

# **Mechatronikai rendszerek**

Oktatási segédlet a PLC programozásához



**Készítette:**

Cservenák Ákos

PhD hallgató

2017



AZ EMBERI ERŐFORRÁSOK MINISZTERIUMA ÚNKP-17-3 KÓDSZÁMÚ ÚJ NEMZETI KIVÁLÓSÁG

PROGRAMJÁNAK TÁMOGATÁSÁVAL KÉSZÜLT”

## **Tartalomjegyzék**

<b>1. PLC definíciója .....</b>	<b>2</b>
<b>2. PLC hardverkörnyezet .....</b>	<b>2</b>
<b>3. Projekt létrehozása, felépítése .....</b>	<b>5</b>
<b>4. Programnyelvek ismertetése .....</b>	<b>12</b>
<b>5. Bemenet-másoló PLC program létrehozása LD programnyelven .....</b>	<b>16</b>
<b>6. PLC feladatok.....</b>	<b>23</b>
<b>6.1 Feladatok megoldása LD programnyelven .....</b>	<b>23</b>
<b>6.1.1 A 9. elektropneumatikai feladat vezérlése PLC-vel .....</b>	<b>23</b>
<b>6.1.2 A 10. elektropneumatikai feladat vezérlése PLC-vel .....</b>	<b>25</b>
<b>6.1.3 A 16. elektropneumatikai feladat vezérlése PLC-vel .....</b>	<b>26</b>
<b>6.1.4 Időzítő alkalmazása LD programnyelven .....</b>	<b>28</b>
<b>6.1.5 Számláló alkalmazása LD programnyelven .....</b>	<b>29</b>
<b>6.1.6 Felfutó él, lefutó él alkalmazása LD programnyelven .....</b>	<b>30</b>
<b>6.2 Feladat megoldás ST programnyelven .....</b>	<b>31</b>
<b>6.3 Feladatok megoldása SFC programnyelven .....</b>	<b>32</b>
<b>6.3.1 Munkahenger ki- és visszafutásának vezérlése SFC programnyelven.....</b>	<b>32</b>
<b>6.3.1 Három munkahenger vezérlése SFC programnyelven.....</b>	<b>33</b>
<b>6.3.2 A 16. szimulációs tábla megoldása SFC programnyelven .....</b>	<b>34</b>
<b>6.3.3 A 17. szimulációs tábla megoldása SFC programnyelven .....</b>	<b>36</b>
<b>6.3.4 A 18. szimulációs tábla megoldása SFC programnyelven .....</b>	<b>38</b>
<b>6.4 Feladat megoldás CFC programnyelven.....</b>	<b>40</b>
<b>6.5 Feladat megoldás FBD programnyelven .....</b>	<b>41</b>
<b>7. Vizualizáció .....</b>	<b>44</b>
<b>Irodalomjegyzék .....</b>	<b>51</b>

## **Hetekre tervezett PLC feladatok**

### **1. hét:**

PLC hardver, programozási nyelvek ismertetése, LD másoló program

### **2. hét:**

Elektropneumatika 9. 10. és 16. feladatának megoldása LD programnyelven

### **3. hét:**

LD számláló, ST nyelv ismertetése, 9. feladat, SFC nyelv ismertetése, egyszerű program

### **4. hét:**

SFC nyelv gyakorlás, 16-17. gyakorlótábla megoldása

### **5. hét:**

18. gyakorlótábla megoldása, CFC nyelv ismertetése, feladat megoldás

### **6. hét:**

FBD nyelv ismertetése, feladat megoldás

### **7. hét:**

Vizualizáció

## 1. PLC definíciója

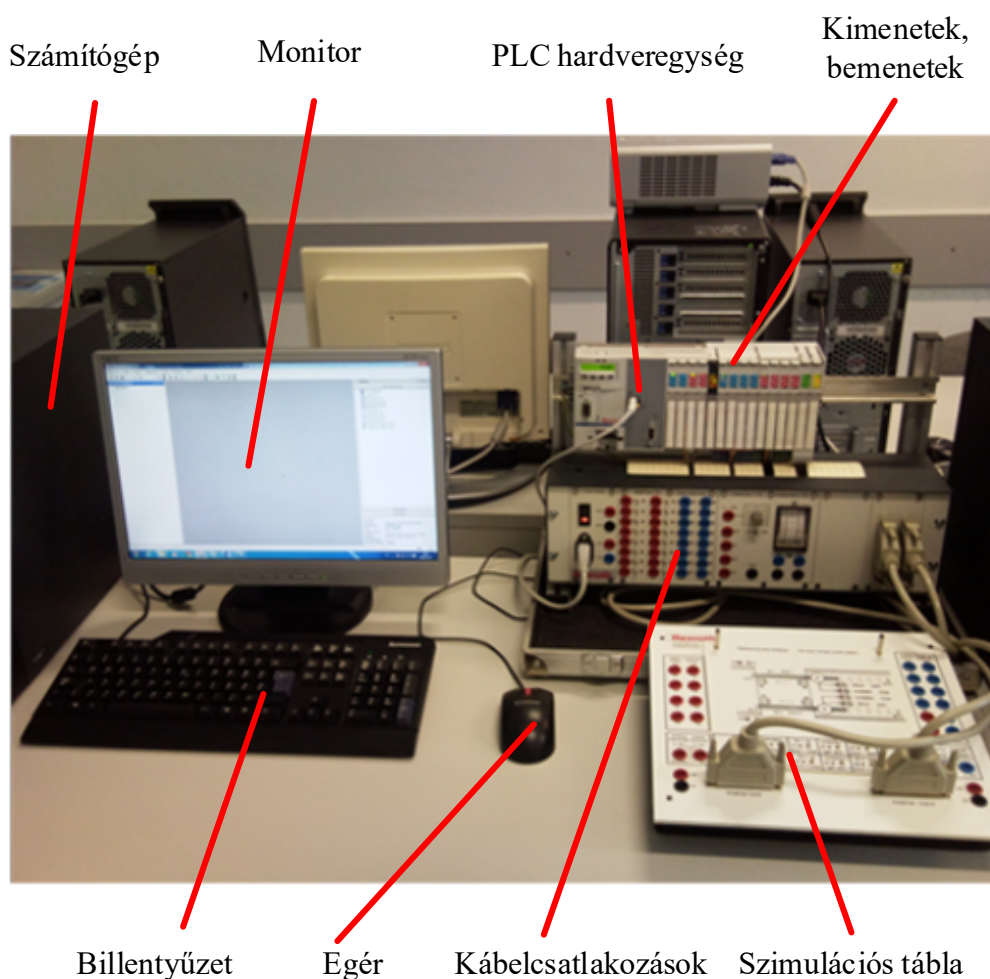
PLC = Programmable Logic Controller, jelentése magyarul programozható logikai vezérlő [1]. Nem összekeverendő a PIC vagy a mikrovezérlővel.

A PLC egy hardveregység, amely rendelkezik egy mikroprocesszorral, be- és kimenetekkel, illetve kommunikációs porttal. Leginkább az iparban használatos, különböző vezérlési, automatizálási feladatokra, illetve oktatás során is hasznosítják.

## 2. PLC hardverkörnyezet

A Tanszékünk Bosch Rexroth IndraControl L40 PLC-eket használ.

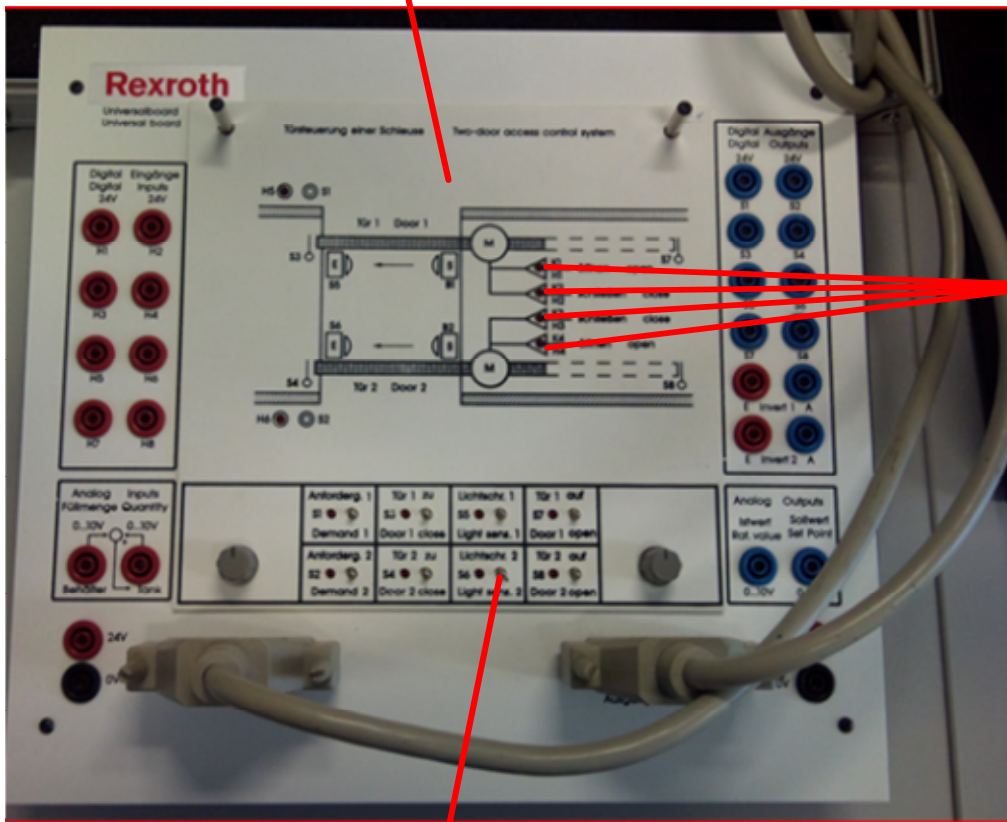
A PLC munkahely környezetének felépítését az 1. ábra szemlélteti [2].



1. ábra: PLC programozási munkahely

A monitoron a projekt-létrehozó, illetve programozó szoftver (IndraWorks és IndraLogic), a szimulációs táblán az aktuális feladat ábrája, a PLC hardveregységen pedig a PLC egységeinek működése látható. A szimulációs tábla az automatizálandó üzemet modellezi kapcsolókkal és LED-ek segítségével.

Aktuális feladat



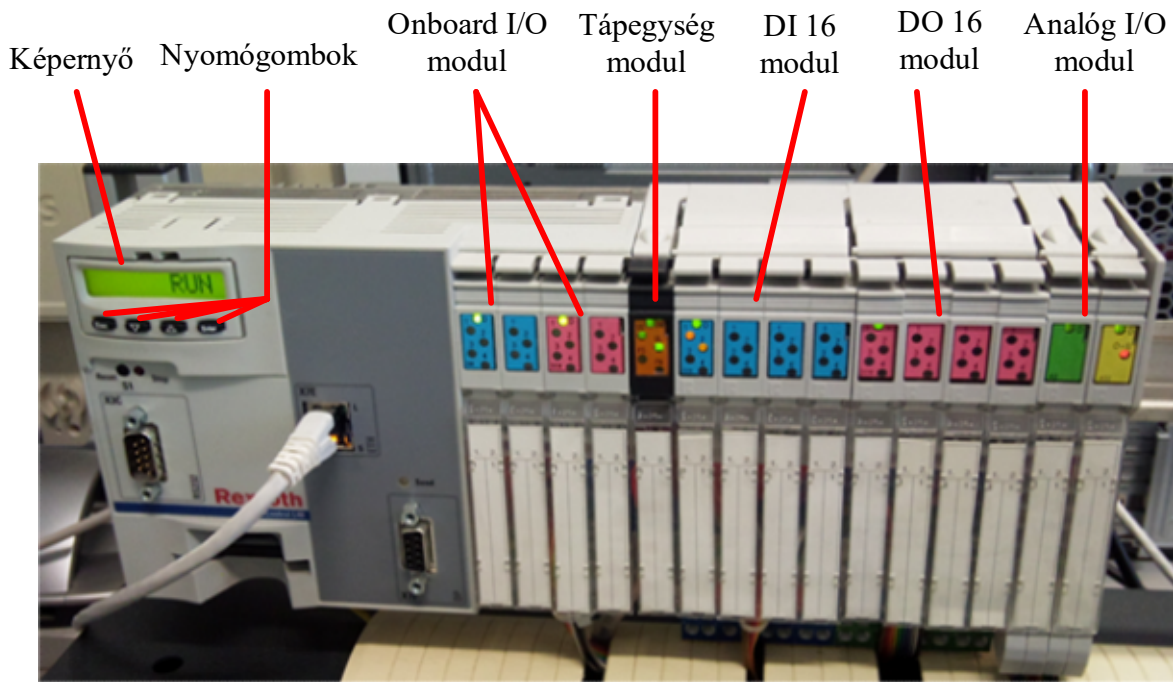
LED-ek/  
Kimenetek

Kapcsolók/Bemenetek

2. ábra: Szimulációs tábla

A szimulációs táblát a 2. ábra mutatja, ahol a következő elemek találhatóak meg [2]:

- Kapcsolók: ezekkel lehet vezérelni a folyamatot, ezek az Onboard I/O modulnak a bemenetei,
- LED-ek: ezek segítségével lehet látni a folyamatot, ezek az Onboard I/O modulnak a kimenetei,
- Aktuális feladat: cserélhető lapok, a PLC programozás egyes nyelveinek gyakorlásához különböző feladatok helyezhetők fel,
- Kommunikációs kábelek: ezeken keresztül kapcsolódik a PLC hardveregységhez,
- Bal és jobb oldalt további ki- és bemenetek vannak.



3. ábra PLC hardveregység

A PLC hardveregység felépítését a 3. ábra részletezi [2].

A következő elemek találhatók meg a hardveregységen:

- Képernyő: üzemállapot, IP-cím, Firmware verzió megfigyelésére alkalmas
- Nyomógombok: ezzel lehet kezelni a menüt
- Onboard I/O modul: gyárilag adott 8-8 ki- és bemenet
- Tápegység modul: a PLC tápellátásáért felelős
- DI 16 modul: kiegészítő 16 digitális bemenet
- DO 16 modul: kiegészítő 16 digitális kimenet
- Analóg I/O modul: kiegészítő analóg ki- és bemenetek

A projekt létrehozása az IndraWorks Engineering szoftver alatt tehető meg, a programozás pedig az IndraLogic szoftverben lehetséges.

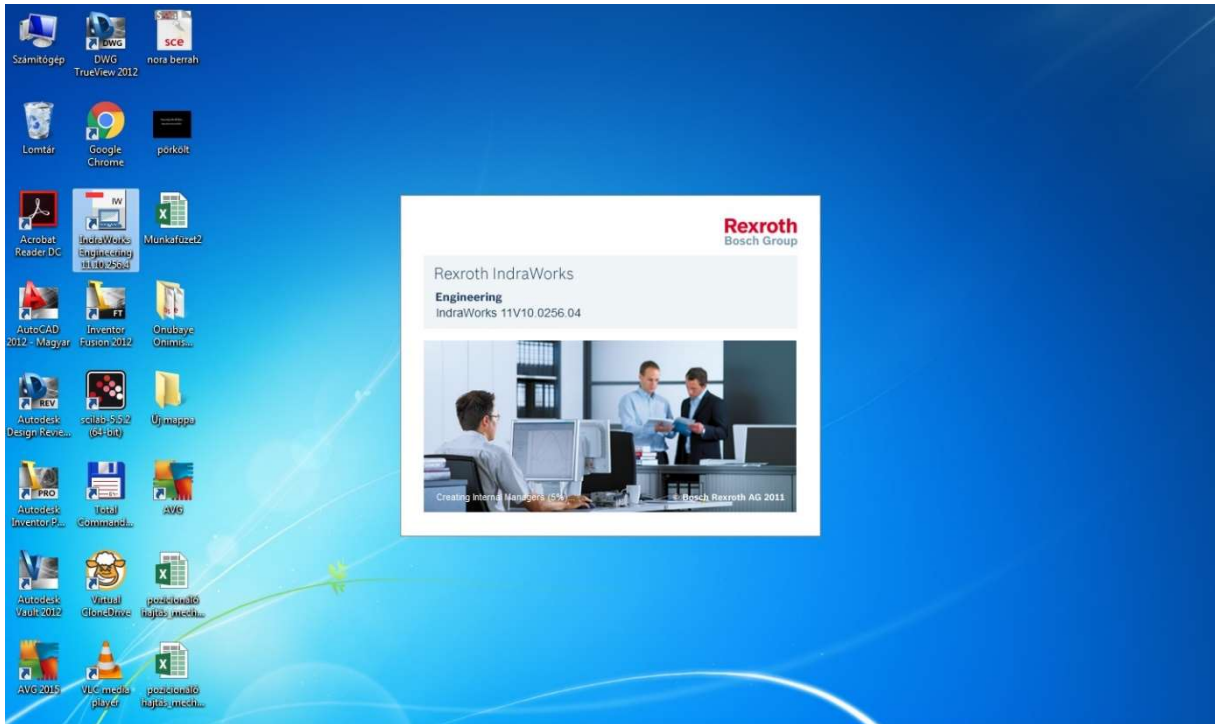
Az adott PLC-n futó IndraLogic szoftverben a következő PLC nyelvek érhetők el a [4] alapján:

- IL (Instruction List): Utasításlistás programnyelv
- ST (Structured Text): Strukturált szövegű programnyelv
- SFC (Sequential Function Chart): Sorrendvezérlési diagram programnyelv
- FBD (Function Block Diagram): Funkcióblokk-diagram programnyelv
- CFC (Continuous Function Chart): Grafikus funkcióblokkos programnyelv
- LD (Ladder Diagram): Létradiagram programnyelv

### 3. Projekt létrehozása, felépítése

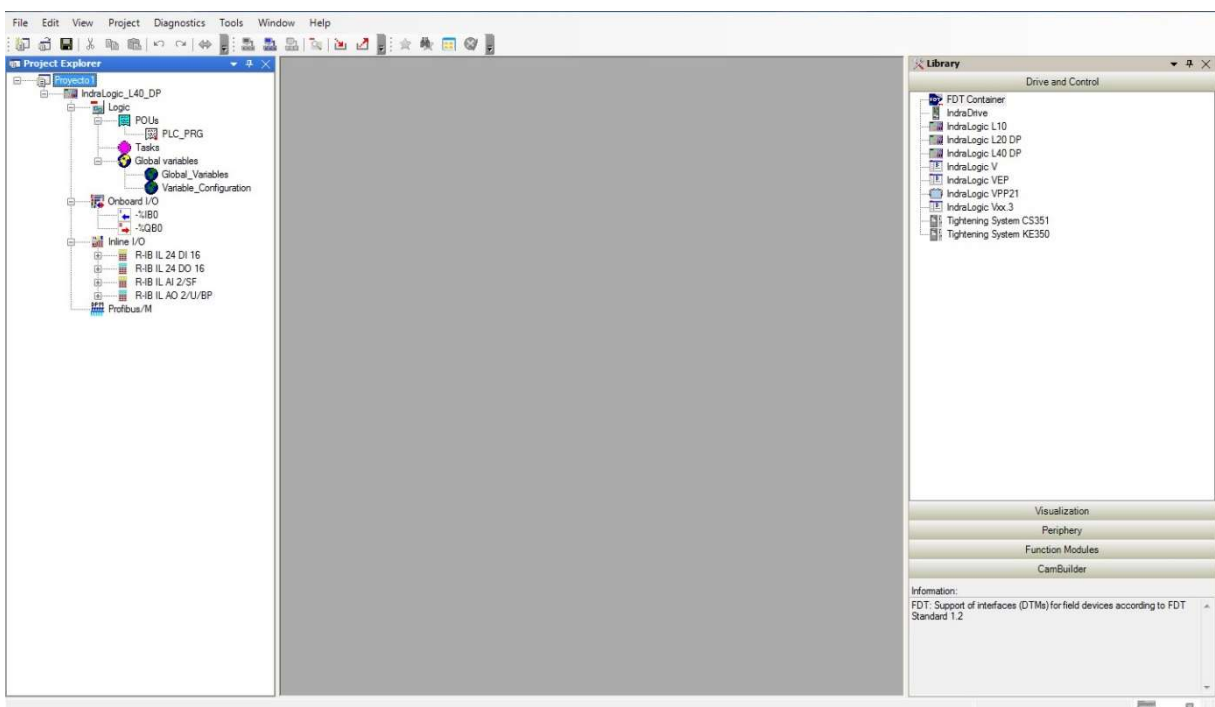
A fejezet leírása a [3] alapján készült el.

A projekt felépítéséhez legelőször a Rexroth IndraWorks szoftvert kell elindítani. Indítás után a 4. ábrán látható ablak fogad.



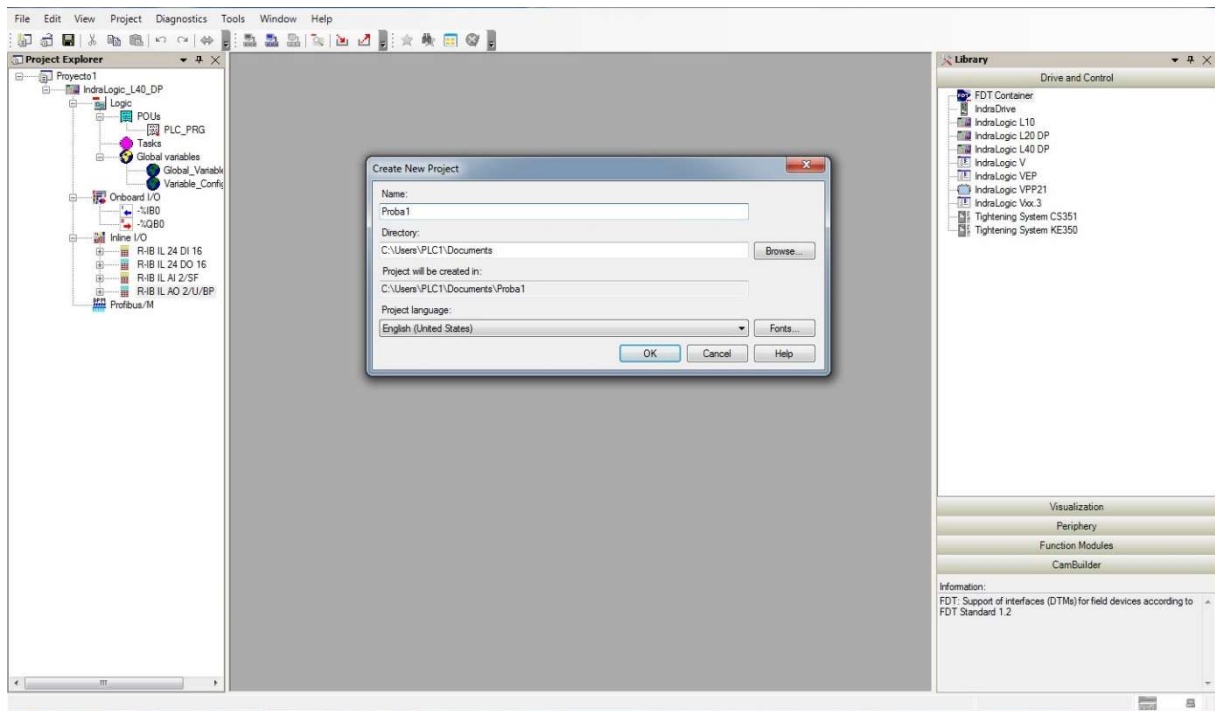
4. ábra: Rexroth IndraWorks szoftver elindítása után

A szoftver betöltése után az 5. ábra szerinti felhasználói ablak jelenik meg.

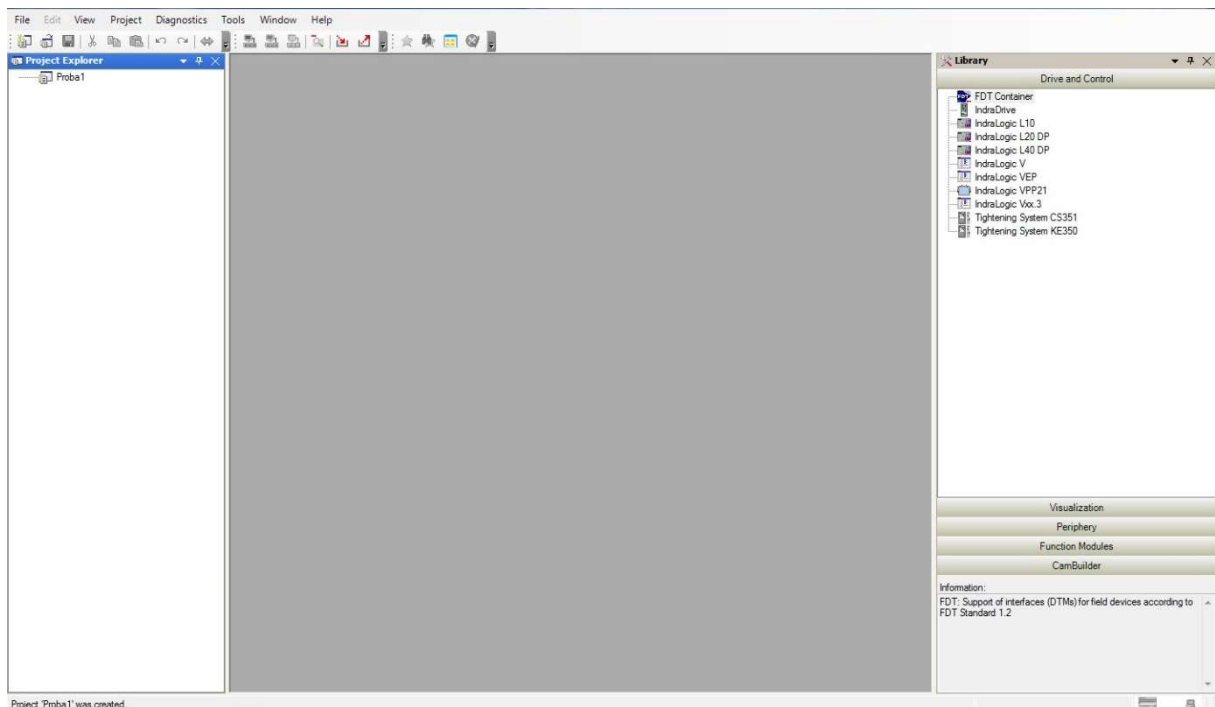


5. ábra: Rexroth IndraWorks szoftver betöltése után

Új projektet létrehozni a „File / New / Project” menüpont segítségével lehet. Ekkor a 6. ábrán látható ablak jelenik meg. A megfelelő név, mentési hely, és nyelv beállítása után a 7. ábrán megfigyelhető ablak látható.



6. ábra: Új projekt létrehozása

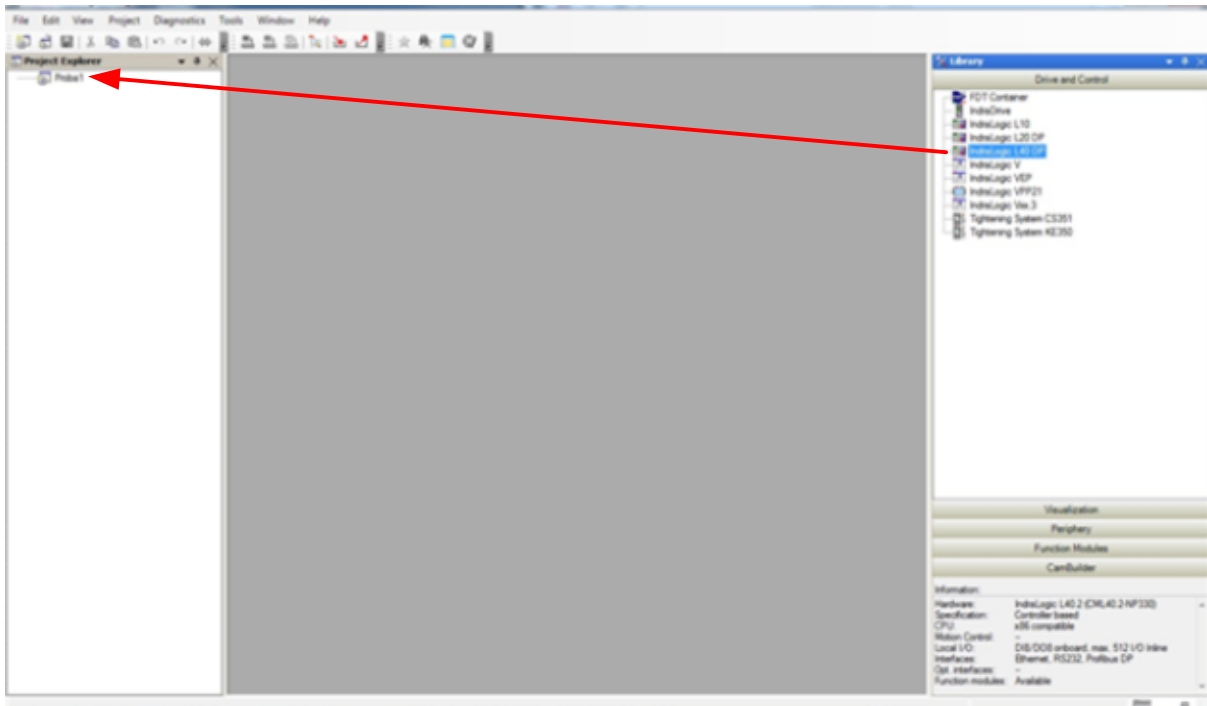


7. ábra: Új projekt létrehozása után

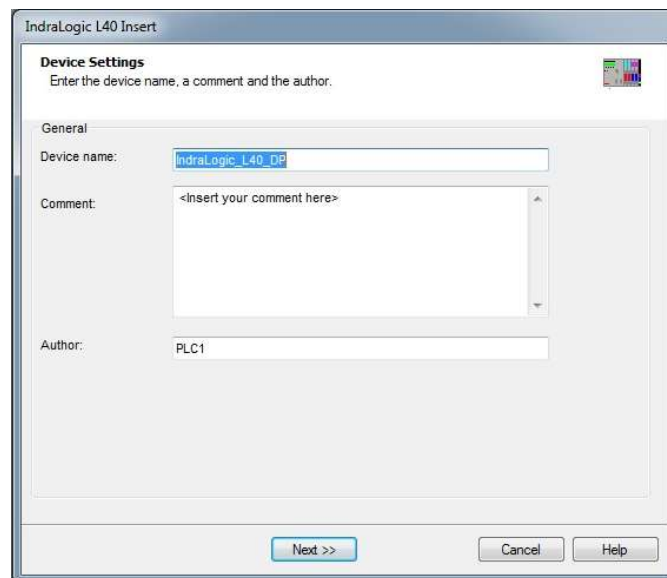
Ezután a PLC hardveregység egyes elemeit kell definiálni a projekthez. Elsőként a PLC fő egységet kell hozzárendelni, majd az egyes modulokat.



Az Indralogic L40 egységet a 8. ábrán látható ablak jobb oldalán található listából lehet kiválasztani, és a bal egérgombot nyomva tartva lehet behúzni a bal oldalra. A behúzás után a 9. ábrán látható ablak fogad.



8. ábra: Indralogic L40 hozzáadása a projekthez



9. ábra: Indralogic L40 egység behúzása után feljövő ablak

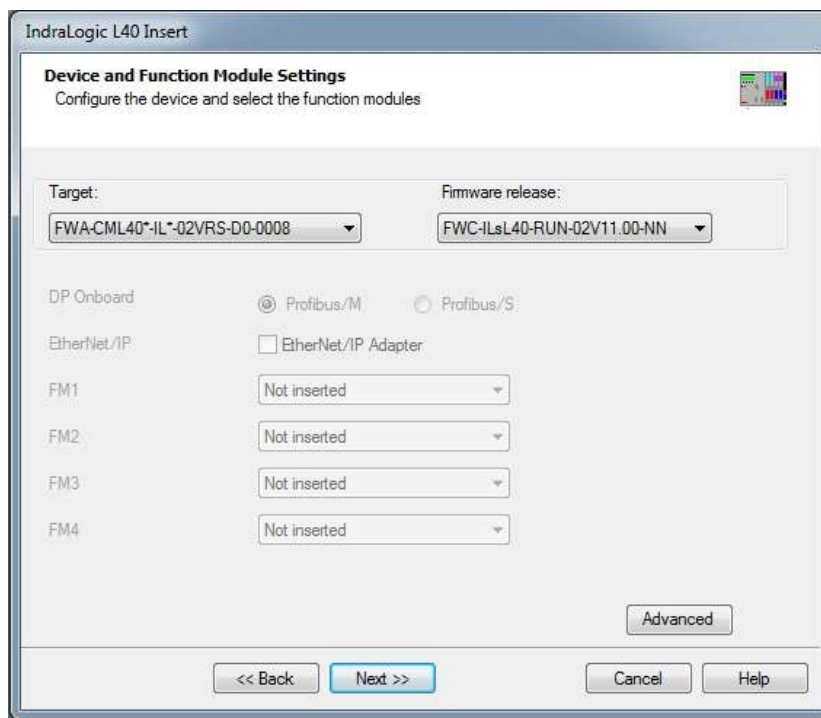
A behúzás után a PLC fő egységhez tartozó beállításokat kell elvégezni. A név adott, ezután a második lépésben a Firmware verzióját kell kiválasztani.

A Firmware-t a PLC hardveregységből lehet kiolvasni (lásd 10. ábra). Az Enter gomb megnyomása után a kurzorgombok segítségével a Firmware opcióra lépve, majd Enter gomb megnyomása után lehet a Firmware verziót kiolvasni.



10. ábra: PLC Firmware verziójának kiolvasása

Az ábrán látszik, hogy 02V13 verzió jelent meg. A szoftverben valószínűleg nem pontosan ugyanilyen verzió van, így javasolt pl. a 02V11 verziót kiválasztani a 11. ábrán látható módon.

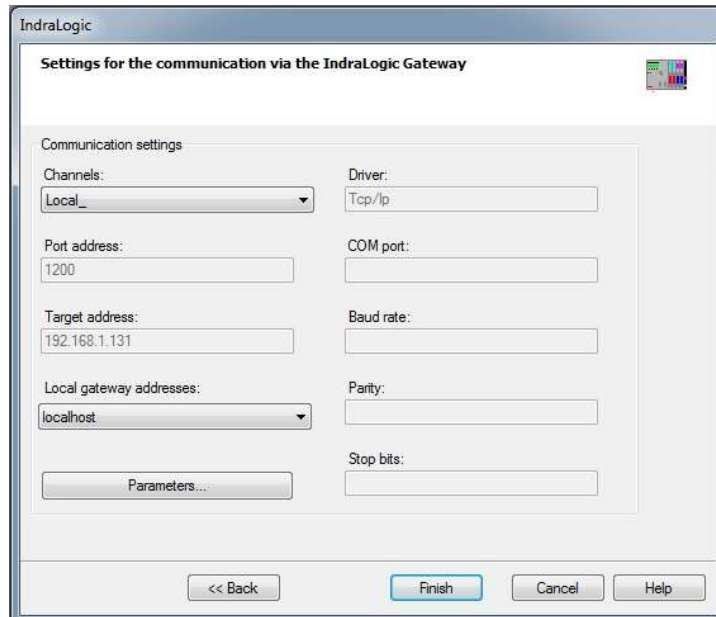


11. ábra: Firmware kiválasztás

A Firmware megadása után a kommunikációs portot kell kiválasztani. A PLC hardveregység IP-címének kiolvasása a Firmware-hez hasonló módon történik a 12. ábrán megfigyelhető módon: „Enter gomb / Ethernet menüpont / Enter gomb / IP-cím”. Ez az IP-cím szükséges a szoftverben való beállításhoz. A beállítás lépését a 13. ábra illusztrálja.

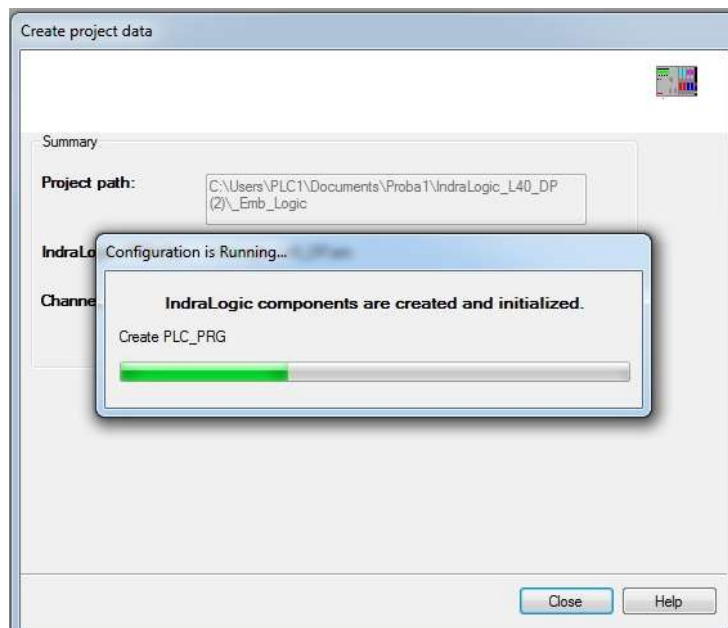


12. ábra: IP-cím kiolvasása a PLC hardveregységen

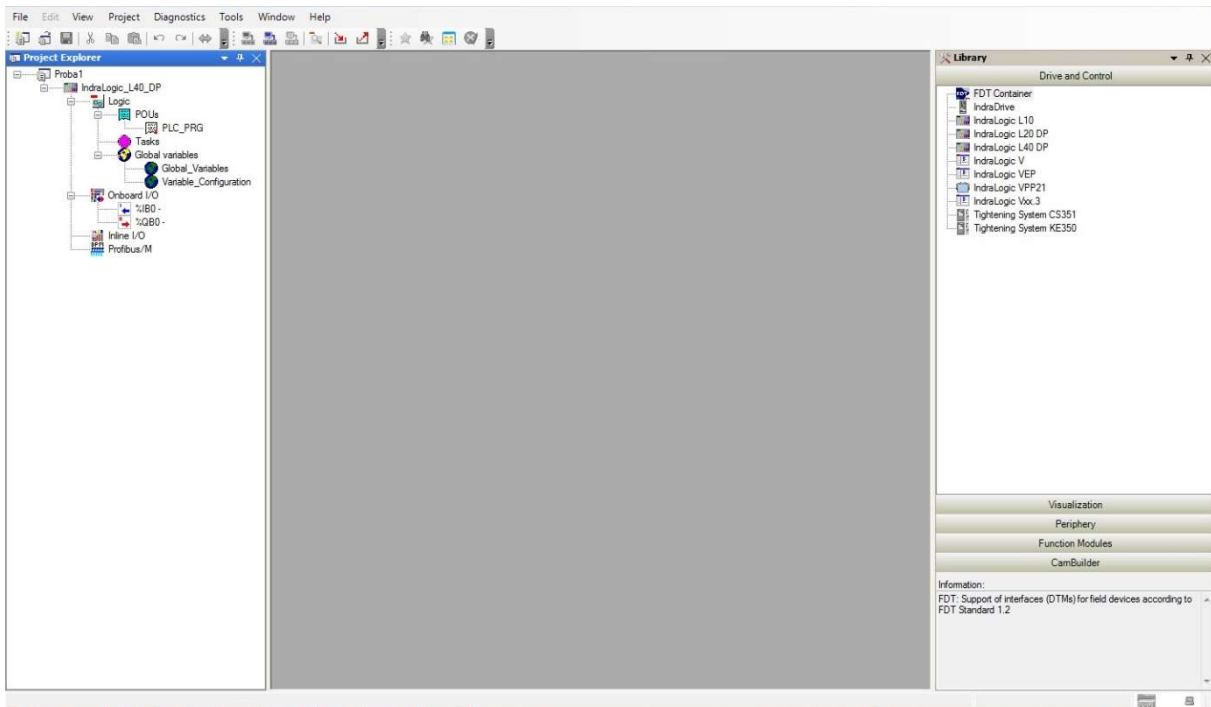


13. ábra: Kommunikáció beállítása

A „Finish” gombra rákattintva létrehozza a szoftver a projektben a PLC egységet a 14. ábrának és 15. ábrának megfelelően.



14. ábra: PLC egység hozzáadásának folyamata a projekthez



15. ábra: PLC egység projekthez való hozzáadása után

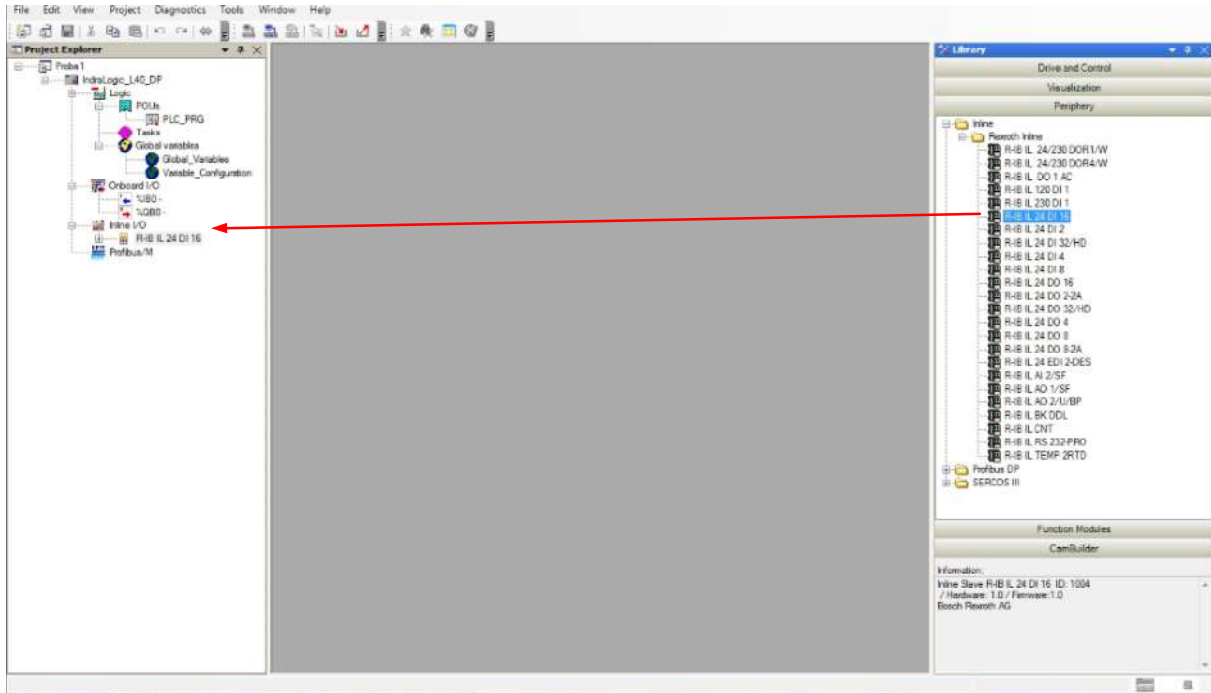
A PLC fő egység hozzáadása után következik az egyes modulok definiálása. A következő modulok hozzárendelésére lesz szükség a 3. ábra alapján:

- Onboard I/O modul
- R-IB IL 24 DI 16 digitális bemeneti modul
- R-IB IL 24 DO 16 digitális kimeneti modul
- R-IB IL AI 2/SF analóg bemeneti modul
- R-IB IL AO 2/U/BP analóg bemeneti modul

Az Onboard I/O modul létrehozása az IndraLogic L40 PLC főegység hozzáadásával együtt valósul meg, ahogy a 15. ábrán is megfigyelhető.

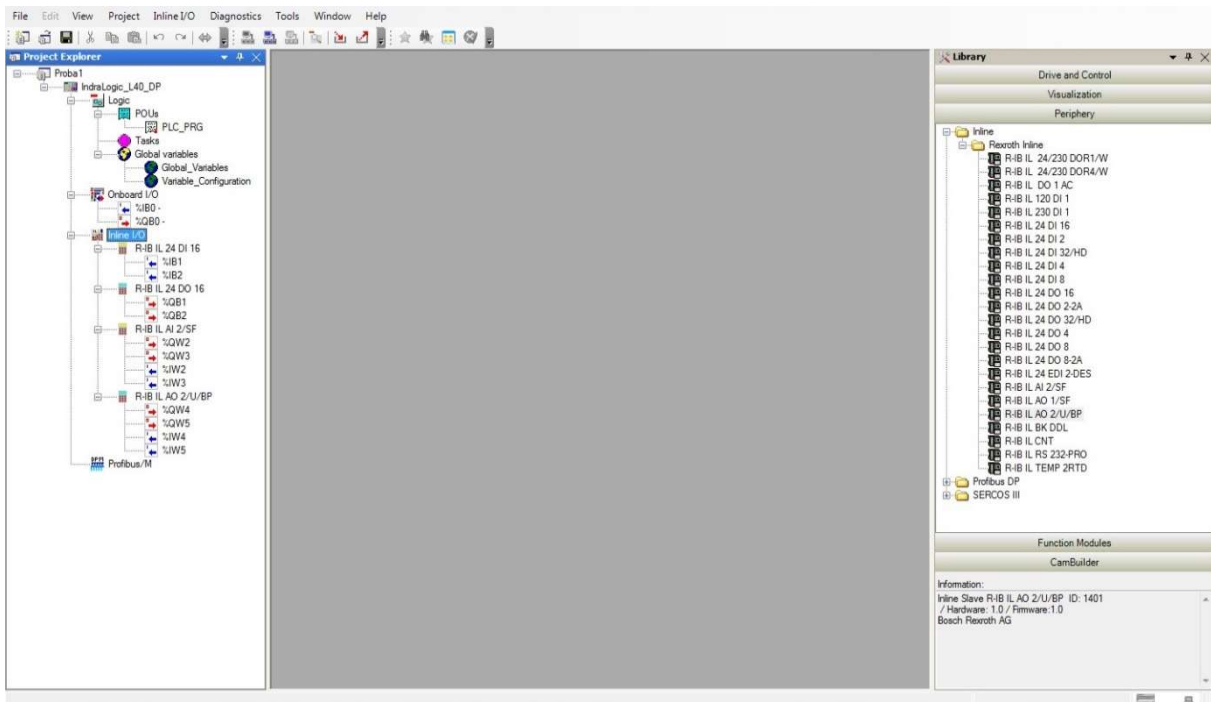
A DI 16, DO 16, AI 2/SF és AO 2/U/BP modulokat már külön kell hozzáadni a projekthez. Mind a négyet hasonló módon kell hozzárendelni, így csak az R-IB IL 24 DI 16 hozzáadásának ismertetését mutatjuk be.

A bemeneti és kimeneti modulok a jobb oldali ablakban a „Periphery/Inline/Rexroth Inline” részben található meg. Innen a kiválasztott modult a jobb oldalról a bal oldali ablakban található Inline I/O részbe kell behúzni a 16. ábrán látható módon.



16. ábra: DI 16 modul hozzáadása

A modulok hozzáadása után a kész projektet a 17. ábrán megfigyelhető ablak mutatja.



17. ábra: Projekt felépítése után

#### 4. Programnyelvek ismertetése

Az IndraLogic szoftverben a következő PLC nyelvek érhetőek el az [1] és a [4] alapján:

- IL (Instruction List): Utasításlistás programnyelv
- ST (Structured Text): Strukturált szövegű programnyelv
- SFC (Sequential Function Chart): Sorrendvezérlési diagram programnyelv
- FBD (Function Block Diagram): Funkcióblokk-diagram programnyelv
- CFC (Continuous Function Chart): Grafikus funkcióblokkos programnyelv
- LD (Ladder Diagram): Létradiagram programnyelv

A programnyelveket az alábbiakban röviden ismertetjük az [1] és a [4] felhasználásával.

IL (Instruction List): Utasításlistás programnyelv leírása:

Ez a programozási nyelv egyidős a PLC-k kialakulásával és az assembly nyelvű programozásból alakult ki és a bitszervezésű PLC-kenél erősen kötődött a hardverstruktúrához. Az ilyen felépítésű PLC-kben egy-egy utasítás a memória egy címén helyezkedett el. Programvezérlési utasítások egymásutánjából áll a program. Az utasítások két részből állnak, először a műveleti (operációs) rész, majd az operandus rész következik. A műveleti részben határozzuk meg, hogy a processzornak milyen műveletet kell elvégeznie (pl. LD – Load, adat betöltése). Az operandus részben azt határozzuk meg, hogy mivel (pl. melyik bemeneti értékkel, változóval) kell a kiadott műveletet elvégezni. Lehetőség van úgynevezett kiegészítő részek megadására is, melyek a negáció (N) és a zárójelek. Az utasításlistás programnyelvet azok a programozók választják, akik assembly nyelvben jártasak.

Példa a [4] alapján:

LD	17	
ST	lint	(* comment *)
GE	5	
JMPC	next	
LD	idword	
EQ	istruct.sdword	
STN	test	
next:		

ST (Structured Text): Strukturált szövegű programnyelv leírása:

A strukturált programnyelv egy magas szintű nyelv a vezérlési és szabályozási funkciók leírására az REPEAT-UNTIL; WHILE-DO, IF-THEN-ELSE, CASE, stb. parancsok és matematikai függvények segítségével.

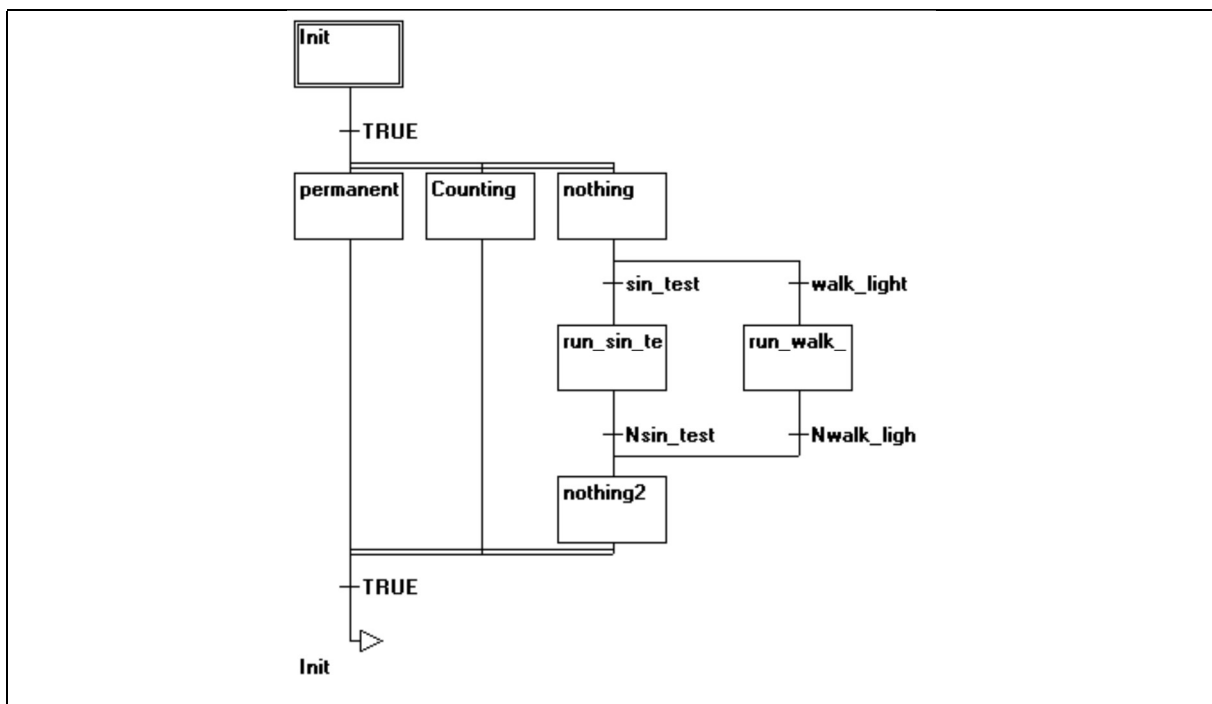
Példa a [4] alapján:

```

IF value < 7 THEN
WHILE value < 8 DO
value:=value+1;
END_WHILE;
END_IF;
    
```

SFC (Sequential Function Chart): Sorrendvezérlési diagram programnyelv leírása:

A folyamatábrán a sorrendi hálózat egy-egy belső állapotát, az ezen állapotba kerülés feltételeit, valamint az állapothoz rendelt kimeneti eseményeket tüntetik el. Ez a módszer az „1 az n”-ből típusú állapotkódolásból fejlődött ki. A következő lépésre való áttérés – illetve elágazás – feltétele található két lépés között. A folyamat felülről lefelé, az elágazás jobbra halad. Az SFC programnyelven elsősorban a szervező (Organizer Block – OB) feladatokat ellátó programokat célszerű megírni. Alkalmas szekvenciális (sorrendi) hálózatok programozására is, ez esetben a pneumatikus léptetőláncokhoz hasonló logikát követ. Példa a [4] alapján a 18. ábra:

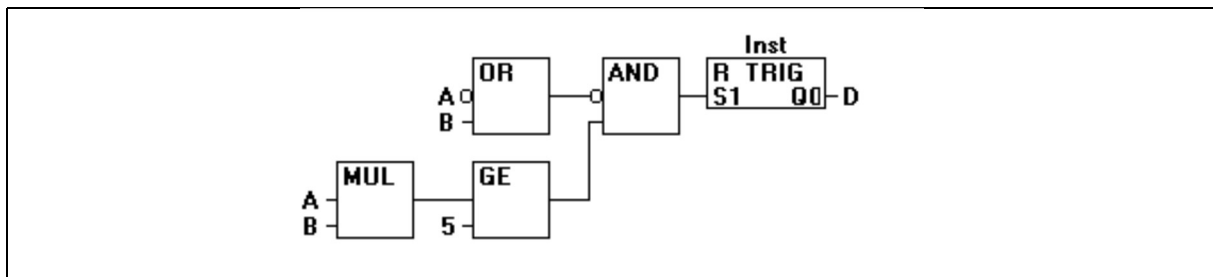


18. ábra: SFC programnyelvre példa

FBD (Function Block Diagram): Funkcióblokk-diagram programnyelv leírása:

A funkcióblokk-diagram tulajdonképpen huzalozott logikában az SSI, MSI áramköröknél használt szimbólumokból kialakított, erősen hardverorientált nyelv. A programban az egyes funkciókhoz blokkok tartoznak. A blokkok bal oldalán a bemenetek, jobb oldalán a kimenetek vannak feltüntetve. A blokkokat vezetékekkel kell összekötni, ami a jelfolyamot is ábrázolja. Az FBD programnyelv a logikai áramkörök tervezésében jártas szakemberek körében is kedvelt és Európában igen elterjedt programnyelv (pl. a Siemens PLC-k támogatják).

Példa a [4] alapján a 19. ábra:



19. ábra: FBD programnyelvre példa

LD (Ladder Diagram): Létradiagram programnyelv leírása:

A létradiagramos programozási nyelv az áramútervből alakult ki, így a létradiagramok az áramútervek szoftver megfelelői. Az áramúterv készítésében jártas irányítástechnikai szakemberek számára az új eszköz (a PLC) programozásának elsajátítása könnyebb volt. A létradiagram a logikai függvények áramutas leírására szolgál, a kontaktusok (érintkezők) megfelelő összekötésével.

A létradiagramok főbb elemei:

- kontaktusok,
- huzalozás,
- logikai kimenetek (Output, Flag, MERKER),
- időzítők,
- számlálók,
- különleges elemek (pl. léptetőregiszter, PID blokk).

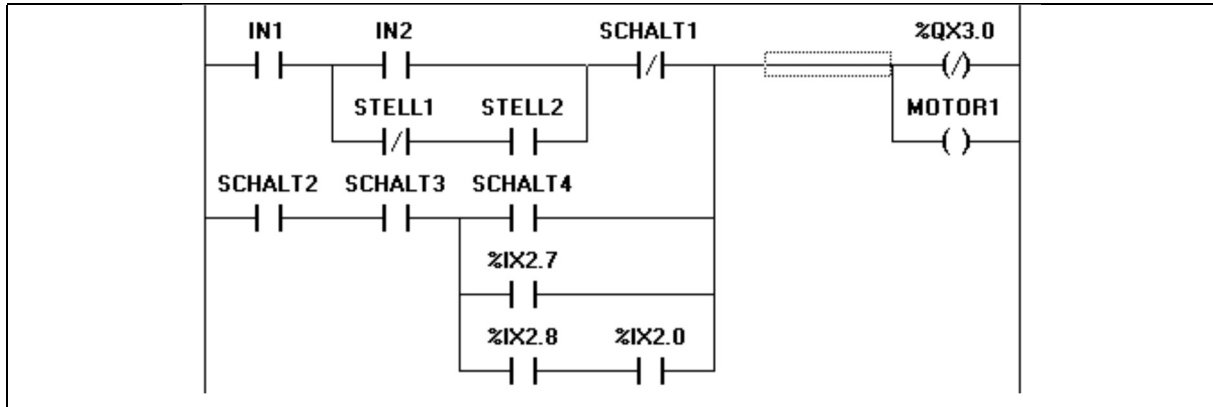
A létradiagram programnyelv főbb szabályai:

- Az áramutakat (létrákat) vízszintesen kell rajzolni a két függőleges vonal (bal – táp, jobb – föld) között.
- Egy létra vezetéke bal oldalon kontaktussal kezdődik (logikai feltételek), jobb oldalon pedig kimenetet (vagy más elemet) működtető tekerccsel végződik.



- A vezetékek nem kereszteződhetnek, a párhuzamos ágakkal való bővítés lefelé történik.
- Az egyes ágakat feszültségmentes állapotban kell ábrázolni.
- Az áramutakat a működési sorrendben (felülről lefelé) ábrázolják.

Példa a [4] alapján a 20. ábra:

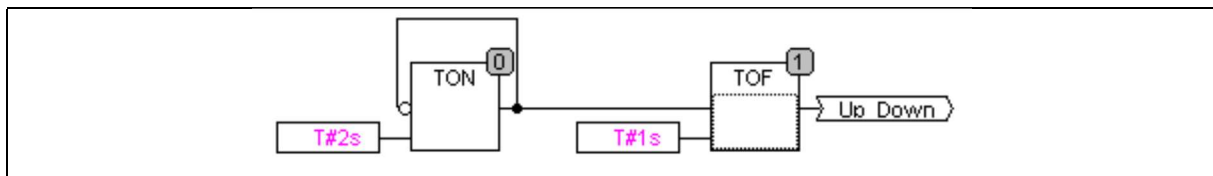


20. ábra: LD programnyelvre példa

CFC (Continuous Function Chart): Grafikus funkcióblokkos programnyelv leírása:

A CFC nyelv a funkcióblokkos programozásnak egy szabadabban alkalmazható változata. Közvetlen visszakötések alkalmazhatóak benne, egy-egy hálózatnak több kimenete is lehet, ezen kívül programrészekből makrók hozhatók létre, valamint a műveletek végrehajtási sorrendje szabadon megválasztható.

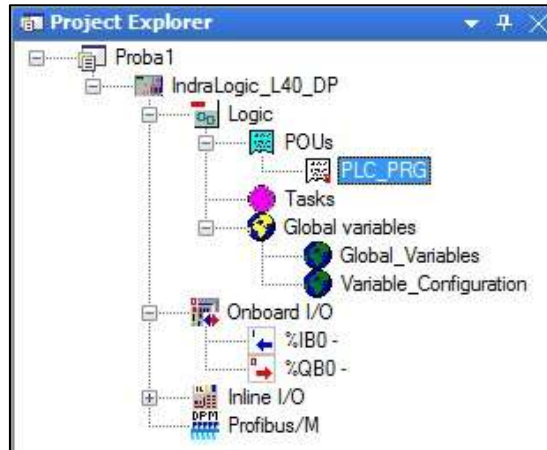
Példa a [4] alapján a 21. ábra:



21. ábra: CFC programnyelvre példa

## 5. Bemenet-másoló PLC program létrehozása LD programnyelven

A PLC programozása az IndraLogic szoftverrel lehetséges. Az indítást az IndraWorks Engineering szoftverből érdemes megtenni, a projektben található PLC\_PRG-re kattintva a 22. ábrán látható módon.



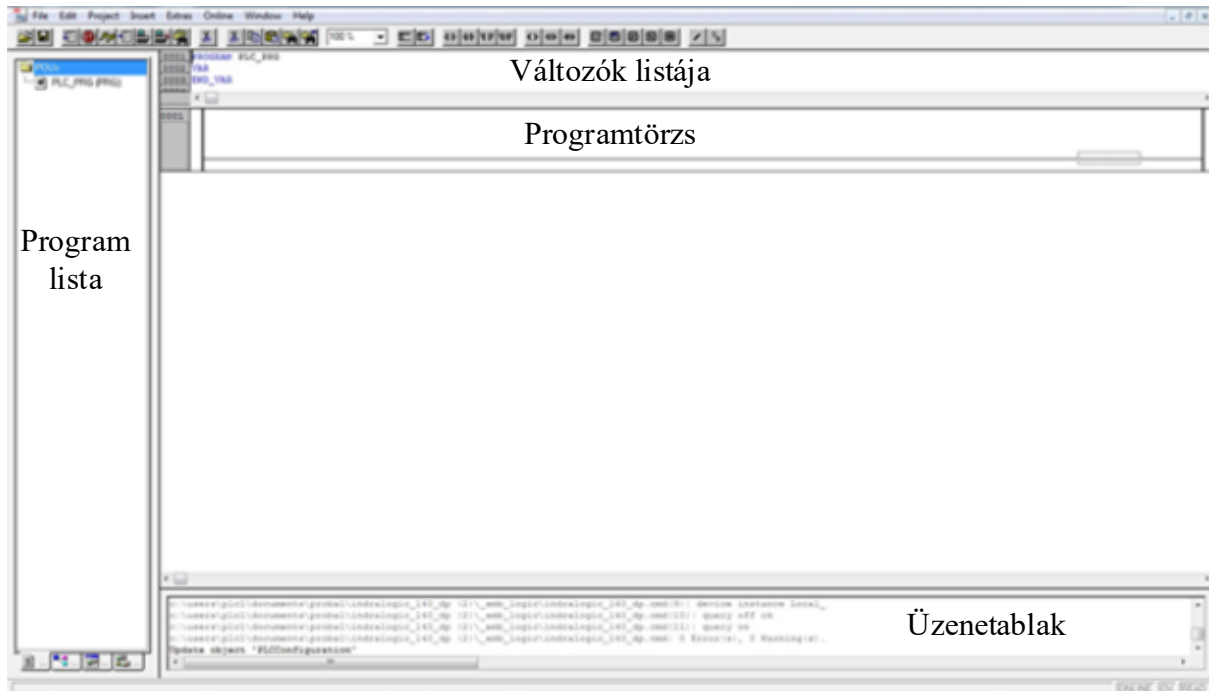
22. ábra: IndraLogic szoftver elindítása a PLC\_PRG-re kattintva

A kattintás után az IndraLogic induló képernyője fogad (lásd 23. ábra).



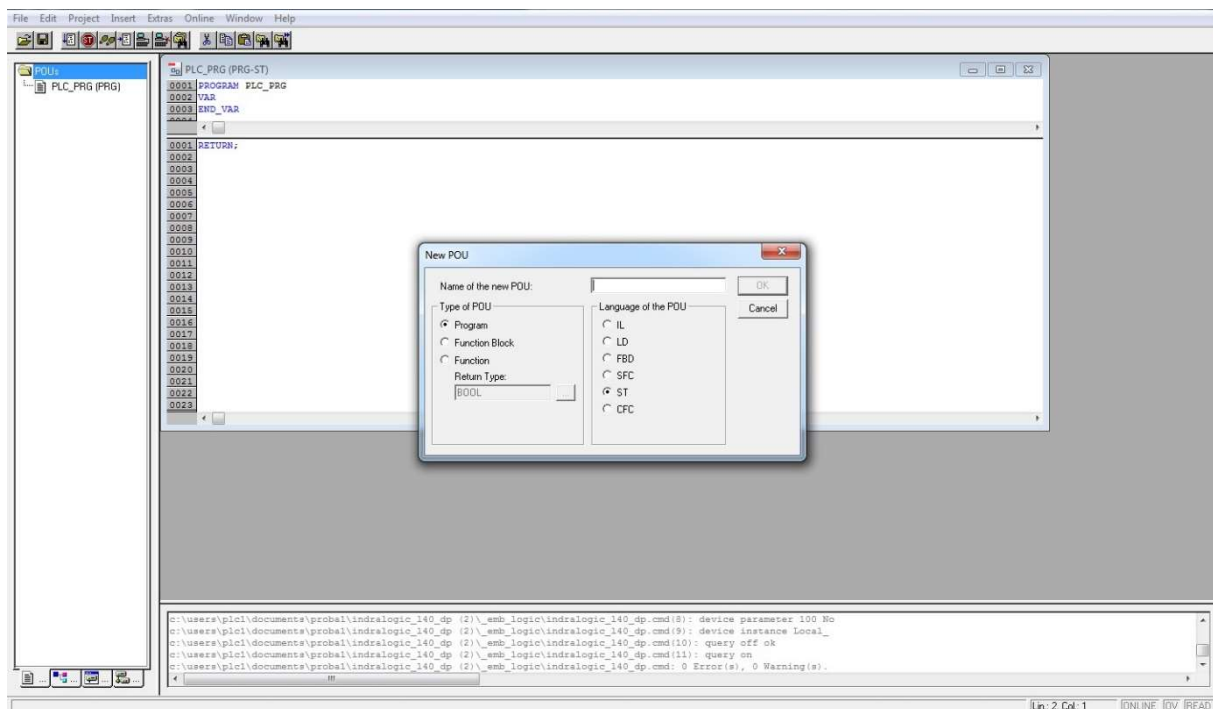
23. ábra: IndraLogic induló képernyője

Ezt követően a 24. ábrán látható ablak jelenik meg. Ugyanezen az ábrán látható az ablak felépítése is. Itt a programtörzs még üres.



24. ábra: Programozási ablak felépítése

A POU (Program Organization Unit) jelentése „Program Szervezési Egység”. A POU-k többnyire alprogramok, de PLC\_PRG néven főprogrammá válik. Új POU-t létrehozni a „Project/Object/Add” menüponttal lehetséges, ekkor a 25. ábrán látható ablak fogad. Itt meg kell adni a POU nevét, a POU típusát, valamint a programozási nyelvet. A „PLC\_PRG” mindig a főprogram lesz, innen lehet meghívni az alprogramokat.



25. ábra: Új POU hozzáadása

Az egyszerű program felépítése a következőkben LD programnyelven kerül bemutatásra.

A program lényege, hogy a szimulációs táblán levő kapcsolók a mellettük levő LED-eket felvillantsák, ha a kapcsoló aktív (bemenet-másoló).

Következő lépésként bemeneteket és kimeneteket kell megírni a „Változók listája” nevű részbe.

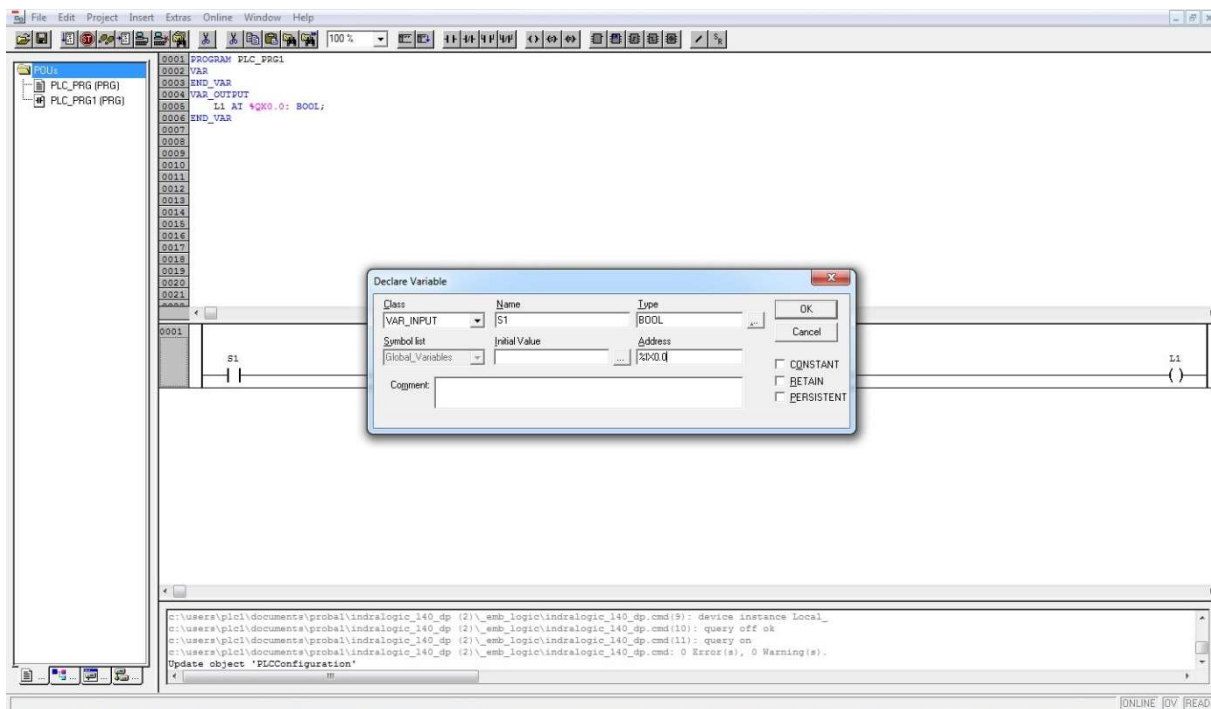
Input hozzáadását egy kontakt elnevezése után fogja felajánlani a szoftver a 26. ábrának megfelelően. Itt meg kell adni a változó osztályát, típusát, nevét, és címét.

A címet az IndraWorks Engineering szoftver Onboard I/O megfelelő részére rákattintva lehet kiolvasni.

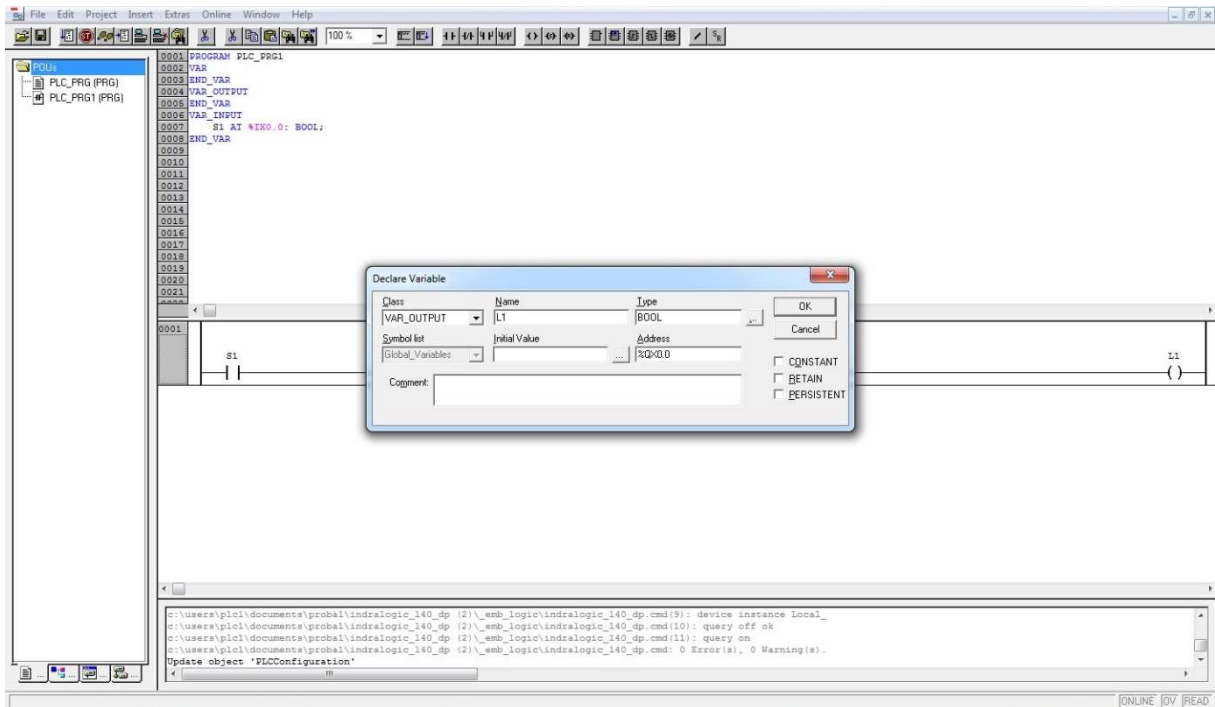
Output hozzáadása az Input hozzáadásához hasonló módszerrel történik a 27. ábrán látható módon.

Pl.

- S1 nevű változó %IX0.0 címet kapja,
- L1 nevű változó %QX0.0 címet kapja.

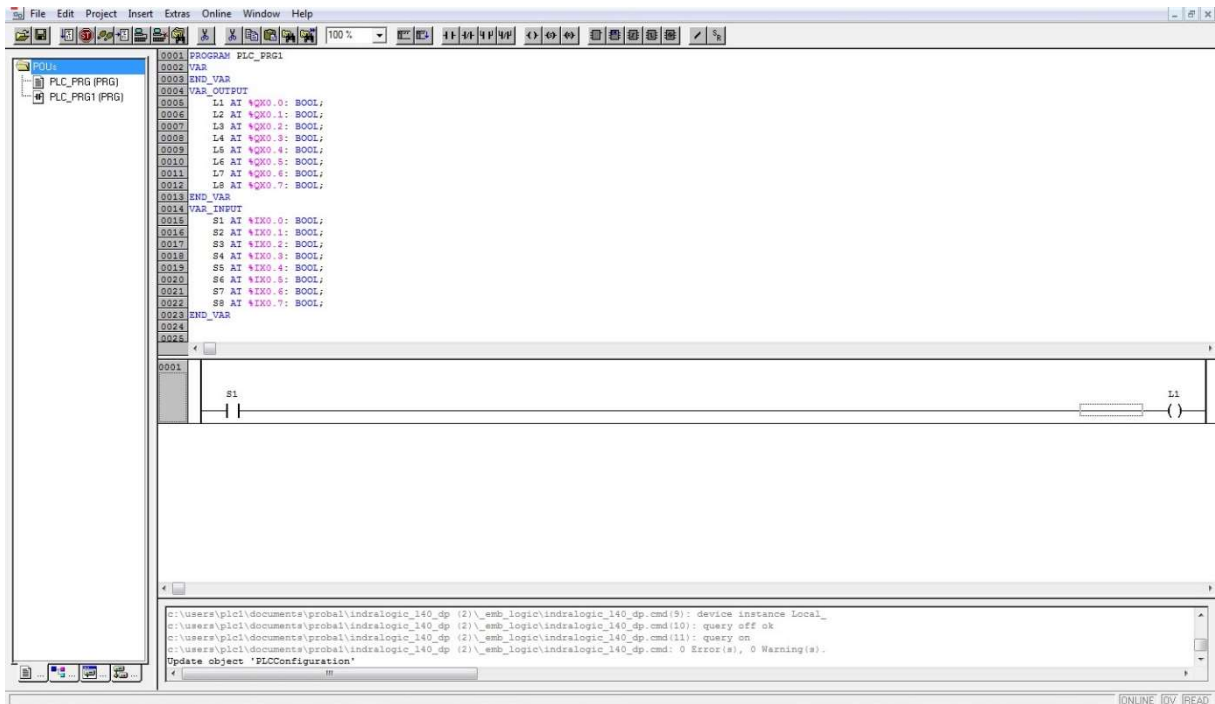


26. ábra: Input hozzáadása



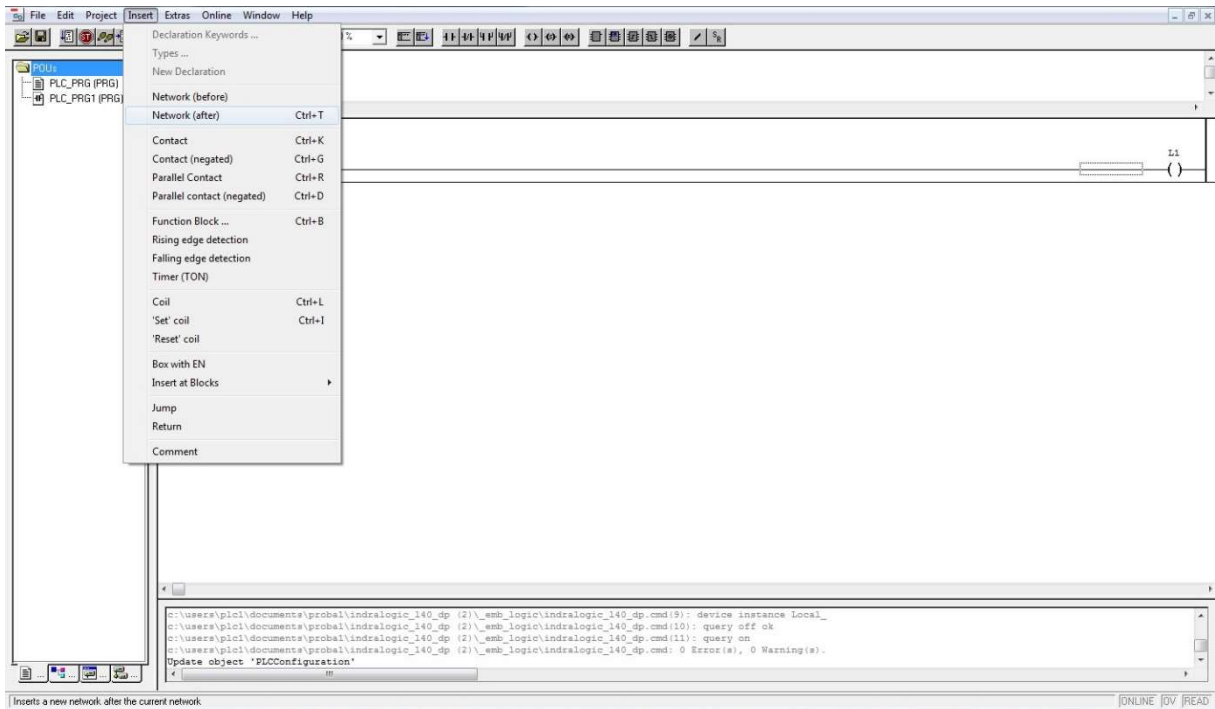
27. ábra: Output hozzáadása

Az egyszerű programhoz szükséges 8-8db Input és Output definiálása után a 28. ábrán szemléltetett változó lista fog megjelenni.



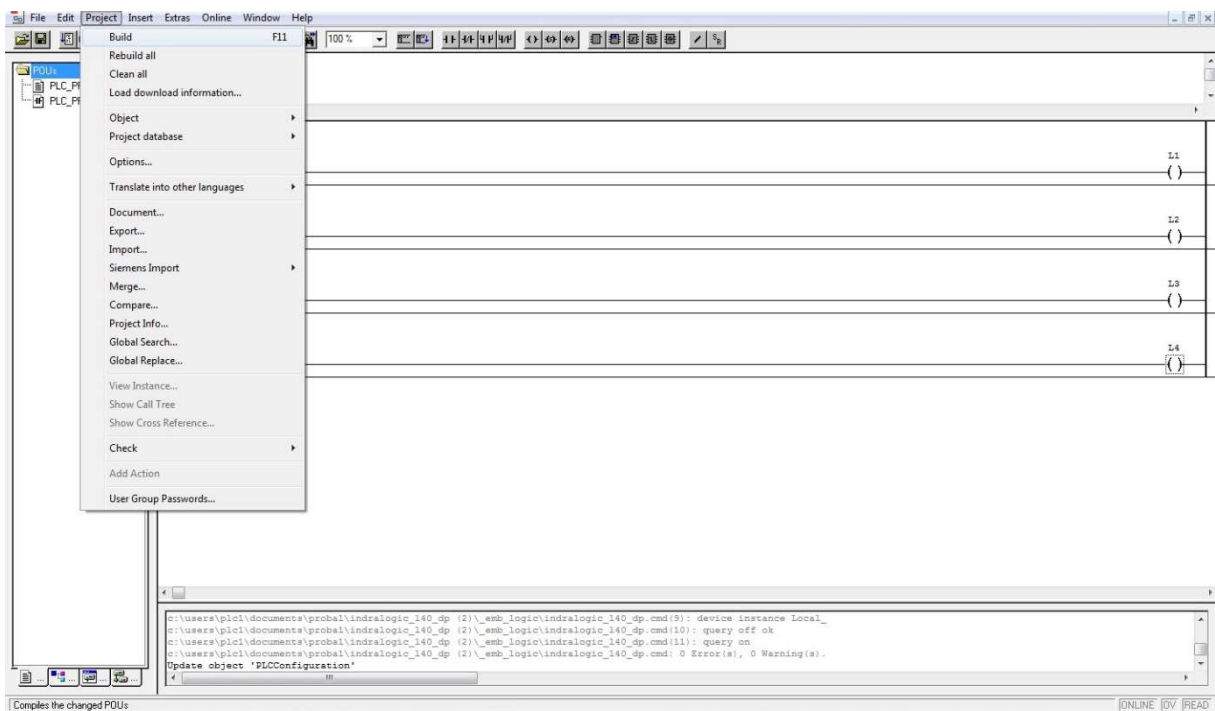
28. ábra: Input és output hozzáadása után

Az LD programnyelvben új sor hozzáadása az „Insert/Network After” menüponttal lehetséges (lásd 29. ábra).



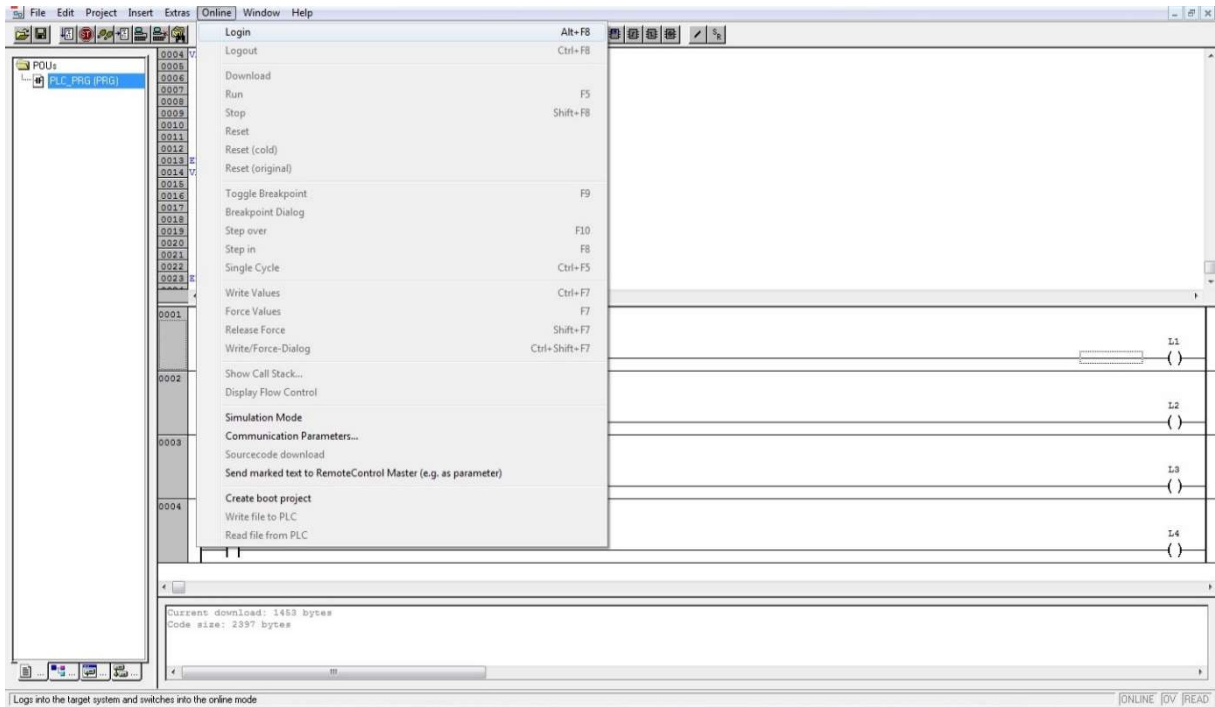
29. ábra: LD programnyelvben új sor hozzáadása

A program elkészülése után „Project/Build” menüponttal lehet ellenőrizni és felépíteni a futtatható programot (30. ábra).



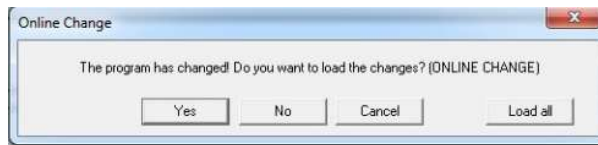
30. ábra: Futtatható program létrehozása

A PLC-hez csatlakozni az „Online/Login” menüponttal lehetséges (lásd 31. ábra).



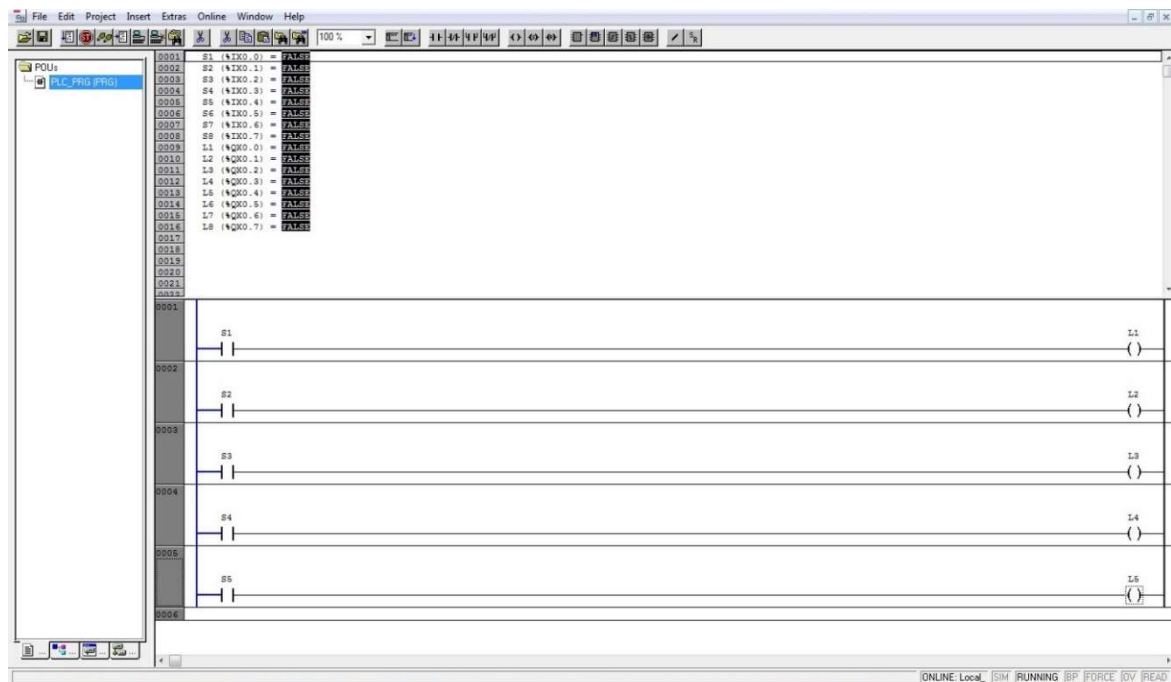
31. ábra: PLC-hez csatlakozás

A PLC-re csatlakozás esetén, ha felugrik a 32. ábrán látható üzenet, ekkor „Yes”-t kell nyomni.



32. ábra: "Program has changed" üzenetablak

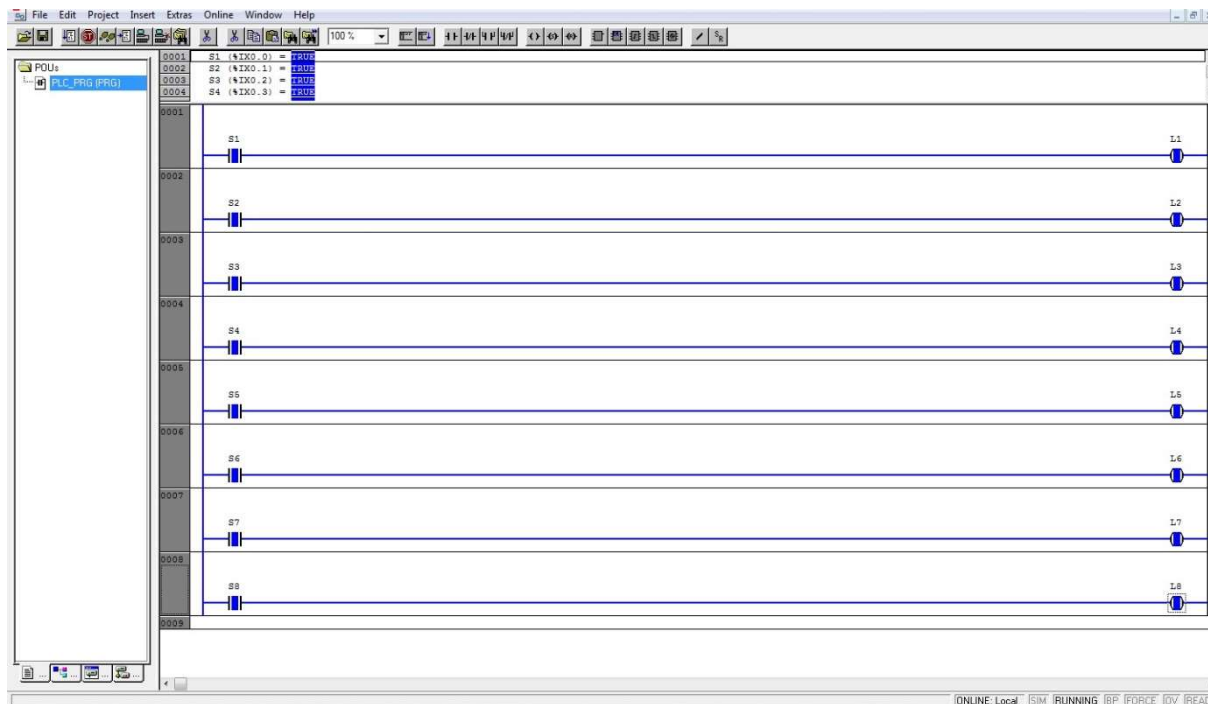
A PLC-re csatlakozás után a 33. ábra szerinti ablak jelenik meg.



33. ábra: PLC-re csatlakozás után

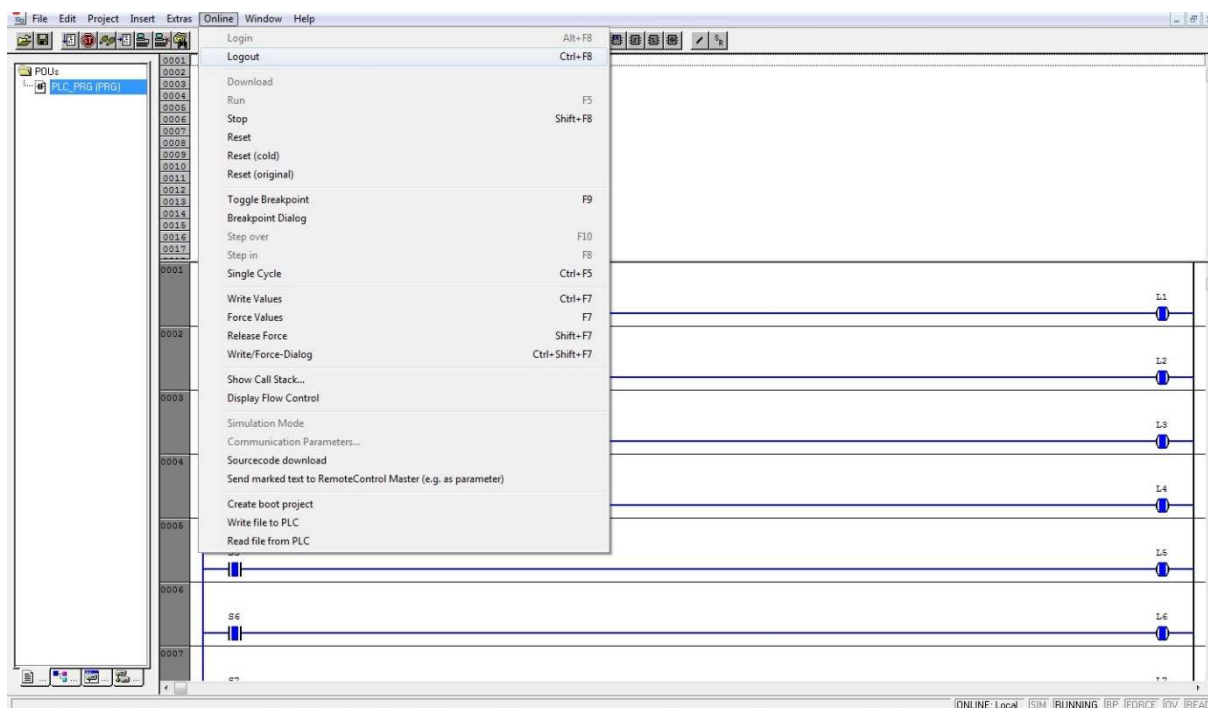
A 33. ábrán lehet látni, hogy az egyes bemenetek, kimenetek „True” (igaz) vagy „False” (hamis) értéket vesznek fel.

A másoló kapcsolás programját LD programnyelven a 34. ábra szemlélteti. Az ábrán mind a 8 kapcsoló aktív, így mind a 8 LED világítani fog.



34. ábra: Másoló kapcsolás programja LD programnyelven

A program módosításához mindig ki kell jelentkezni a PLC-ből. Ezt az „Online/Logout” menüponttal lehet elérni (35. ábra).



35. ábra: Kijelentkezés a PLC-ből



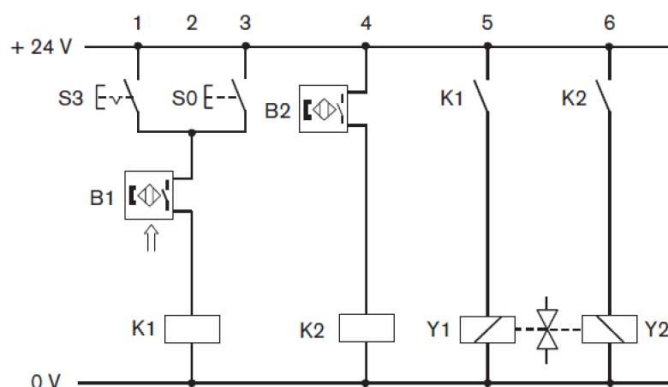
## 6. PLC feladatok

### 6.1 Feladatok megoldása LD programnyelven

Az elektropneumatikában megoldott feladatgyűjtemény alapján lettek kiválasztva a következő PLC feladatok.

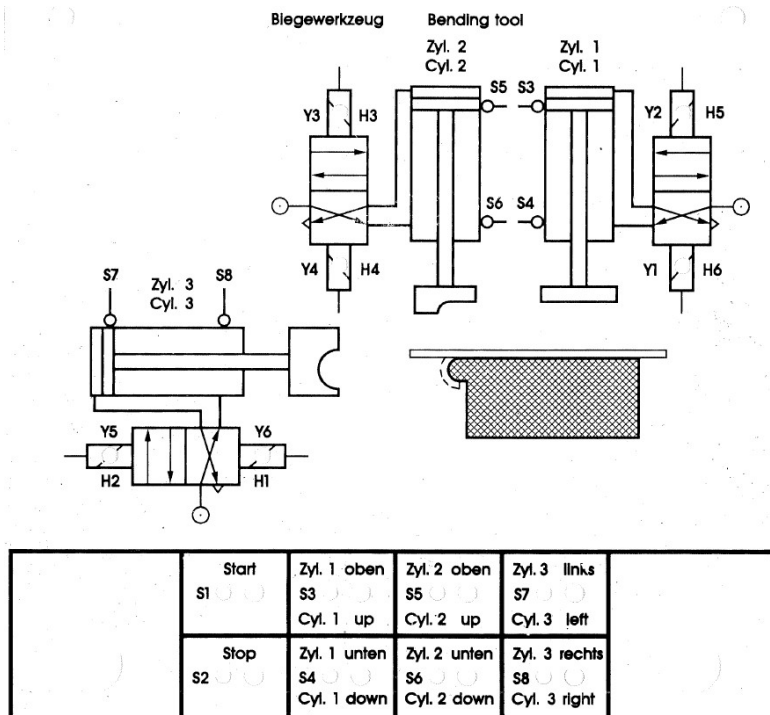
#### 6.1.1 A 9. elektropneumatikai feladat vezérlése PLC-vel

A feladatgyűjtemény 9. feladatának címe „*Útfüggő vezérlés, kettős működésű henger, impulzus-szelep, elektromos érzékelők*”. A feladat villamos kapcsolása a 36. ábrán látható.



36. ábra: Elektropneumatika feladatgyűjtemény 9. feladatának villamos kapcsolása

A PLC programozás során az S3 kapcsoló nem kerül felhasználásra. A feladathoz a PLC szimulációs lapok közül a 19. számút választjuk, ahol csak az 1. munkahengert fogjuk működtetni, a munkahenger feladata a munkadarab leszorítása és felengedése.



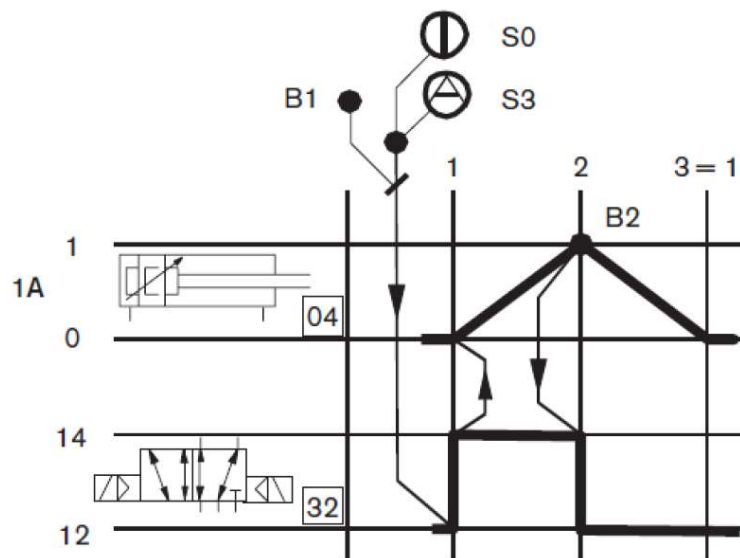
37. ábra: Elektropneumatikus kapcsolás táblája

Az elektropneumatika során használt S0 jelű nyomógomb a PLC programozás során S1 jelű nyomógomb, a B1 jelű Reed-relé S3 jelű bemenet, a B2 jelű Reed-relé S4 jelű bemenet lesz.

A működtetési sorrend:

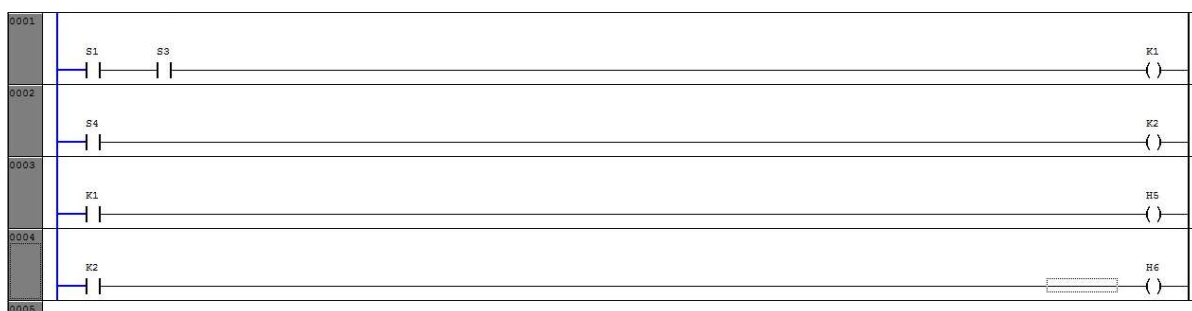
1. A munkahenger indítás feltétele az S1 nyomógomb és S3 végállás érzékelő működtetése
2. A munkahenger kifutását a K1 relén keresztül a H5 LED kivillanása fogja jelezni
3. A visszafutás feltétele az S4 végállás érzékelő működtetése
4. A munkahenger visszafutását a K2 relén keresztül a H6 LED kivillanása fogja jelezni

A feladat ütemdiagramját a 38. ábra részletezi.



38. ábra: A 9. elektropneumatikai feladat ütemdiagramja

A kapcsolást LD programnyelven programozva a 39. ábra illusztrálja. A működtetés során az aktív elemek kék színnel világítanak a számítógépen.



39. ábra: Elektropneumatika 9. feladatának megoldása LD programnyelven

A definiált változókat a 40. ábra szemlélteti.

```

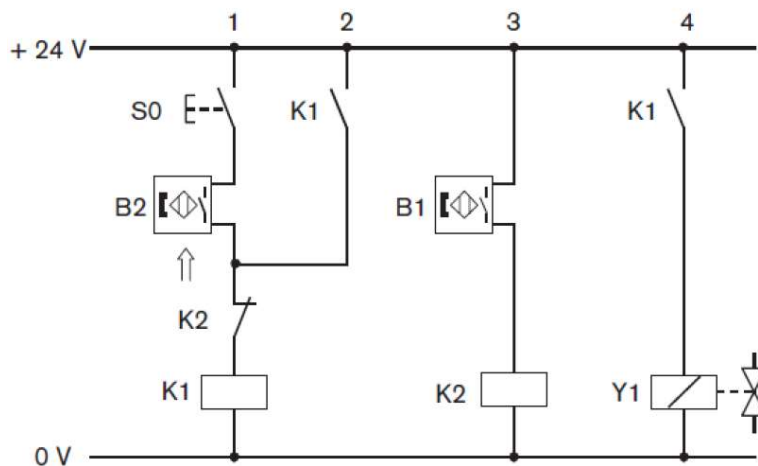
PROGRAM Bistabil_1_munkahenger
VAR
  K1: BOOL;
  K2: BOOL;
END_VAR
VAR_OUTPUT
  H1 AT %QX0.0: BOOL;
  H2 AT %QX0.1: BOOL;
  H3 AT %QX0.2: BOOL;
  H4 AT %QX0.3: BOOL;
  H5 AT %QX0.4: BOOL;
  H6 AT %QX0.5: BOOL;
  H7 AT %QX0.6: BOOL;
  H8 AT %QX0.7: BOOL;
END_VAR
VAR_INPUT
  S1 AT %IX0.0: BOOL;
  S2 AT %IX0.1: BOOL;
  S3 AT %IX0.2: BOOL;
  S4 AT %IX0.3: BOOL;
  S5 AT %IX0.4: BOOL;
  S6 AT %IX0.5: BOOL;
  S7 AT %IX0.6: BOOL;
  S8 AT %IX0.7: BOOL;
END_VAR

```

40. ábra: Elektropneumatika 9. feladatának megoldása során használt változók

### 6.1.2 A 10. elektropneumatikai feladat vezérlése PLC-vel

Ebben a feladatban az elektropneumatikai feladatgyűjtemény 10. feladatát oldjuk meg, amely hasonlít a 6.1.1.-hez, de a „Kettős működésű munkahenger útfüggő vezérlése, rugós visszatérítésű szeleppel, elektromos érzékelőkkel” címmel jelölt útváltó szelepet alkalmazzuk egyetlen munkahengerre. A feladat villamos kapcsolása a 41. ábrán látható.



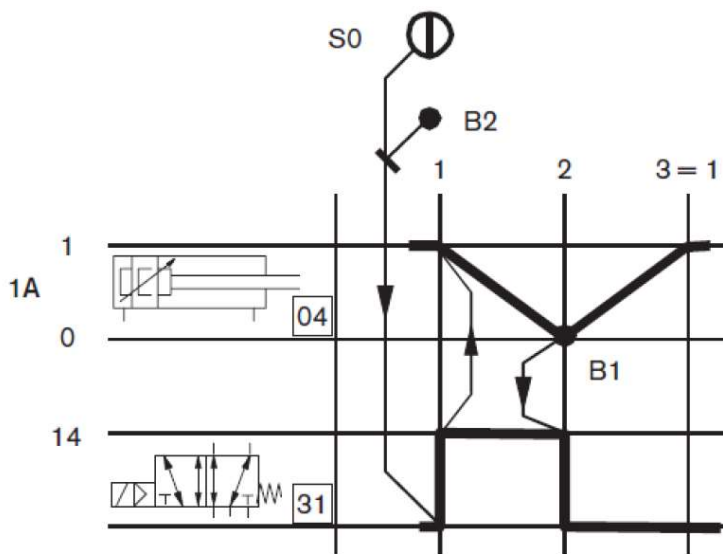
41. ábra: Elektropneumatika feladatgyűjtemény 10. feladatának villamos kapcsolása

Az elektropneumatika során használt S0 jelű nyomógomb a PLC programozás során ismét S1 jelű nyomógomb, a B1 jelű Reed-relé S3 jelű bemenet, a B2 jelű Reed-relé S4 jelű bemenet lesz.

A működtetési sorrend:

1. A munkahenger indítás feltétele az S1 nyomógomb és S3 végállás érzékelő működtetése, ez létrehozza az öntartó funkciót

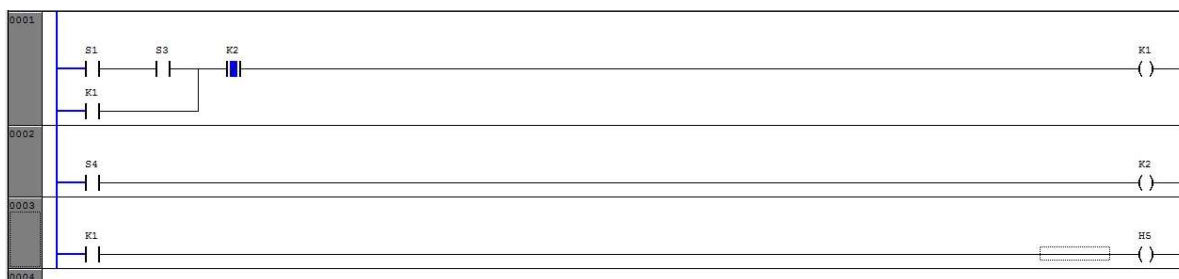
2. A munkahenger kifutását a K1 relén keresztül a H5 LED kivillanása fogja jelezni
3. A visszafutás feltétele az S4 végállás érzékelő működtetése, bontja az öntartást



42. ábra: A 10. elektropneumatikai feladat ütemdiagramja

A feladat ütemdiagramját a 42. ábra mutatja.

Az LD programnyelven megírt PLC programot a 43. ábra illusztrálja.

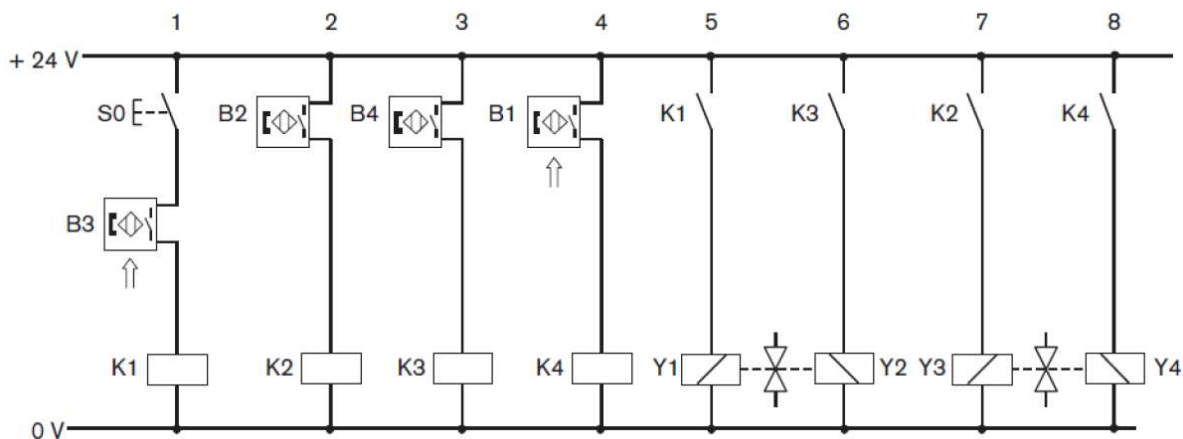


43. ábra: Elektropneumatika 10. feladatának megoldása LD programnyelven

### 6.1.3 A 16. elektropneumatikai feladat vezérlése PLC-vel

Ebben a feladatban az elektropneumatikai feladatgyűjtemény 16. feladatát oldjuk meg, amelyben már 2 munkahengert 2 útváltóval vezérlünk. A feladat címe „Sorrind vezérelt kettős működésű munkahengerek vezérlése impulzus szelepekkel”. A fizikai feladat a 37. ábra felső részének feleltethető meg. Az elektropneumatika során használt S0 jelű nyomógomb a PLC programozás során ismét S1 jelű nyomógomb, a B1 jelű Reed-relé S3 jelű bemenet, a B2 jelű Reed-relé S4 jelű bemenet, a B3 jelű Reed-relé S5 jelű bemenet, a B4 jelű Reed-relé S6 jelű bemenet lesz.

A feladat villamos kapcsolását a 44. ábra mutatja.

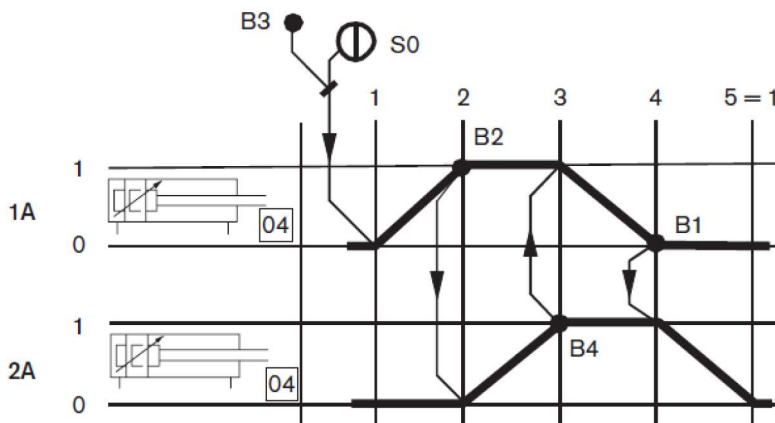


44. ábra: Elektropneumatika feladatgyűjtemény 16. feladatának villamos kapcsolása

A működtetési sorrend:

1. A 1. munkahenger indítás feltétele az S1 nyomógomb és S5 végállás érzékelő működtetése
2. A 1. munkahenger kifizetését a K1 relén keresztül a H5 LED kivillanása fogja jelezni
3. A 2. munkahenger indítás feltétele az S4 végállás érzékelő működtetése
4. A 2. munkahenger kifizetését a K2 relén keresztül a H3 LED kivillanása fogja jelezni
5. A 1. munkahenger visszafutás feltétele az S6 végállás érzékelő működtetése
6. A 1. munkahenger visszafutását a K3 relén keresztül a H6 LED kivillanása fogja jelezni
7. A 2. munkahenger visszafutás feltétele az S3 végállás érzékelő működtetése
8. A 2. munkahenger visszafutását a K4 relén keresztül a H4 LED kivillanása fogja jelezni

A feladat ütemdiagramját a 45. ábra részletezi.



45. ábra: A 16. elektropneumatikai feladat ütemdiagramja

Az LD programnyelven megírt programot a 46. ábra illusztrálja.



46. ábra: Elektropneumatika 16. feladatának megoldása LD programnyelven

A definiált változókat a 47. ábra szemlélteti.

```

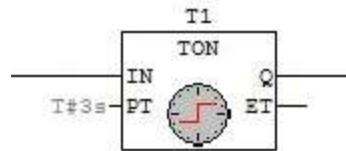
PROGRAM PLC_PRG
VAR
    K1: BOOL;
    K2: BOOL;
    K3: BOOL;
    K4: BOOL;
    T1: TON;
END_VAR
VAR_OUTPUT
    H1 AT %QX0.0: BOOL;
    H2 AT %QX0.1: BOOL;
    H3 AT %QX0.2: BOOL;
    H4 AT %QX0.3: BOOL;
    H5 AT %QX0.4: BOOL;
    H6 AT %QX0.5: BOOL;
    H7 AT %QX0.6: BOOL;
    H8 AT %QX0.7: BOOL;
END_VAR
VAR_INPUT
    S1 AT %IX0.0: BOOL;
    S2 AT %IX0.1: BOOL;
    S3 AT %IX0.2: BOOL;
    S4 AT %IX0.3: BOOL;
    S5 AT %IX0.4: BOOL;
    S6 AT %IX0.5: BOOL;
    S7 AT %IX0.6: BOOL;
    S8 AT %IX0.7: BOOL;
END_VAR
    
```

47. ábra: 16. elektropneumatika feladat megoldása során használt változók

A szimulációs táblán a S3 és S4 érzékelőket, illetve S5 és S6 érzékelőket értelemszerűen egyszerre nem szabad működtetni. A létradiagramban az aktív elemek kékkel világítanak.

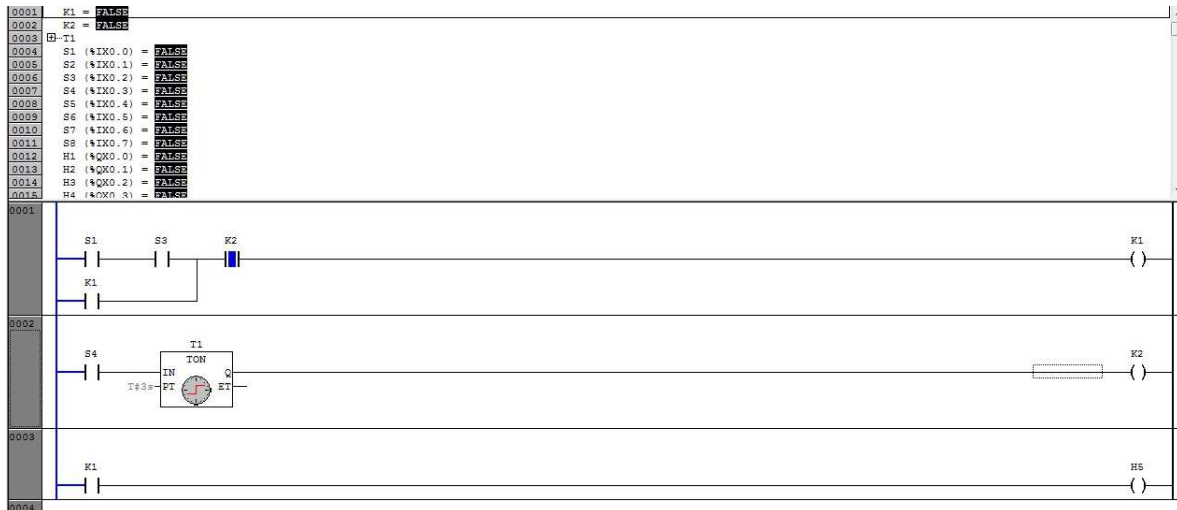
### 6.1.4 Időzítő alkalmazása LD programnyelven

Ha egy PLC kapcsolatban azt szeretnénk, hogy a munkahenger visszafutása késleltetéssel történjen meg, akkor időzítőt kell beépíteni. Ennek LD programbeli blokkja a 48. ábrán látható.



48. ábra: Időzítő blokkja LD programnyelvben

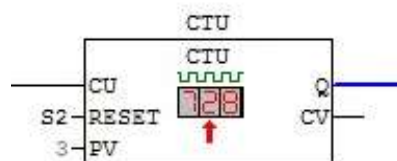
A 10. elektropneumatika feladat kapcsolásba (azaz 6.1.2. alfejezetbe) építjük be az időzítőt. Az elkészült programot és a futtatás közbeni állapotát a 49. ábra szemlélteti.



49. ábra: 10. elektropneumatika feladat megoldása LD programnyelven időzítővel

### 6.1.5 Számláló alkalmazása LD programnyelven

A programozás során szükség lehet olyan feladatra, amikor gombnyomás után automatikusan egymás után egy előre meghatározott számú alkalommal fut ki és vissza a munkahenger. Ezt számláló (angolul „Counter”) segítségével lehet megvalósítani. Az LD programnyelvbeli számláló blokkja az 50. ábrán látható.



50. ábra: Számláló LD programnyelvben

Az 50. ábrán látható jelölések:

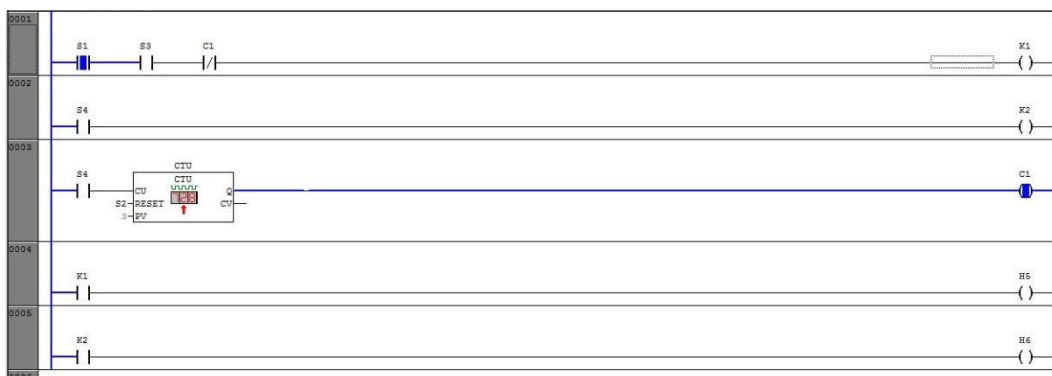
- CU: számláló bemenete
- Q: számláló kimenete
- RESET: ezzel a bemenettel lehet nullázni a számláló értékét
- PV: számlálás értéke
- CTU/CTD/CTUD: számláló típusa: felszámláló, leszámláló és fel-, leszámláló

- Felül: számláló neve (az ábrán CTU névvel)

A számláló használatához a 9. elektropneumatikai feladat (azaz 6.1.1 fejezet) került átalakításra. Itt a 1. munkahenger S4 bemenete lépteti a számláló értékét. Amennyiben a számláló értéke elérte a 3-at, akkor a C1 változó segítségével megszakítja az 1. sorban levő indítás feltételét.

A számlálót az S2 nyomógommbal lehet nullázni.

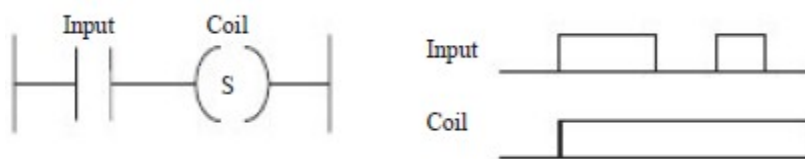
Az elkészült feladatot az 51. ábra mutatja.



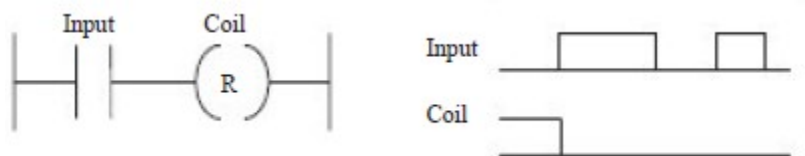
51. ábra: Számlálóval bővített feladat megoldása LD programnyelven

### 6.1.6 Felfutó él, lefutó él alkalmazása LD programnyelven

A 10. elektropneumatikai feladatot (6.1.2 fejezet) „SET Coil” és „RESET Coil” segítségével is meg lehet oldani. A „SET Coil” és „RESET Coil” működését az 52. ábra és az 53. ábra mutatja. „SET Coil” esetén, ha az Input értéke egy pillanatra is „1” lesz, akkor a továbbiakban „Coil” végig „1” marad. „RESET Coil” esetén, ha az Input értéke egy pillanatra is „1” lesz, akkor a továbbiakban „Coil” végig „0” marad, még akkor is, ha előtte „1” volt az értéke.



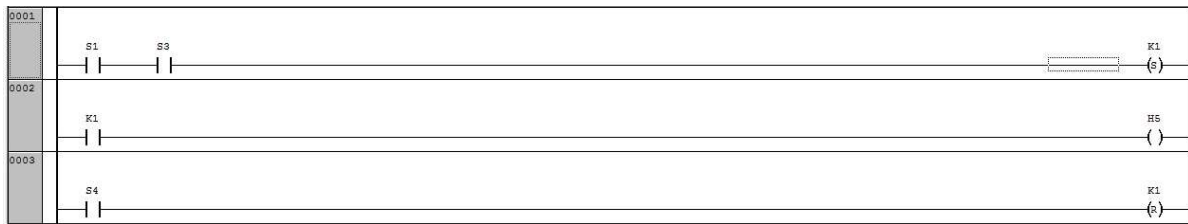
52. ábra: SET Coil működése



53. ábra: RESET Coil működése

Az így megoldott feladatot LD programnyelvben az 54. ábra szemlélteti.





54. ábra: Elektropneumatika 10. feladat megoldása SET-RESET Coil segítségével

## 6.2 Feladat megoldás ST programnyelven

Az ST nyelv a C nyelvhez hasonlatos, ezért a számítógép programozásban általánosan járatos mérnökök kedvelik.

Egy feltétel megadása itt az „IF - THEN – ELSE” szerkezettel lehetséges az 55. ábrához hasonlóan.

```

IF S1 = 1
    THEN L1 := TRUE;
    ELSE L1 := FALSE;
END_IF;
    
```

55. ábra: ST nyelvben egy feltétel megadása

Az 5. fejezet 34. ábra környezetében már ismertetett bemenet másoló program ST-ben való megfogalmazását az 56. ábra mutatja.

```

0001 PROGRAM PLC_PRG
0002 VAR_OUTPUT
0003     L1 AT %QX0.0: BOOL;
0004     L2 AT %QX0.1: BOOL;
0005     L3 AT %QX0.2: BOOL;
0006     L4 AT %QX0.3: BOOL;
0007     L5 AT %QX0.4: BOOL;
0008     L6 AT %QX0.5: BOOL;
0009     L7 AT %QX0.6: BOOL;
0010     L8 AT %QX0.7: BOOL;
0011 IF S1 = 1
0012     THEN L1 := TRUE;
0013     ELSE L1 := FALSE;
0014 END_IF;
0015 IF S2 = 1
0016     THEN L2 := TRUE;
0017     ELSE L2 := FALSE;
0018 END_IF;
0019 IF S3 = 1
0020     THEN L3 := TRUE;
0021     ELSE L3 := FALSE;
0022 END_IF;
0023 IF S4 = 1
0024     THEN L4 := TRUE;
0025     ELSE L4 := FALSE;
0026 END_IF;
0027 IF S5 = 1
0028     THEN L5 := TRUE;
0029     ELSE L5 := FALSE;
0030 END_IF;
0031 IF S6 = 1
0032     THEN L6 := TRUE;
0033     ELSE L6 := FALSE;
0034 END_IF;
0035 IF S7 = 1
0036     THEN L7 := TRUE;
0037     ELSE L7 := FALSE;
0038 END_IF;
0039 IF S8 = 1
0040     THEN L8 := TRUE;
0041     ELSE L8 := FALSE;
0042 END_IF;
0043
0044
    
```

PLC\_PRG LD (8): L8 AT %QX0.7  
 PLC\_PRG (37): L8 AT %QX0.7  
 PLC\_PRG (38): L8 AT %QX0.7  
 Current download: 1205 bytes

Lin: 17, Col: 19 | ONLINE | RV | READ

56. ábra: Másoló program ST programnyelven

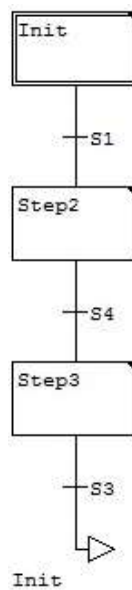
### 6.3 Feladatok megoldása SFC programnyelven

Az SFC programnyelv jelentése „Sequential Function Chart”.

A programnyelv lényege, hogy sorrendben hajtódnak végre az egyes feltételek függvényében az utasítások.

#### 6.3.1 Munkahenger ki- és visszafutásának vezérlése SFC programnyelven

Az első feladat a 6.1.1. alfejezetben már ismertetésre került, azaz a munkahenger ki- és visszafutásának egymás utáni vezérlése történik meg. A munkahenger vezérlésére egy bistabil útváltó szelep áll rendelkezésre. Az elkészült SFC program az 57. ábrán látható.



57. ábra: Munkahenger ki- és visszafutásának egymás utáni vezérlése

Az „Init” lépésen belül meg kell írni a program elejére szükséges utasításokat, amely jelen esetben a bistabil szelep beállítását hajtja végre. A blokkra kétszer rákattintva ki lehet választani a lépés alprogramjának programnyelvét, és abban megírni a szükséges utasításokat. Bistabil szelep vezérléshez célszerű az ST programnyelvet használni. A megírt „Init” lépést az 58. ábra mutatja.

```

    H5 := FALSE;
    H6 := FALSE;
  
```

58. ábra: Munkahenger vezérlésénél kezdeti feltételek megadása

A „Step2” és „Step3” feltételek megadását az 59. ábra részletezi.

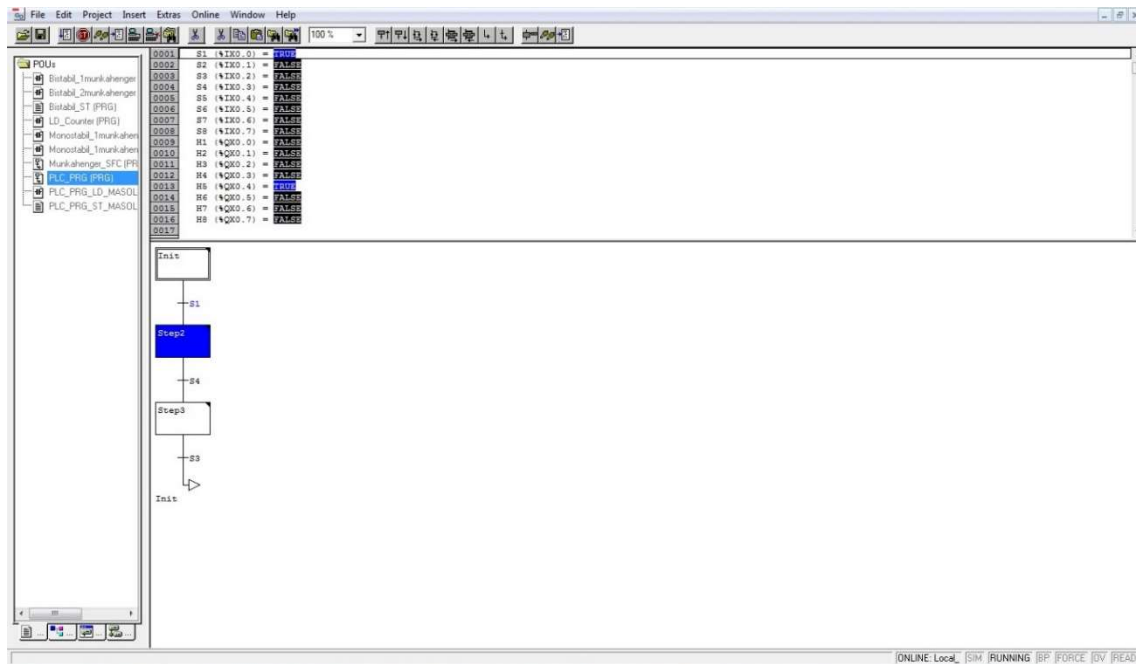
<pre>         H5 := TRUE;         H6 := FALSE;       </pre>	<pre>         H5 := FALSE;         H6 := TRUE;       </pre>
---	---

„Step2” lépés

„Step3” lépés

59. ábra: Munkahenger vezérlésénél további lépések

A program futtatás közbeni állapotát az 60. ábrán figyelhetjük meg.



60. ábra: Munkahenger vezérléséhez megírt program futtatás közben.

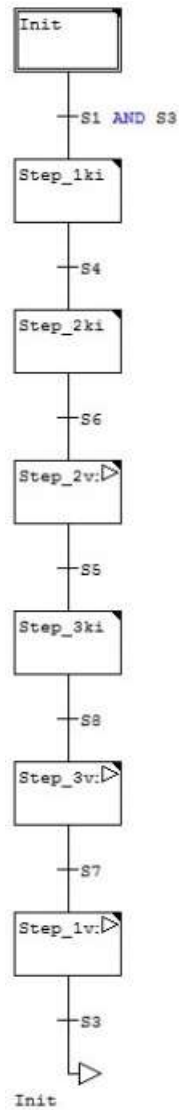
### 6.3.1 Három munkahenger vezérlése SFC programnyelven

Ebben a feladatban egy 3 munkahengeres kapcsolás vezérlését SFC programnyelv segítségével valósítjuk meg. Az SFC programnyelv előnye a LD programnyelvvvel szemben, hogy sorrendi működésű, így kiküszöbölhető a párhuzamosságból adódó működési rendellenességek. A szimulációs tábla a 37. ábrán látszik.

A gyakorlati feladat egy lemez két lépcsőben való hajlítását tartalmazza. A lépések sorrendben a következők:

1. Leszorítás az 1. munkahenger segítségével
2. Előhajlítás a 2. munkahenger segítségével
3. Előhajlító 2. munkahenger visszafutása
4. Készrehajlítás a 3. munkahenger segítségével
5. Készrehajlító 2. munkahenger visszafutása
6. Leszorító 1. munkahenger visszafutása

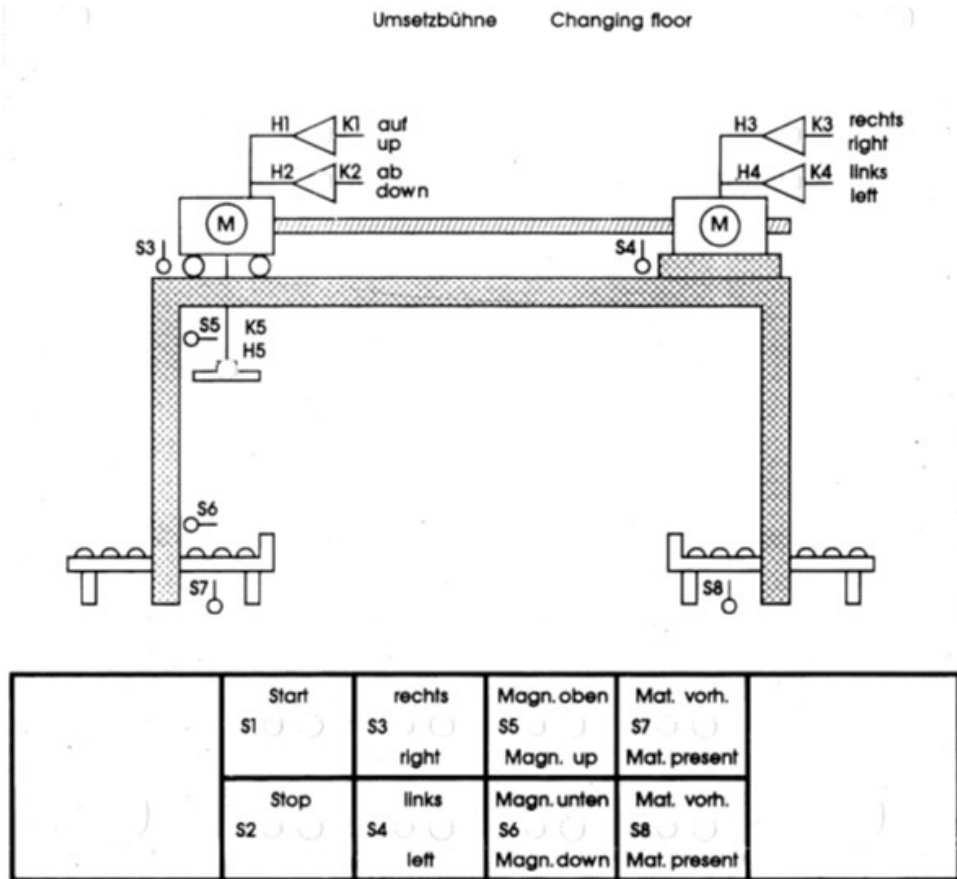
A feladat SFC programját a 61. ábra mutatja.



61. ábra: 3 munkahengeres kapcsolás SFC programnyelven

### 6.3.2 A 16. szimulációs tábla megoldása SFC programnyelven

Következő feladat a szimulációs tábla 16. feladat kapcsolásának vezérlése SFC programnyelv segítségével. A szimulációs tábla kinézete a 62. ábrán látszik.

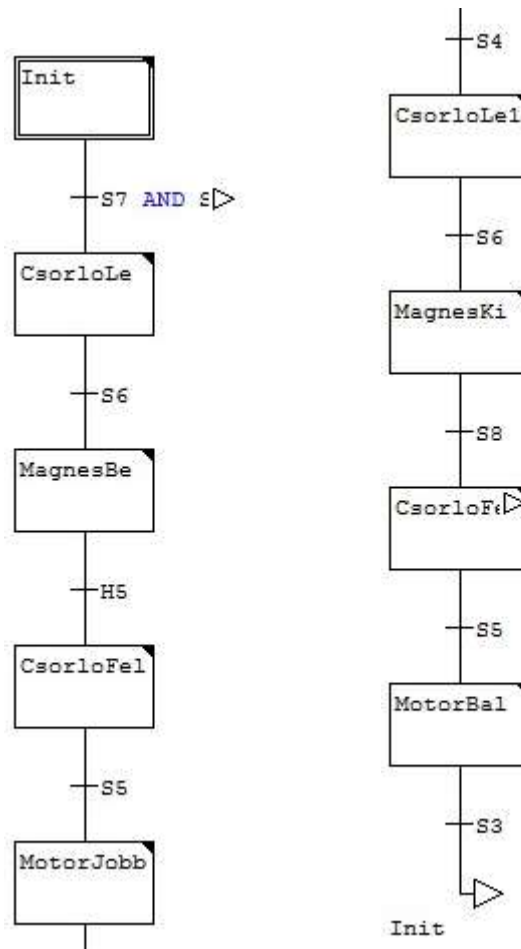


62. ábra: Szimulációs tábla 16. feladata

A feladat egy munkadarabnak bal oldalról jobb oldalra történő áthelyezése. A lépések sorrendben a következők:

1. Munkadarab megérkezése a bal oldalon, amelyet az S7 érzékel
2. Emelő csörlő lemegy, S6 érzékeli
3. Mágnes bekapcsol, azaz H5 aktív lesz
4. Emelő csörlő felmegy, S5 érzékeli
5. Vízszintes mozgató csörlő jobbra megy, S4 érzékeli
6. Emelő csörlő lemegy, S6 érzékeli
7. Mágnes kikapcsol, azaz H5 lekapcsol
8. Emelő csörlő felmegy, S5 érzékeli
9. Visszafut a kiinduló helyzetbe a rendszer, S3 érzékeli

A megoldott feladatot SFC programnyelvben megírva a 63. ábra mutatja.

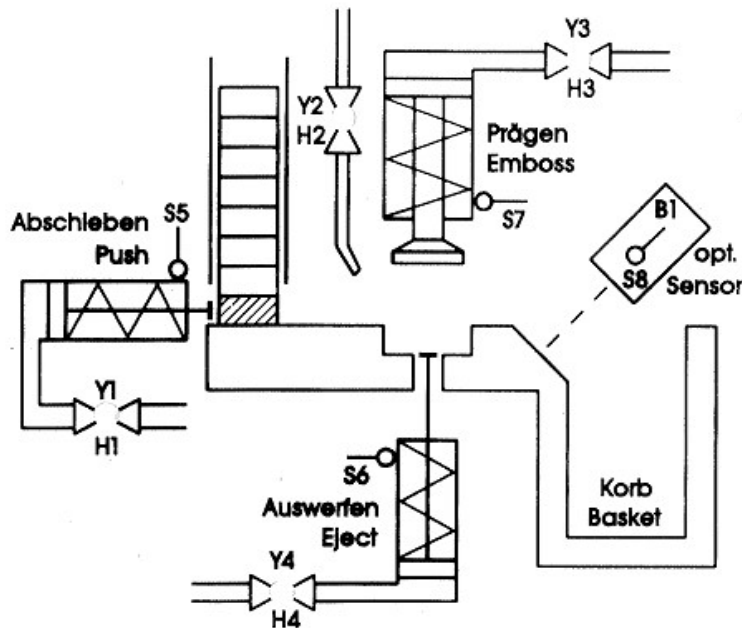


63. ábra: Szimulációs tábla 16. feladat kapcsolása SFC programnyelven

### 6.3.3 A 17. szimulációs tábla megoldása SFC programnyelven

Ez a feladat a szimulációs tábla 17. feladat kapcsolásának vezérlését valósítja meg SFC programnyelven. A szimulációs táblát a 64. ábra szemlélteti.

Prägemaschine - Automatik    Embossing machine - automatic



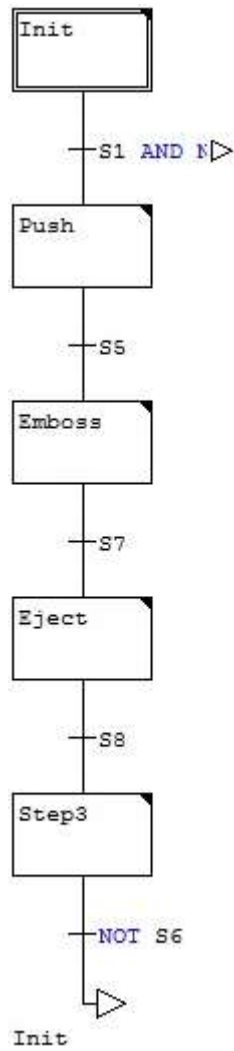
	Start S1	Hand S3	Abschleiben S5 Push	Prägen S7 Emboss	
	Stop S2	Automatic S4	Auswerfen S6 Eject	opt. Sensor S8 B1	

64. ábra: Szimulációs tábla 17. feladata

A feladat egy munkadarab préselése. A lépések sorrendben a következők:

1. Nyomó munkahenger a munkatérbe tolja a munkadarabot, S5 érzékel
2. Préselés elindul, S7 érzékel, majd időzítés után visszamegy
3. Alulról munkahenger feltolja a munkadarabot, S6 érzékel
4. H2 kifúvató a munkatérből eltávolítja a munkadarabot
5. S8 optikai szenzor érzékel
6. Visszafut kiinduló helyzetbe a rendszer

A megoldott feladatot SFC programnyelvben megírva a 65. ábra mutatja.

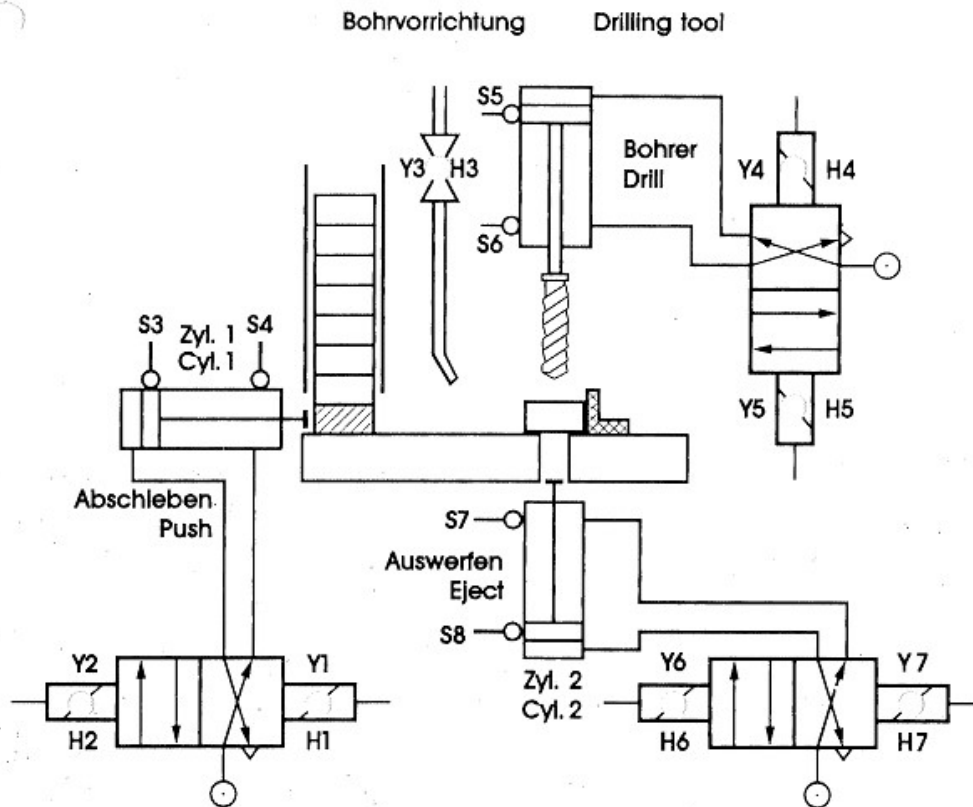


65. ábra: Szimulációs tábla 17. feladat kapcsolása SFC programnyelven

#### 6.3.4 A 18. szimulációs tábla megoldása SFC programnyelven

A jelenlegi feladat a 18. feladat kapcsolásának vezérlése SFC programnyelv segítségével. A szimulációs táblát a 66. ábra illusztrálja.





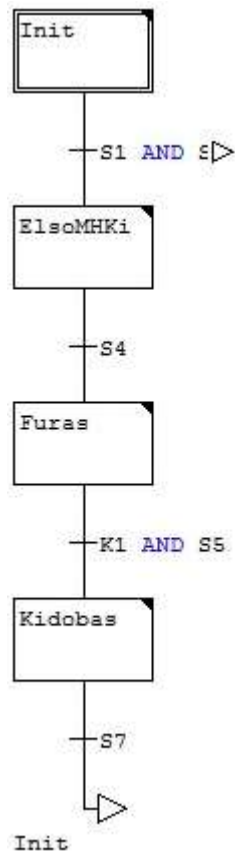
Start	Zyl. 1 links S3 <input type="radio"/> <input type="radio"/>	Bohrer oben S5 <input type="radio"/> <input type="radio"/>	Zyl. 2 oben S7 <input type="radio"/> <input type="radio"/>
S1 <input type="radio"/> <input type="radio"/>	Cyl.1 left	Drill up	Cyl. 2 up
Stop	Zyl. 1 rechts S4 <input type="radio"/> <input type="radio"/>	Bohrer unten S6 <input type="radio"/> <input type="radio"/>	Zyl. 2 unten S8 <input type="radio"/> <input type="radio"/>
S2 <input type="radio"/> <input type="radio"/>	Cyl.1 right	Drill down	Cyl. 2 down

66. ábra: Gyakorlótábla 18. feladata

A feladat egy munkadarab fúrása. A lépések sorrendben a következők:

1. Nyomó munkahenger a munkatérbe tolja a munkadarabot, S4 érzékel
2. H3 hűtés-kenés bekapcsol, kitoló munkahenger visszafut
3. Fúró munkahenger 3x egymás után le-felmegy, S5 és S6 érzékelők által
4. H3 hűtés-kenés kikapcsol
5. Alulról munkahenger kitolja a munkadarabot, S7 érzékel
6. Visszafut kiinduló helyzetbe a rendszer

A megvalósított feladatot SFC programnyelvben megírva a 67. ábra mutatja.



67. ábra: Szimulációs tábla 18. feladat kapcsolása SFC programnyelven

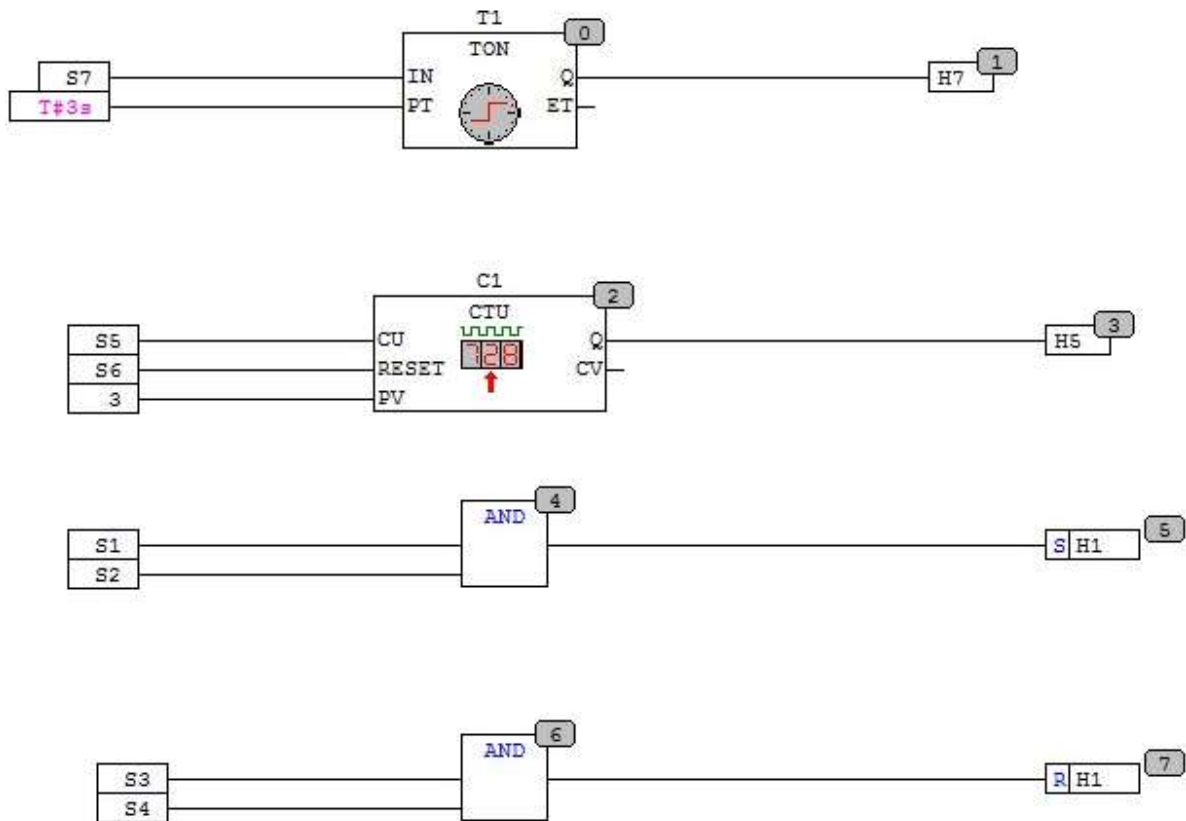
#### 6.4 Feladat megoldás CFC programnyelven

A CFC programnyelv jelentése „Continuous Function Chart”.

A programnyelv lényege, hogy Matlab-szerűen lehet programozni, és párhuzamosan hajtódnak végre az egyes feltételek függvényében az egyes utasítások.

A programozást először a „Box”-ok létrehozásával célszerű kezdeni, majd ehhez a „Box”-hoz hozzárendelni az egyes ki- és bemeneteket.

A feladatban beépítésre kerül egy időzítő, egy számláló és két darab ÉS kapu a 69. ábrán látható módon.



68. ábra: Egyszerű feladat megoldása CFC programnyelven

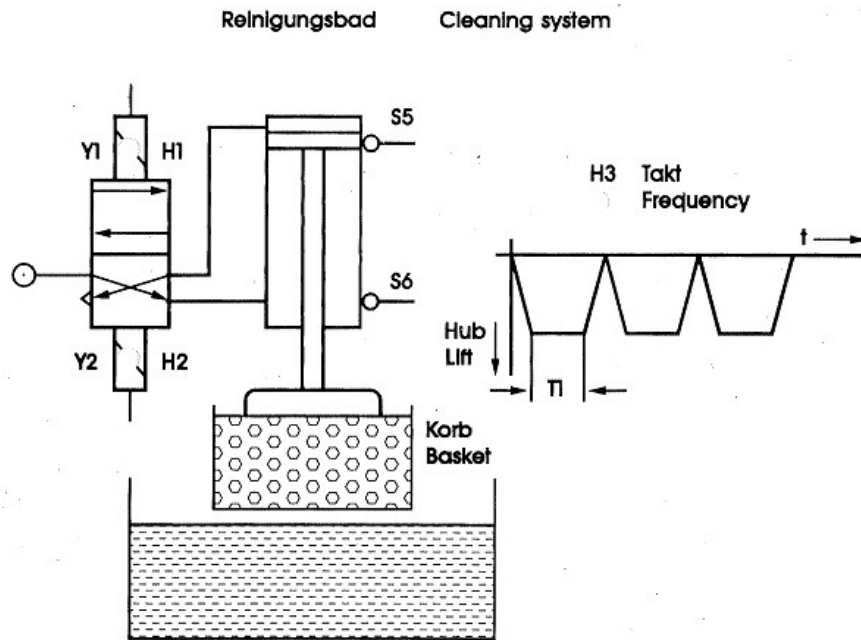
## 6.5 Feladat megoldás FBD programnyelven

Az FBD programnyelv jelentése „Function Block Diagram”.

A programnyelv lényege, hogy ötvözi az LD és a CFC programnyelv tulajdonságait. Így Matlab-szerűen lehet programozni, és létraszerűen párhuzamosan hajtódnak végre az egyes sorok.

Az FBD programnyelv gyakorlására a szimulációs tábla 13. feladat kapcsolását használjuk fel.

A szimulációs táblát a 69. ábra illusztrálja.

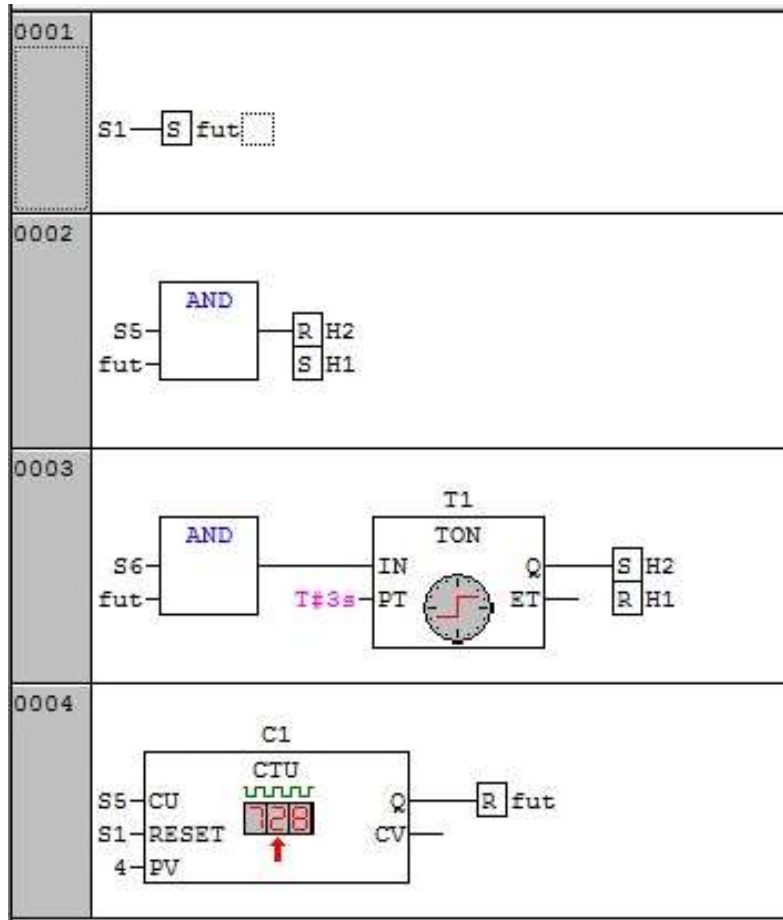


	Start S1	Hand S3	Korb oben S5 Basket up	S7
	Stop S2	Autom. S4	Korb unten S6 Basket down	S8

69. ábra: Gyakorlótábla 13. feladata

A feladat egy tisztítási folyamat megvalósítása, egy kosár folyadékba való merítése háromszor egymás után, ahogy azt a 69. ábra jobb felső részében levő ütemdiagram mutatja.

Az FBD programnyelven megírt feladatot a 70. ábra mutatja.

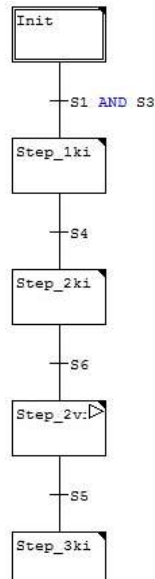


70. ábra: Gyakorlótábla 13. feladat kapcsolása FBD programnyelven

## 7. Vizualizáció

Amennyiben PLC hardveregység nem áll rendelkezésre, és szimulálni szeretnénk a megírt PLC programot, vizualizációt lehet használni. Ennek lényege, hogy a PLC be- és kimeneteit egy grafikus felületen jeleníthetők meg, és így tesztelhető a megírt PLC program.

A vizualizáció használatához elő kell venni a 6.3.1. alfejezetben ismertetett 19. szimulációs tábla 71. ábra szerinti megoldását. A feladat egy munkadarab hajlítása 3 munkahenger segítségével.



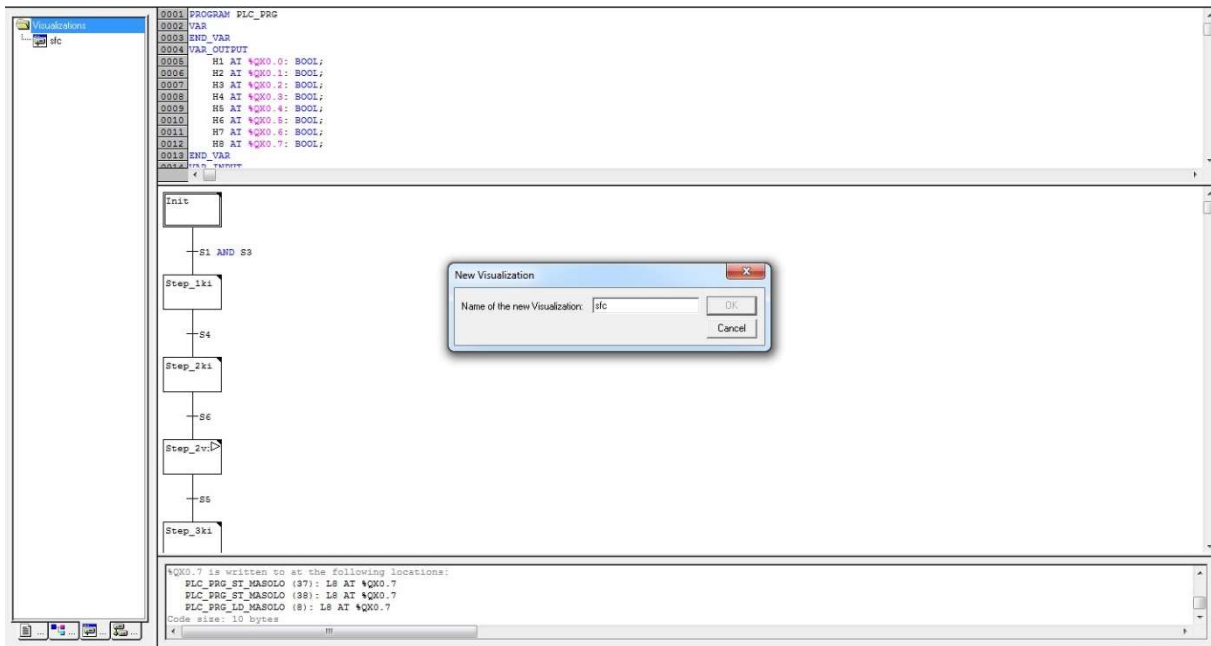
71. ábra: Szimulációs tábla 19. feladatának megoldása SFC programnyelven

A vizualizációs feladat létrehozásához át kell lépni az ablak bal oldal alján található 3. ikonra (lásd 72. ábra).



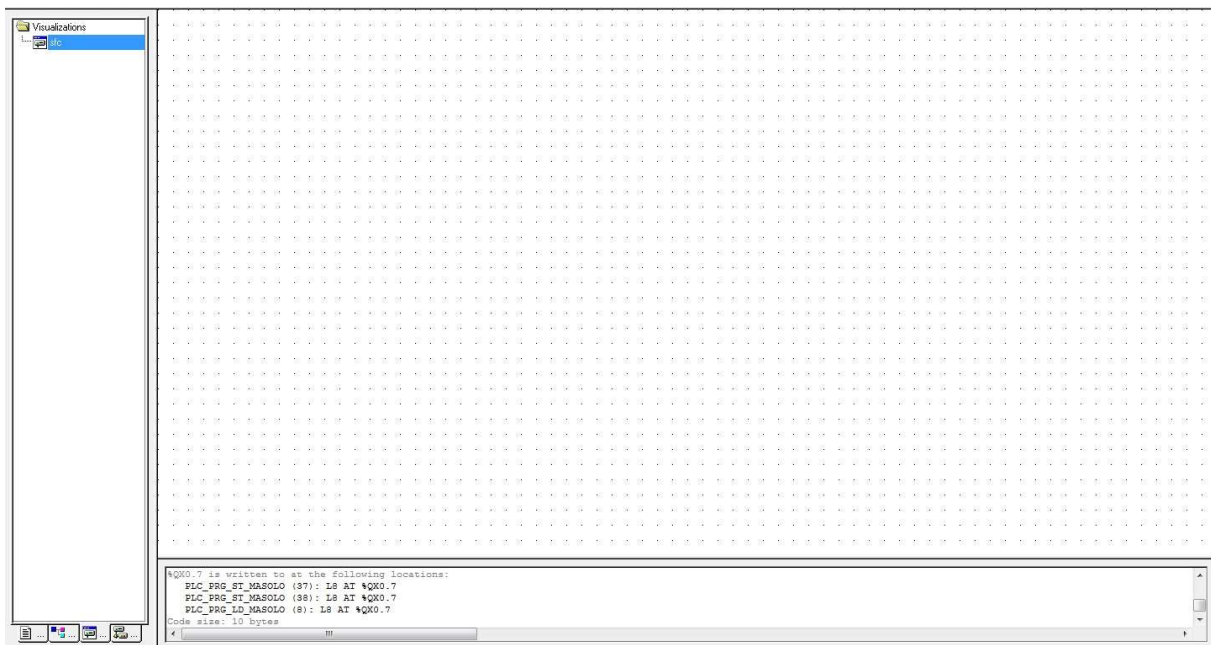
72. ábra: Vizualizációs ikon

Itt egy új vizualizációs feladat létrehozását a 73. ábra mutatja.



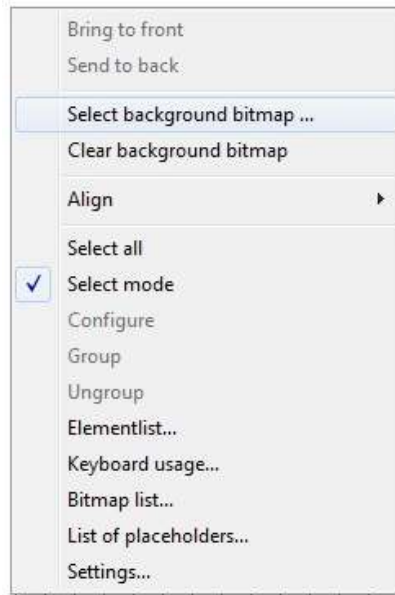
73. ábra: Új vizualizációs feladat létrehozása

Az új feladat létrehozása után egy pontozott rácsos felület jelenik meg a 74. ábrán látható módon.



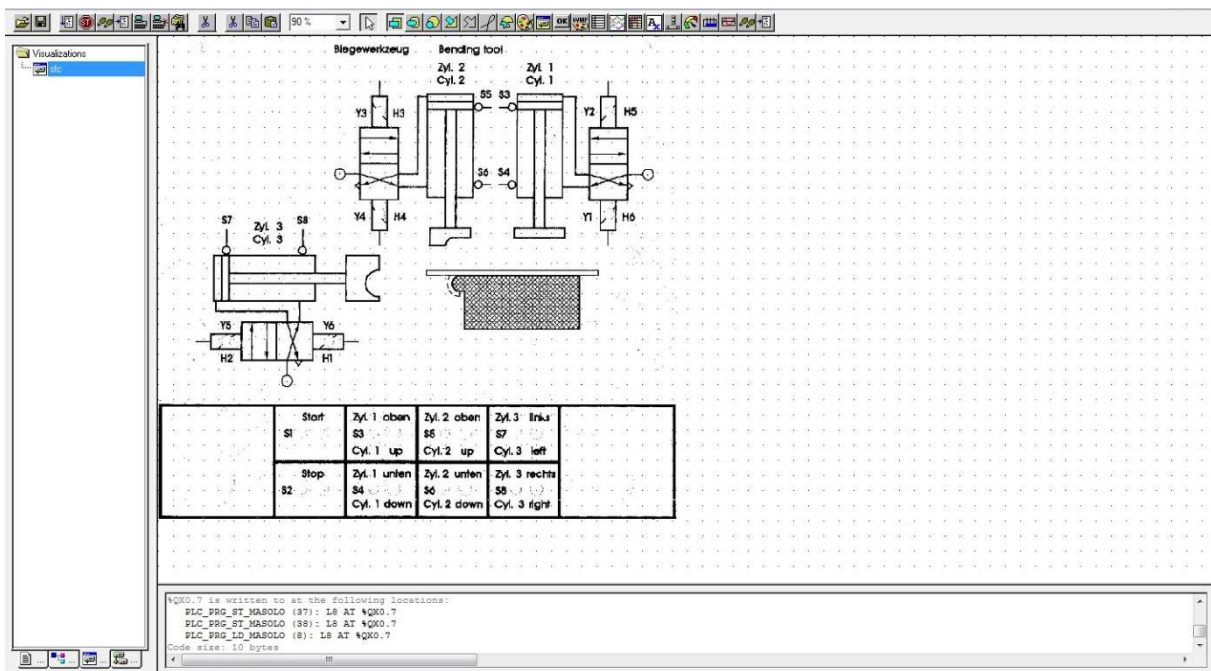
74. ábra: Új vizualizációs feladat létrehozása után

Ezután háttérként hozzá kell adni a szimulációs tábla 19. feladatának beszkenelt ábráját. Ezt a „jobb klikk/Select background bitmap ...” menüponttal hajtuk végre (lásd 75. ábra).



75. ábra: Háttér hozzáadása

A háttér hozzáadása utáni állapotot a 76. ábra illusztrálja.

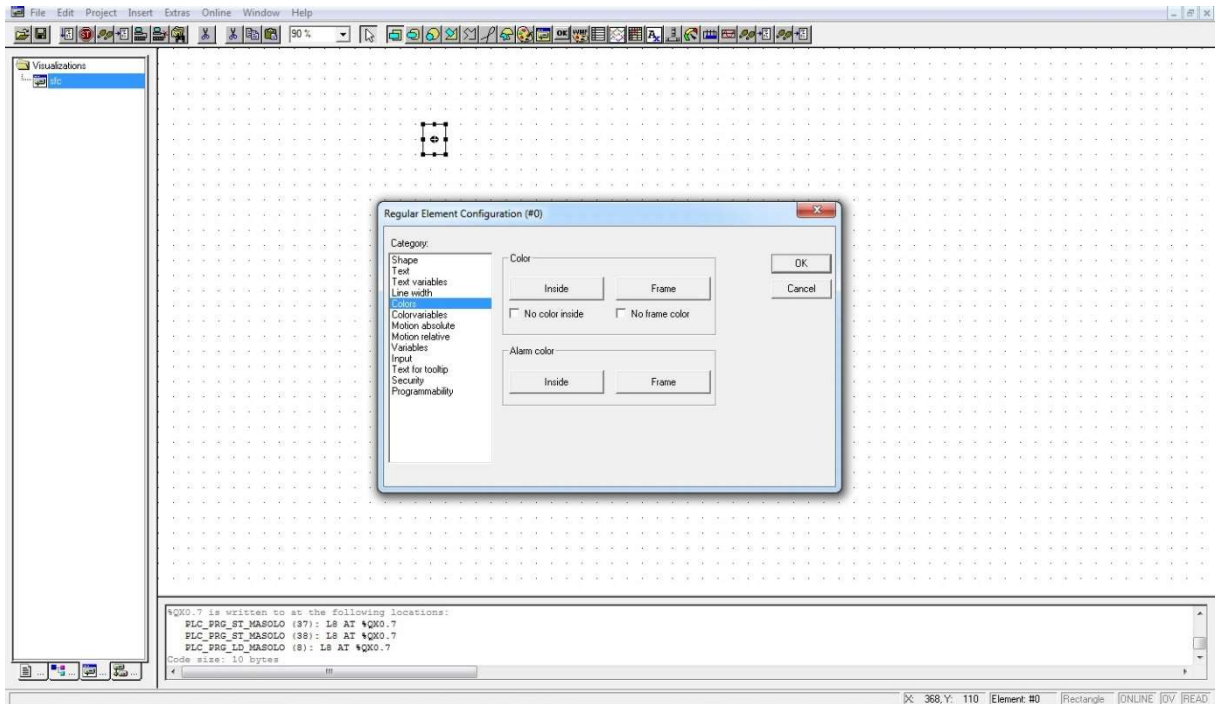


76. ábra: Háttér hozzáadása után

A háttér hozzáadása után következik a bemenetek és kimenetek kialakítása a felületen. A kimeneteket valamilyen síkidommal célszerű megjeleníteni. A kimenetek két állását a síkidom két különböző színe jelzi. A jelenlegi feladat során a kimenet alakja egy kör lesz, valamint fekete színű a kikapcsolt, és piros színű a bekapcsolt állapot.

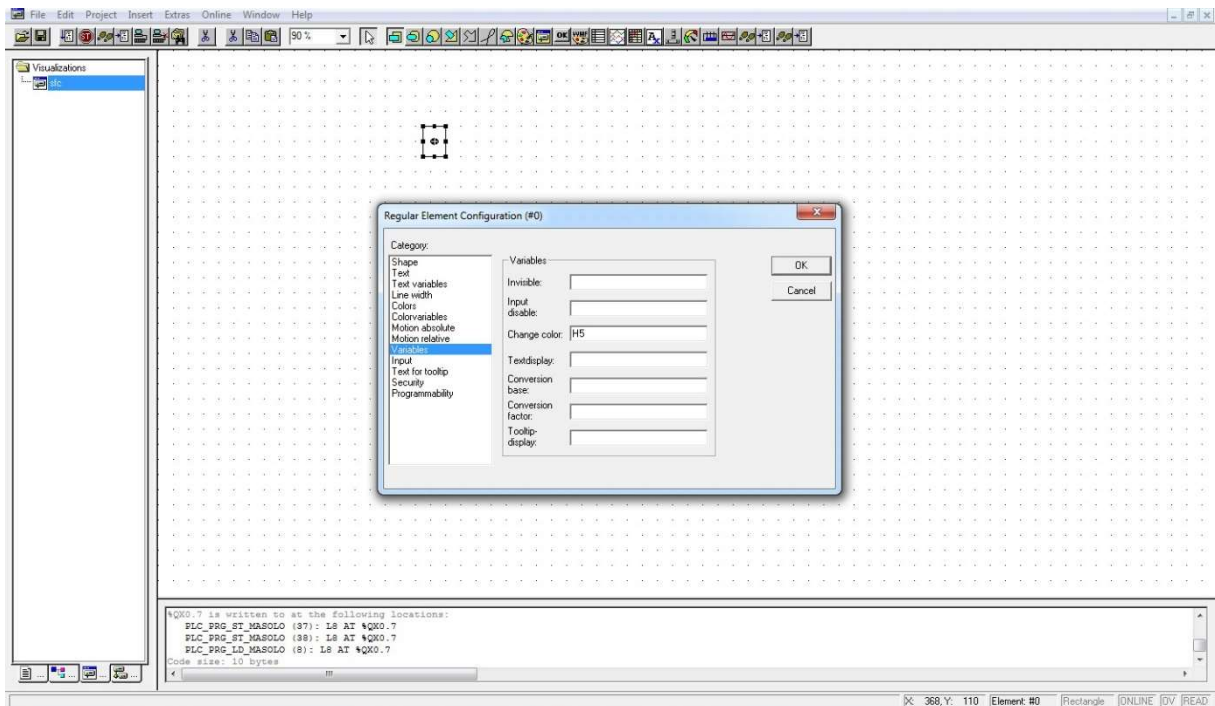
A kimeneti síkidomra való kattintás után a „Colors” fülnél történik a színek beállítása (lásd 77. ábra).





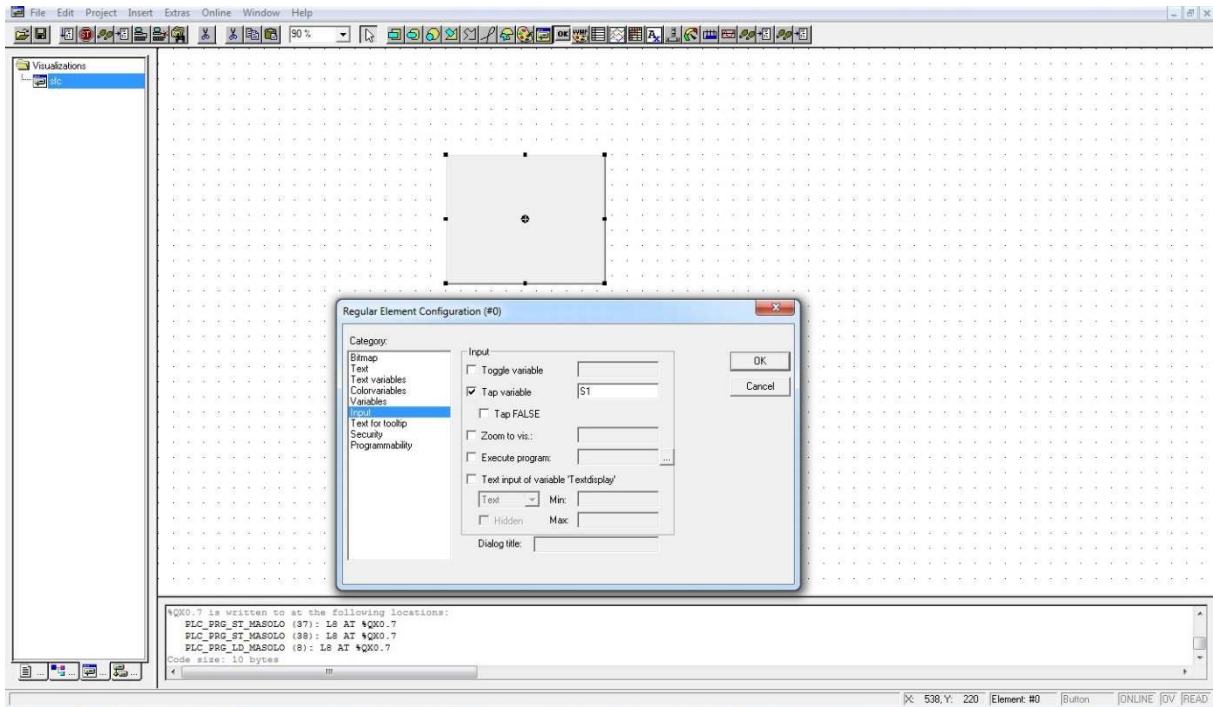
77. ábra: Kimeneti színek beállítása

Ezután következnek a kimeneti színek változtatásához szükséges változó hozzárendelése (lásd 78. ábra).



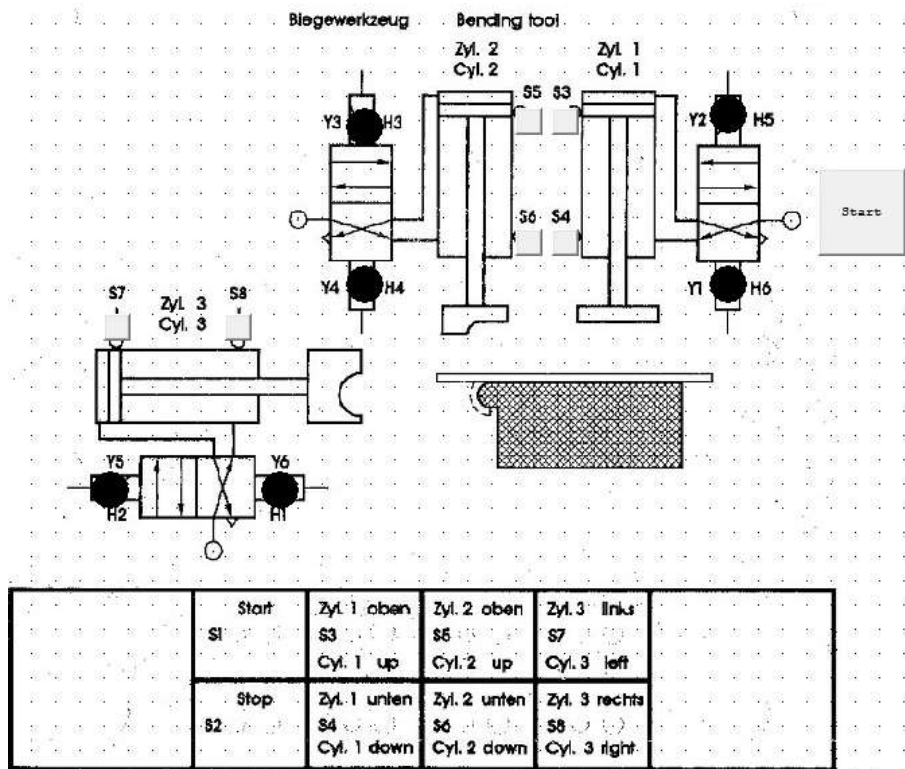
78. ábra: Kimeneti színek változásához változó rendelése

Bemenet esetén nincs szükség színváltoztatásra, csak egy bemeneti változó beállítására. Itt kétféle lehetőség nyílik: az egyik egy nyomógomb („Toggle Variable”) vagy a másik egy kapcsoló („Tap Variable”) szimulálása. A beállítás az „Input” fülnél történhet meg a 79. ábra alapján.



79. ábra: Bemenet beállítása

A bemenet és kimenet beállíthatóságának ismeretében már el lehet készíteni a feladatot (80. ábra).



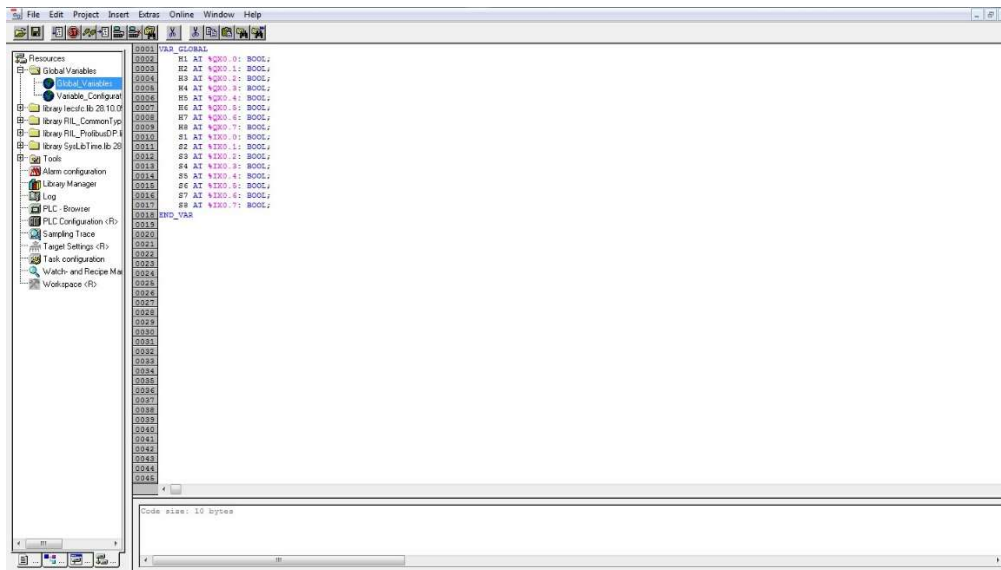
80. ábra: Vizualizációs feladat elkészítése után

A futtatáshoz még két lépésre van szükség. Az első a változók globálissá tétele, a másik a szimulációs mód aktiválása.

A globális változókra azért van szükség, mert a PLC program során deklarált változók csak az adott programra érvényesek, a vizualizációra már nem terjednek ki. A globális változók az ablak bal oldali részénél a 4. ikonnál érhetőek el (lásd 81. ábra). Ide be kell írni a be- és kimeneti változókat (82. ábra).

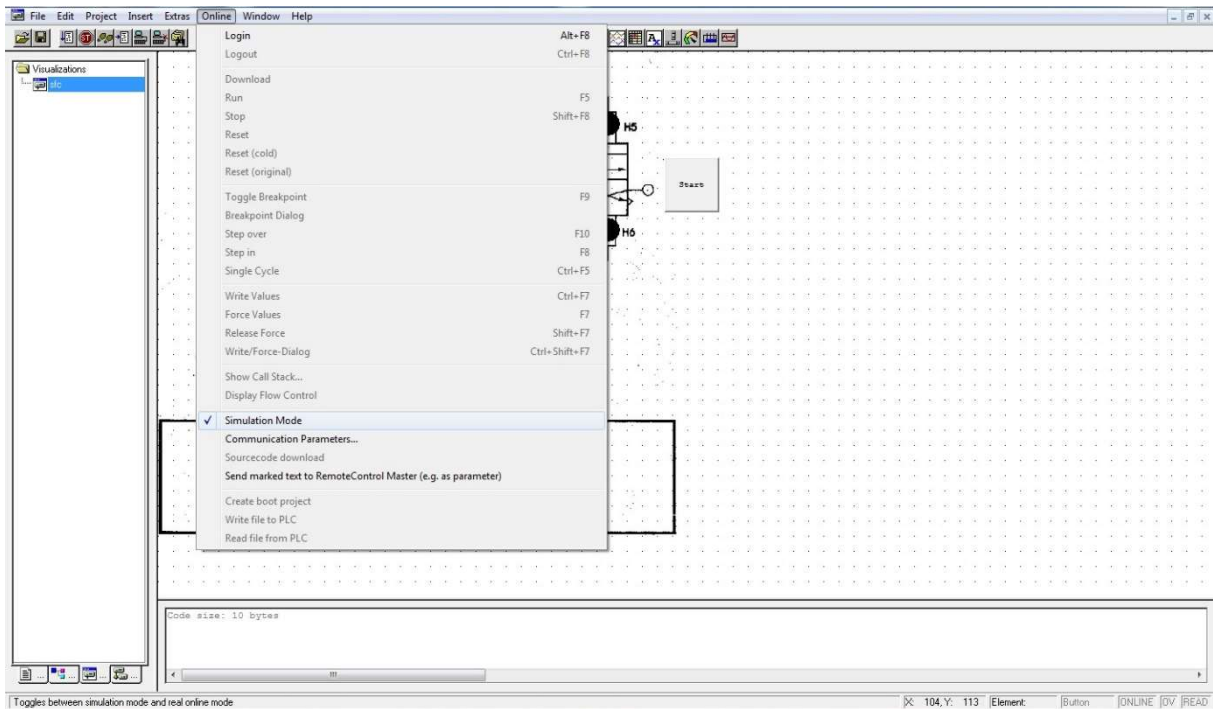


81. ábra: Globális változók állításához szükséges ikon



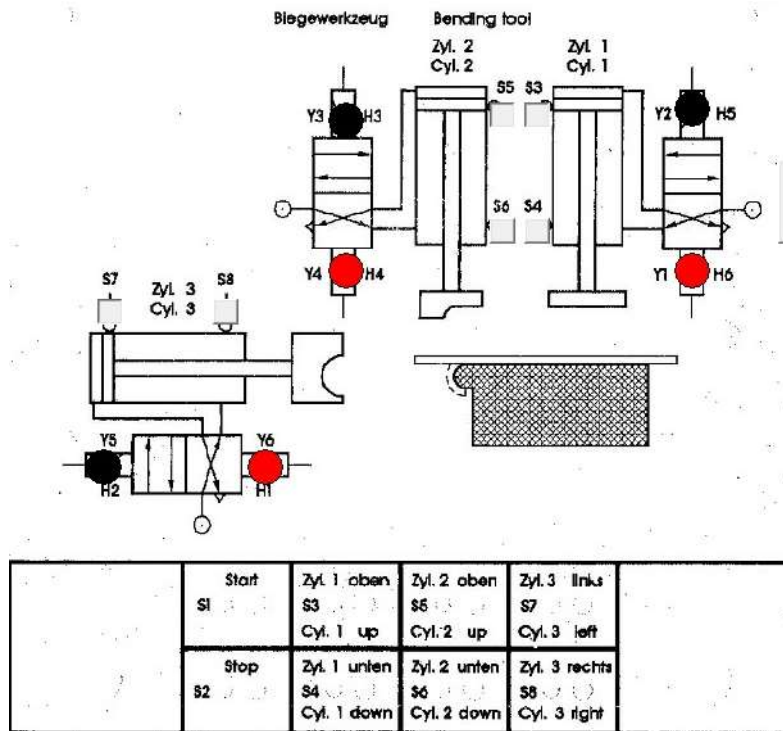
82. ábra: Globális változók beállítása

A szimulációs mód beállítása felül az „Online/Simulation Mode” menüponttal lehetséges a 83. ábrának megfelelően.



83. ábra: „Simulation Mode” aktiválása

A futtatás utáni alaphelyzeti állapot a 84. ábrán figyelhető meg.



84. ábra: Vizualizációs feladat futtatása után

## **Irodalomjegyzék**

- [1]. Dr. Ajtonyi István – Dr. Gyuricza István: **Programozható irányítóberendezések, hálózatok és rendszerek**, Műszaki Könyvkiadó Kft., 2007
- [2]. Rexroth Bosch Group: **Rexroth IndraLogic L40 System Description**, R911309372, Operating and Programming Guide, 2004
- [3]. Rexroth Bosch Group: **Rexroth IndraWorks Engineering 07VRS**, R911320741, Operating and Programming Guide, 2006
- [4]. Rexroth Bosch Group: **PLC Programming with Rexroth IndraLogic 1.0**, R911305036, Operating and Programming Guide, 2004
- [5]. Moeller: **Bevezető a MOELLER easy Soft CoDeSys / XSoft PLC-program fejlesztő használatához**