	21
A mozgásdiagram.....	21
Az elektro-pneumatikus elemek bekötése .....	22
A PLC elektromos bekötése .....	23
A szimbólumtáblázat.....	23
A Grafcet program.....	24
A főprogram (g0).....	25
Folyamatos üzemmód / lépés üzemmód.....	25
Egy ciklus üzemmód / több ciklus üzemmód.....	25
Rögzítő vészstop.....	26
Alaphelyzetbe hozó vészstop .....	26

## Az SFC programozási nyelvről

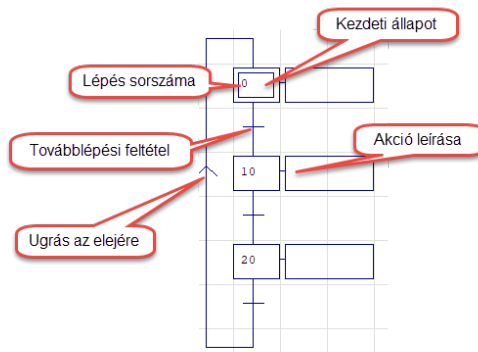
Lefutó típusú vezérlések programozásához célszerű SFC PLC programozási nyelvet választani, hiszen itt egy feltétel teljesülése elindít egy akciót, ami teljesülésekor vár egy újabb feltétel teljesülésére, ami elindít egy újabb akciót, és így tovább az eseménysor végéig, amikor majd visszaugrik a program az elejére.

A legelső lépésnél (nulladik lépés) kezdeti állapotokat adhatunk meg (pl.: számláló nullázása, kimenetek alaphelyzetbe állítása).

Egy feltétel több akciót is elindíthat, amiket vesszővel választunk el egymástól. Egy akció elindításához esetleg több feltételnek is teljesülnie kell, amik egymással különféle logikai kapcsolatban (nem, és, vagy) lehetnek.

A feltételek jelei és az akciók leírásai üresen is maradhatnak, ez nem okoz hibát a program futásában.

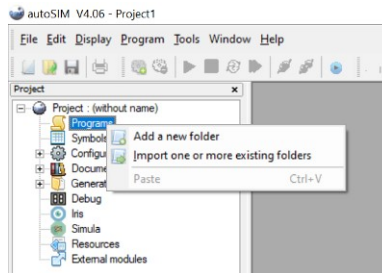
Az SFC programnyelv kiváló megoldást jelent pneumatikus munkahengerek összetett mozgássorának programozásához.



# Programkészítés

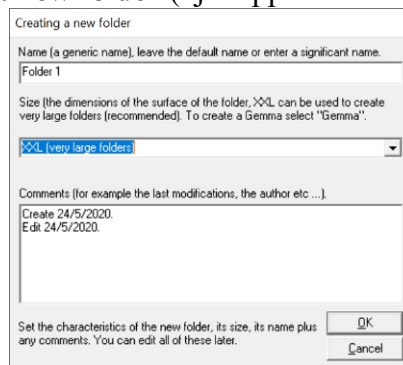
## A programozás indítása

Miután elindítottuk a szoftvert, a „**Project**” ablakban jobb egérgombbal kattintunk a „**Programs**” menüpontra, és az „**Add a new folder** (adjon hozzá egy új mappát)” lehetőséget választjuk.



## Új mappa létrehozása

Erre megnyílik a „**Create a new folder** (új mappa létrehozása)” ablak.



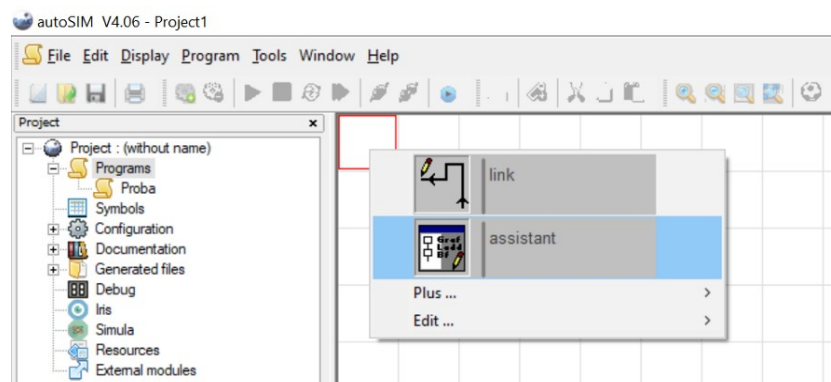
A „**Name** (név) mezőben nevezzük el mappánkat! Az alapértelmezett „**Folder1**” értéket cseréljük ki valami jellemzőbbre!

A „**Size** (méret) értéket hagyjuk **XXL**-en.

A „**Comments**” mezőbe jellemző megjegyzéseket írhatunk (pl.: készítés időpontja, utolsó szerkesztés időpontja).

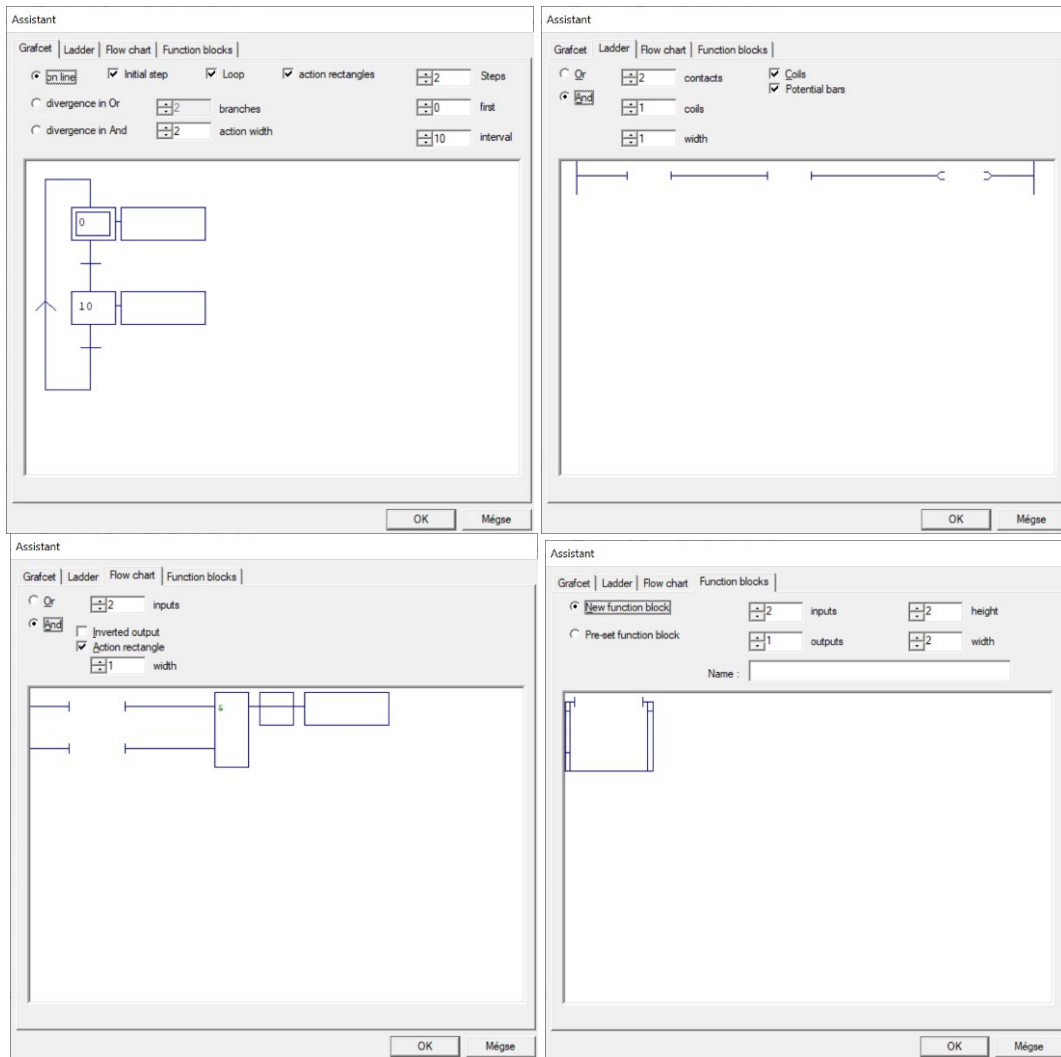
## Assistant használata

A megjelenő négyzettrácsos szerkesztő felület bal felső cellájára jobb egérgombbal kattintva válasszuk az „**assistant**” lehetőséget!



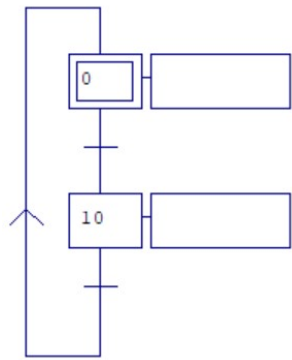
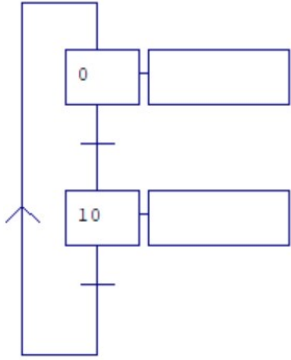
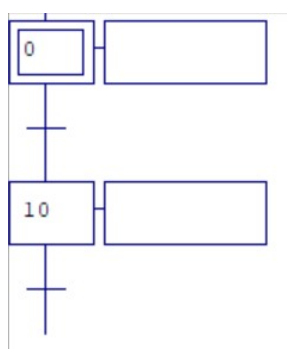
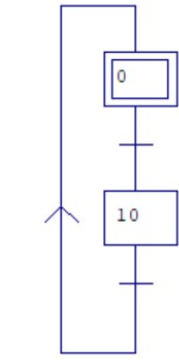
## Az „Assistant (asszisztens)” ablak beállítási lehetőségei

Négyféle programozási lehetőség közül választhatunk: **Grafcet** (gráf), **Ladder** (létra), **Flow chart** (folyamatábra), **Function block** (funkció blokk)



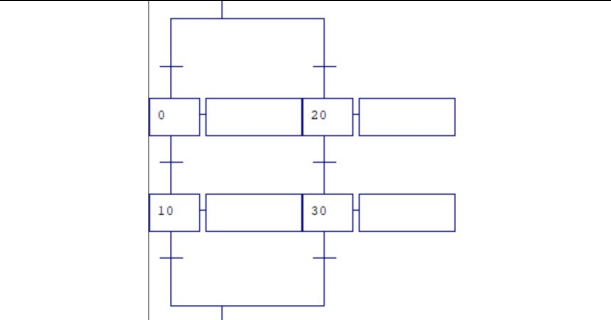
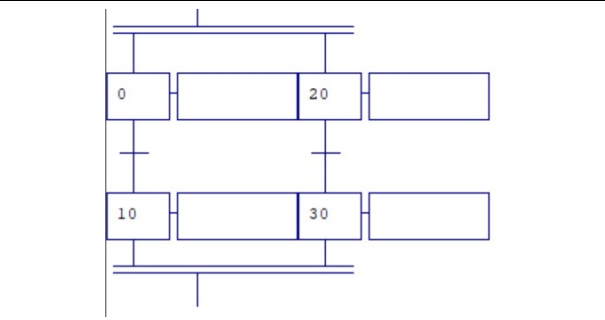
## A Grafcet típusú programozás lehetőségei

**On line:** Logikai elágazás nélküli folyamat főbb beállításai

Minden jelölőnégyzet bekapcsolva	<input type="checkbox"/> Initial step	<input type="checkbox"/> Loop	<input type="checkbox"/> action rectangles
Teljes funkcionalitás	Kezdeti beállítások kikapcsolása	Ciklusos ismétlődés kikapcsolása	Akciókat jelölő négyzetek kikapcsolása
			

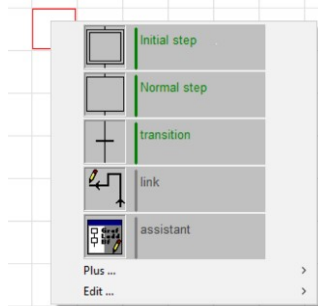
**On line:** logikai elágazás nélküli folyamat egyéb beállításai

<input type="text" value="2"/> Steps	<input type="text" value="0"/> first	<input type="text" value="10"/> interval	<input type="text" value="2"/> action width
Itt állíthatjuk be, hogy hány lépéses legyen a program	Az első lépés sorszámát állíthatjuk be (a 0 megfelelő)	Hánysával növekedjen a sorszámozás (a 10 megfelelő)	Az akciókat jelölő négyzet szélességét tudjuk beállítani

<b>Divergence in OR:</b> OR típusú program-elágazásos folyamat főbb beállításai	<b>Divergence in AND:</b> AND típusú program-elágazásos folyamat főbb beállításai
<input type="text" value="2"/> branches	<input type="text" value="2"/> branches
Hány elágazás legyen	Hány elágazás legyen
	

## Asszisztens nélküli szerkesztés

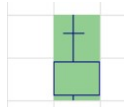
Bármely cellára (a bal felsőt kivéve) jobb egérgombbal kattintva beszúrhatunk kezdeti beállításokat, normál akciót, továbblépési feltételt, összekötő vonalat, de elindíthatjuk az asszisztent is.



Az akciót jelölő szimbólum melletti négyzetrácsra jobb egérgombbal kattintva lehet beszúrni az akciót leíró négyzetet.



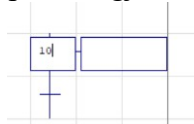
Kijelölni területet az egér bal gombját lenyomva tartva, vonszolással lehet. A kijelölt terület zöld háttérrel kap. A kijelölt területet bal egérgomb lenyomása mellett arrébb lehet vonszolni, „Del” gombbal törölni.



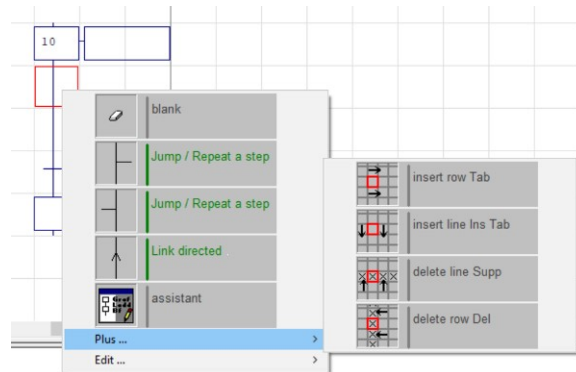
Ha egy cella fölé visszük az egeret, piros keretet kap, jobb egérgombbal tudjuk törölni (blank).



Belekattintva egy akciót jelölő jelképbe, megjelenik a kurzor, beírható annak sorszáma.



Négyzetrács sort vagy oszlopot beszúrni vagy törölni úgy lehet, hogy a kérdéses helyen jobb egérgombbal kattintunk, a „**Plus...**” lehetőséget választjuk, majd a megfelelő lehetőséget választjuk.



Ha egy egyszerű vonalra kattintunk jobb egérgombbal, beszúrhatunk egyszerű összekötést, vagy bal, vagy jobb leágazó összekötést.

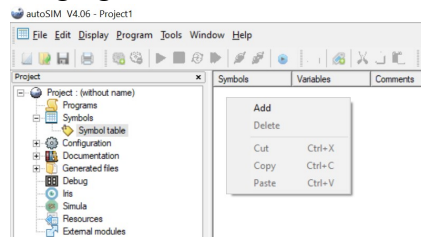
## Szimbólum táblázathasználata

Miért fontos a szimbólum táblázat? Ugyan a szimbólumok használata nem kötelező, de ha a programunkban csak a változók neveit látjuk, nehezebben tudjuk értelmezni az összefüggéseket, főleg ha sok idő elteltével újra megnyitjuk programunkat.

## Indítás

Miután elindítottuk a szoftvert, a „Project” ablakban jobb egérgombbal kattintunk a „Symbols” menüpontra, és az „Create a symbol table (készítsen egy szimbólum táblázatot)” lehetőséget választjuk.

Az üres munkaterületen jobb egérgombbal kattintsunk és válasszuk az „Add” lehetőséget!



## Szimbólum tulajdonságai (Symbol properties)

The 'Symbol properties' dialog box has three input fields: 'Name' (containing 'Start'), 'Associated variable' (containing 'i0'), and 'Associated comments' (empty). At the bottom, there is a note: 'The name can contain any character with the exception of \'. A long is restricted to 512 characters. A variable name must comply with IEC-1131-3 or autoSIM syntax.' and 'OK' and 'Cancel' buttons.

## Név (Name)

A „Name” mezőbe írjuk be az általunk választott szimbólum elnevezését!

- A név csak az **angol abc** kis- és nagybetűit, továbbá arab számjeleket tartalmazhat!
- **Nem kezdődhet számmal!**
- Hossza maximum 512 karakter lehet.

## Változó megadása (Associated variable)

**Kétféle módon lehet megadni, az IEC-1131 szabvány szerintit használjuk, a másik hibára fut!!!**

Változó neve	Változó jele (utána nincs szóköz)	Kezdő érték	Befejező érték
Bemenet	%i	0	9999
Kimenet	%q	0	9999
Időkésleltető	%t	0	9999
Számláló	%c	0	9999
Belső memória bit	%m	100	9999
Belső memória 16 bites szó	%mw	200	9999
Belső memória 32 bites dupla szó	%md	100	4998
Belső memória Tört szám	%mf	100	4998



### További változó, ami szimbólumtáblán nem adható meg, csak programozáskor

Változó neve	Változó jele (utána nincs szóköz)	Értéke
A Grafcet lépés sorszáma	%X	Amelyik lépésre hivatkozunk

- A betűjelek lehetnek kisbetűk és nagybetűk is, a program keverve is elfogadja és be is azonosítja.

### Magyarázat (Associated comments)

A szimbólumhoz magyarázat is fűzhető, de nem kötelező.

### A szimbólum táblázat és szerkesztése

Symbols	Variables	Comments
Start	i0	Indítás

- A szimbólumsorra kettőt kattintva szerkeszthető lesz, visszatér a „Symbol properties” táblázat.
- **Attól, hogy átírjuk a szimbólum nevét a táblában, még a programban nem cserélődik le, ott a régít törölni kell, majd ezután az újat hozzárendelni!**
- A szimbólumsorra jobb egérgombbal kattintva hozzáadhatunk, törölhetjük, kivághatjuk, másolhatjuk, beilleszthetjük.

Symbols	Variables	Comments
Start	i0	Indítás

Add		
Delete		
Cut	Ctrl+X	
Copy	Ctrl+C	
Paste	Ctrl+V	

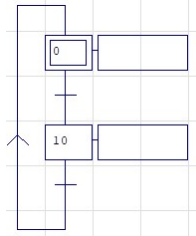
### Operátorok

Az operátorok feladata a feltételek, vagy az akciók formázása, kapcsolatba hozása.

Feltétel operátorok		Akcio operátorok	
/	NOT művelet	N	Invertálás
.	AND művelet	S	Bekapcsolás
+	OR művelet	R	Kikapcsolás
↑	Felfutó élre tűjel	P1	Impulzus a felfutó élnél
↓	Lefutó élre tűjel	P0	Impulzus a lefutó élnél
()	Zárójel a paramétereknek	+	Hozzáadás
=	Egyenlő	-	Kivonás
<>	Nem egyenlő	,	Elválasztás
<	Kisebb	{	Kód kezdete
>	Nagyobb	}	Kód vége
<=	Kisebb-egyenlő		
>=	Nagyobb egyenlő		
=1	Értéke mindig 1		

## Az SFC program készítése

1. Az asszisztens segítségével a feladatnak megfelelően kiválasztjuk a gráf típusát, lépésszámát.



2. Elkészítjük a szimbólumtáblát.

Symbols	Variables	Comments
Nyomogomb	%I0	
Lampa	%Q0	

3. Felvesszük a gráfba a szimbólumokat (ezzel a változókat is).

The sequence of diagrams illustrates the process of adding symbols to the SFC diagram. The first diagram shows the initial state. The second diagram shows a red box around the transition between steps 0 and 10. The third diagram shows the symbol 'Nyomogomb' added to the transition. The fourth diagram shows a red box around the symbol box of step 10. The fifth diagram shows the symbol 'Lampa' added to the symbol box of step 10. The sixth diagram shows the final state with both symbols added. To the right is a screenshot of the 'Editing a test' dialog box, showing the 'Symbols' category selected in the tree view, with 'Nyomogomb' and 'Lampa' listed in the table below.

Symbols	Variables	Comments
Nyomogomb	%I0	
Lampa	%Q0	

4. A feltétel vonalka cellájára kattintunk egyet, amikor az piros körvonalú, majd a 3 db pont ikonra. A megjelenő szerkesztő ablakban kinyitjuk a „Symbols” kategóriát, majd kétszer kattintunk a megfelelő szimbólumra. Ekkor a szimbólum neve felkerül a szerkesztő ablakba, amit az OK gombbal el is fogadhatunk. Ugyanígy járunk el az akció téglalapjánál is.

Nem fontos a szimbólumot használni, kiválaszthatjuk közvetlenül az EIC változót is, bár ekkor értelmét veszíti a szimbólumtábla.



Munkahengerek mozgásakor a véghelyzet-szenzorok és a mágnes-tekercesek beazonosítására viszont nélkülözhetetlen.

5. Ha kell, a szimbólumok elé, közé, mögé operátorokat helyezünk el. Ezt a jelet vagy fejből tudjuk és beírjuk a szerkesztő ablakba a megfelelő helyre, vagy lenyitjuk az „Operators” kategóriát, és kettőt kattintunk a megfelelőre. **Vigyázzunk a szóközökre!**

```
/%Q2, Nyomogomb, %t1(20)
```

Az operátor **kék** színű, a szimbólum **zöld**, a változó **lila**, az érték **fekete**.

## Szimuláció

A program futásának szimulációját az eszköztár **zöld körben fekete háromszög** ikonjával indíthatjuk el és állíthatjuk is meg . A szimuláció futása közben egy **forgó futball labda ikont** láthatunk .

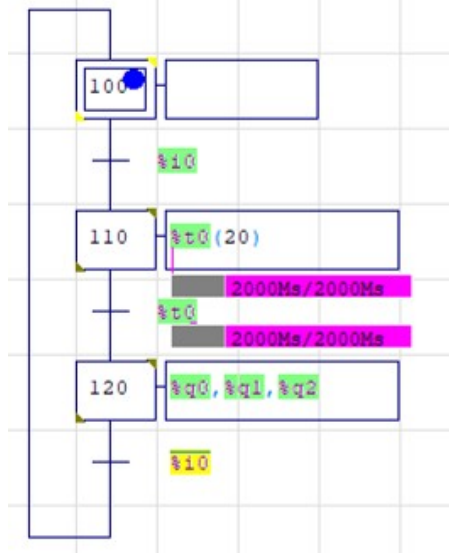
A szimulációban egy **kis ék, pattogó labda** jelzi, hogy melyik lépésnél tart a program, továbbá a lépést jelző négyszög jobb felső és bal alsó sarkán **kis sárga háromszögek** láthatóak.

A lépéseket jelző négyszög jobb felső és bal alsó sarkában egyébként **kis barna színű háromszögek** vannak, bármelyikre egérrel rákattintva a lépés aktívvá tehető (függetlenül attól, hol tart a program futása).

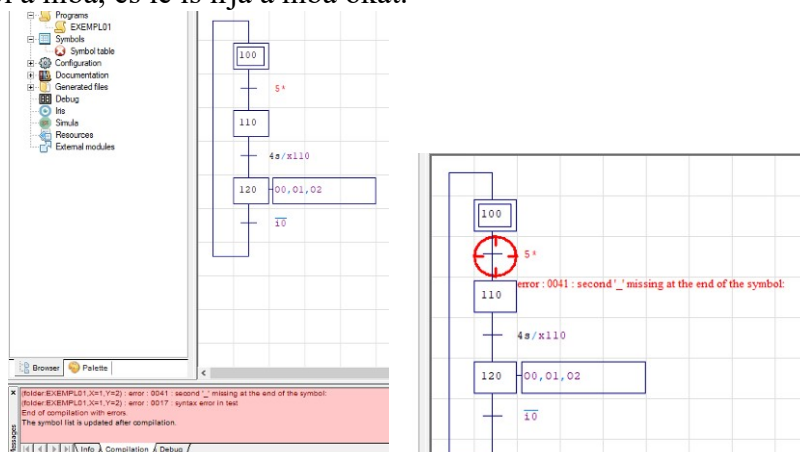
A továbblépési feltételeket jelző szimbólumokra, jelekre egér bal gombbal egyszer rákattintva azok aktívvá tehetőek, azok úgy is maradnak a következő kattintásig.

A passzív elemek (hamis érték) **zöld hátteret**, az aktív elemek (igaz érték) **sárga hátteret** kapnak.

Az időzítők, számlálók aktuális értékei **rózsaszín hátteret** kapnak, az eltelt időt egy csökkenő **szürke sáv** jelzi. Ezek a színek természetesen a beállításokban megváltoztathatók.



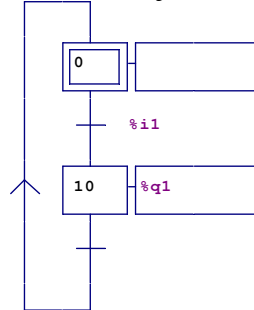
A szimuláció indításakor lefut egy ellenőrzés, ha ez hibát talál a „Compilation” részben rózsaszín háttérben piros betűszínnel jelzi azt. Az üzenet sorára kettőt kattintva a program megmutatja, hol a hiba, és le is írja a hiba okát.



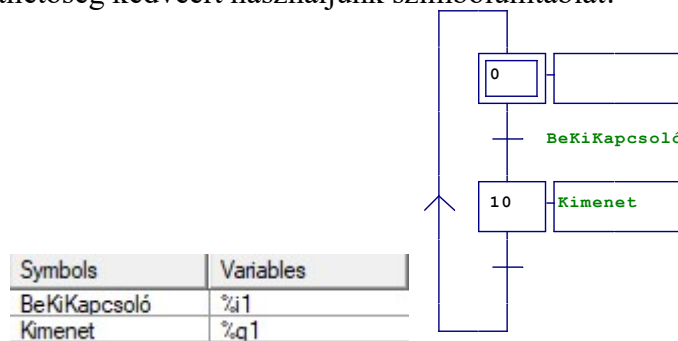
# Példák SFC programozáshoz

## Egyszerű be-kikapcsolás

%i1 bemenetre kötött reteszelés kapcsolóval kapcsoljuk be-ki %q1 kimenetet! A lépésszám kezdődjön 0-nál és tízesével növekedjen!



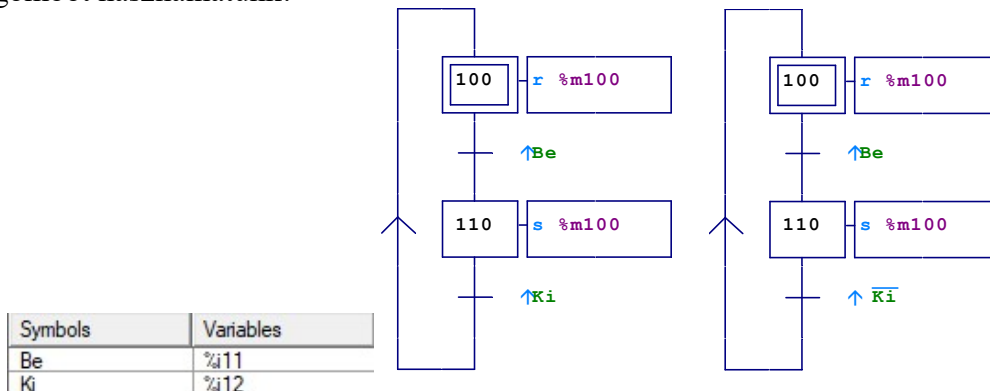
A könnyebb érthetőség kedvéért használjunk szimbólumtáblát!



## Öntartás

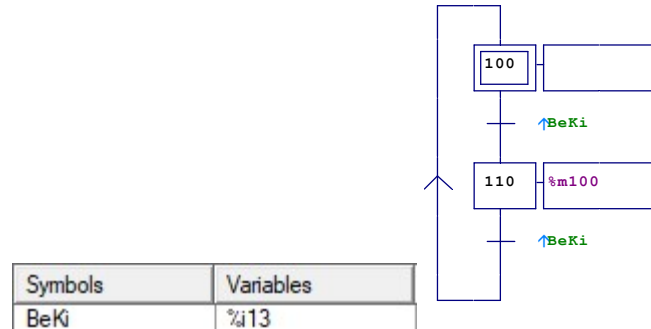
A lépésszám kezdődjön 100-nál és növekedjen tízesével! %m100 belső memóriabitet kapcsoljuk be %i11 bemeneti nyomógommbal, majd ki %i12 bemeneti nyomógommbal! Használjuk a r, s operátorokat! A nyomógombok prellezését szűrjük ki a ↑ felfutó él operátorral! Használunk szimbólumtáblát! Egy bonyolultabb mozgássornál a %m100-ra hivatkozva ezt fel tudjuk majd használni be- és kikapcsoláshoz.

Ha a kikapcsolást tagadjuk ( $\uparrow \overline{Ki}$ ), a megépítésnél szabványos NC kikapcsoló nyomógombot használhatunk!



## Flip-flop

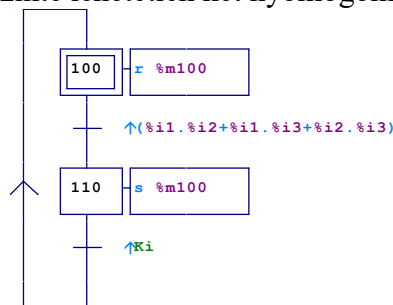
A lépcsőszám kezdődjön 100-nál és növekedjen tízesével! **%m100** belső memóriabitet kapcsoljuk be **%i13** bemeneti nyomógomb egyszeri lenyomásával, majd ki a még egyszeri lenyomásával! A nyomógombok prellezését szűrjük ki a  $\uparrow$  felfutó él operátorral! Használunk szimbólumtáblát!



## Logikai függvények használata

Az egyszerű öntartást változtassuk meg úgy, hogy **%i1**, **%i2**, **%i3** nyomógombokból bármely kettőt lenyomva kapcsoljon be **%m100**! Bekapcsolási feltételnek a megfelelő Boole-algebrai egyenlet kell, de ne feledkezzünk el a prellezés kiiktatásáról sem!

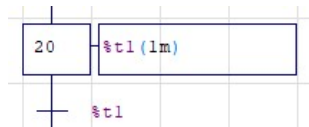
**Figyelem!** Ne egyenként használjunk felfutó él figyelést, hanem az egész feltételre egyszerre, ugyanis egyébként szinte lehetetlen két nyomógombot egy időpontban lenyomni!



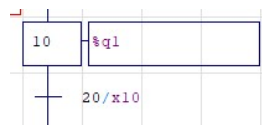
## Időkésleltetés

Kétféleképpen adhatjuk meg

1. Az akcióba beírjuk a késleltető változó elnevezését és utána zárójelben az értékét. Ezt követően a feltételben hivatkozunk a változóra.



2. A feltételben megadjuk a késleltetés értékét (20), majd egy perjel (`()`), továbbá melyik lépéshez képest (`x10`).

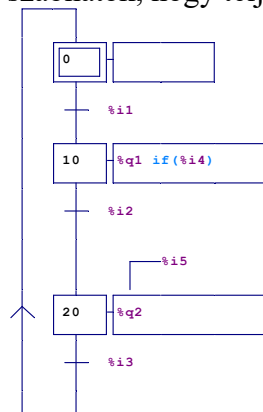


Ha nem írunk mértékegységet, **tizedmásodpercben** értendő a késleltetés, ha írunk, akkor abban, amit írunk. Az időtartam napokban, órákban, percekben, másodpercben, milliszekundumban kifejezhető a «d», «h», «m», «s» és «ms» mértékegységekkel.

Például: 1d30s = 1 nap és 30 másodperc.

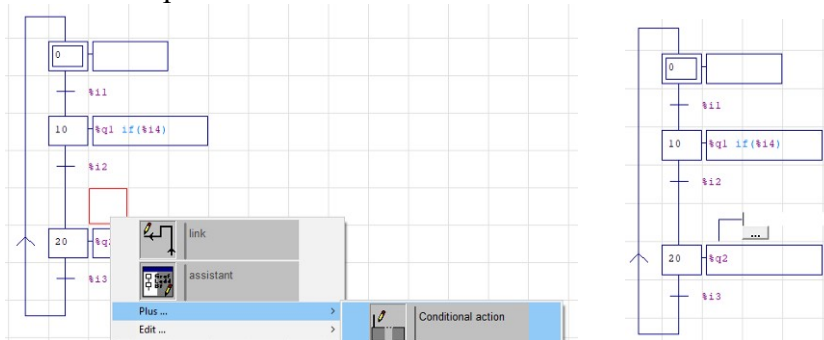
## Feltételes akció

Az akcióknak további feltételek szabhatók, hogy teljesüljenek.



A példában **%i1** feltétel teljesülésekor a program továbblép a 10 lépésre, de **%q1** akció csak akkor lesz aktív, ha **%i4** feltétel is teljesül. Hasonlóan **%i2** feltétel teljesülésekor a program továbblép a 20 lépésre, de **%q2** akció csak akkor lesz aktív, ha **%i5** feltétel is teljesül. A két megoldás egyenértékű, csupán az egyik karakteres, míg a másik grafikus megjelenésű.

A karakteres megoldás a szerkesztőben gépelhető be, míg a grafikus megoldásnál a kérdéses lépés sorát először egy osztással lejjebb kell tolni, majd az akciót jelölő négyszög fölött, ha piros a kijelölő, jobb egérgombbal kattintunk, és a „Plus... Conditional action” lehetőséget választjuk. A megjelenő „sarok vonalkára” kattintva megjelenik a három pont ikon, amire újból kattintva paraméterezhető a feltétel.



## Számláló (Counter)

A számlálót célszerű a kezdeti lépésnél nullázni: **r%c0** (ahol **r** a reset, **%c0** a 0 sorszámú számláló).

Felfelé számlálás (incremental) a **+%c0** utasítással történik. Annak a lépésnek az aktív állapotait számolja, ahová írtuk.

Lefelé számlálás (decremental) a **-%c0** utasítással történik. . Annak a lépésnek az aktív állapotait számolja, ahová írtuk.

A számláló lépés után elágazást kell tervezni, és a feltételekben el kell dönteni, hogy mi legyen, mielőtt a számláló eléri a célértéket, illetve miután elérte a célértéket.

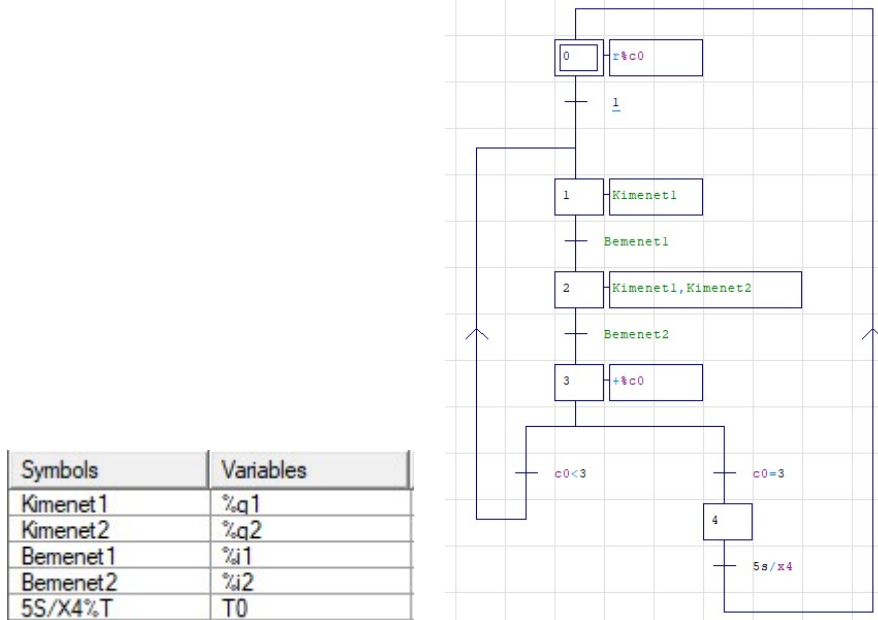
Felhasználási lehetőségek:

- Bizonyos ciklusszám után automatikus leállítás
- Bizonyos ciklusszám után más eseménysor futtatása

## Példa számlálóra

Az alábbi példában, amíg  $\%c0 < 3$ , a **Bemenet1** és a **Bemenet2** kapcsolgatására **Kimenet1** és **Kimenet2** kapcsolódik fel, és le.

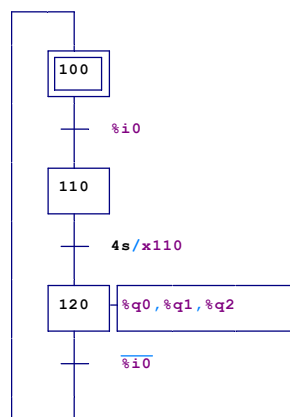
Ha  $\%c0 = 3$  értéket eléri, az elágazás másik oldala lesz aktív, 5 másodperces késleltetés következik, majd a program az elejére ugrik, és nullázza  $\%c0$ -t.



## Késve bekapcsoló

$\%i0$  reteszelt kapcsolót bekapcsolva 4 másodperc késleltetéssel kapcsoljon be egyszerre  $\%q1$ ,  $\%q2$ , és  $\%q3$  kimenet!  $\%i0$  kapcsolót lekapcsolva azonnal kapcsoljanak le a kimenetek! A szimbólumtábla a késleltetőről magától létrejön.

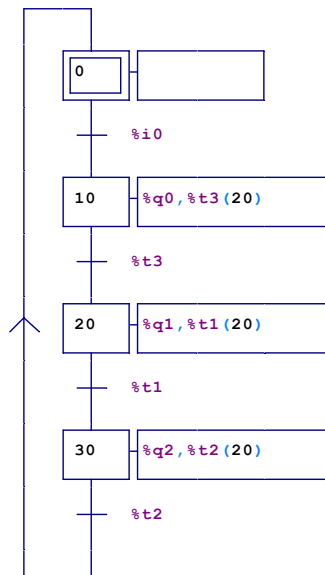
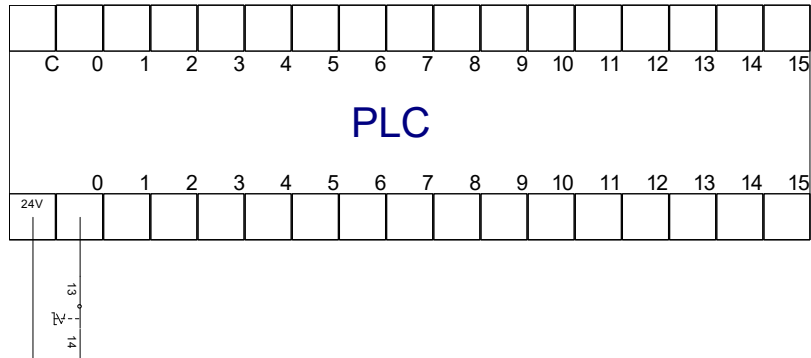
Symbols	Variables
4S/X110%T	T0



## Futófény

$\%i0$  reteszelt kapcsolót bekapcsolva fusson végig a fény többször, ciklusosan  $\%q0$  –  $\%q1$  –  $\%q2$  kimeneteken! A felvillanások 2 másodpercig tartsanak!

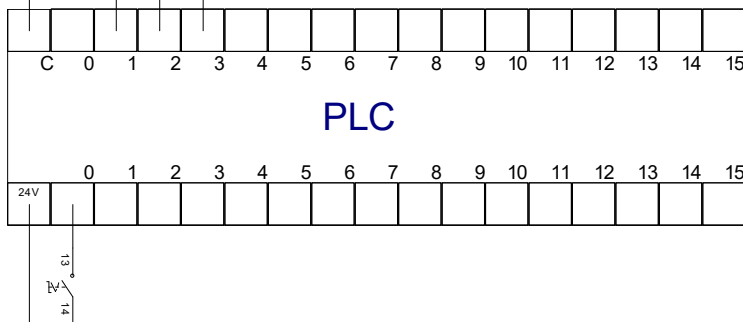
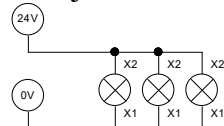
Most használjuk a másik típusú időzítőt!



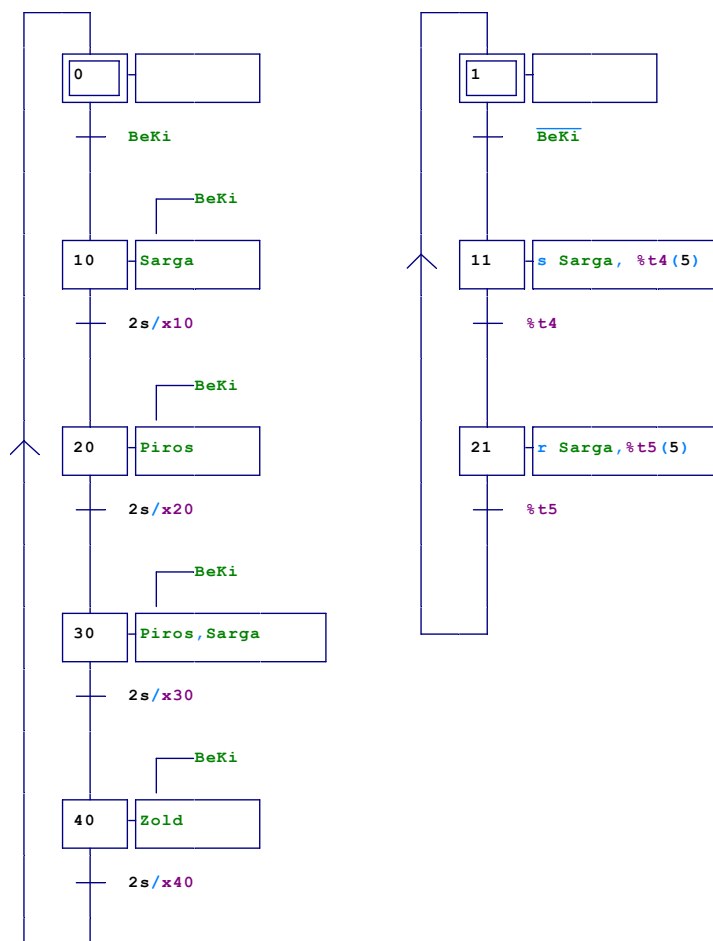


## Háromfényes közlekedési jelzőlámpa

%i0 BeKi reteszelt kapcsolót bekapcsolva fusson folyamatosan a sárga – piros – pirossárga – zöld ciklus, mindegyik esemény 2 másodpercig tartson! %i0 BeKi kapcsolót kikapcsolva a sárga lámpa villogjon (0,5 másodpercig világít, 0,5 másodpercig nem)! A feladathoz használjunk feltételes akciót!

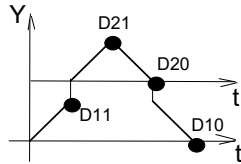


Symbols	Variables
BeKi	%i0
Sarga	%q2
Piros	%q1
Zold	%q3
2S/X40%T	T0
2S/X10%T	T1
2S/X20%T	T2
2S/X30%T	T3



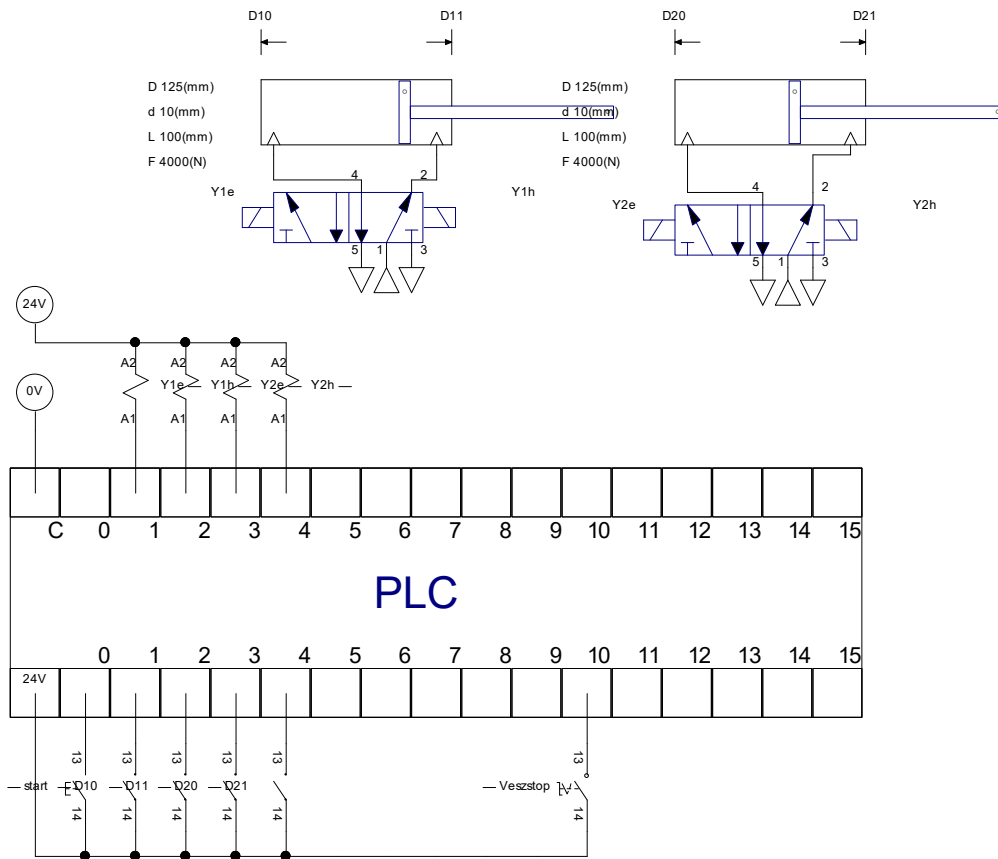
## Két munkahenger mozgása 3 ciklusig

Start nyomógomb lenyomására a két munkahenger végezzen mozgást az alábbi út-lépés diagram szerint, 3 ciklus után automatikusan álljon le! Használjunk 5/2-es bistabil szelepeket, és mechanikus végállás-kapcsolókat!



Vészstop reteszelt kapcsoló lenyomására annak a mozdulatnak a végén álljon le, ahol éppen tart, feloldására folytassa ciklusát! A feladathoz használjunk feltételes akciót!

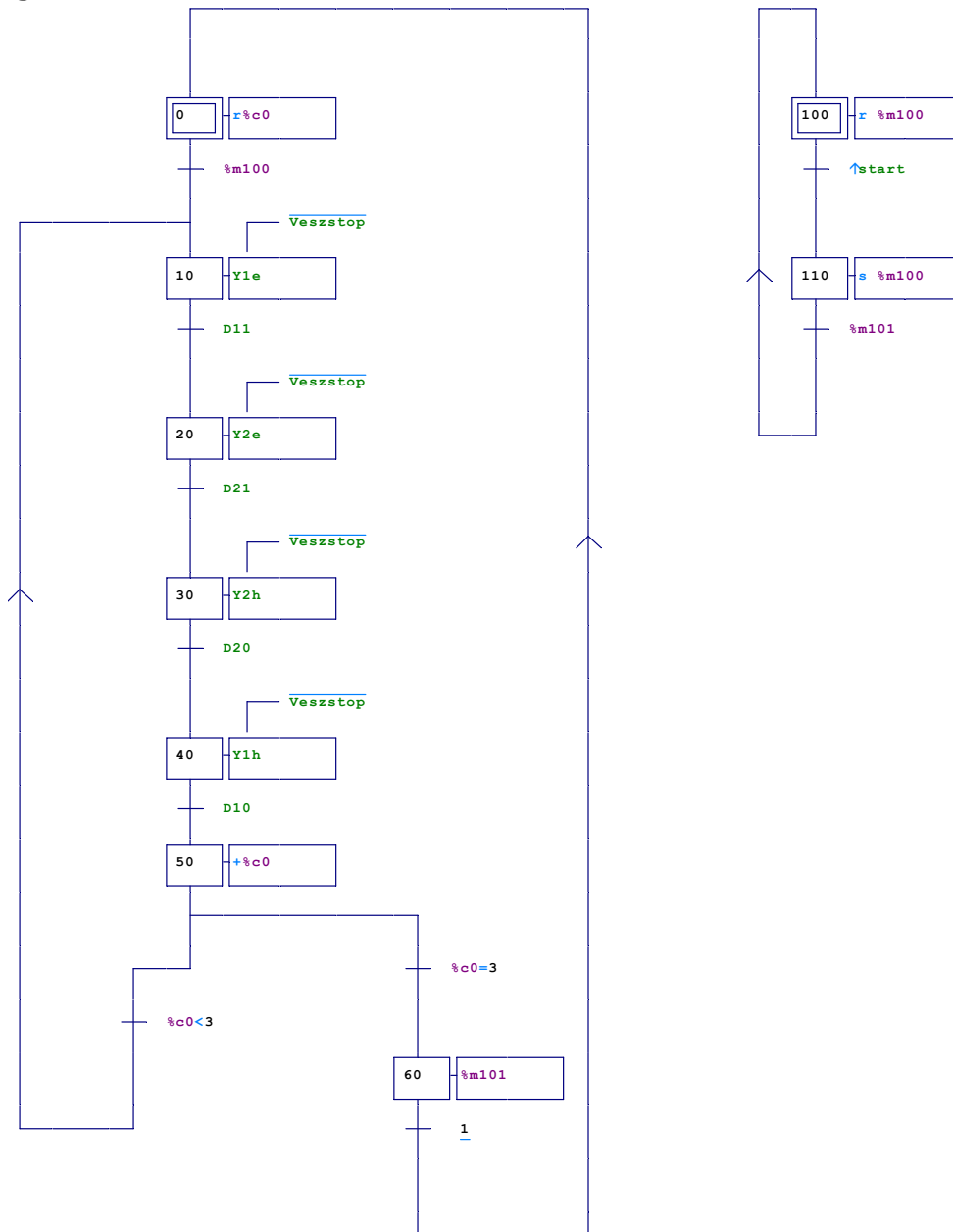
## A bekötés



## A szimbólumtábla

Symbols	Variables
D10	%i1
D11	%i2
D20	%i3
D21	%i4
Y1e	%q1
Y1h	%q2
Y2e	%q3
Y2h	%q4
start	%i0
Veszstop	%i10

## A program



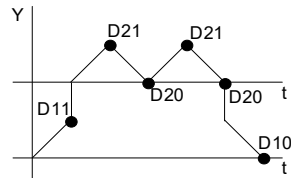
A program elején nullázom a számlálót `r%c0` paranccsal, majd figyelem, hogy van-e `%m100` jel, vagyis a `start` gombot lenyomtam-e, ugyanis ez bekapcsolja a `s%m100` memória területet.

Amennyiben nincs bekapcsolva a `Vészstop` kapcsoló, a mozgássor lefut, és a végén `+%c0` miatt a számláló egyet fölfelé számol.

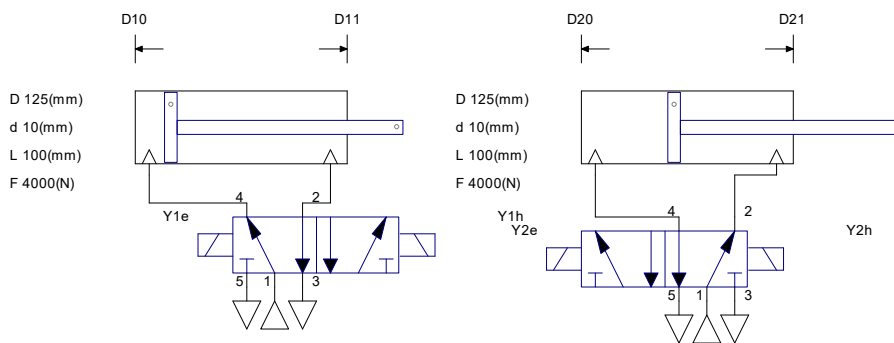
Ha `%c0<3`, a program az első mozdulat elé ugrik, és újból lefut a ciklus. Ha `%c0=3`, a program bekapcsolja `%m101` memória területet, ami viszont lekapcsolja `r%m100` memóriaterületet, továbbá a főprogram a legelejére ugrik, így a programfutás leáll.

## Eseményvezérelt lefutó vezérlésközpont ismétlődéssel

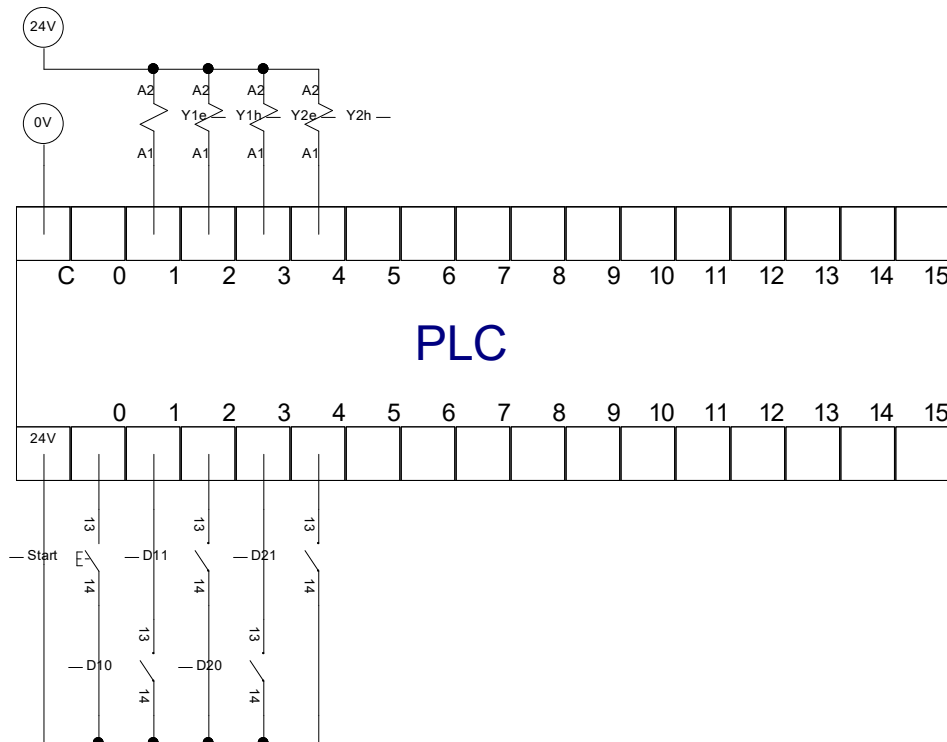
Két kettős működésű munkahenger végezzen mozgást az alábbi út-lépés diagram szerint! A munkahengereket egy-egy 5/2-es bistabil szelep vezérelje! Használjunk mechanikus végállás-érzékelőket az ábra szerinti jelöléssel! A „Start” gomb lenyomására egy ciklus fusson le!



## A munkahengerek és a szelepek bekötése



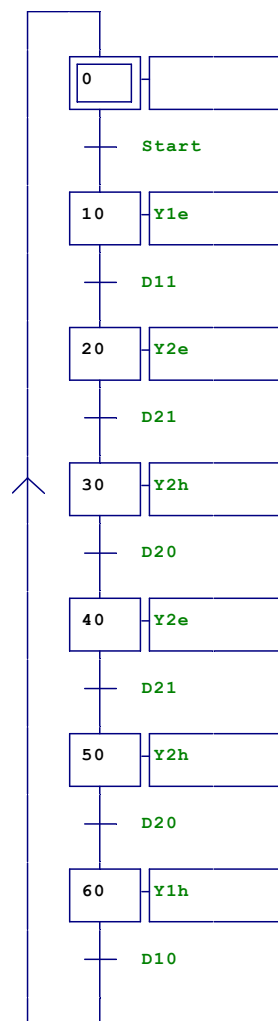
## A PLC elektromos bekötése



## A szimbólumtábla

Symbols	Variables
Start	%i0
D10	%i1
D11	%i2
D20	%i3
D21	%i4
Y1e	%q1
Y1h	%q2
Y2e	%q3
Y2h	%q4

## Az SFC program

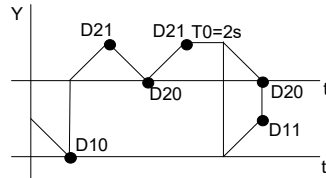


Mivel a mozgás hat lépésből áll, kezdőlépés plusz hat lépéses gráfot kell felvenni, a kezdő lépésszám legyen 0, a növekmény 10.

Az út-lépés diagram alapján csupán sorban fel kell venni az érzékelt eseményeket és az ezekre adott válasz akciókat. A kezdő lépésnél nem szükséges semmit definiálni, a mozgás a „Start” jelre indul. A cikluson belül tetszőleges ismétlődés lehet, erre nincsenek speciális szabályok, csak meg kell ismétetni az eseményeket és az akciókat az út-lépés diagram alapján.

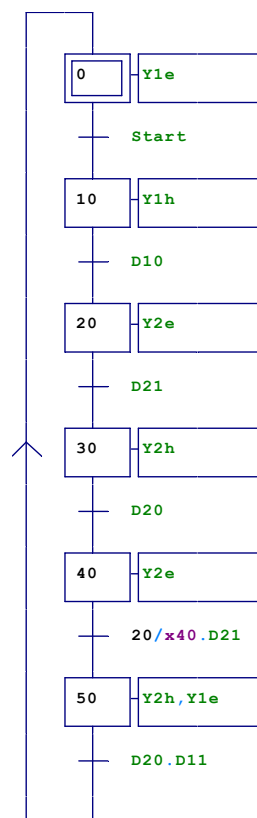
## Belső ismétlődés, inverz mozgás, belső várakozás, egyidejűség

Változtassuk meg a programunkat, hogy az alábbi út-lépés diagram szerint mozogjon! T0 várakozás legyen 2 másodperc!



A pneumatikus bekötés, és a PLC bekötés nem változik, a szimbólumtábla bővül T0 értékével (20/X40%T), de ez automatikusan létrejön az időzítő parancs kiadásakor.

20/X40%T	T0
----------	----



Azt, hogy az első munkahenger alaphelyzete kitolt (inverz mozgás) úgy programozzuk, hogy a kezdőlépésben ezt beállítjuk. Mivel van egy egyidejűség, a lépések száma hatról ötre csökken.

Figyeljünk a végállás-kapcsolók és a mágnes-tekercek sorrendjére, haladjunk az út-lépés diagram szerint! A két másodperces késleltetést a negyedik lépés után a 20/x40 parancssal állítjuk be, amit és (•) kapcsolatba hozunk a D21 végállás-kapcsolóval, hiszen mindegyiknek teljesülnie kell a továbblépéshez.

Az ötödik lépésben két akció is van, amiket vesszővel (,) választunk el.

Az utolsó feltétel a D20 és a D11 végállás-érzékelők és (•) kapcsolata, hiszen mindegyiknek teljesülnie kell a továbblépéshez, vagyis a ciklus elejére ugráshoz.

## Összetett feladat

Az első munkahenger fogja be a munkadarabot egy satuba! Ezt követően a második munkahenger gyorsjárattal közelítse meg a munkadarabot, majd a löket vége előtt 20 mm-el lassítson le, hogy a befogott fúró kifúrassa azt, majd gyorsjárattal térjen vissza! Ezt követően a harmas munkahenger egy bélyegzőt szorítson a munkadarabhoz 2 másodpercig, majd térjen vissza! Ezután az első munkahenger térjen vissza, vagyis engedje el a munkadarabot a satuban! Végül a negyedik munkahenger egy tárolóból toljon új munkadarabot a satuhoz, miközben az elkészültet ezzel lökje ki egy másik tárolóba!

A kettős működésű munkahengereket 5/2-es monostabil szelepek vezéreljék, a végállás-kapcsolók passzív mechanikusak legyenek! A lassítást 5/2-es bistabil szelep, és reed-relék vezéreljék!

A program tudjon lépés-üzemmódot, egy-ciklus üzemmódot és folyamatos üzemmódot (**Lépés** nyomógomb, **Folyamatos üzem be** nyomógomb, **Folyamatos üzem ki** nyomógomb)! A program indításakor **Folyamatos üzem ki** üzemmódban van, a **Start** gomb és a **Start egy-ciklus** gomb nem aktív, ilyenkor a **Lépés** gombot lenyomva a mozgás mindig csak egy lépést tesz meg.

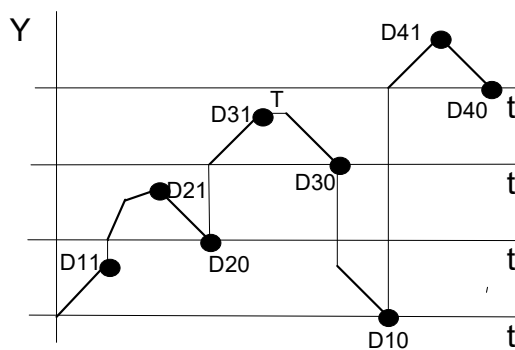
**Folyamatos üzem be** állapotban a **Start egy-ciklus** gombot lenyomva egy ciklus fut le, majd a mozgás leáll!

**Folyamatos üzem be** állapotban a **Start** nyomógombra elindul a folyamatos ciklus-ismétlődés, **Stop** nyomógombra az éppen futó ciklus végén leáll!

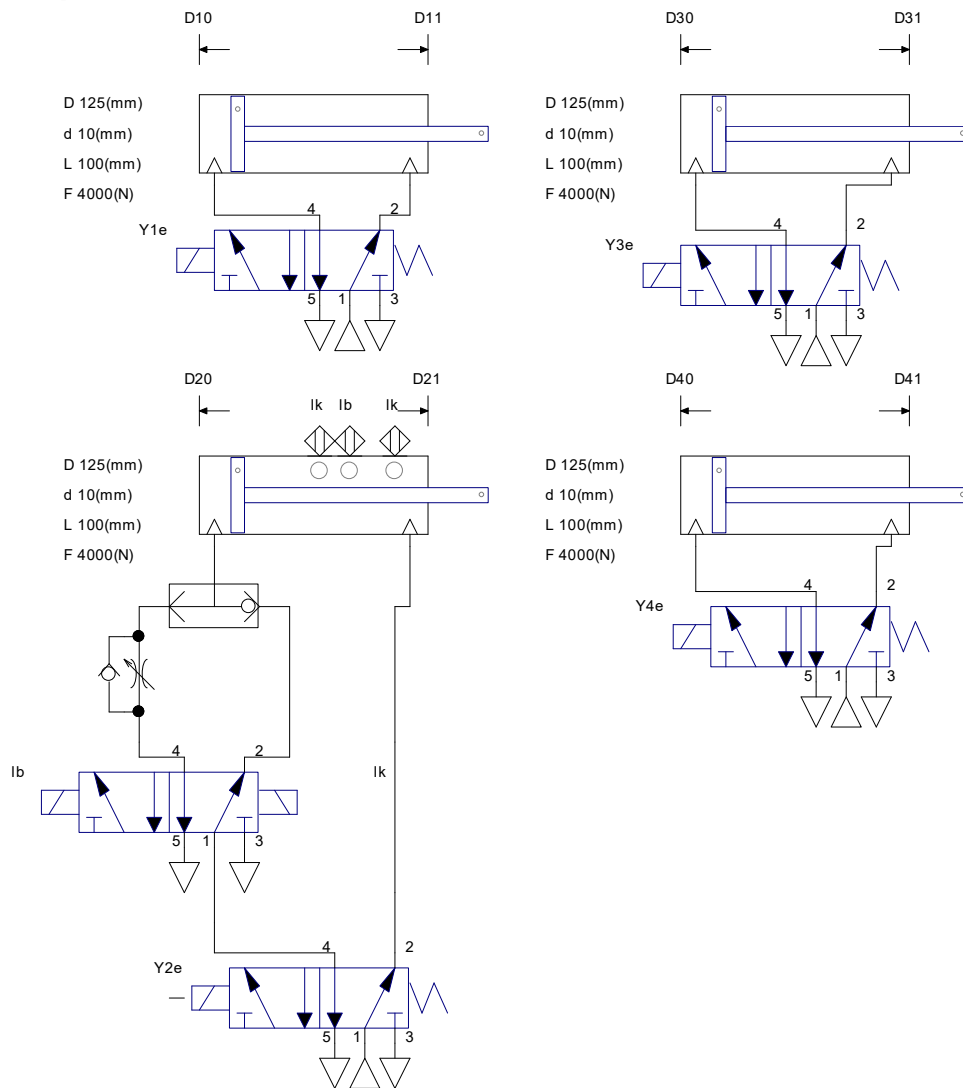
A ciklus bármely pontján az **Alaphelyzet** nyomógombot lenyomva minden munkahenger térjen alaphelyzetbe, ezután a folyamat **Start** jelre indulhat újra!

A **Vészstop** gombot lenyomva a ciklus álljon meg az éppen végzett mozdulat befejezése után, ezután **Start** jelre folytathatja mozgását!

## A mozgásdiagram

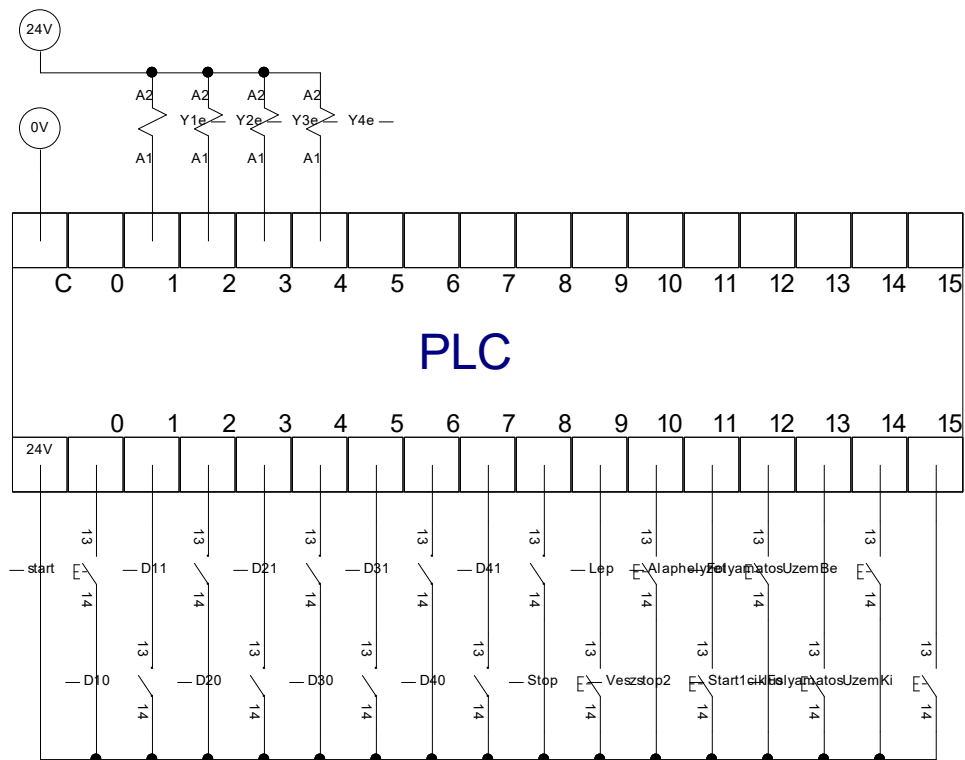


## Az elektro-pneumatikus elemek bekötése





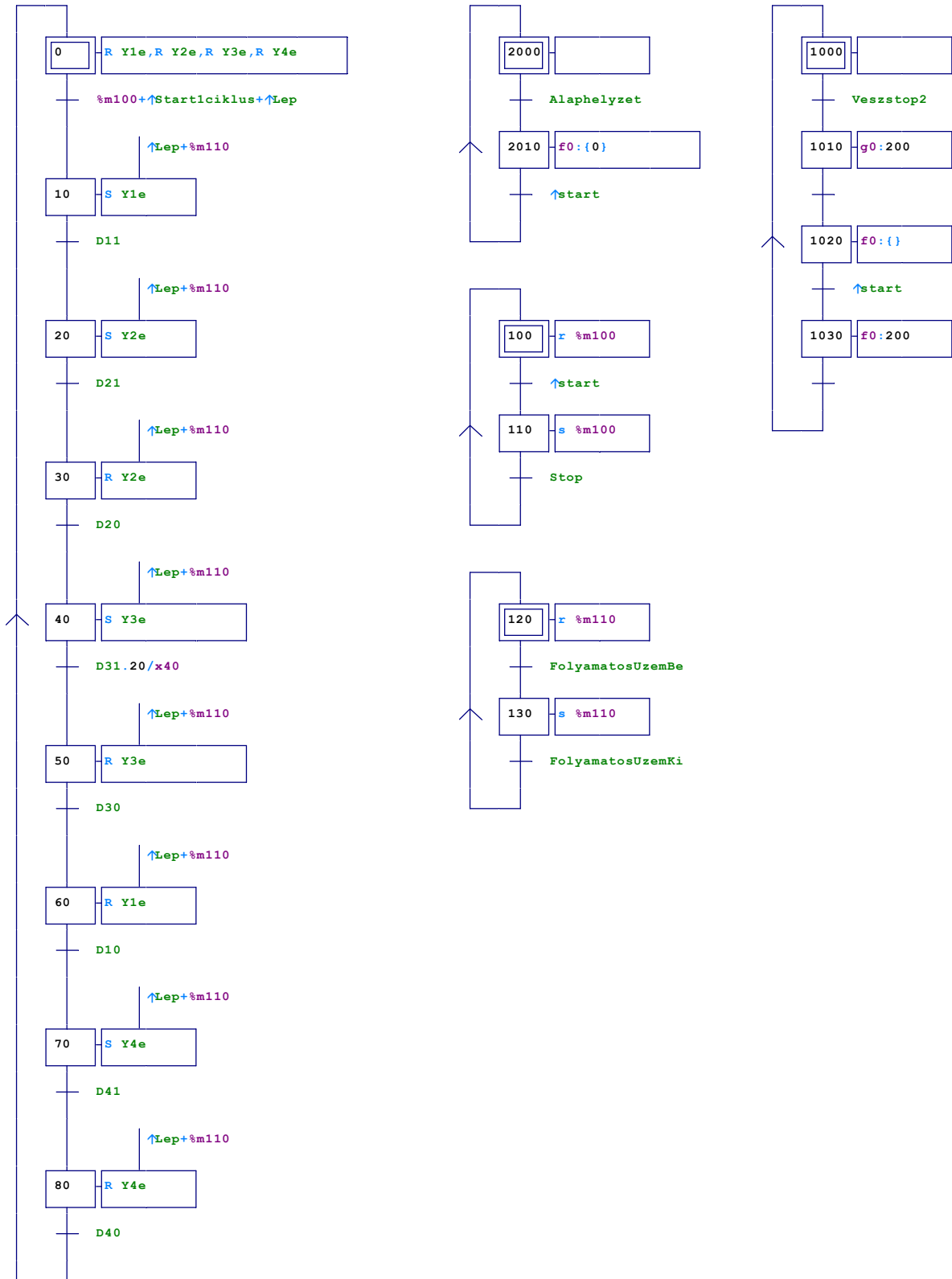
## A PLC elektromos bekötése



## A szimbólumtáblázat

Symbols	Variables
D10	%i1
D11	%i2
D20	%i3
D21	%i4
D30	%i5
D31	%i6
D40	%i7
D41	%i8
Y1e	%q1
Y2e	%q2
Y3e	%q3
Y4e	%q4
start	%i0
stop	%i9
Veszstop2	%i11
Alaphelyzet	%i12
Start1ciklus	%i13
FolyamatosUzemBe	%i14
FolyamatosUzemKi	%i15
Lep	%i10
20/X40%T	T0

# A Grafcet program



## A főprogram (g0)

Az asszisztens segítségével, mivel a lefutó mozgássor nyolc lépésből áll, válasszunk kezdeti lépés, plusz nyolc lépést, a kezdő lépés legyen 0, a lépésköz legyen 10!

A kezdeti lépésnél, mivel monostabil szelepeket választottunk, hozzuk őket alaphelyzetbe a Reset (R) paranccsal, egymástól vesszővel (,) elválasztva!

A kezdeti beállítások után az indítási feltételeket kell megadnunk. Először csak egy egyszerű nyomógombos indítást válasszunk „↑Start1ciklus”, majd később ezt kibővítjük a különféle üzemmódokkal!

Az út-lépés diagramnak megfelelően szépen sorban vegyük fel a mozgás-parancsokat, majd a végállás-kapcsoló feltételeket! Set (S) és Reset (R) utasításokat kell használnunk, mert a szelepek monostabilak!

A „D40” utolsó feltételt a pneumatikus megoldásoktól eltérően nem a kezdetre kell felvenni, hanem a végére.

A programban egy helyen, a 40-es lépés után szerepel 2 másodperces időkésleltetés, ezt a 20/x40 paranccsal programozzuk, és a D31 végállás-kapcsoló érzékelésével és (•) logikai kapcsolatba hozzuk.

Ha már működik az alapmozgásunk, csak utána tegyük fel a feltételeket a lépésekhez!

## Folyamatos üzemmód / lépés üzemmód

Válasszuk a 120-at kezdeti lépésnek, mert az még nem szerepelt sehol. Az ehhez kapcsolódó akcióban kikapcsoljuk (reset, r) a %m110 memória területet. Ez a memória terület fogja eldönteni, hogy folyamatos, vagy lépésenkénti üzemmódot választunk. Ha lenyomjuk a „FolyamatosUzemBe” nyomógombot, a 130-as lépés előtt, a %m110 memória területet bekapcsoljuk (set, s). A „FolyamatosUzemKi” nyomógommbal újra alaphelyzetbe hozzuk a %m110 memória területet, vagyis kikapcsoljuk.

A fő gráfon az alaphelyzet kivételével mindegyik akciót feltételes akcióvá tesszük, amelyeknél a továbblépés feltétele vagy (+) a folyamatos üzem bekapcsolása (%m110), vagy a „↑Lep” nyomógomb felfutó él figyelés lenyomása. Azért kell felfutó él figyelés, hogy ne lehessen lesúlyozva, leragasztva a nyomógombot folyamatosra tenni az üzemmódot. Ne felejtjük el az indítási feltételnél is megadni a „+↑Lep” lehetőséget! Ennek hatására, ha a „FolyamatosUzemKi” üzemmódot választjuk, csak a „+↑Lep” nyomógomb nyomogatásával lép egyet-egyed előre a mozgássor. Ha a „FolyamatosUzemBe” lehetőséget választjuk, a %m110 memória területet aktív lesz, és engedő végigfutni a ciklust.

## Egy ciklus üzemmód / több ciklus üzemmód

Az egy ciklus üzemmód elindításához elegendő az első feltételnél a „↑Start1ciklus” nyomógomb figyelése, persze csak ha folyamatos üzemben vagyunk. A nyomógomb vagylagos kapcsolatban van (+) a másik két féle indítási móddal. Azért kell felfutó él figyelés, hogy ne lehessen lesúlyozva, leragasztva a nyomógombot többciklusossá tenni az üzemmódot.

A többciklusos üzemmódhoz kell egy öntartás, ahol a %m100 memória területet a „↑start” nyomógommbal bekapcsoljuk, míg a „Stop” nyomógommbal kikapcsoljuk. Ez a %m100 memória területet is vagylagosan (+) elindíthatja a fő gráfot. Válasszuk a 100-at kezdeti lépésnek, mert az még nem szerepelt sehol!

## Rögzítő vészstop

A vészstop jelentse ebben az esetben azt, hogy a mozgássor még befejezi a „Vészstop2” nyomógomb lenyomásakor aktív lépést, majd rögzül a folyamat, ahol éppen tart.

Kezdő lépésnek célszerű egy olyan értéket megadni, ami biztosan nincs a mozgássorban, például 1000. Ezután következik a „Vészstop2” nyomógomb lenyomásának figyelése, ha ez megtörténik, a következő akció a programfutás aktuális értékeinek letárolása a „g0:200” paranccsal. Itt a „g” a letárolást jelenti, a „0” azt, hogy milyen kezdő lépésszámú gráfot (példánkban a 0-val kezdődőt), 200 pedig hogy hová történjen a mentés. Itt olyan értéket kell adni, amit a többi gráf nem használ (0-80 nem jó, 100-130 nem jó, 1000-1030 nem jó, 2000-2010 nem jó).

A következő lépésben az `f0: {}` paranccsal, a 0-val kezdődő gráfban minden lépést inaktívvá teszünk.

Ezután következik a „↑start” nyomógomb lenyomásának figyelése (↑felfutó él figyeléssel), ha ez megtörténik, a következő akció a programfutás „g0:200” –ba lementett értékeinek visszaolvasása a „f0:200” paranccsal. Ennek hatására a program onnan folytatódik, ahol abbamaradt.

## Alaphelyzetbe hozó vészstop

Ez a vészstop azon esete, amikor az „Alaphelyzet” nyomógomb lenyomására minden mozgás megszakad, és alaphelyzetbe kerülnek az elemek.

Kezdő lépésnek célszerű egy olyan értéket megadni, ami biztosan nincs a mozgássorban, például 2000. Ezután következik a „Alaphelyzet” nyomógomb lenyomásának figyelése, ha ez megtörténik, a következő lépésben az `f0: {0}` paranccsal, a 0-val kezdődő gráfban minden lépést inaktívvá teszünk, kivéve a 0-dik lépést, vagyis az alaphelyzetet. Ennek hatására minden alaphelyzetbe áll.

Ezután következik a „↑start” nyomógomb lenyomásának figyelése (↑felfutó él figyeléssel), ha ez megtörténik, a programfutás újból előről, elindul.