

SZIMULÁCIÓS SZOFTVERRENDSZER ALKALMAZÁSA A PLC TECHNOLÓGIA OKTATÁSÁBAN

Koszár András

PLC szakmérnök, mérnök informatikus, mérnöktanár, szakvizsgázott pedagógus
Szombathelyi Műszaki SZC III. Béla Szakgimnáziuma és Szakközépiskolája
9970 Szentgotthárd, Honvéd u. 10.
mobil: +36 30/824-8867, email: akoszar@sztgnet.hu, andras.koszar@gmail.com

Összefoglalás

Manapság az automatizálási szakterületen is megjelentek a korszerű számítógépes szoftverrendszerek különféle alkalmazásai. Immár nem csak fejlesztőkörnyezetként tekinthetünk ezekre, hanem széleskörű szakmai igényeket is kielégítenek a modellezési és szimulációs funkcionalitásaikkal, melyeket az oktatásban is hatékonyan fel tudunk használni.

A szakképzésben és felnőttképzésben dolgozó elméleti és gyakorlati oktatóként már korábban megfogalmazódott bennem a kíváncsi, hogy rátaláljak olyan szoftveres és hardveres technológiai lehetőségekre, melyek segítségével hatékonyan és szemléletesen modellezhetők az iparban használt berendezések lényegi komponensei. Természetesen fontos volt, hogy mindezt a drága mechatronikai rendszerek komolyabb károsodásának következménye nélkül lehessen megtenni a tanórákon, tanfolyamokon és vállalati tréningeken.

Írásomban ez irányú szakmódszertani tapasztalataimat szeretném megosztani, és egy konkrét változatát is felvázolom.

Kulcsszavak: PLC, szimuláció, hardver, szoftver

Abstract

In the scope of automation various applications of advanced computerized software systems have appeared. Not only are they regarded as development environment, but they meet professional requirements with their simulation and modelling functions and can be used effectively during the education.

I have been teaching theory and professional practice in vocational training, in-service training and adult education for years. I have always dealt with the demand for finding software and hardware technologies, and with the help of them the components of systems used in industry can be modelled or demonstrated without causing any damage to expensive mechatronic systems during classes, courses and in-service trainings.

I intend to share my professional-methodological experience, and give a concrete example of implementation.

Keywords: PLC, simulation, hardware, software

1 Bevezetés

Munkahelyemen több éve oktatom az automatikai technikus, valamint mechatronikai technikus tanulóknak a PLC technológiával összefüggő elméleti és gyakorlati tantárgyakat. A diákok munkaerőpiacon való elhelyezkedési esélyeit nagyban javítja, ha az iskolákban és a gyakorlati képzőhelyeken az iparban bevált és alkalmazott technológiák vonatkozásában a lehető legtöbb ismeretet és tapasztalatot szerzik meg. Ehhez korszerű, ugyanakkor költséges tárgyi eszköz feltételrendszert szükséges biztosítani és fenntartani (Koszár 2014). Napjainkban gyakran felmerülő probléma a szakképzési rendszerünkkel összefüggésben, hogy a gyakorlati oktatáshoz szükséges, csúcstechnológiát képviselő eszközök nem állnak rendelkezésre az optimális mennyiségben, illetve meghibásodásuk esetén nehéz a pótlásukra forrásokat biztosítani.

A csúcstechnika oktatásáról szóló szakmódszertani publikációk nagyon kevés számban állnak rendelkezésre. Ez viszonylag érthető, hiszen a gyorsan változó technikai-technológiai fejlődés követésére a nyomtatott könyvek, jegyzetek nem képesek. Ezen enyhítenek az elektronikus úton elérhető dokumentumok, illetve publikációk. Témánk szempontjából meghatározóan fontos szakirodalom a TÁMOP-4.1.2 B2 keretében kidolgozott *Szaktananyagfejlesztés az elektronikus villamos szakmacsoportos mérnökök számára* c. elektronikus könyv, amelyben találhatunk példát és alapelveket is (Lükő – Molnár 2015).

A fentiekből következően a gyakorlati foglalkozásokon lényegi szerepet kapnak az innovatív szimulációs technológiák. Ezek a számítógéppel támogatott alkalmazások képesek interaktívan modellezni az összetett irányítástechnikai rendszerek elemeinek működését.

Az alternatívák számbavételekor a következő alapvető lehetőségek kínálkoznak:

- szoftveres implementációk;
- hardveralapú szimulációs rendszerek.

A szoftveralapú rendszerek esetén kulcsfontosságú a költséghatékony, szakmai igényeknek megfelelő számítógépes szoftverek kiválasztása. Valós hardveregységeket (pl.: valós PLC CPU-t) tartalmazó összeállítás esetén az automatizált rendszer komplex összefüggéseit is reálisan modellezhetjük pl.: szenzorok és aktuátorok valós reakciói, a működés időadatai, a teljesítőképesség vizsgálata, stb. A különféle szimulációs aspektusok lényegi összefüggéseit az 1. sz. táblázat ismerteti.

Szimuláció aspektusa	Alacsony költségek	Lefutási paraméterek manipulálhatósága	Iparban bevált megoldások alkalmazhatósága	Valós PLC működés	Valós reakciók a működtetett komponens-től
Szoftveres (számítógépes szoftver-rendszer)					
Szoftveres (PLC hardver, szoftveres működtetett komponens)					
Hardveres (PLC hardver, hardveres működtetett berendezés)					

1. sz. táblázat: A szimulációs aspektusok összefüggései (Forrás: saját szerkesztés)

Az alábbi táblázatban (2. sz. táblázat) összefoglaltam, hogy mely szakmák és témakörök tanításánál használhatók fel az előzőekben felvázolt szimulációs alkalmazások.

Szakma	Követelménymodul	Tantárgyak
Automatikai technikus (OKJ 54 523 01)	10001-16 Ipari folyamatok irányítása PLC-vel	PLC ismeretek PLC programozási gyakorlat
Automatikai berendezés karbantartó (OKJ 35 523 01)		
PLC programozó (OKJ 51 523 01)		
Mechatronikai technikus (OKJ 54 523 04)	11584-16 Vezérléstechnikai alapok	PLC alkalmazása gyakorlat
	10191-12 Mechatronikai villamos feladatok	Mechatronikai villamos feladatok Mechatronikai villamos feladatok gyakorlat
Mechatronikus karbantartó	10019-12	Programozás

(OKJ 34 523 01)	Irányítás, programozás	<i>Programozási gyakorlat</i>
-----------------	------------------------	-------------------------------

2. sz. táblázat: *A szimulációs rendszerek alkalmazhatósága a szakképzésben* (Forrás: saját szerkesztés; adatok: 29/2016. (VIII. 26.) NGM rendelet, 30/2016. (VIII. 31.) NGM rendelet)

A következőkben ismertetem a tisztán szoftveres szimulációs rendszer egy lehetséges változatának fő összetevőit és sajátosságait. Majd szakmódszertani szempontból vizsgálva részletezem a szakmai gyakorlati tanórákon javasolt lépések sorrendiségét és tartalmi elemeit egy konkrét szimulációs feladat kialakításának tükrében.

2 A szimulációs környezet komponensei

A szoftveres szimulációs aspektust olyan esetekben célszerű választani, amikor már a tervezési, illetve a kódolási tevékenységek során, a számítógépünkön dolgozva is információkat szeretnénk kapni a PLC-vel vezérelt ipari folyamat dinamizmusáról, működéséről. Ezeket a technikákat alkalmazva elsősorban a rendszer adatáramlási és adatfeldolgozási állapotai modellezhetők részletesen. Előnyt jelent továbbá, hogy a kritikus működtetési szakaszok kimerevíthetők, lépésekre bonthatók, illetve a kritikus adatokat tároló változók nyomon követhetők a fejlesztés tesztelési fázisaiban. Ezen funkcionalitások felhasználásával hatékonyan támogathatjuk a didaktikai folyamatokat is. Virtuális tanulási környezetet alakíthatunk ki a diákoknak, melyben programozottan biztosíthatjuk a feladatok megoldását és ellenőrzését segítő konzisztens adatok és információk folyamatos rendelkezésre állását.

2.1 A CoDeSys fejlesztőkörnyezet bemutatása

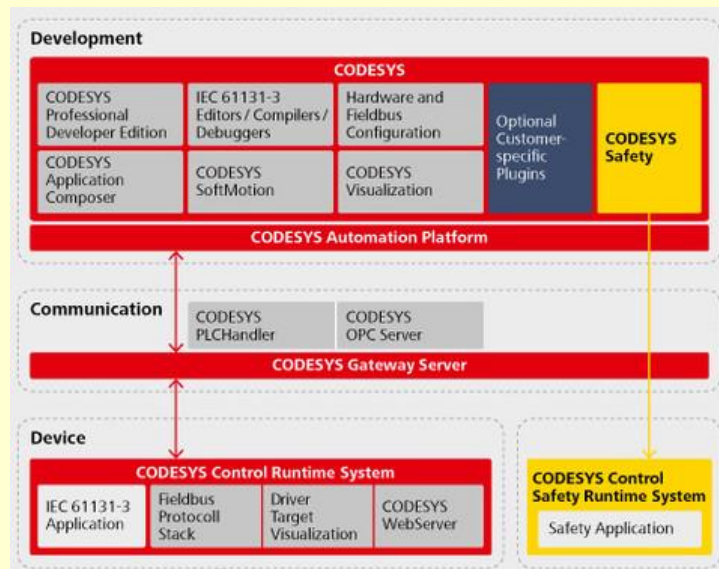
A *CoDeSys (Controller Development System)* a *3S – Smart Software Solutions* cég által létrehozott PLC-s vezérlő és programfejlesztő környezet, mely számos gyártó által támogatott. A felhasználó cégek között van az *ABB*, a *FESTO*, a *Mitsubishi Electric*, a *Moeller*, a *Schneider Electric* és még számos PLC gyártó.

A programcsomag két felhasználási komponensre tagolható:

- a programfejlesztői környezet, mely a felhasználói alkalmazások integrált tervezését és kódolását támogatja, valamint
- a runtime összetevő, mely biztosítja a *CoDeSys* végrehajtási környezet működését az intelligens vezérlőkön.

A szoftverrendszer funkciók szerinti felosztása alapján három réteg definiálható (CoDeSys 2015):

- a fejlesztői réteg,
- a kommunikációs réteg, valamint
- az eszköz réteg (lásd 2.1. ábra).



2.1. ábra: A CoDeSys rétegei (Forrás: www.codesys.com – CoDeSys Worldwide, The System; 2016.09.25)

A fejlesztői réteg tartalmazza a PLC programfejlesztői eszközeit. A különböző IEC szerkesztők könnyen kezelhető eszközt nyújtanak programjaink implementálásához. Számos vezérlőhöz léteznek fordítóprogramok (*compiler-ek*), melyek egy univerzális fejlesztői környezetté teszik a CoDeSys-t. A szoftverfejlesztés mind *online*, mind *offline* módon történhet, köszönhetően a magas szintű szimulációs módnak és a nyomkövetési (*debugger*) funkcióknak. A fejlesztői környezet részeként beállíthatjuk a PLC megfelelő hardver paramétereit, különböző moduljainak eszköz-specifikus konfigurációit. Számos ipari kommunikációs megoldást is támogat, amelyek minden paramétere könnyen beállítható. A konfigurációs és programozási funkciók mellett fejlett vizualizációs rendszert is tartalmaz.

Egy IEC 61131-3 CoDeSys program futtatásához szükség van a *runtime rendszer* (CoDeSys SP) alkalmazására, ezt az eszköz réteg tartalmazza. Egyéb összetevők is hozzáadhatók a futtatási környezethez, mint például a *WEB vizualizáció*, amely webes távfelügyeletet tesz lehetővé a CoDeSys-es eszközökön.

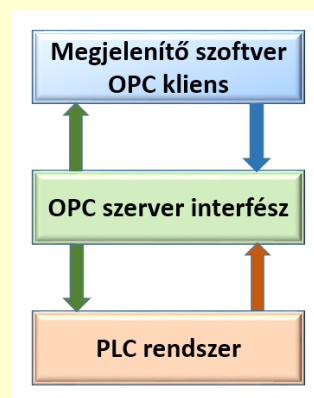
A fejlesztői réteg és az eszköz réteg között a CoDeSys Gateway Server teremti a kapcsolatot. A kommunikációs réteg OPC szervert is tartalmaz, amely az ipari eszközök egy szabványos

kommunikációs módja. Ennek a használatával a *CoDeSys* programok könnyen kommunikálhatnak más eszközökkel, illetve szoftverekkel.

2.2 A szimulációs működést támogató egyéb szoftverösszetevők

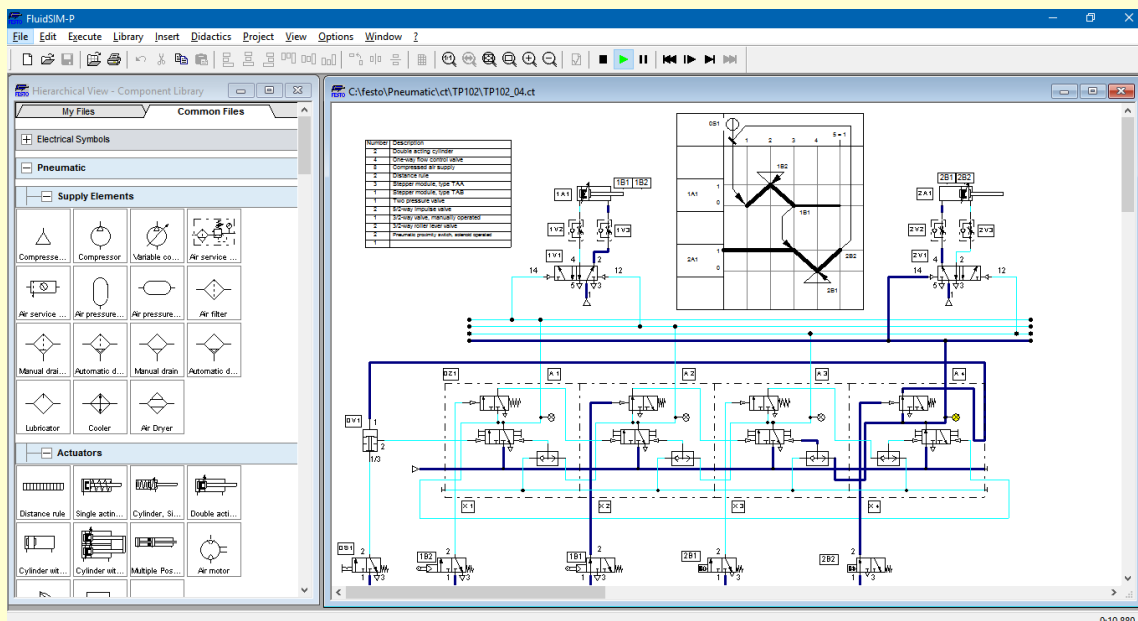
A szoftverrendszer komponenseinek összeállításakor meg kellett határoznom a szimulációs munkateret megjelenítő összetevőt. Mivel a *CoDeSys* v2.3 fejlesztőkörnyezet *OPC* kapcsolódási lehetőséget is biztosít, ezért kézenfekvő volt számomra olyan szoftvert választani, amely *OPC* kliens oldali szolgáltatást is tartalmaz.

Az *OPC* (*Object linking and embedding for Process Control*) a Windows operációs rendszerre épülő alkalmazások közötti szabványos kommunikációt biztosító objektum- és eljáráshalmazra vonatkozó ajánlás. Ezen ajánlás alapján készítik el a szoftvergyártók a saját implementációikat. Az *OPC* ajánlás szerint: a számítógépen a megjelenítő szoftvertől elkülönülten fut egy *OPC* szerver interfész, amely átveszi a PLC rendszerrel – annak „anyanyelvén” – való kommunikálás feladatát. Az *OPC* szerver egy definiált szoftveres felületen utasításokat forgalmaz (küld és fogad) az *OPC* kliens felé (lásd 2.2. ábra).



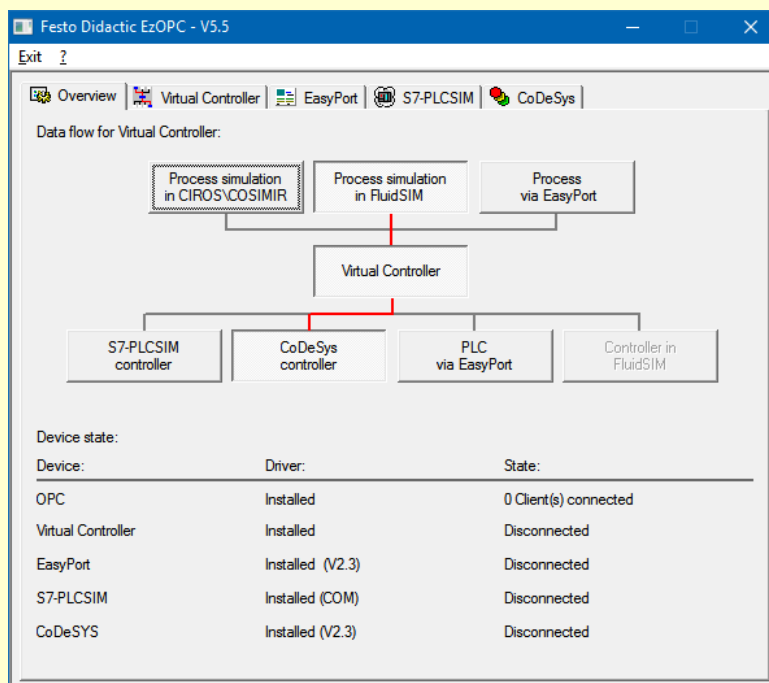
2.2. ábra: Általános OPC modell (Forrás: saját szerkesztés)

A megjelenítő szoftver tekintetében a választásom az előzőekben leírtak alapján a *FESTO FluidSIM Pneumatics* v4.2 alkalmazásra esett, mellyel pneumatikus, illetve elektropneumatikus kapcsolási modellek készíthetők. A modellek működése az interaktív szimulációs opció segítségével tesztelhető (lásd 2.3. ábra).



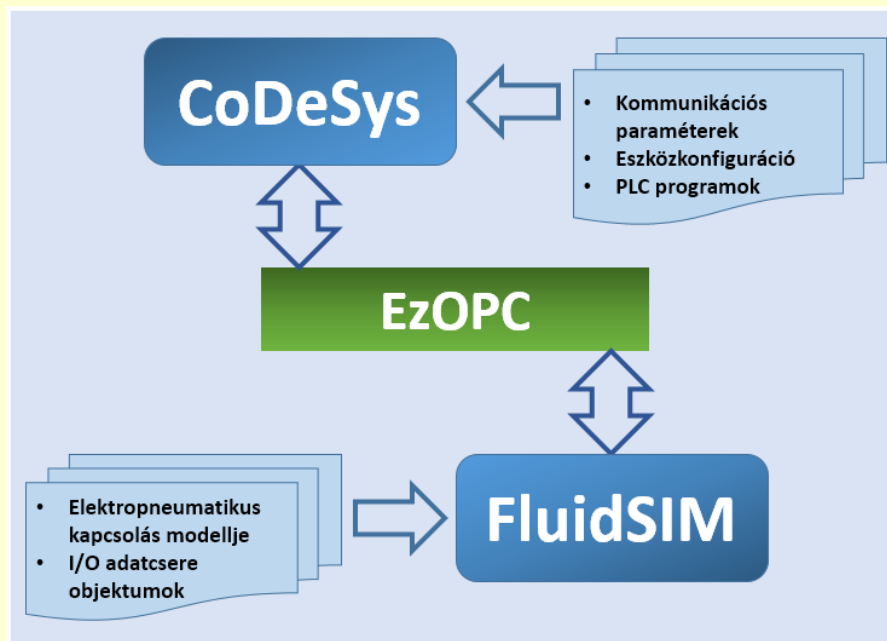
2.3. ábra: FluidSIM Pneumatics (Forrás: saját szerkesztés)

A szimulációs környezet futtatásához szükséges volt még egy OPC szerver interfész implementáció kiválasztása is. Működési és kompatibilitási paramétereit alapján a *FESTO EzOPC v5.5* szoftvert találtam alkalmasnak a feladatra (lásd 2.4. ábra). A komponens külön telepítést igényelt. A szoftver ingyenesen letölthető a *FESTO Didactic* honlapjáról.



2.4. ábra: EzOPC (Forrás: saját szerkesztés)

Az előzőekben felvázoltak alapján tehát összeállt egy teljesen szoftveres alapokon működő, interaktív modelleket alkalmazó szimulációs rendszer koncepciója (lásd 2.5. ábra).



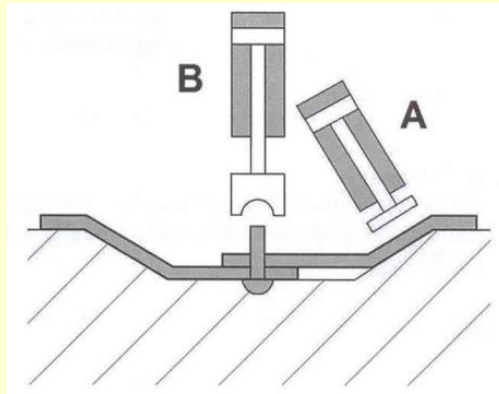
2.5. ábra: A szimulációs szoftverkörnyezet kommunikációs modellje (Forrás: saját szerkesztés)

3 A szimulációs szoftverrendszer használati lehetőségeit bemutató mintaalkalmazás kialakítása/előállítása

Az alábbiakban egy gyakorlati foglalkozáson alkalmazható feladat kidolgozásának ismertetése következik, amely az automatikai technikus szakmához kapcsolódó PLC programozási gyakorlat c. tantárgy tematikájához illeszkedik. A megvalósítás algoritmus, ábrázolási és modellezési technikái, valamint a magyarázatok részletezése a szerző módszertani megközelítését tükrözik.

3.1 A feladat megfogalmazása, célkitűzése

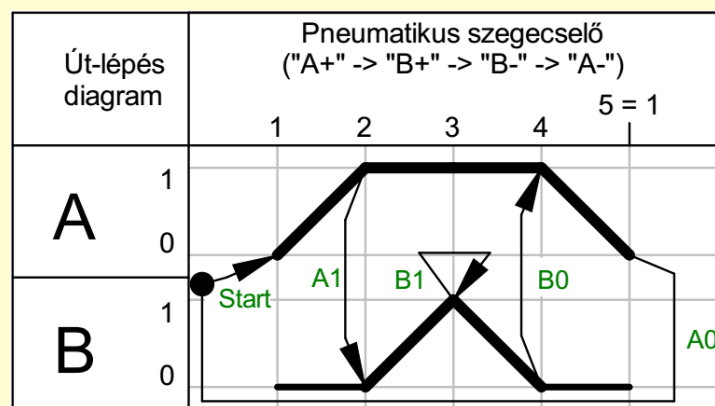
Lemezek szegecselését kell elvégezni pneumatikus szegecselővel (lásd 3.1. ábra). A készüléket a technológiának megfelelő sorrendi vezérléssel kell működtetni (FESTO 2014: 23).



3.1. ábra: *A készülék vázlata* (Forrás: FESTO, 2014 p. 23, 1. ábra)

A működési feltételek megfogalmazása, pontosítása (lásd 3.2. ábra):

- Az indítást egy „Start” jellel kezdeményezhessük!
- A folyamat többször ismétlődő legyen!
- Az „A” munkahenger a szorítást, a „B” munkahenger a szegecselést végezze!

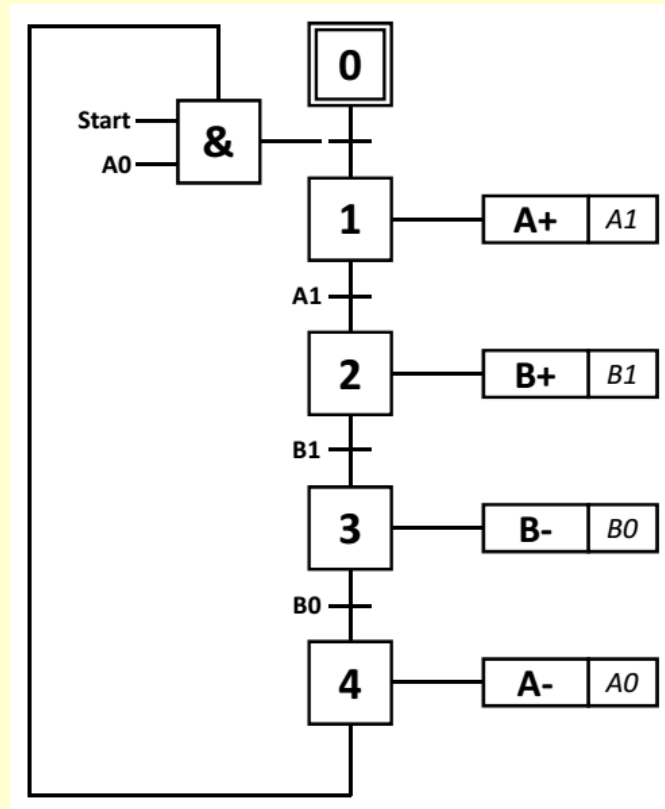


3.2. ábra: *A működési feltételeknek megfelelő út-lépés diagram* (Forrás: saját szerkesztés)

Az aktuátorok (munkahengerek) működési helyzetét mágneses közelítéskapcsolókkal figyeljük az alábbiak szerint:

- A0: Az „A” munkahenger hátsó (-) helyzetét jelző mágneses közelítéskapcsoló (Reed-relé).
- A1: Az „A” munkahenger kinti (+) helyzetét jelző mágneses közelítéskapcsoló (Reed-relé).
- B0: A „B” munkahenger hátsó (-) helyzetét jelző mágneses közelítéskapcsoló (Reed-relé).
- B1: A „B” munkahenger kinti (+) helyzetét jelző mágneses közelítéskapcsoló (Reed-relé).

A folyamatkövető vezérlések működési sorrendjét egyszerűen, áttekinthetően kell ábrázolni az összefüggések gyors felismerésének, megértésének segítése céljából. A sorrendi vezérlések esetén a legegyszerűbben *GRAFCET* (sorrendi folyamatábra) felírásával szemléltethetjük a vezérlési logikát (lásd 3.3. ábra). Az ábrázolási technika programnyelv független, tehát bármely vezérlőre könnyen átírható a fejlesztés kódolási fázisában (FESTO 2014).



3.3. ábra: A vezérlési logikát leíró GRAFCET (Forrás: saját szerkesztés)

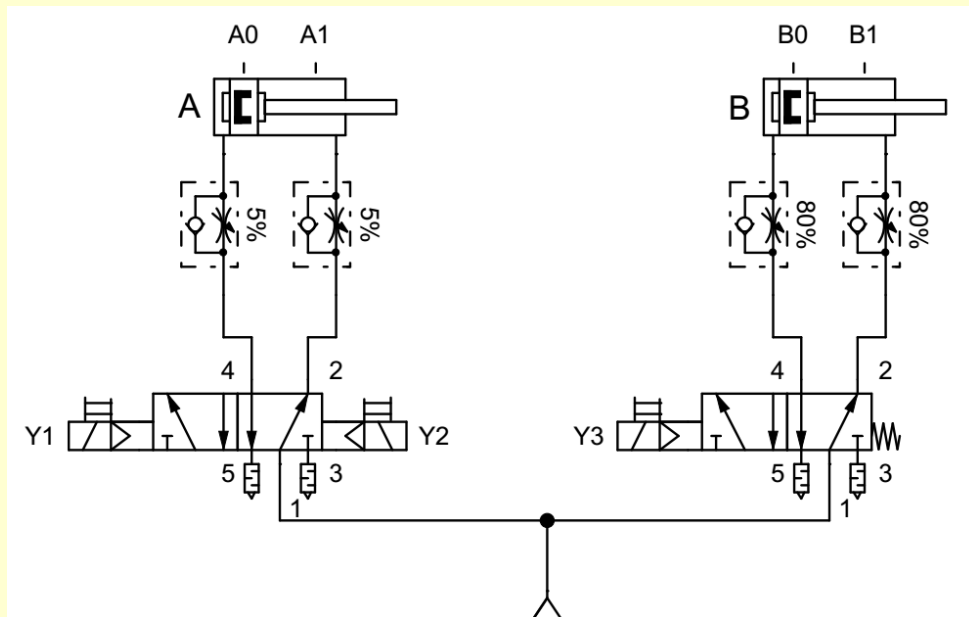
3.2 A FluidSIM kapcsolási modell elkészítése

A kapcsolási modell szakszerű elkészítése alapvető pneumatikai és elektrotechnikai ismereteket feltételez. Ezek megléte esetén a szoftver elemkészletéből kiválasztva „fogd és vidd” módszerrel összeállítható az elektropneumatikus kapcsolás. A kialakítás ezen fázisa megegyezik egy hagyományos kapcsolási rajz elkészítésével.

A pneumatikus rendszerhez (lásd 3.4. ábra) felhasznált elemek (Csík – Váradi 2007: 226-241):

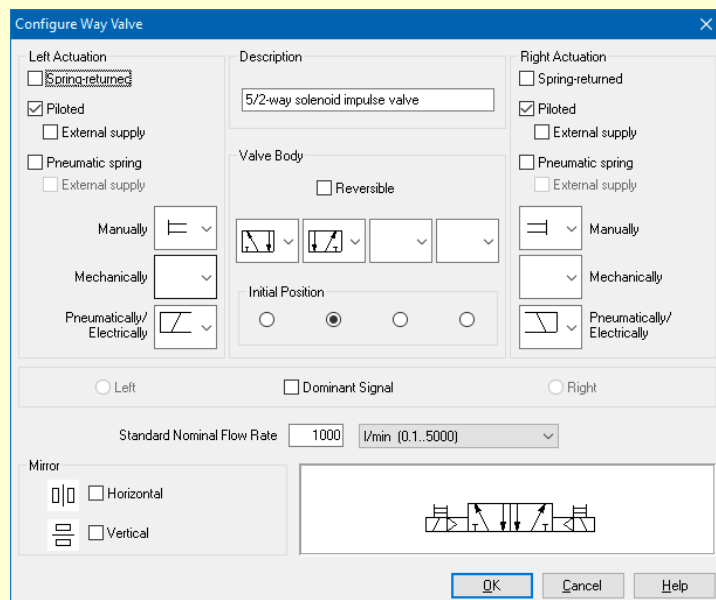
- 2 db kétoldali működésű pneumatikus munkahenger (A és B);
- 4 db fojtó-visszacsapó szelep (munkahengerenként 2 db), melyek segítségével kimentí fojtást alkalmazunk a munkahengerek sebességszabályozhatóságának biztosítása végett;

- 1 db bistabil 5/2-es útszelep (mágnesszelep) az „A” munkahenger működtetéséhez;
- 1 db monostabil 5/2-es útszelep (mágnesszelep) a „B” munkahenger működtetéséhez;
- 4 db Reed-relé (munkahengerenként 2 db) A: -A0, +A1; B: -B0, +B1.



3.4. ábra: A kapcsolási modell pneumatikus kiépítése (Forrás: saját szerkesztés)

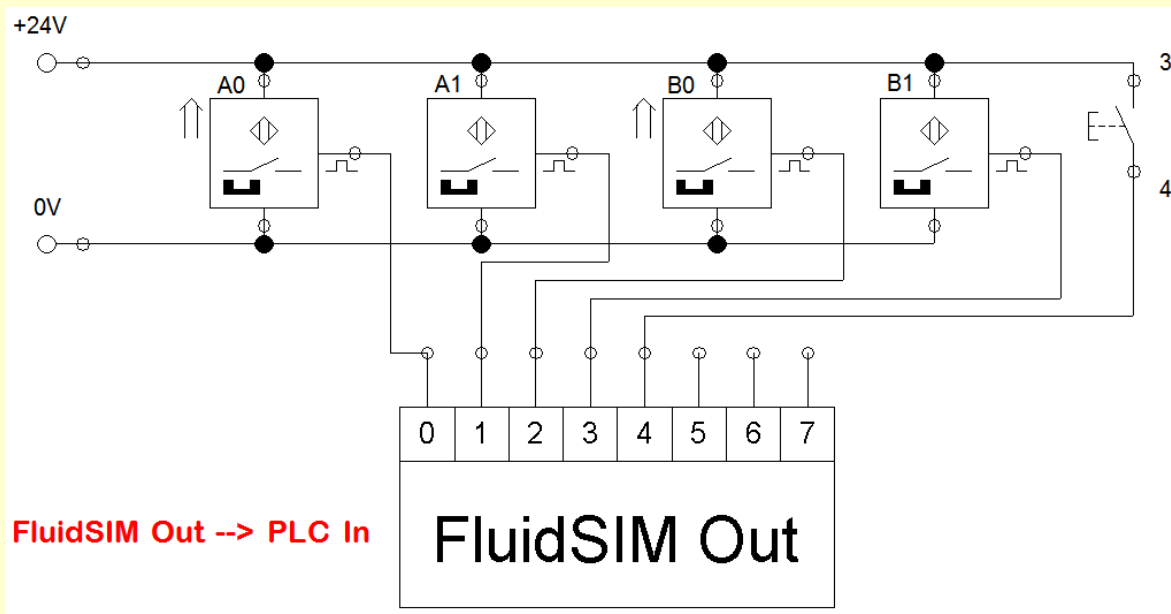
A pneumatikus elem tulajdonságait és hivatkozási címkeit az elemen történő dupla kattintás után felbukkanó párbeszédablakokban állíthatjuk be (lásd 3.5. ábra).



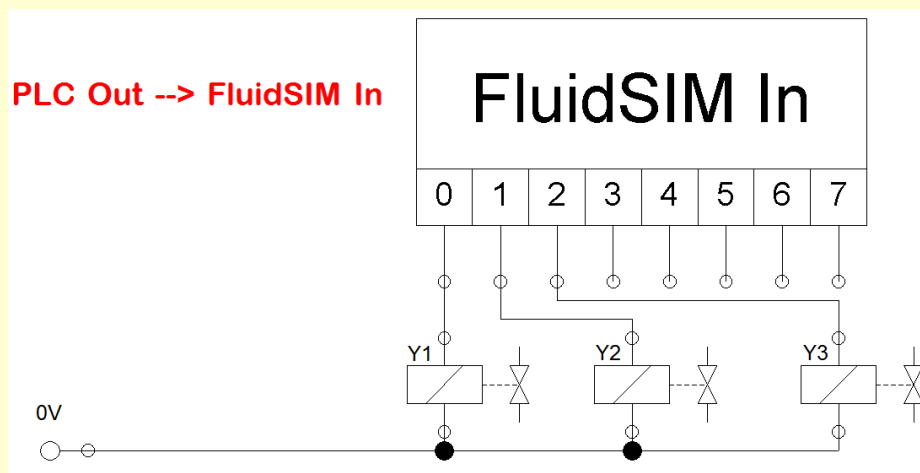
3.5. ábra: Pneumatikus elem (5/2-es mágnesszelep) tulajdonságpanelje (Forrás: saját szerkesztés)

A következőkben el kell készíteni a kapcsolási modell elektromos huzalozási tervét. Az elektronikus alkatrészeket ún. *FluidSIM Port* objektumokba kötjük be. Amennyiben a

szerelést a PLC szemszögéből nézzük, akkor ügyeljünk rá, hogy a szenzorokat *FluidSIM Output Port* objektumba (lásd 3.6. ábra), az aktuátorokat *FluidSIM Input Port* objektumba (lásd 3.7. ábra) huzalozzuk!



3.6. ábra: Virtuális PLC bemenetek huzalozása (Forrás: saját szerkesztés)



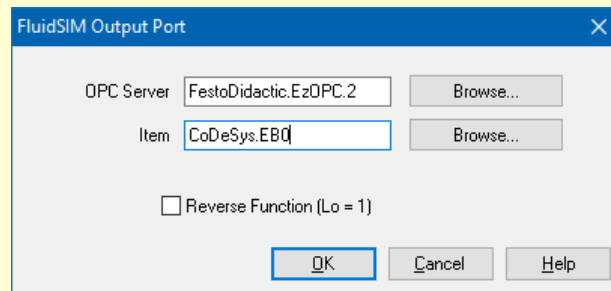
3.7. ábra: Virtuális PLC kimenetek huzalozása (Forrás: saját szerkesztés)

Ezután a *FluidSIM Port* objektumok konfigurációját szükséges beállítani. Azonosítani kell a használni kívánt OPC szervert, valamint az objektum sorkapocs elemeihez rendelt PLC-s memóriaterületet (bitek vagy bájtok).

A beállítandó adatok a *FluidSIM Out Port* objektumpéldányhoz (lásd 3.8. ábra):

- OPC szerver: *FestoDidactic.EzOPC.2*
- Memóriaterület: *CoDeSys.EB0*

Tehát a szimulációs modell beállított kommunikációs objektuma a *CoDeSys* PLC szimulátor első bemeneti bájtjához fog kapcsolódni a megadott OPC szerveren keresztül.

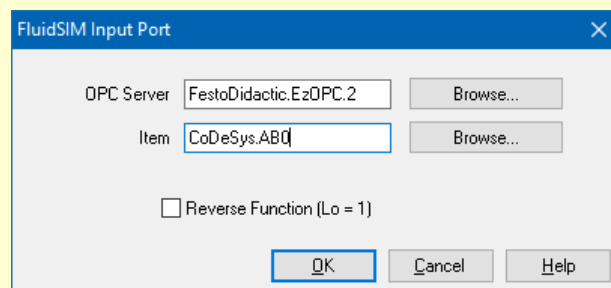


3.8. ábra: *FluidSIM Out Port* beállításai (Forrás: saját szerkesztés)

A beállítandó adatok a *FluidSIM In Port* objektumpéldányhoz (lásd 3.9. ábra):

- OPC szerver: *FestoDidactic.EzOPC.2*
- Memóriaterület: *CoDeSys.AB0*

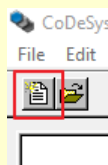
Tehát a szimulációs modell beállított kommunikációs objektuma a *CoDeSys* PLC szimulátor első kimeneti bájthoz fog kapcsolódni a megadott OPC szerveren keresztül.



3.9. ábra: *FluidSIM In Port* beállításai (Forrás: saját szerkesztés)

3.2.1 A PLC projekt elkészítése a CoDeSys fejlesztőkörnyezetben

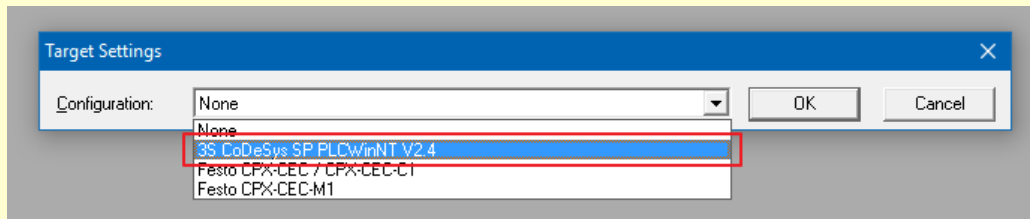
Első lépésben létre kell hozni egy új projektet, miután megnyitottuk a keretrendszert. Ehhez az eszköztár *New* elemét kell választani (lásd 3.10. ábra).



3.10. ábra: Új projekt létrehozása (Forrás: saját szerkesztés)

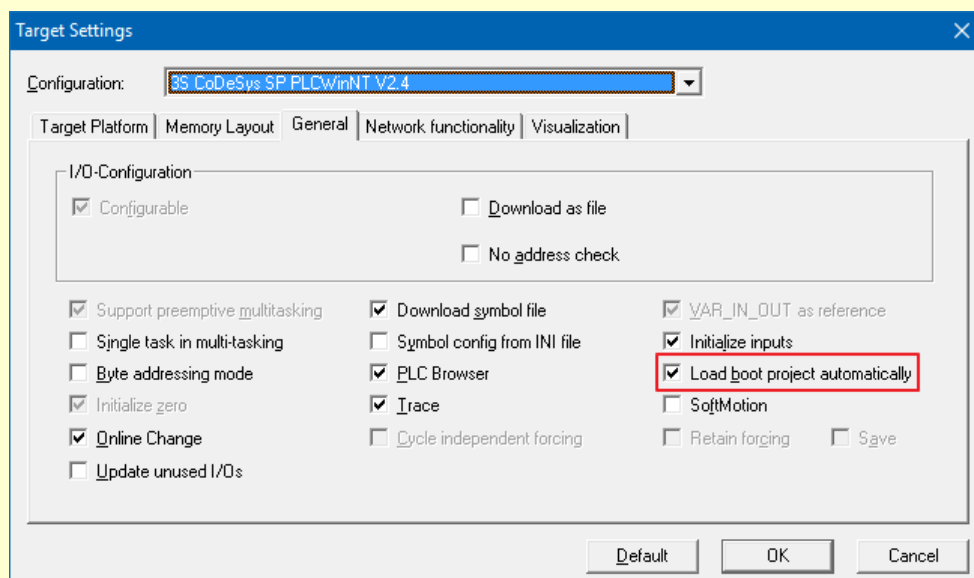
Ezután meg kell határozni a céleszköz típusát, mely lenyíló listából választható. A lista tartalma a *CoDeSys* disztribúció függvényében változhat. Minden disztribúció standard eleme egy virtuális PLC, mellyel szimulációs működés valósítható meg. A virtuális eszközt valós

PLC-nek fogja látni a rendszer, így lényegében hardver emulációt állíthatunk be. A 3S *CoDeSys SP PLCWinNT* elemet kell választani (lásd 3.11. ábra).



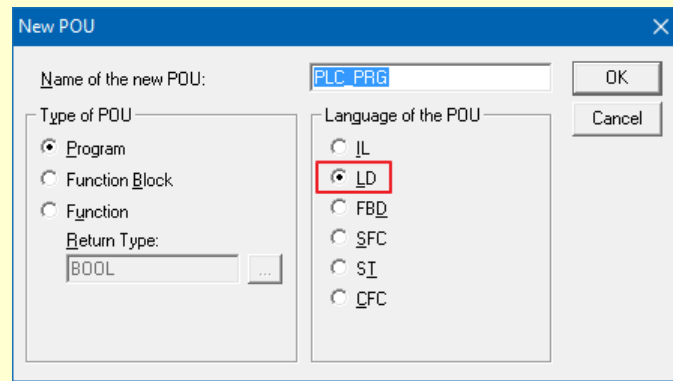
3.11. ábra: Céleszköz kiválasztása (Forrás: saját szerkesztés)

A következőkben a céleszköz beállításait adhatjuk meg. Fontos, hogy a PLC-be áttöltött program az eszköz „futtatás állapotba” kapcsolásakor automatikusan elinduljon. Ezt a beállítási panel *General* fülén jelölhetjük be (lásd 3.12. ábra).



3.12. ábra: A céleszköz beállításai (Forrás: saját szerkesztés)

A PLC program kódolása előtt ki kell választani a programnyelvet (lásd 3.13. ábra). A mintaprogram implementálásához a létradiagram nyelvet használtam. A PLC-s vezérléseknél a leggyakrabban alkalmazott, valamint a sorrendi vezérlések hatékonyan és átláthatóan programozhatók a segítségével. A *CoDeSys*-ben az alapértelmezett programnév a *PLC_PRG*. Amennyiben más nem indokolja, érdemes ezt megtartani.



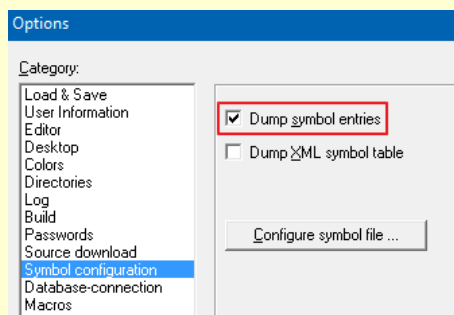
3.13. ábra: Új program létrehozása (Forrás: saját szerkesztés)

Mivel a program OPC szervertől fogja vezérelni a FluidSIM kapcsolási modellt, és ott korábban az *EB0* és *AB0* szimbólumokkal hivatkoztunk a bemeneti és kimeneti bájtokra, így ezt a fejlesztőkörnyezetben is deklarálni kell. Ezt a projektablak *Resources* fülén a *Global_Variables* szekcióban tehetjük meg (lásd 3.14. ábra).



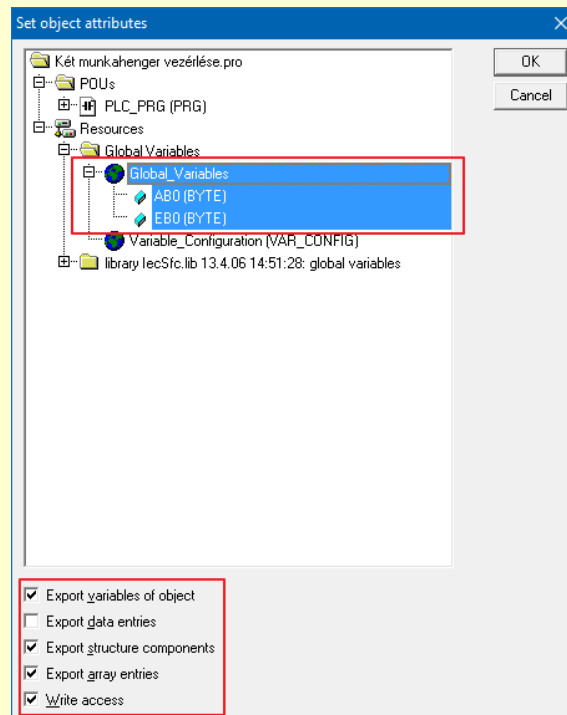
3.14. ábra: Globális változók deklarálása (Forrás: saját szerkesztés)

Az OPC server abban az esetben tudja átadni a változók aktuális értékeit a kliensalkalmazásnak, amennyiben engedélyezzük ezt a projekt beállításában (*Project* menü - > *Options...*) (lásd 3.15. ábra).



3.15. ábra: Változók engedélyezése adatcserére (Forrás: saját szerkesztés)

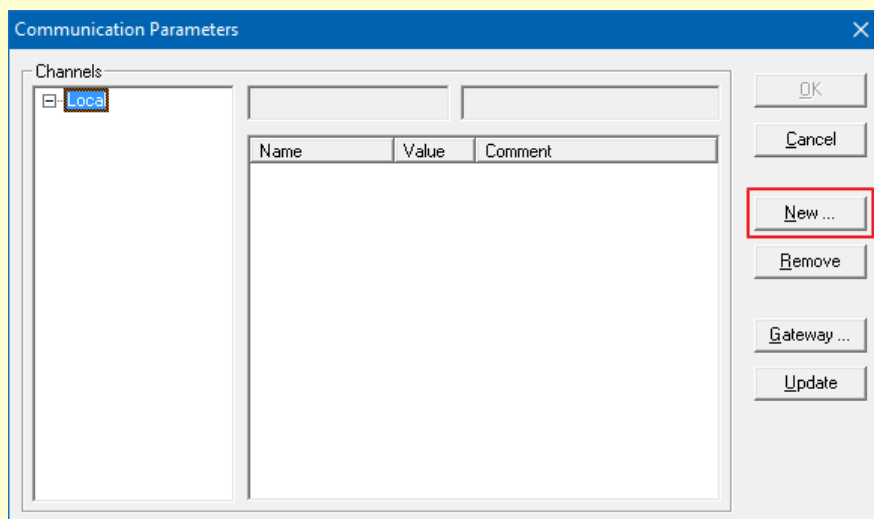
A továbbiakban a *Configure symbol file ...* nyomógomb használata után meg kell adni az adatcserére engedélyezett szimbólumokat és azok attribútumait (lásd 3.16. ábra).



3.16. ábra: Szimbólumok kijelölése és attribútumaiknak beállítása (Forrás: saját szerkesztés)

3.2.2 A fejlesztőkörnyezet és a virtuális PLC eszköz kommunikációjának beállítása

A virtuális PLC eszköz (PLCWinNT) egy valódi PLC berendezés szoftveres absztrakciója, működése modellezi és emulálja a hardver működését. Tehát a fejlesztőkörnyezettel bonyolított kommunikációja is hasonló. A kommunikációs beállítások az *Online* menü -> *Communication Parameters...* menüponton keresztül érhetők el. A felbukkanó panel *New ...* nyomógombjával hozhatunk létre új kommunikációs csatorna definíciót (lásd 3.17. ábra).



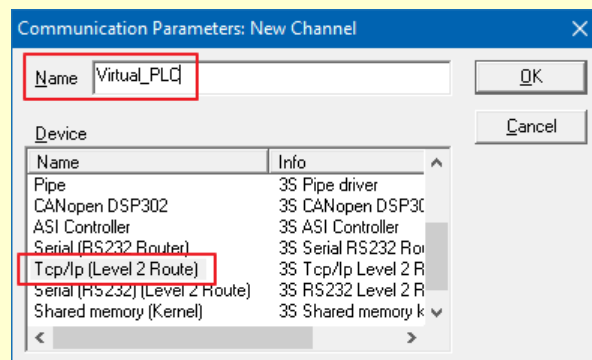
3.17. ábra: Új kommunikációs csatorna létrehozása (Forrás: saját szerkesztés)

Napjainkban a korszerű PLC-k *Ethernet* keretekbe ágyazott üzenetekkel kommunikálnak. Ehhez a *TCP/IP* protokollt használják (lásd 3.18. ábra) (Tanenbaum-Wetherall 2013: 60-64).

	OSI	TCP/IP
7	Application	Applications (FTP, SMTP, HTTP, etc.)
6	Presentation	
5	Session	
4	Transport	TCP (host-to-host)
3	Network	IP
2	Data link	Network access (usually Ethernet)
1	Physical	

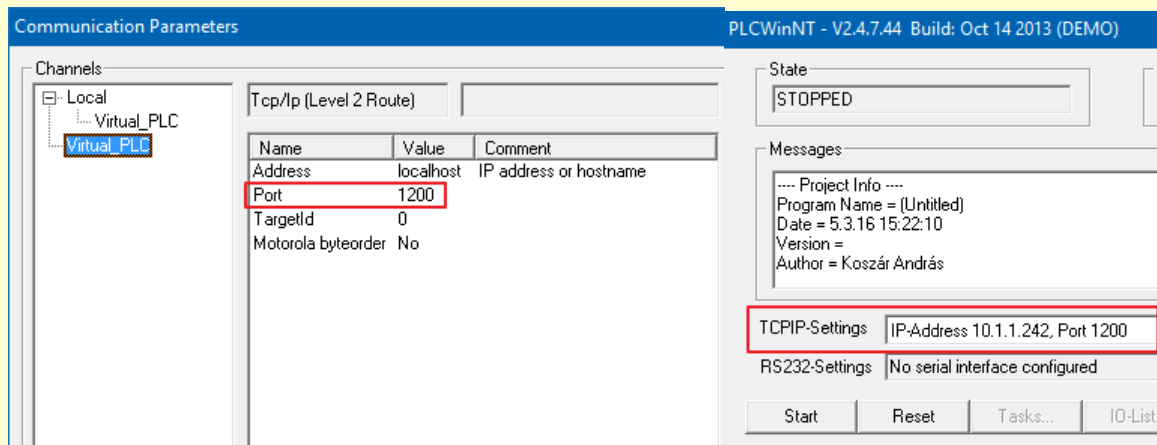
3.18. ábra: Az ISO-OSI és a TCP/IP összefüggésmodellje (Forrás: electronicdesign.com, What's The Difference Between The OSI Seven-Layer Network Model And TCP/IP?, 2013. 10.)

Az új kommunikációs csatornának tehát egy a *TCP/IP* protokollt alkalmazó, a virtuális PLC-vel kompatibilis drivert választottam: *TCP/IP (Level 2 Route)* (lásd 3.19. ábra).



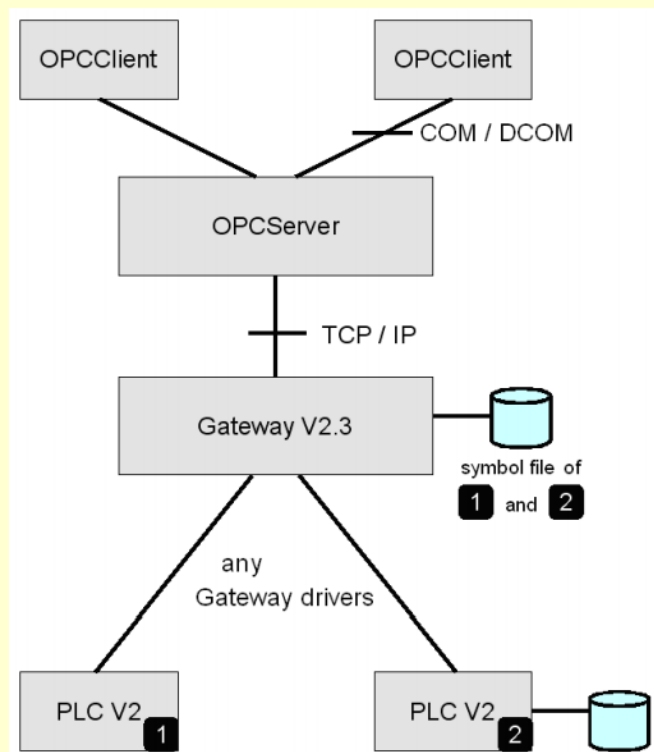
3.19. ábra: A kommunikációs csatorna beállítása (Forrás: saját szerkesztés)

A beállítások aktualizálása után a fejlesztőkörnyezetből is elérhetjük a virtuális PLC eszközt a *CoDeSys Gateway Server* szolgáltatáson keresztül, mely a keretrendszer telepítése során beépül a számítógép operációs rendszerébe. A *PLCWinNT* az 1200-as portot használja a kommunikációhoz, ezt a beállítást szükséges egyeztetni a fejlesztőkörnyezet és az emulátor között, valamint engedélyezni kell az operációs rendszerhez kapcsolódó tűzfal szoftverben is (lásd 3.20. ábra).



3.20. ábra: Kommunikációs beállítások egyeztetése a CoDeSys és a PLCWinNT között (Forrás: saját szerkesztés)

A CoDeSys Gateway Server nem csak a virtuális PLC és a fejlesztőkörnyezet közötti kapcsolatot vezérli. Lehetőséget ad több valós vagy virtuális, esetleg valós-virtuális PLC kapcsolatra, illetve az OPC szerverrel is az átjáró szolgáltatáson keresztül kommunikálhatunk (CoDeSys, 2015). Tehát, ugyancsak így valósul meg az engedélyezett szimbólumok általi adatcsere a FluidSIM kapcsolási modellnél (lásd 3.21. ábra).



3.21. ábra: A CoDeSys rendszer hálózati modellje (Forrás: CoDeSys, 2015 p. 8; 1. ábra)

3.2.3 A folyamatkövető vezérlés PLC programjának elkészítése

Mivel a programot létradiagram nyelven valósítottam meg, illetve sorrendi végrehajtás a követelmény, ezért kézenfekvő volt a hagyományos relés vezérléseknél is alkalmazott léptetőláncos jelkapcsolás logikájából kiindulni. A működés algoritmusából adódó lépésekhez bites memóriatárolókat, ún. *merkereket* rendeltem. Ezek alapján a program lokális deklarációs részét a 3.22. ábra szemlélteti.

```

0001(*
0002PNEUMATIKUS SZEGECSELŐ
0003Feladat: "A+" -> "B+" -> "B-" -> "A-", a folyamatot a "START" nyomógomb jelével indítjuk.
0004Merkeres léptetőláncot alkalmazunk az implementációban.
0005A léptetőláncot öntartó körökkel valósítjuk meg.
0006*)
0007PROGRAM PLC_PRG
0008VAR
0009    M0: BOOL:=0; (* "START állapot" merker *)
0010    M1: BOOL:=0; (* 1. lépés merker *)
0011    M2: BOOL:=0; (* 2. lépés merker *)
0012    M3: BOOL:=0; (* 3. lépés merker *)
0013    M4: BOOL:=0; (* 4. lépés merker *)
0014    START_el: R_TRIG;
0015END_VAR
0016(*
0017    EB0.0: "A" henger dugattyúja "-" helyzetben
0018    EB0.1: "A" henger dugattyúja "+" helyzetben
0019    EB0.2: "B" henger dugattyúja "-" helyzetben
0020    EB0.3: "B" henger dugattyúja "+" helyzetben
0021    EB0.4: "START" nyomógomb
0022    AB0.0: "A" henger dugattyúja "KITOL" szelep (Y1 bistabil)
0023    AB0.1: "A" henger dugattyúja "VISSZA" szelep (Y2 bistabil)
0024    AB0.2: "B" henger dugattyúja "KITOL" szelep (Y3 monostabil)

```

3.22. ábra: A program lokális deklarációs szakasza (Forrás: saját szerkesztés)

Léptetőláncos megvalósítások esetén az első indítást megelőzően az utolsó lépést jelző tárolót be kell írni (a működtetett állapotot mutató merker negált értéke esetén szükséges megtenni), a többit törölni kell (a deklarációnál alkalmazott kezdőértékek megadása biztosítja). A „Start állapot” segédrelét a nyomógomb jelének felfutó éle fogja beírni (lásd 3.23. ábra).



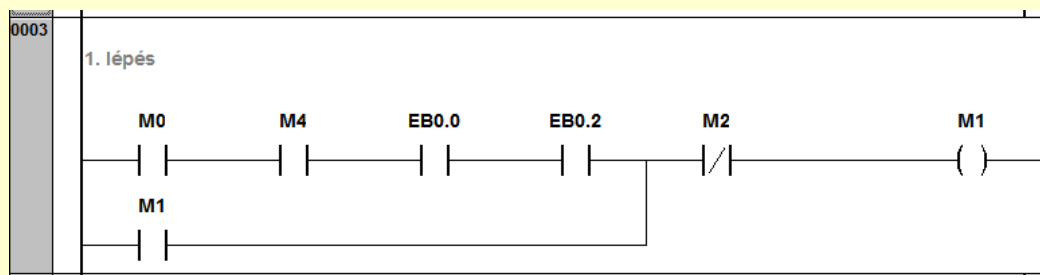
3.23. ábra: Az alaphelyzet felvétele és a START nyomógomb lekezelése (Forrás: saját szerkesztés)

A tárolók öntartó kapcsolások kialakításával működtethetők. A léptetőláncnak mindig csak egy eleme lehet aktív, a többi ki kell zárni.

Az öntartásokhoz tervezett feltételrendszer az előzők alapján:

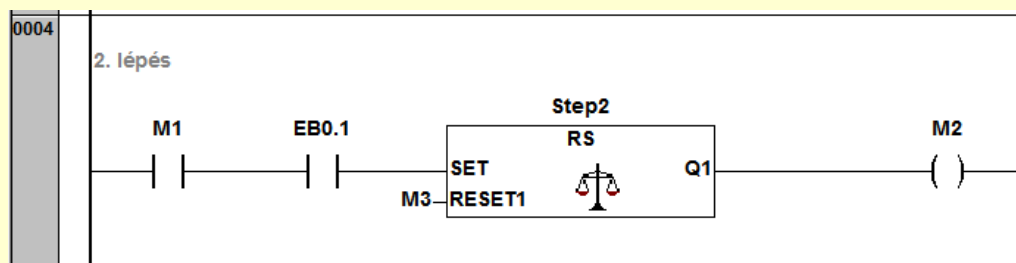
- az aktuális lépés végrehajtását az előző lépés engedélyező jele (*a segédreléhez tartozó kontakt ponált alakú lekérdezése*), valamint a kiegészítő feltételek teljesülése indíthatja el;
- az első lépésnél az utolsó lépés számít előzőnek a ciklikus működtetés biztosításának érdekében;
- a kiegészítő feltételeket a „Start állapot” merker értéke, illetve a szenzorok jelei határozzák meg; a szenzorok jeleinek állapotai az *EB0* bitjei által kérdezhetők le;
- az aktuális lépés végrehajtását a következő lépés blokkoló jele (*a segédreléhez tartozó kontakt negált alakú lekérdezése*) állíthatja meg (*dominánsan törlő megoldásban*);
- a merker beírt állapotban tartását a saját kontakt öntartó ágba szervezett ponált alakú lekérdezése valósítja meg.

Egy a feltételrendszer alapján kialakított hálózatot szemléltet a 3.24. ábra.



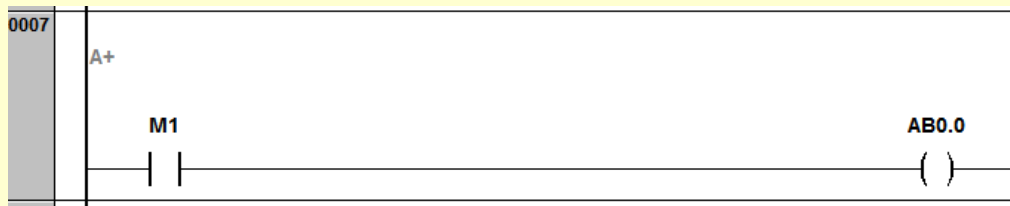
3.24. ábra: Vezérlési lépés kialakítása dominánsan törlő öntartó kapcsolással (Forrás: saját szerkesztés)

A vezérlési lépések hálózatai megvalósíthatók *RS Flip-flop* blokkok alkalmazásával is. Működése egyenértékű az előzőekben tárgyalt alternatívával, viszont tömörebb leírási formát biztosít. A törlésre domináns RS tárolós változatot mutatja be a 3.25. ábra.



3.25. ábra: Vezérlési lépés kialakítása RS Flip-flop alkalmazásával (Forrás: saját szerkesztés)

A programkészítés utolsó fázisában be kell állítani a kimeneteket reprezentáló *AB0* bitjeit a merkek állapotának megfelelően (lásd 3.26. ábra).



3.26. ábra: *A kimenetek (AB0) beállítása (részlet)* (Forrás: saját szerkesztés)

3.2.4 A kapcsolási modell vezérlése a PLC program futtatásával

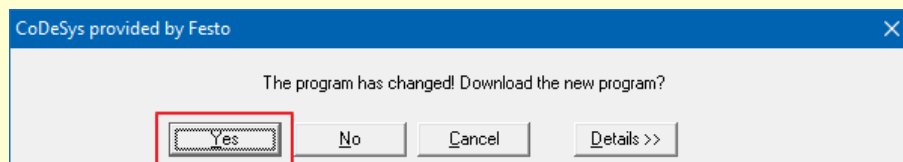
Indítsuk el a *PLCWinNT* programot a *Start* menüben található *CoDeSys* fejlesztői mappából (*a pontos neve disztribúciónként eltérő lehet*), valamint nyissuk meg a korábban elkészített kapcsolási modellt a *FluidSIM* programmal!

Az elkészült és szintaktikailag helyes létradiagram programot át lehet tölteni a virtuális PLC eszközre. Ehhez az eszköztár *Login* elemét kell használni (lásd 3.27. ábra).



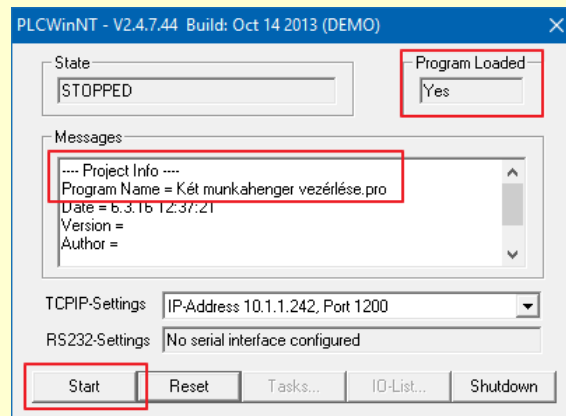
3.27. ábra: *A program áttöltése a PLC-re* (Forrás: saját szerkesztés)

Amennyiben korábban a virtuális PLC-be már töltöttünk át programot, a 3.28. ábra szerinti párbeszédablakkal találkozhatunk. A továbblépéshez válasszuk a *Yes* opciót!



3.28. ábra: *Előző program felülírása a PLC-ben* (Forrás: saját szerkesztés)

Az áttöltés sikerességét ellenőrizhetjük a *PLCWinNT* felhasználói felületén. Majd kezdeményezzük a PLC program futtatását a *Start* nyomógomb alkalmazásával (lásd 3.29. ábra)!



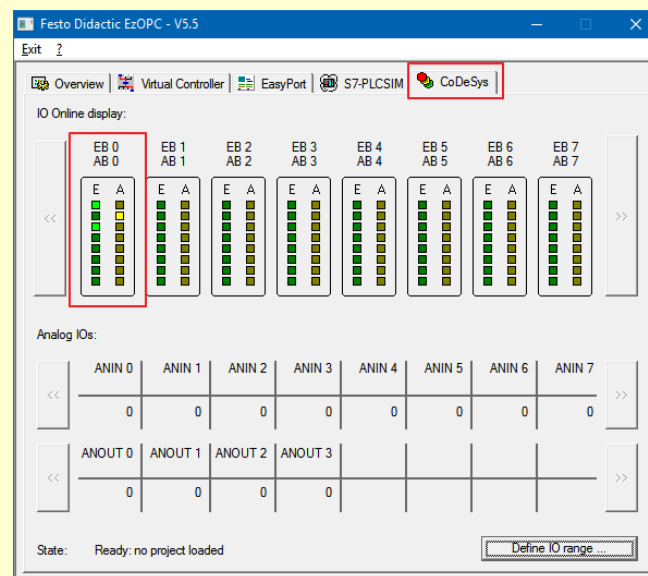
3.29. ábra: A PLCWinNT kezelői felülete (Forrás: saját szerkesztés)

Ezután már indítható a kapcsolási modell szimulációja a *FluidSIM* programban. Ez a *Start* nyomógomb használatával kezdeményezhető (lásd 3.30. ábra).



3.30. ábra: A FluidSIM szimuláció futtatása (Forrás: saját szerkesztés)

A szimuláció indítását követően a *FluidSIM* ellenőrzi a kapcsolási modellt szintaktikai és szemantikai aspektusból.

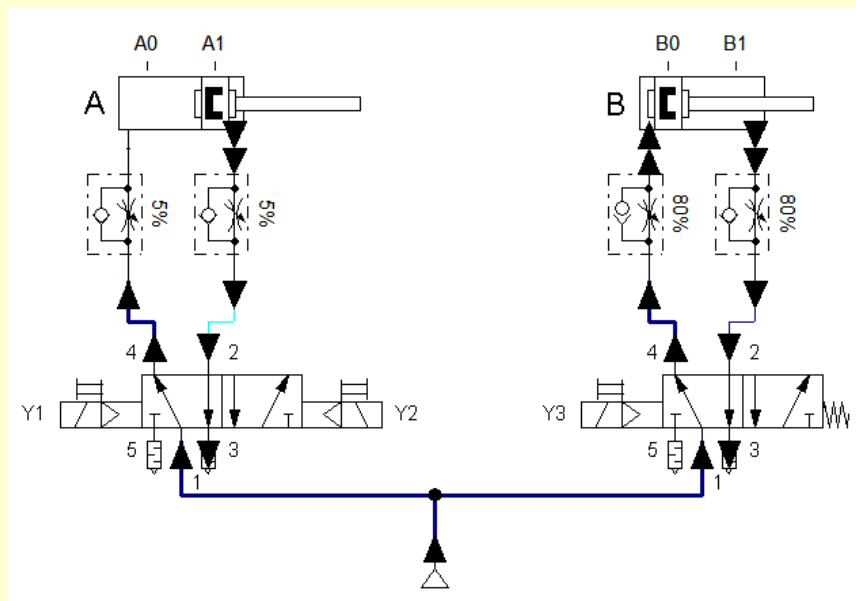


3.31. ábra: Az OPC szerver állapotjelző ablaka (Forrás: saját szerkesztés)

Mivel az elektropneumatikus kapcsolás elektronikus elemei *FluidSIM* I/O objektumokhoz vannak huzalozva a szimulációban, ezért a rendszer elindítja a korábban beállított *EzOPC* szerveret, mely összekapcsolja a *FluidSIM* modellt és a virtuális PLC-ben futó programot. Ekkor a szerverhez tartozó állapotjelző párbeszédablak is megjelenik a tálcára kicsinyített

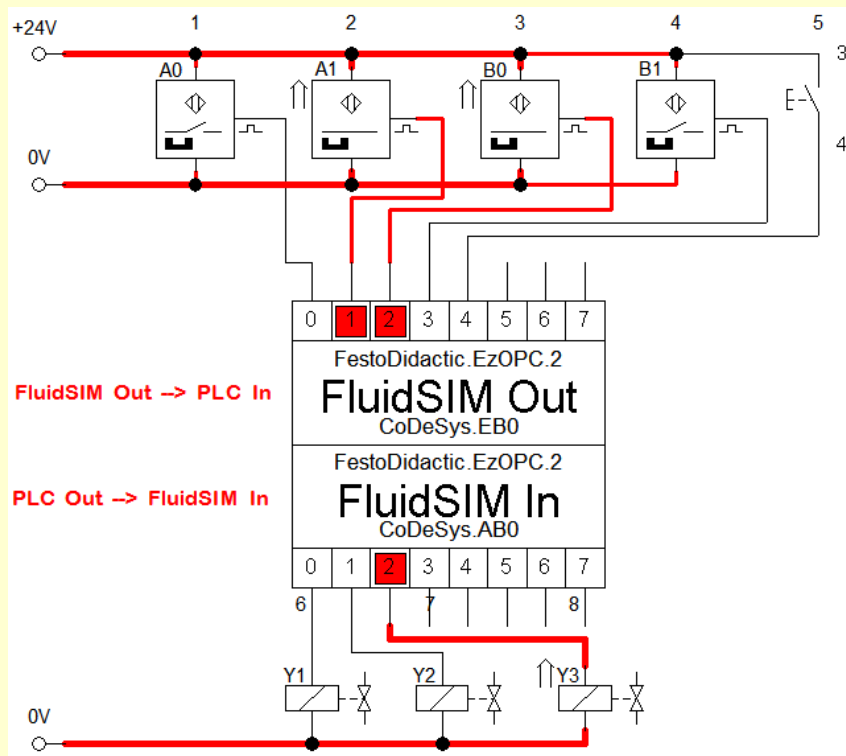
formában. Itt célszerű ellenőrizni a virtuális be- és kimenetek állapotát, azaz az *OPC* szerver által, és a korábban definiált szimbólumokon (*EB0*, *AB0*) keresztül lebonyolított adatcserét. Összekapcsolt állapotban, a biteket reprezentáló *IO Online display* panelelemek zöld, illetve sárga visszajelzést adnak a monitoron (lásd 3.31. ábra).

A vezérlés szimulációs működtetése a *FluidSIM*-ben (lásd 3.32. ábra) a kapcsolási modellben huzalozott nyomógomb (lásd 3.33. ábra – 5. áramút) jelére indítható.



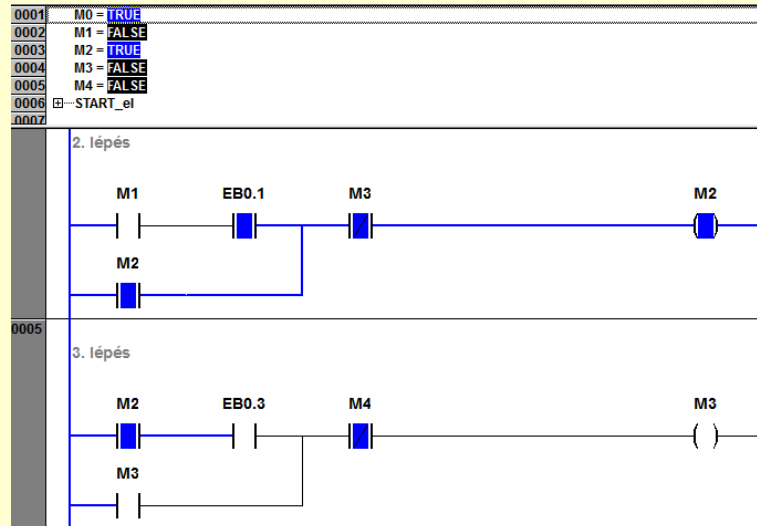
3.32. ábra: A kapcsolási modell pneumatikus rendszere a szimulációs működtetés közben
(Forrás: saját szerkesztés)

A szimuláció futtatása interaktív módon történik, tehát folyamatos visszajelzést kapunk a pneumatikus, illetve az elektromos alrendszer állapotváltozásairól. Valamint bármikor beavatkozhatunk a folyamat lefutásába a *FluidSIM* szimulációs eszköztár vezérlőgombjainak működtetésével: leállítható, pillanatmegállítás alkalmazható, szakaszos működési mód kapcsolható, stb. Mindeközben a *CoDeSys* felületén (online állapotban) is nyomon követhetjük a program működését és a változók aktuális értékeit (lásd 3.34. ábra). A PLC program futtatása a *PLCWinNT* virtuális PLC eszköz párbeszédpaneljén állítható le.



3.33. ábra: A kapcsolási modell elektromos huzalozása a szimulációs működtetés közben

(Forrás: saját szerkesztés)

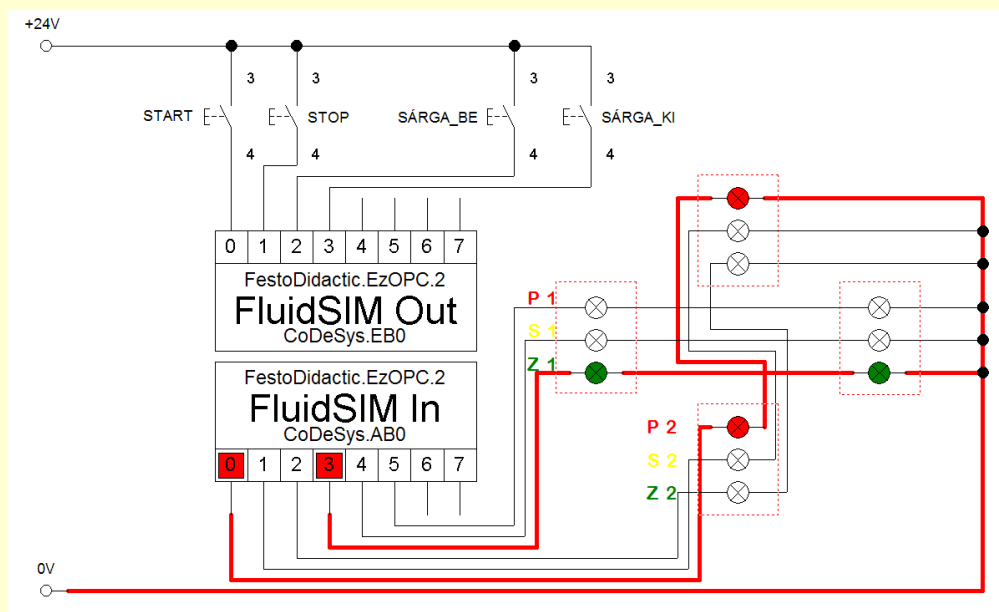


3.34. ábra: A CoDeSys fejlesztőkörnyezet visszajelzései Online állapotban (Forrás: saját szerkesztés)

3.3 A szimulációs szoftverrendszer alkalmazásának kiterjesztése

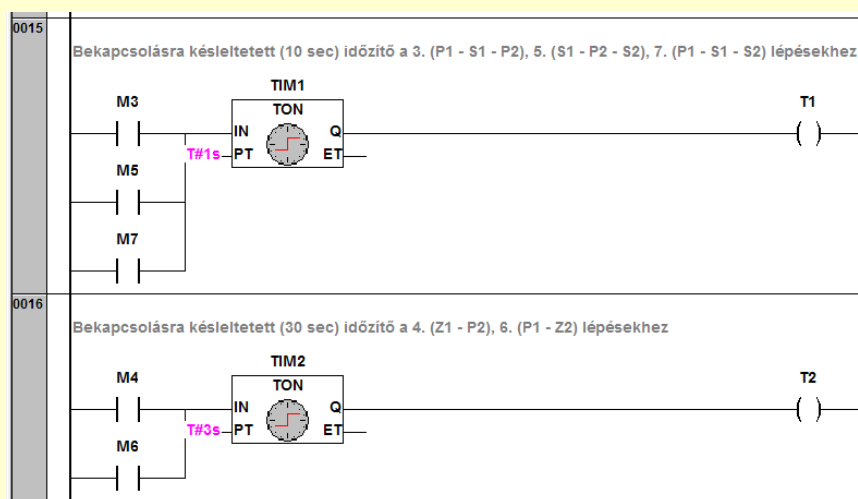
Az előzőekben ismertetett szimulációs technikát és a pedagógiai tevékenységem során kialakított, hozzá kapcsolódó módszertani eszközkészletet több éve eredményesen

alkalmazom a szakképzésben, valamint a felnőttképzésben. Tapasztalataim szerint a felvázolt szoftver-összeállítás és módszertana nem csak a pneumatika/hidraulika tárgykörébe eső modellfeladatok kidolgozására alkalmas. Különböző gépipari és villamosipari – PLC-vel vezérelhető – készülékek, illetve berendezések absztrakciója alakítható ki a felhasználásával. A 3.35. ábra egy újabb alkalmazási példa logikai huzalozási vázlatát mutatja be, mely egy útkeresztvező forgalomirányító lámpáinak PLC-s vezérlését modellezi.



3.35. ábra: Közlekedési lámpák kapcsolási modellje működés közben (Forrás: saját szerkesztés)

A 3.2 alfejezetben ismertetett eljárás konzekvens implementálásával kidolgozható a megoldás szimulációs környezete és PLC-s vezérlőprogramja (lásd 3.36. ábra) is.



3.36. ábra: A modellt működtető létradiagramos program - részlet (Forrás: saját szerkesztés)

4 Összegzés

A számítógéppel támogatott automatizálási technológiák megjelenésével a szakterület robbanásszerű fejlődésének lehattünk tanúi az elmúlt évtizedekben. Az ipari csúcstechnológiák hatékony felhasználásának alapfeltétele a magasan képzett, specializált munkaerő, melynek biztosításához elengedhetetlen a korszerű szakmai oktatási trendek követése a szakmai oktatók részéről.

Az automatikai-mechatronikai szakterület szakmai elméleti és gyakorlati oktatásában a valóságos eszközök és berendezések innovatív és eredményes kiegészítőjének bizonyulhat a szimulációs szoftverek absztrakciós funkcionalitása. Ezek segítségével olyan virtuális tanulási környezetek alakíthatók ki, melyek a pedagógusok, valamint a tanulók iskolai, tanműhelyi és otthoni munkáját hatékonyan támogatják.

Az *OPC* technológiát gyakran és hatékonyan alkalmazzák irányítástechnikai feladatok (pl.: folyamatvezérlés, adatgyűjtés, stb.) megvalósítására az iparban, így a szimulációs rendszerem alapjának ezt választottam. Korábbi tapasztalataim szerint az *OPC* megoldások remekül együtt tudnak működni a *CoDeSys* alapú PLC-s vezérlésekkel, tehát a programfejlesztő környezet is adott volt. Mivel a rendszerhez PLC futtatási környezet tartozik, így a vezérlőberendezés működését is meg tudtam valósítani szoftveres alapokon. Kellett még egy szenzorokat és aktuátorokat, azaz virtuális berendezést emuláló alkalmazást is választanom a szimulációs környezet kialakításához. A *FluidSIM-P*-ben pneumatikus és elektropneumatikus kapcsolási modellek hozhatók létre, melyekben *OPC* interfészmodulok is definiálhatók, tehát megfelelőnek bizonyult az integráláshoz.

Írásomban az előzőekben ismertetett komponensekből kialakított szimulációs környezet jellemzőit foglaltam össze. Elemeinek konfigurálását, valamint egy tanórai használati esetét, azaz egy szimulációs mintafeladat létrehozását vezettem végig, melynek során a hatékony és korszerű szakmódszertani rutinok gyakorlati alkalmazását volt szándékomban bemutatni a Tisztelt Olvasónak.

Véleményem szerint és tapasztalataim alapján úgy gondolom, hogy a tanulmányban felvázolt technikák és módszerek eredményesen alkalmazhatók a szakképzésben, elsősorban a PLC programozás és hibadiagnosztika témaköreit magában foglaló elméleti és gyakorlati órákon. Használatukkal fejleszthető a tanulók programozási készsége, biztosítható a gyakori és ismétlődő hibajelenségek elhárításában való alapvető jártasságuk a valós oktatási berendezés rendelkezésre állása vagy meghibásodásának veszélye nélkül. Ugyanakkor hasznos

kiegészítő, illetve szemléltető lehetőségeket rejt magában a vezérléstechnikát érintő egyéb tantárgyak esetében is, mivel bemutatathatók, valamint vizsgálhatók a valóságos vezérelt rendszereket tükröző absztrahált modellek statikus állapotai és dinamikus folyamatai is.

A szimulációs szoftverrendszer használatával közvetlenül összefüggő tanítási-tanulási egység cél- és feladatrendszere a következő:

- A tanuló legyen tisztában a PLC-hez tartozó informatikai alapokkal.
- Tudjon számítógép-virtuális PLC kapcsolatfelvételt elvégezni.
- Tudjon Ethernet-kommunikációs feladatokat elvégezni.
- Alkalmazni tudja a felprogramozási, feltöltési funkciókat.
- Alkalmazni tudja a használt PLC adottságait.
- Legyen képes program-hibakeresésre virtuális környezetben.
- A tanuló ismerje a létradiagramos programozási nyelvet, a kapcsolódó dokumentációs technikákat.
- Tudjon egyszerű PLC programot készíteni digitális jelek fogadására.
- Megfelelően alkalmazni tudja a programozási lépéseket.
- Tudjon aritmetikai és logikai műveleteket, valamint funkcióblokkokat kezelni.

A tanítási-tanulási egység feldolgozásához az alábbi tematikus előzmények szükségesek:

- A PLC-típus kiválasztásának szempontjai.
- Szenzorok és aktuátorok illesztése.
- PC-PLC kommunikáció kialakítása.
- IEC 61131-3 szabvány szerinti programozási nyelvek.
- Öntartások.
- Logikai vezérlés, élvezérlés.
- Időzítők és számlálók alkalmazása.

5 Irodalomjegyzék

Könyvek, jegyzetek:

Dr. Ajtonyi István – Dr. Gyuricza István (2007): *Programozható irányítóberendezések, hálózatok és rendszerek*, Budapest, Műszaki Könyvkiadó.

Andrew S. Tanenbaum – David J. Wetherall (2013): *Számítógép-hálózatok*, Budapest, Panem Könyvek.

Csík József – Váradi György (2007): *Irányítástechnikai gyakorlatok*, Budapest, Műszaki Könyvkiadó.

FESTO (2014): *PLC alapismeretek*, Tanfolyami jegyzet, Budapest, Festo Kft.

Farkas András – Nagy Lóránt – Tverdota Miklós (2010): *Automatika*, Budapest, Képzőművészeti Kiadó.

Koszár András: *A korszerű csúcstechnológiák témaköreinek szakelméleti oktatása*, Tanulmány, TÁMOP 4.1.1.C, 2014.

Lükő István – Molnár György (2015): *Szaktudásértékelési ismeretek villamos szakmacsoportos mérnökök számára*, Elektronikus könyv, BME Tanárképző Központ.

Dr. Szaladnya Sándor – Telek Péter (2009): *A pneumatikus automatizálás eszközei, a tervezés módszerei, exkluzív példatár*, Miskolc, Miskolci Egyetem

Gyártói dokumentáció:

CODESYS (2015): *CODESYS OPC Server V3, Installation and Usage*, 3S – Smart Software Solutions GmbH

Jogszabályok:

29/2016. (VIII. 26.) NGM rendelet a nemzetgazdasági miniszter hatáskörébe tartozó szakképesítések szakmai és vizsgakövetelményeiről szóló 27/2012. (VIII. 27.) NGM rendelet módosításáról

30/2016. (VIII. 31.) NGM rendelet a szakképzési kerettantervekről