

Debreceni Egyetem
Informatika Kar
Vállalati információs rendszerek szakirány

Festő gyártósor modellezése és vezérlése,
programozható logikai vezérlővel és szimulációja
Vijeo Citect 7.20 segítségével

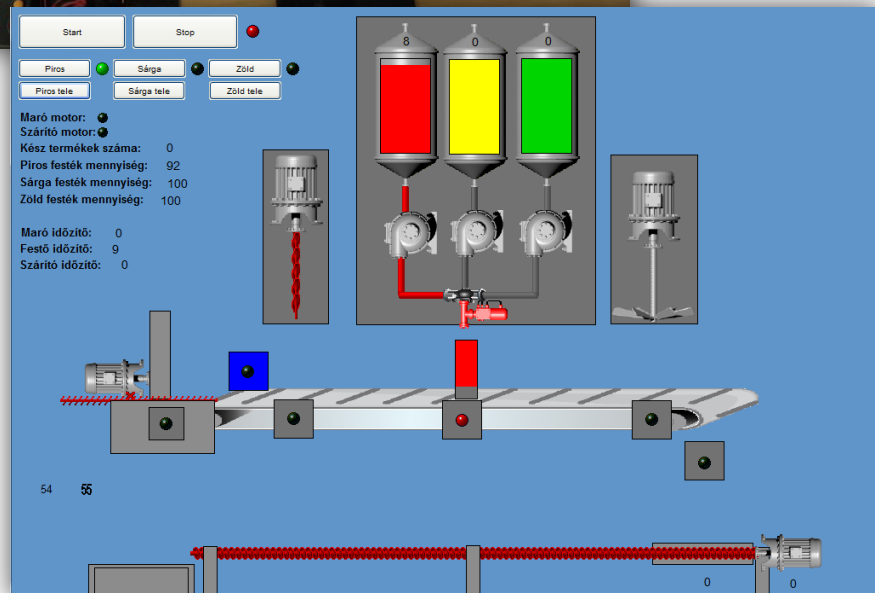
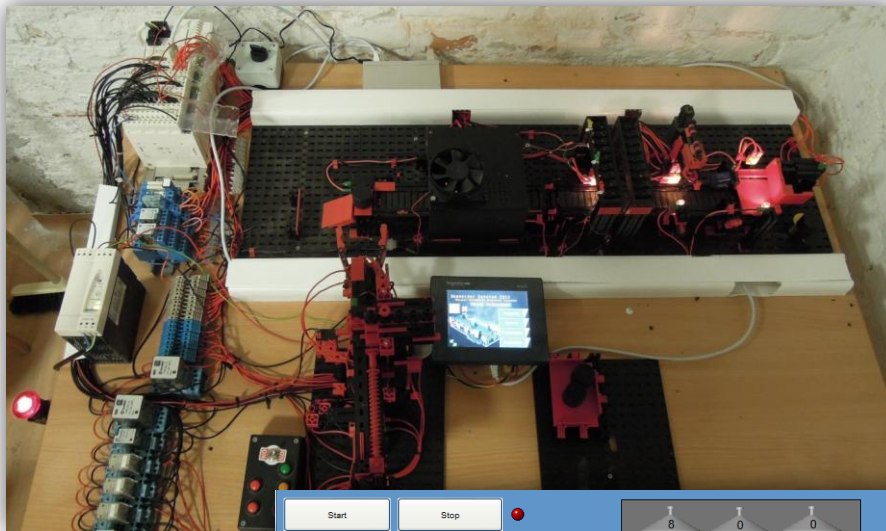
TÉMAVEZETŐ: **VITÉZ ATTILA**
TANÁRSEGÉD

KÉSZÍTETTE: **PAPP ZOLTÁN**
MÉRNÖK INFORMATIKUS
HALLGATÓ

Debrecen
2014

Nyilatkozat

Alulírott **Papp Zoltán**, a Debreceni Egyetem Informatikai Karának mérnök informatikus hallgatója kijelentem, hogy a „Festő gyártósor modellezése és vezérlése, programozható logikai vezérlővel és szimulációja Vijeo Citect 7.20 segítségével” című szakdolgozat, és a hozzá kapcsolódó PLC program, bekötés és szimuláció közös szerzemény, melyet **Vincze Jánossal** és **Fekete Attila Sándorral**, szintén a Debreceni Egyetem Informatikai Karának mérnök informatikus hallgatóival közösen készítettünk el.



Tartalomjegyzék

Nyilatkozat	2
Bevezetés	5
Gyártósor alkotóelemeinek bemutatása	7
A gyártósor elemei:.....	7
Gyártósor részei állomáskora bontva.....	7
Anyagkitoló egység	8
Induktív érzékelő	8
Maró egység	9
Festő részleg	9
Szárító részleg.....	10
Vezérlő és vészleállító gombok	11
Robotkar.....	11
Schneider M340 PLC.....	12
Fejlesztői környezet.....	15
Vijeo Citect 7.20.....	16
A megépített gyártósortól független gyártási folyamat szimulációja Vijeo Citect 7.20 segítségével.....	17
Kitoló motor vezérlése.....	17
Anyagmozgás (anyagmozg, anyagmozg2)	18
Érzékelők és hozzá tartozó motorjaik, időzítők.....	20
Festési folyamat	24
Festést vezérlő gombok:.....	25
Piros Gomb:.....	25
Sárga gomb:.....	27
Zöld gomb:	28
Csövek festési irányának beállítása:.....	29
A festő időzítő és az elfogyasztott festék mennyiség közötti kapcsolat:	29
Resetelés:	30
Fischertechnik gyártósor vezérlése és szimulációja Vijeo Citect 7.20 segítségével OPC szerverten keresztül	31
Az OPC szervert bemutatása	31

Schneider OPC Factory Server konfigurálása a Vijeo Citect 7.20-hoz.....	32
Vijeo Citect 7.20 konfigurációja.....	33
Fischertechnik gyártósor vezérlése és szimulációja Vijeo Citect 7.20 segítségével.	36
Gyártás indítás	37
PLC-s Start/Stop gomb	37
Szimulációs Start/ Stop gomb.....	37
Kitoló egység	38
Kitoló egység és az anyag érzékelő a szimulációban.....	40
Első futószalag.....	41
Maró egység.....	42
Festési részleg	43
Szelep:	44
Pirosra színezés	46
Sárgára színezés:	48
Zöldre színezés.....	49
PLC nyomógomb:	51
Második futószalag.....	52
Szárítás.....	52
Induktív érzékelő működése és a fémes anyagok megszámlálása.....	54
Anyagmozgás megvalósítása a szimulációban	54
Anyagláthatóság.....	56
Vészleállító	58
Stop és vészleállító gomb közötti működési különbségek.....	59
Információs panelek.....	59
Vezérlő panelek	60
Összefoglalás	61
Függelék	62
Irodalomjegyzék	65
Köszönetnyilvánítás	66

Bevezetés

Mérnök informatikai tanulmányaink során Vállalati Információs Rendszerek szakirányra szakosodva lehetőségünk adódott megismerni az informatika gyárakban és üzemekben lévő fontosságát.

Úgy gondoltuk, hogy a megszerzett elméleti tudásunk mellé szükséges az ehhez tartozó gyakorlati tudás elsajátítása is, ami az évek folyamán háttérbe szorult. A célunk egy működő gyártósor létrehozása és szimulációja volt, amivel szemléltetni tudjuk egy üzem gyártási folyamatát. Ezért diplomamunkánk témájának egy festő gyártósor modell elkészítését, vezérlését és szimulációját választottuk.

Az ötlet önálló labor 1 alkalmával dőlt el, de akkor még egy kuka robot segítségével akartuk véghezvinni a folyamatot. Később jött a lehetőség, hogy „Fischertechnik” eszközöket használjunk és ennek segítségével kezdtük el tervezni gyártósorunkat.

A projekt megvalósításához az egyetemtől egy Schneider M340 típusú PLC-t kaptunk, melynek elektronikai bekötését és beszerelését Vincze János és Fekete Attila Sándor valósította meg.

Vizsgálatomhoz, hogy az Én feladatommal is boldoguljak, Nekem is szükséges volt a bekötés alapvető ismerete.

Elsődleges feladatom a gyártósor vezérlése és szimulációjának létrehozása volt. A feladat megvalósítására a Vijeo Citect 7.20 nevű szimulációs programot választottam.

Munkámat ezen szimulációs program megismerésével kezdtem. A program működési alapjait néhány oktató videó segítségével sajátítottam el. Kezdetben az egyszerű LED-ek konfigurálását és anyagok mozgását valósítottam meg. Amint sikerült egyre több funkcióját megismerni a programnak, elkezdtem összerakni a gyártósorhoz szükséges szimulációs alkatrészeket. A gyártósor megvalósításának folyamatát kisebb részfeladatokra bontottam. Részfeladatonként haladva oldottam meg az egyes részfeladatokhoz tartozó részegységek vezérlését, szimulációját.

Először egy gyártósortól független szimulációt hoztam létre, amiben csak lokális változókat használva valósítottam meg a gyártósor működését. Ezen feladat befejeztével elég tapasztalatot és tudást szereztem, hogy OPC szerver segítségével kapcsolódva a PLC-hez, létrehozam a „Fischertechnik”-es gyártósor vezérlését, és a globális PLC változókat

használva, a szimulációs egységek egyidejű és azonos mozgását a valós „Fischertechnik”-es egységekkel.

A szakdolgozatom célja bemutatni a szimulációs egységek tervezésének és létrehozásának fázisait, valamint azok részletes konfigurációját és működését. Szakdolgozatomban kifejtem a gyártósori modell működését, bemutatom az egyes részegységeket és a hozzájuk tartozó szimulációs egységek beállításait. Továbbá bemutatom az általam elkészített programrészleteket, melyek segítségével vezérlöm a gyártósort szimulációban és valóságban is.

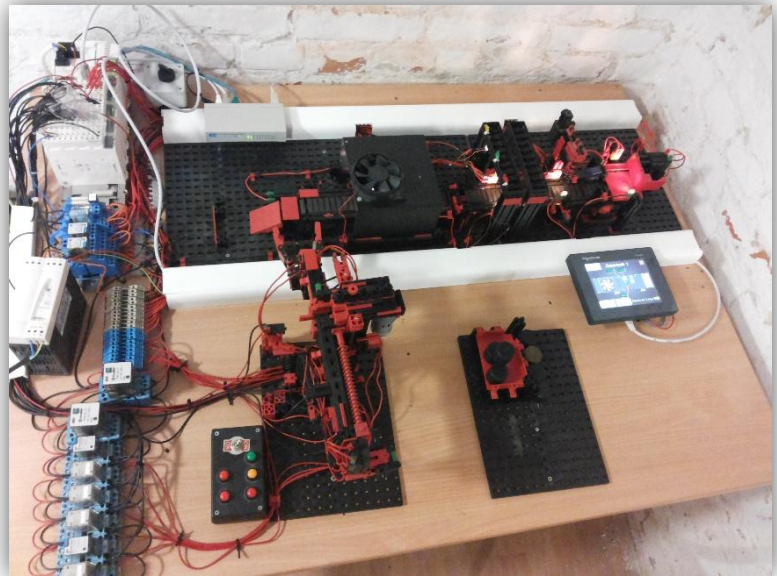
Gyártósor alkotóelemeinek bemutatása

A gyártósor elkészítéséhez a műszaki kartól kapott Fischertechnik-es alkatrészeket használtunk.

A „Fischertechnik” céget Németországban alapították, és ott is gyártja modelljeit, több mint negyven éve. A cég mottója, hogy megismertesse a fiatalokkal a technológia világát. Annyira elterjedtek a cég modelljei, hogy egy 30000 tagot számláló rajongói klub is létrejött.

A gyártósor elemei:

1. Kitoló egység
2. Induktív érzékelő
3. 2db futószalag
4. Maró egység
5. Festést szimuláló egység
6. Vezérlőgombok
7. Vészleállító gomb
8. Száritást szimuláló egység
9. RTT robotkar
10. Schneider M340 PLC
11. HMI panel („Magelis”)
12. 24V DC Tápegység
13. KTI Network switch



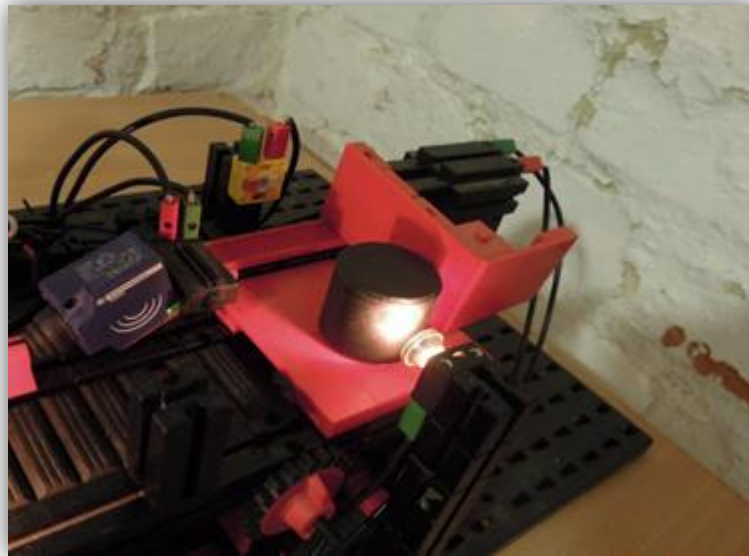
Gyártósor részei állomáskora bontva

1. állomás: Anyagkitoló egység
2. állomás: Induktív érzékelő
3. állomás: Maró egység
4. állomás: Festő részleg
5. állomás: Száritó részleg
6. állomás: Robotkar
7. állomás: Tároló egység



Anyagkitoló egység

A kitoló egység feladata, hogy az itt elhelyezett anyag érzékelő által érzékelt új terméket a futószalag felé tolja. Az egységet egy 24V-os motor hajtja meg ami egy relére van kötve, mivel két irányú mozgást is kell végeznie, de ahhoz, hogy ezt a feladatát elvégezze különböző feltételeknek teljesülnie kell. A kitoló egység végén elhelyezett végállás kapcsoló benyomása után indul az első futószalag működése.



Induktív érzékelő

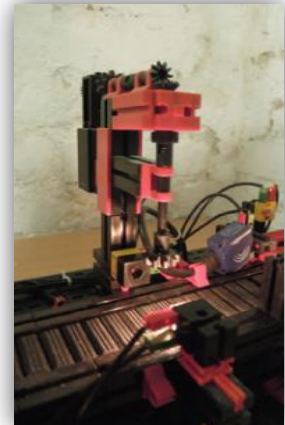
Induktív érzékelőnek egy Schneider Osiconcept típus lett beépítve, ami a fém érzékelésre szolgál. Az érzékelőn levő "tanítás" gomb megnyomásával megtörténik a termék automatikus konfigurálása az összes érzékelési helyzetre.



Maró egység

A 3. állomáson egy marási folyamatot bemutató egység található, ami egy optikai érzékelő segítségével érzékeli a termék érkezését. Ha érkezett anyag, akkor a szalag megáll és a marófej forgása elindul. Az egységet egy 24V-os motor mozgatja.

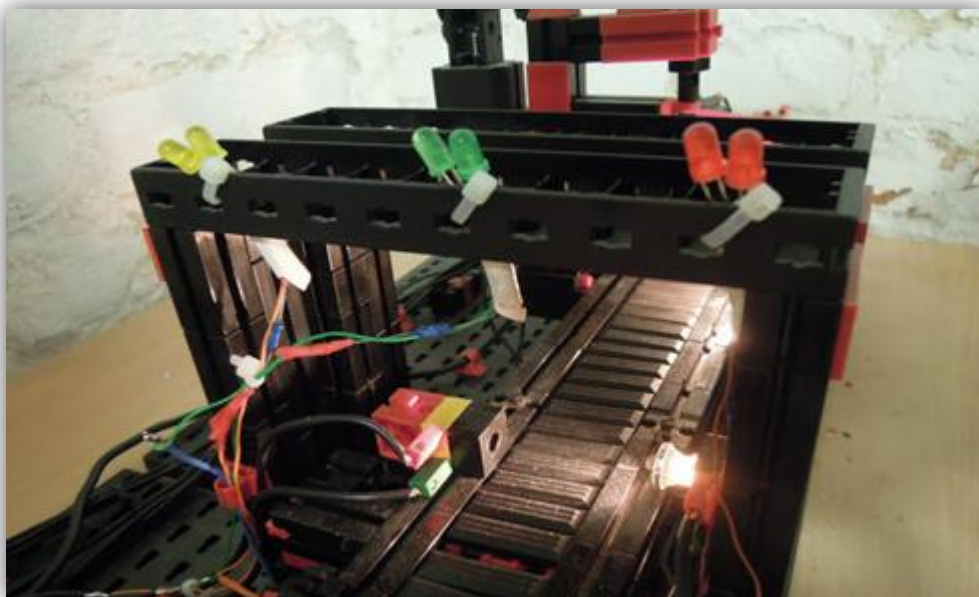
Marás után a futószalag a megmunkált terméket tovább szállítja a festési részleghez.



Festő részleg

A festési részlegnél ugyancsak egy optikai érzékelő található, ami ha érzékel megállítja a futószalagot. A festést szimbolizáló "kapu" tetején 3 LED-et helyeztünk el (piros, sárga, zöld), azt szimbolizálva, hogy a terméket milyen színűre festjük. A festék színét megadhatjuk a PLC-s nyomógombokkal vagy a szimulációban létrehozott gombokkal.

Festés után az első szalag újraindul és ezzel egy időben indul a szárító egység futószalagja is. Az első szalag mozgását a szárító érzékelő érzékelése állítja meg.

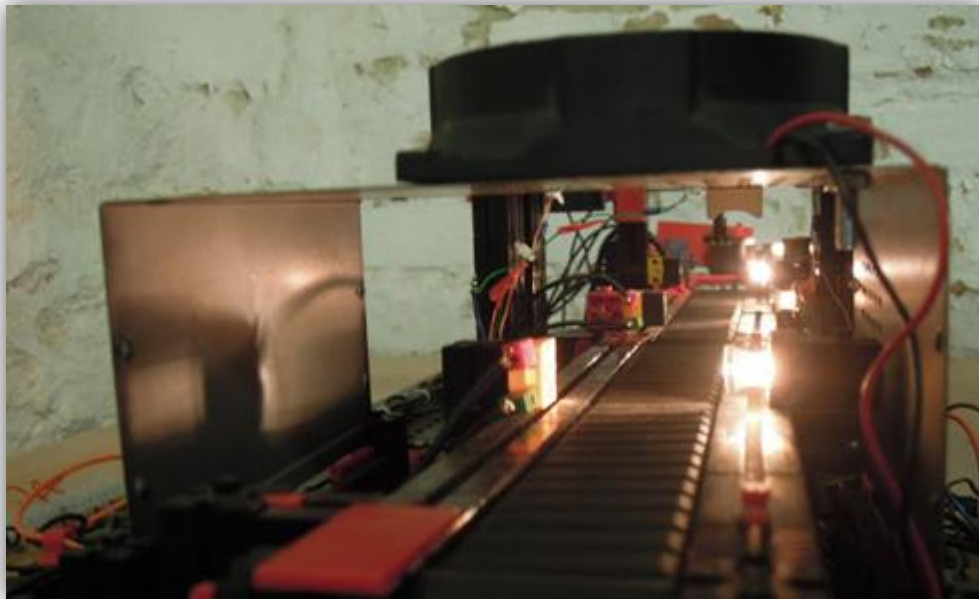
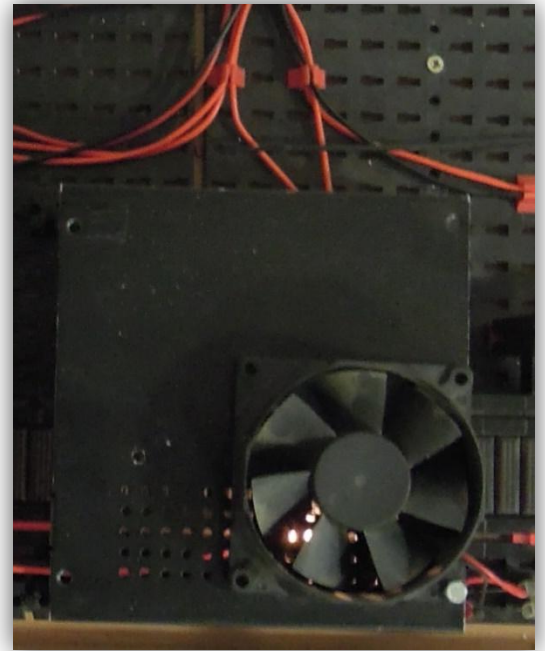


Szárító részleg

A szárító egység egy asztali számítógép tápegységének a burkolatából lett elkészítve, tetejére pedig egy tápegység hűtő ventilátora lett felfűrva.

A szárító résznél elhelyezett optikai érzékelő, termék érzékelése esetén, megállítja a futószalagot. A szárító rész tetején elhelyezett ventilátor elindul a szárítási folyamatot szimbolizálva.

Szárítás után a futószalag elindul és a termék a futószalag végéhez ér.



Vezérlő és vészleállító gombok

A vezérlő gombok segítségével tudjuk a gyártást elindítani, leállítani, illetve beállítani a kívánt festési szint (piros, sárga, zöld).

A Vészleállító (beragadó) gomb megnyomásakor minden éppen futó folyamat megáll, és csak akkor indul el, ha a gomb újból vissza nem kerül alap állapotában. Ha ez megtörtént a start gomb megnyomásakor minden ugyan onnan indul, ahol abbahagyta.



Robotkar

A robotkar 2 tengelyen lineáris mozgásra képes: le-fel, előre-hátra, és egy tengelyen pedig rotációs mozgásra: jobbra-balra irányuló elfordulást tud végezni. A vízszintes lineáris tengely végén található egy fogó, mely segítségével megfogja a gyártásból kikerülő kész terméket, és a tárolására kijelölt helyre teszi.



Schneider M340 PLC

A gyártósor vezérlését mindenképpen PLC segítségével szeretnénk volna megoldani. Ezért esett a választás a Schneider cég Modicon M340 kompakt PLC-re. Ideális megoldás az ipari automatizálás területén fellépő automatizálási feladatok megoldására. Külön szempont volt számunkra, hogy a feladatot olyan eszközökkel oldjuk meg, amellyel egy összetettebb, nagyobb volumenű valós automatizálási feladat megoldása is lehetséges.



Teljesítmény

- 7000 utasítás/ms
- 4 MB programmemória
- 256 KB adatmemória

Kompakt felépítés

- 3 db, processzorba épített kommunikációs port
- Magasság x szélesség x mélység = 100 x 32 x 93 mm
- Nagy sűrűségű, 64 csatornás, diszkrét I/O-modulok 32 mm széles formátumban

Beépített portjainak

- CANopen gépi- és terepi busz
- Ethernet TCP/IP hálózat – Transparent Ready
- Modbus soros kapcsolat vagy karaktermód
- Távoli hozzáférés STN, GSM, rádió vagy ADSL útján

További hasznos funkciói

- Számlálómodulok használatra kész funkciókkal
- Mozgásvezérléshez szánt funkcióblokk-könyvtár. MFB (Motion Function Blocks mozgásvezérlő funkció-blokkok) a PLCopen szabvány szerint
- Fejlett, gépi berendezésekre orientált folyamatvezérlő blokk-könyvtár
- Az USB port az alapkiépítéshez tartozik
- Beágyazott webkiszolgáló
- Recept fájlkezelés FTP protokollon keresztül
- „Plug and Load” SD memóriakártya
- Beépített elem nélkül

Teherbíró-képesség

- Üzem közbeni modulcserét (Hot-Swap) lehetővé tevő hátlapfelépítés
- Ütésállóság, rezgésállóság, hőmérséklet, magasság és elektromos zavartűrés tekintetében magasan túlteljesíti az IEC szabványok követelményeit.

Az általunk használt PLC konfigurációja a következő:

Egy hátlapos konfiguráció, 6 helyes rekesszel. Ennél a hátlapnál a tápegység modul mindig a CPS helyre kerül. A processzor mindig a 00 helyre, és 01-05 helyekre pedig az I/O modulokat helyezhetjük el.

Mi esetünkben a processzor BMX P34 2030 típusú:

- CAN RUN (CAN működés)
- LED (zöld): a beépített gépi-, illetve terepi busz működőképes
- CAN ERR (CAN hiba)
- LED (piros): a beépített gépi-, illetve terepi busz hibás
 - Mini USB-csatlakozó a programozó terminál (vagy a Magelis XBT GT kezelői interfész) számára
 - Egy flash memóriakártyával beültetett nyílás az alkalmazás biztonsági mentéséhez (a nyílás felett elhelyezkedő LED jelzi a memóriakártya felismerését, illetve az ahhoz történő hozzáférést)

- Egy RJ45 típusú csatlakozó az Ethernet TCP/IP
- 10BASE-T/100BASE-TX hálózatra történő csatlakozáshoz

I/O modulok:

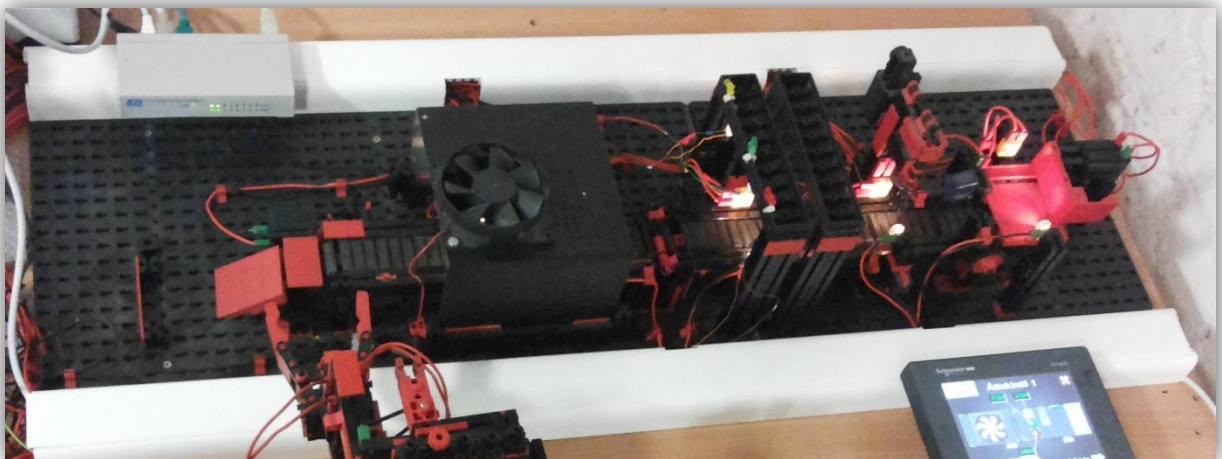
2db BMX DDO 1602 szilárdtest-kimenetű modul:

- Kimenetek száma:16
- Feszültség: 24 DC
- Áram: 0,5 A
- Túlfeszültség elleni védelem Transil diódával
- Fordított bekötés elleni védelem

2db BMX DDI 1602

- Bemenetek száma: 16
- Feszültség: 24 DC
- Áram: 3,5mA
- Bemeneti védelem
- Fordított bekötés elleni védelem

A gyártósor egészében:



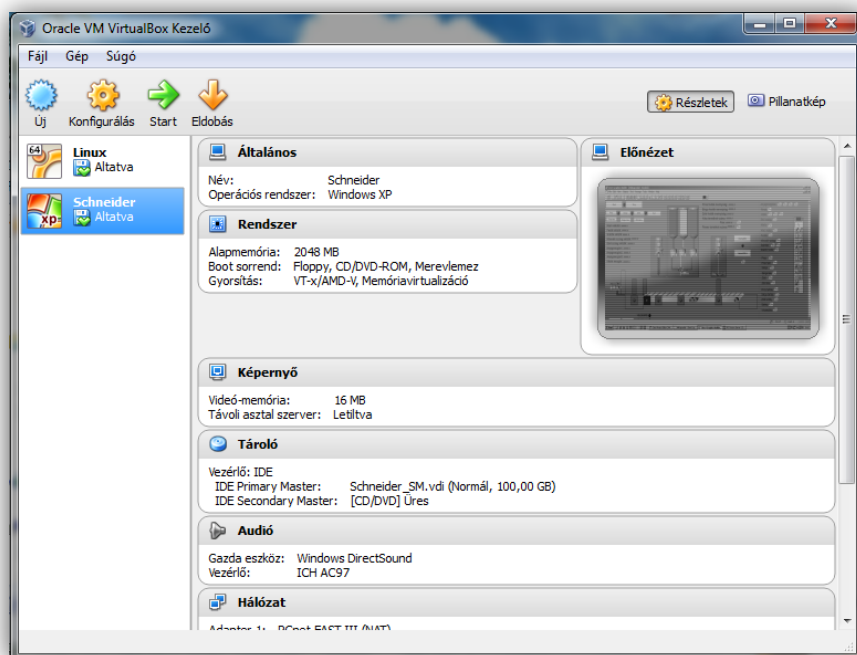
Fejlesztői környezet

Fejlesztői környezetként Fekete Attila Sándor által létrehozott virtuális gépet használtuk, melyen Windows XP operációs rendszer futott. Ezt követően telepítette a munkánkhoz szükséges programokat, amelyeket a munkahelyének vezetői biztosítottak nekünk.

A virtuális gép létrehozásához a Virtualbox nevű programot használtuk, ami ingyenesen letölthető a <https://www.virtualbox.org/> oldalról.

A gyártósor megvalósítására és vezérlésére használt programok:

- Vijeo Citect 7.20
- OPC Factory Server
- Unity Pro XL,
- Vijeo Designer,
- Open Office



Vijeo Citect 7.20

A szimuláció megvalósításához a Vijeo Citect 7.20 nevű programot választottam.

A „speedlink” technológiája lehetővé teszi a Unity programozó szoftverrel való szinkronizációt, melynek segítségével könnyen importálhatjuk az újonnan létrehozott változókat egyik platformról a másikra. A 7.20-as verzióban lehetőség van az OPC Factory Server könnyű integrálhatóságára, aminek segítségével könnyen kapcsolódhatunk a különböző Schneider típusú PLC-khez, esetünkben a Schneider M340-hez.

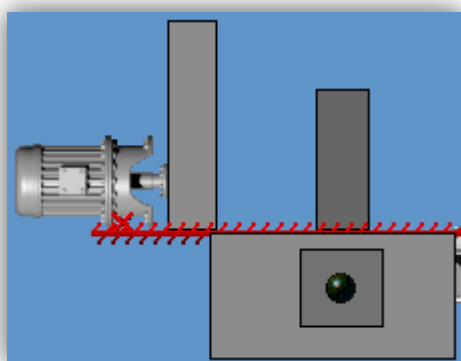
Továbbá támogatja a virtuális gépeken való használatot, támogatja a Vijeo Designer-rel való kapcsolatot.



A megépített gyártósortól független gyártási folyamat szimulációja Vijeo Citect 7.20 segítségével

A start gomb megnyomásával elindul a gyártási folyamat, amit a start változó vezérel. A start gomb a start változót 1-re, a stop gomb 0-ra billenti. A programkód működését az anyagmozgás befolyásolja, tehát minden részegység- kivéve a kitoló motort- működését az anyagmozg változó szabályozza.

Kitoló motor vezérlése



A Start gomb megnyomása után indul a kitoló motor vezérlése.

A kitoló motor mozgását megvalósító változó INT típusú, 0 és 100 közötti értéket vehet fel.

Működését - miszerint a kitoló kifelé és befelé mozgásra képes - a Kitolo_Motor_Kint digitális változó is befolyásolja, aszerint, hogy értéke 0 vagy 1.

Kitolo_Motor_Kint: Ha a kitoló motor kezdő állapotban van, tehát értéke 0, akkor a Kitolo_Motor_Kint változó értéke 0.

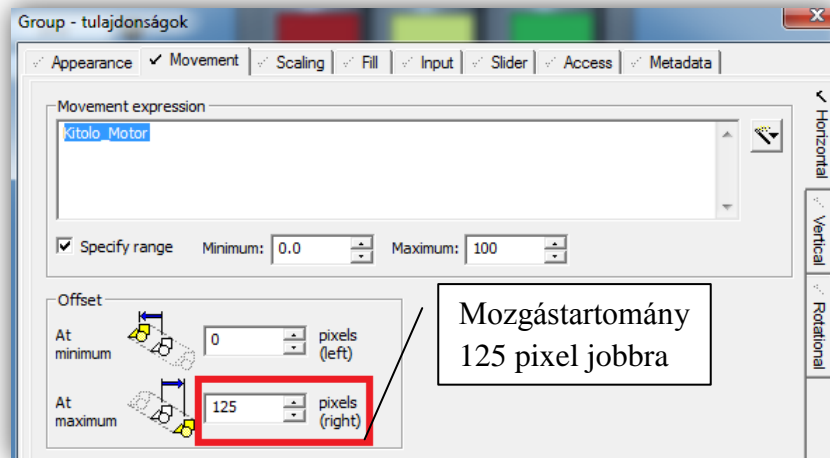
Ha a kitoló motor mozgása során eléri a végállapotot, tehát a 99-et, akkor a Kitolo_Motor_Kint változó 1-re módosul.

```
IF Kitolo_Motor=0 THEN
  Kitolo_Motor_Kint=0;
END
IF Kitolo_Motor=99 THEN
  Kitolo_Motor_Kint=1;
END
```

Ha a start értéke 1, a Kitolo_Motor_Kint értéke 0 és az anyagmozg2 értéke 0, akkor kezdődik a kitoló szerkezet mozgása 3 egységenként. Ha elérte a végállapotot, a 99-et, a Kitolo_Motor értéke csökkenni kezd (a kitoló szerkezet visszafelé mozog) 1 egységenként.

```
IF start=1 AND Kitolo_Motor_Kint=0 AND anyagmozg2=0 THEN
  Kitolo_Motor=Kitolo_Motor+3;
END
IF start=1 AND Kitolo_Motor_Kint=1 THEN
  anyagmozg=anyagmozg+1;
  Kitolo_Motor=Kitolo_Motor-1;
END
```

A kitoló motor beállítása:



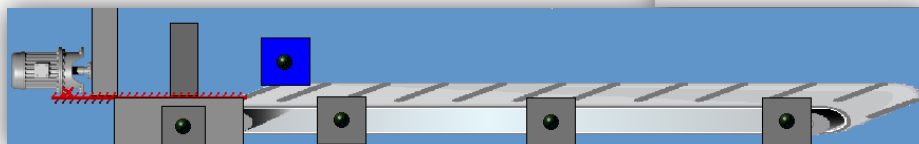
Anyagmozgás (anyagmozg, anyagmozg2)

Int típusú változó, értéke 0-100-ig terjed. Anyagmozgás akkor kezdődik, amikor a kitoló egység a 42. helyzetben áll, értéke nő 1-el. A következő ugrása a kitoló 48. helyzetében történik, majd az 54, 60, 63, 69, 72, 78, 84 és 87. helyzetben. A szimulációban így lehetséges szemléltetni, hogy a kitoló rész folyamatosan maga előtt tolja az anyagot. Ezekben a lépésekben a Kitolo_Motor_Kint értékének 0-nak kell lennie. Így a kitoló visszafele mozgása esetén - amikor a Kitolo_Motor_Kint értéke 1- nem zavarja a futószalagon történő anyagmozgást.

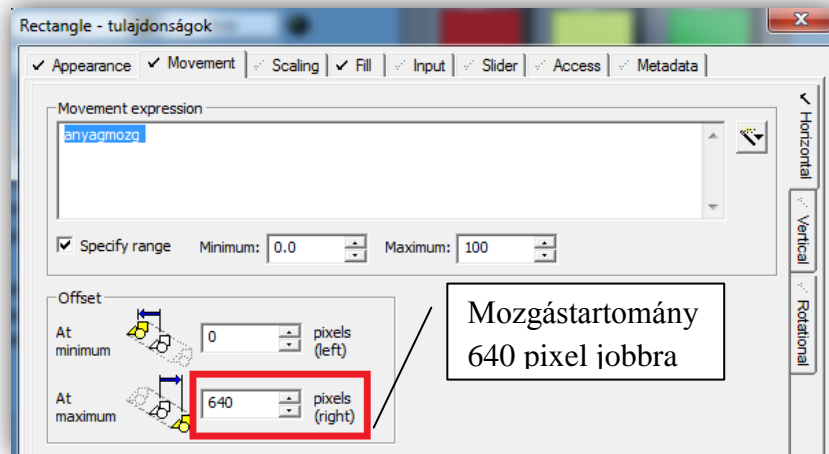
Ha az anyag megérkezett a futószalaghoz, vagyis a kitoló elérte a 99-et (Kitolo_Motor_Kint=1), az anyagmozgás folytatódik tovább 1 egységenként, míg el nem éri a 100-at. Ezzel a futószalag működését szimulálom.

```

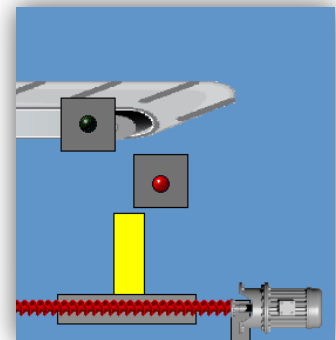
IF start=1 AND Kitolo_Motor_Kint=0 THEN
  IF Kitolo_Motor=42 THEN
    anyagmozg=anyagmozg+1
  END
  IF Kitolo_Motor=48 THEN
    anyagmozg=anyagmozg+1
  END
  IF Kitolo_Motor=54 THEN
    anyagmozg=anyagmozg+1
  END
  IF Kitolo_Motor=60 THEN
    anyagmozg=anyagmozg+1
  END
  IF Kitolo_Motor=63 THEN
    anyagmozg=anyagmozg+1
  END
  IF Kitolo_Motor=69 THEN
    anyagmozg=anyagmozg+1
  END
  IF Kitolo_Motor=72 THEN
    anyagmozg=anyagmozg+1
  END
  IF Kitolo_Motor=78 THEN
    anyagmozg=anyagmozg+1
  END
  IF Kitolo_Motor=84 THEN
    anyagmozg=anyagmozg+1
  END
  IF Kitolo_Motor=87 THEN
    anyagmozg=anyagmozg+1
  END
END
    
```



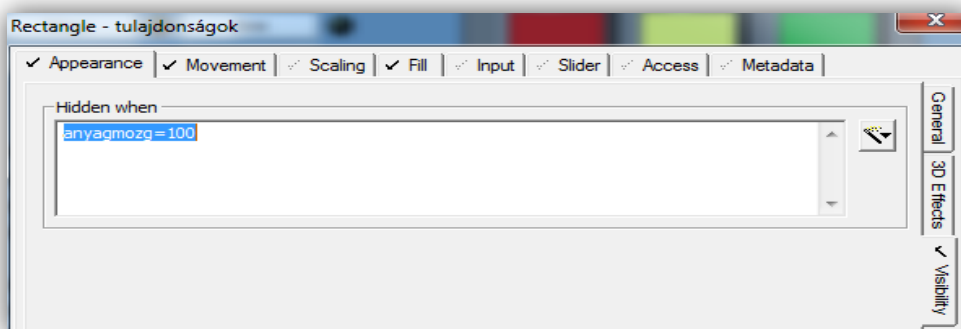
Anyagmozgás beállítása:



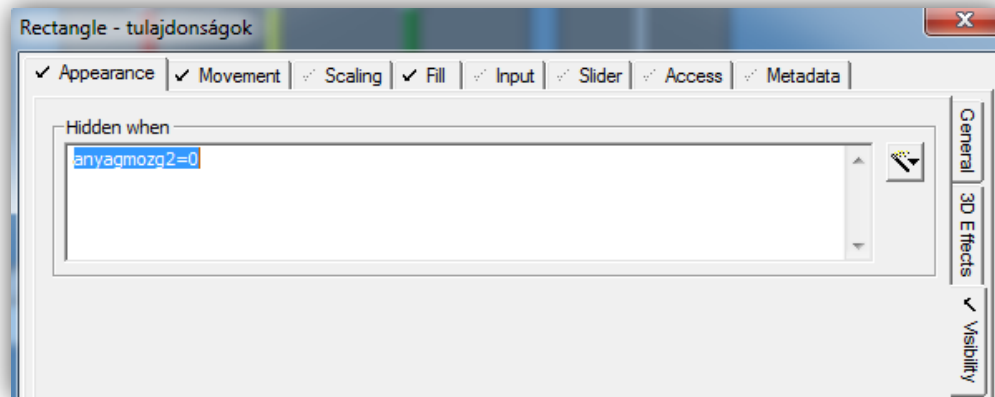
Amikor az anyag végigér az első futószalagon, tehát az anyagmozg változó eléri 100-as értéket, eltűnik (Hidden when anyagmozg=100). Ezzel egy időben a transzport egységen megjelenik egy ugyanazon tulajdonságokkal rendelkező anyag, amelynek mozgását az anyagmozg2 változó befolyásolja. Így sikerült azt a hatást elérni, hogy az anyag az első futószalagon végighaladva áthelyeződött a transzport egységre. A transzport egységen lévő anyag mindaddig láthatatlan, míg értéke 0 (Hidden when anyagmozg2=0).



Az anyagmozg láthatatlanságának beállítása:



Az anyagmozg2 láthatatlanságának beállítása:

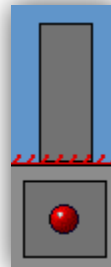


Érzékelők és hozzá tartozó motorjaik, időzítőik

Az érzékelők szimulációjának megvalósítása LED lámpákkal valósult meg.

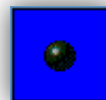
Anyag érzékelő: A kitoló egységen helyezkedik el, jelezve azt amikor az anyag a kezdő állapotban van. Ha az anyagmozgás 0-án van, akkor az Anyag_Erzekelo 1, tehát a LED világít, egyébként 0.

```
IF anyagmozg=0 THEN
  Anyag_Erzekelo=1
ELSE
  Anyag_Erzekelo=0
END
```



Induktív érzékelő: LED-je a 15. anyagmozgásnál villan fel.

```
IF anyagmozg=15 THEN
  Induktiv_Erzekelo=1
ELSE
  Induktiv_Erzekelo=0
END
```



Maró érzékelő:

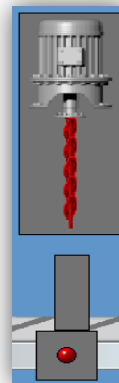
Ha az anyagmozgás a 24. pozícióban tart, a Maro_Erzekelo 1-re vált, LED-je pedig felvillan. A start értéke 0 lesz, tehát a folyamat megáll, elindul a maró motor időzítője. Ha eléri a 10-et a folyamat ismét elindul, a maró időzítő pedig lenullázódik. A szimulációban ez a rész azt eredményezi, hogy amikor az anyag a maró állomásához ér, akkor a futószalag megáll, vár 10 egységnyi időt, majd ismét elindul.

```
IF anyagmozg=24 THEN
  Maro_Erzekelo=1
ELSE
  Maro_Erzekelo=0
END
IF Maro_Erzekelo=1 THEN
  start=0
  Maro_Idozito=Maro_Idozito+1
  IF Maro_Idozito=10 THEN
    start=1
    Maro_Idozito=0
  END
END
```

A várakozási idő alatt elindul a maró motor működése.

Maró motor: A maró motor addig dolgozik, míg a maró érzékelő érzékel, tehát míg a maró időzítő el nem éri a 10 értéket.

```
IF Maro_Erzekelo=1 THEN
  Maro_Motor=1
ELSE
  Maro_Motor=0
END
```

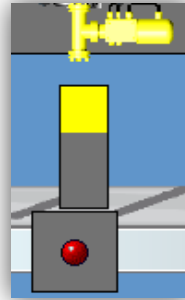


Festő érzékelő: Az anyag következő állomása a festési részleg, ahol szintén egy érzékelő jelzi az anyag érkezését. Az anyagmozgás 55. helyzetében a festő rész érzékelője felvillan, a folyamat megáll (start=0), majd elindul a festő időzítő 0-tól 10 időegységig. Ezután a folyamat ismét elindul (start=1), a festő időzítője pedig visszaáll 0-ra.

```
IF anyagmozg=55 THEN
  Festo_Erzekelo=1
ELSE
  Festo_Erzekelo=0
END
IF Festo_Erzekelo=1 THEN
  start=0
  Festo_Idozito=Festo_Idozito+1
  IF Festo_Idozito=10 THEN
    start=1
    Festo_Idozito=0
  END
END
```

Szelep: A festési részlegnél történő várakozás során történik az anyag festése a festékszóró szelep segítségével. (A festési folyamat részletes bemutatása később olvasható.)

```
IF Festo_Erzekelo=1 THEN
  szelep=1
ELSE
  szelep=0
END
```



Szárító érzékelő: A maró és a festő részhez hasonlóan működik a szárító rész is. Érzékelője a 90. anyagmozgásnál villan fel, a folyamat megáll, elindul a szárító időzítő 20 időegységig, majd a folyamat ismét elindul.

```
IF anyagmozg=90 THEN
  Szarito_Erzekelo=1
ELSE
  Szarito_Erzekelo=0
END
IF Szarito_Erzekelo=1 THEN
  start=0
  Szarito_Idozito=Szarito_Idozito+1
  IF Szarito_Idozito=20 THEN
    start=1
    Szarito_Idozito=0
  END
END
```

Szárító motor: A szárítást egy motorra felszerelt ventilátor szimulálja. Akkor működik, ha a szárító érzékelő érzékel. Működése során a ventilátor forgó mozgást végez.

```
IF Szarito_Erzekelo=1 THEN
  Szarito_Motor=1
ELSE
  Szarito_Motor=0
END
```



Transzport érzékelő: A transzport érzékelő LED-je az anyagmozg2 változó 5. értékénél villan fel.

```
IF anyagmozg2=5 THEN
  Transzport_Erzekelo=1
ELSE
  Transzport_Erzekelo=0
END
```

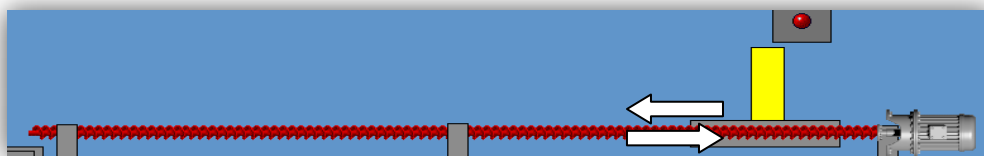
Transzport motor: A transzport egység mozgatásához egy segédváltozó bevezetése volt szükséges, mely az anyagmozg változó 100-as értékénél 1-re vált, az anyagmozg2 változó 100-as értékénél pedig 0-ra.

```
IF anyagmozg=100 THEN
  seged=1
END
IF anyagmozg2=100 THEN
  seged=0
END
```

```
IF start=1 AND seged=1 THEN
  anyagmozg2=anyagmozg2+5
  Transzport_Motor=Transzport_Motor+5
END
IF start=1 AND seged=0 THEN
  anyagmozg2=0
  Transzport_Motor=Transzport_Motor-1
END
```

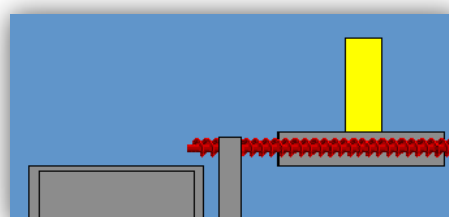
Erre azért volt szükség, hogy a transzport egység két irányba való mozgását váltogassuk. Ha a start és a seged változó értéke 1, akkor megkezdődik az anyag a transzport egységgel való mozgása. Ezzel azt a hatást érjük el, hogy transzport egység szállítja az anyagot. Ha az anyagmozg2 változó értéke 100 lesz, tehát a seged változó 0, akkor az anyagmozg2 változó értéke 0 lesz. Ezzel azt érjük

el, hogy az anyag visszaugrik a transzport egység elejére, ahol láthatatlan (Hide when anyagmozg2= 0). A transzport egység pedig visszafele mozgást végez, amíg eléri a 0 pozíciót.



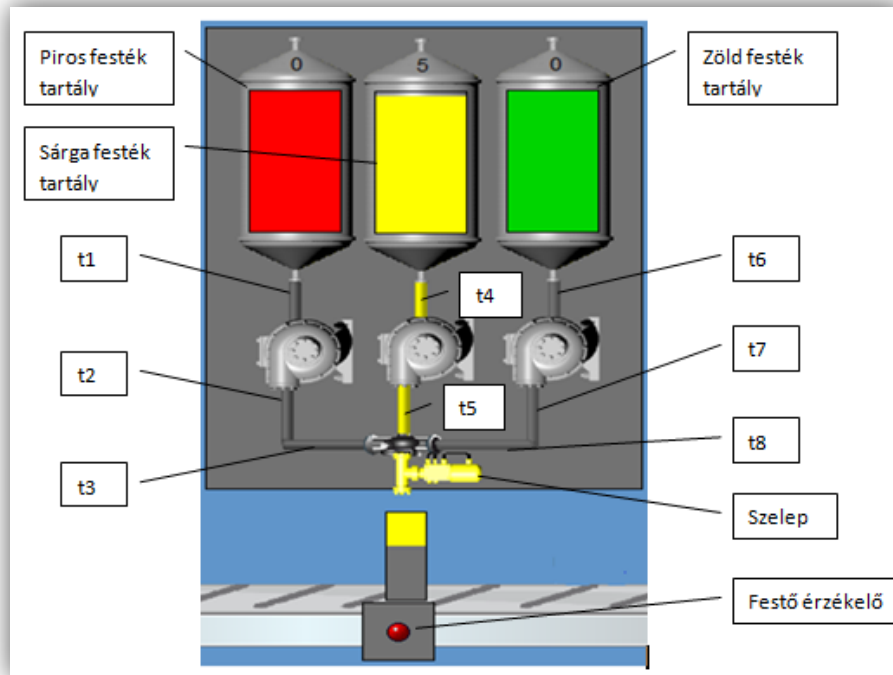
Kész termék: A transzport egység végén a kész termék egy tárolóba kerül. A kész termékeket a Kesz_Termek változó számolja.

```
IF Transzport_Motor=100 THEN
  Kesz_Termek=Kesz_Termek+1
END
```

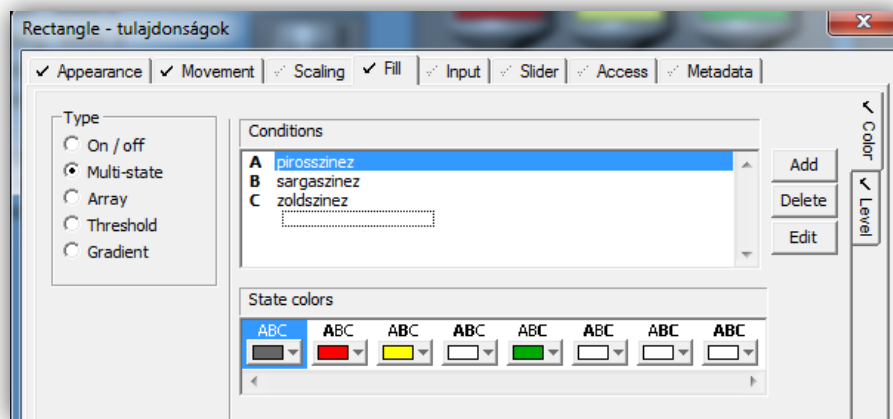


Festési folyamat

Festő állomás részei, elnevezései:

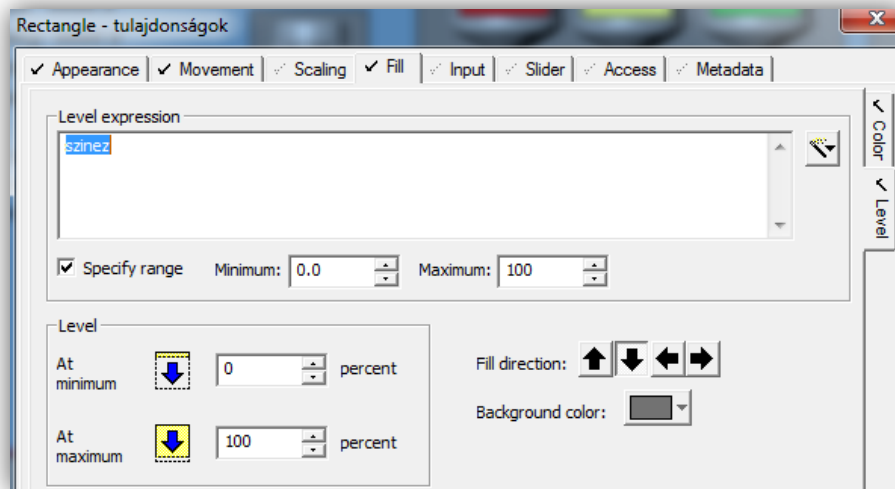


A gombok működésének beállítása előtt szükségesek az alábbi képeken látható beállítások, amit az anyag kitöltési tulajdonságánál tudunk beállítani:



Az anyag kitöltése 3 színnel történik (piros, sárga, zöld), ezért állapotát Multi-state-re kell állítani. Ezt követően meg kell adnunk a 3 lehetséges színezést, amihez 3 változó hozzárendelése szükséges (pirosszinez, sargaszinez, zoldszinez).

Az alábbi beállítás az anyag színezésének irányát és maximum értékét mutatja. Az egész színezés folyamatát a szinez változó irányítja.



Festést vezérlő gombok:

Piros Gomb: A piros gomb lenyomása esetén a t5 (sárga festék csöve), a t7 és a t8 (zöld festék csöve) változó értéke 0 lesz, ami a szimuláció folyamán azt eredményezi, hogy azok üresek lesznek, nem lesznek feltöltve festékekkel. Ezt követően a t1, t2 (felülről lefelé) és t3 (balról jobbra) csövek feltöltődnek piros festékekkel egymás után 10 egységenként, jelezve a festék áramlását a csövek belsejében. Ha a t3-as cső is feltöltődött és ha a szelep értéke 1, ami akkor történik ha a festő érzékelő érzékel, a pirosszinez változó átadódik a szinez változónak, amely 10 egységenként növekedni kezd. Ez azt eredményezi, hogy a szelep alatt álló anyag felülről lefelé 10 egységenként pirosra színeződik.

```

IF Piros_Gomb=1 THEN
    t5=0
    t8=0
    t7=0
    t1=t1+10
    IF t1=100 THEN
        t2=t2+10
        IF t2=100 THEN
            t3=t3+10
            IF t3=100 AND szelep=1 THEN
                pirosszinez=szinez
                szinez=szinez+10
            END
        END
    END
END
END
END

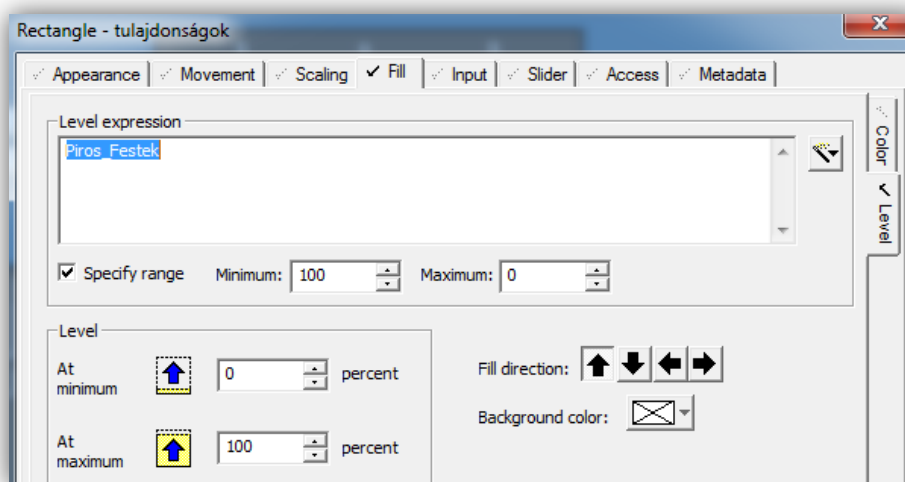
```

Piros festék tartály: A piros festék tartály szintjét a Piros_Festek változó befolyásolja.

Ha a t3-as cső feltöltődött és ha a szelep nyitva van (értéke 1), akkor a piros festék értéke nő (kezdőállapota 0).

```
IF t3=100 AND szelep=1 THEN
  Piros_Festek=Piros_Festek+1
END
```

A piros festék értékének növekedése a hozzá tartozó tartály szintjének csökkenését eredményezi, amihez a tartály képen látható beállítása szükséges:



Ezek alapján a tartályban lévő piros festék mennyiség kiszámítása: $100 - \text{Piros_Festek}$.

Piros tartály feltöltése: A piros tartály akkor töltődik fel, ha megnyomjuk a Piros tele gombot vagy ha a piros festék kifogyott a tartályból. A piros festék 0 szintje esetén szükséges a Piros_tele változó nullázása, mert a továbbiakban is megtörténhet a Piros tele gomb megnyomása.

```
IF Piros_Tele=1 OR Piros_Festek=0 THEN
  Piros_Festek=0
END
IF Piros_Festek=0 THEN
  Piros_Tele=0
END
```

Sárga gomb: Működése hasonló a piros gombéhoz. Megnyomása után a t2, t3 (piros festék csövei) és a t7, t8 (zöld festék csövei) 0-ra állítódnak, így üresek lesznek. Ezután a t4-es cső 10 egységenként felülről lefelé befestődik, majd ugyanígy a t5-ös cső is. Ha a t5-ös cső feltöltődött és ha a szelep nyitva van, akkor a sargaszinez változó átadódik a szinez változónak, ami 10 egységenként növekszik, melynek eredményeképp a festő érzékelőnél álló anyag felülről lefelé 10 egységenként sárgára színeződik.

```
IF Sarga_Gomb =1 THEN
  t2=0
  t3=0
  t8=0
  t7=0
  t4=t4+10
  IF t4=100 THEN
    t5=t5+10
    IF t5=100 AND szelep=1 THEN
      sargaszinez=szinez
      szinez=szinez+10
    END
  END
END
```

Sárga festék tartály: A sárga festék tartály szintjét a Sarga_Festek változó befolyásolja.

Ha a t5-ös cső feltöltődött és ha a szelep nyitva van (értéke 1), akkor a sárga festék értéke nő.

```
IF t5=100 AND szelep=1 THEN
  Sarga_Festek=Sarga_Festek+1
END
```

A sárga festék értékének növekedése a sárga tartály szintjének csökkenését eredményezi, aminek a beállítása hasonló a piros tartályéhoz (lásd a piros tartálynál).

A sárga tartályban lévő sárga festék mennyiség kiszámítása: 100-Sarga_Festek.

Sárga tartály feltöltése: A sárga tartály akkor töltődik fel, ha megnyomjuk a Sárga tele gombot vagy ha a sárga festék kifogyott a tartályból. A sárga festék 0 szintje esetén szükséges a Sárga_tele változó nullázása, mert a továbbiakban is megtörténhet a Sárga tele gomb megnyomása.

```
IF Sarga_Tele=1 OR Sarga_Festek=100 THEN
  Sarga_Festek=0
END
IF Sarga_Festek=0 THEN
  Sarga_Tele=0
END
```

Zöld gomb: A zöld gomb megnyomása után a t5 (sárga festék csöve) és a t2, t3 (piros festék csövei) üresek lesznek, majd a zöld festék csövei feltöltődnek 10 egységenként. A t6 és t7 felülről lefelé, a t8 pedig jobbról balra. Ha a t8-as cső tele van és ha a szelep nyitva van, a zoldszinez változó átadódik a szinez változónak, ami 10 egységenként növekszik. Így a szelep alatt lévő anyag felülről lefelé 10 egységenként zöldre színeződik.

```
IF Zold_Gomb =1 THEN
t5=0
t3=0
t2=0
t6=t6+10
IF t6=100 THEN
t7=t7+10
IF t7=100 THEN
t8=t8+10
IF t8=100 AND szelep=1 THEN
zoldszinez=szinez
szinez=szinez+10
END
END
END
END
```

Zöld festék tartály: A zöld festék tartály szintjét a Zold_Festek változó befolyásolja.

Ha a t8-as cső feltöltődött és ha a szelep nyitva van (értéke 1), akkor a zöld festék értéke nő.

```
IF t8=100 and szelep=1 THEN
Zold_Festek=Zold_Festek+1
END
```

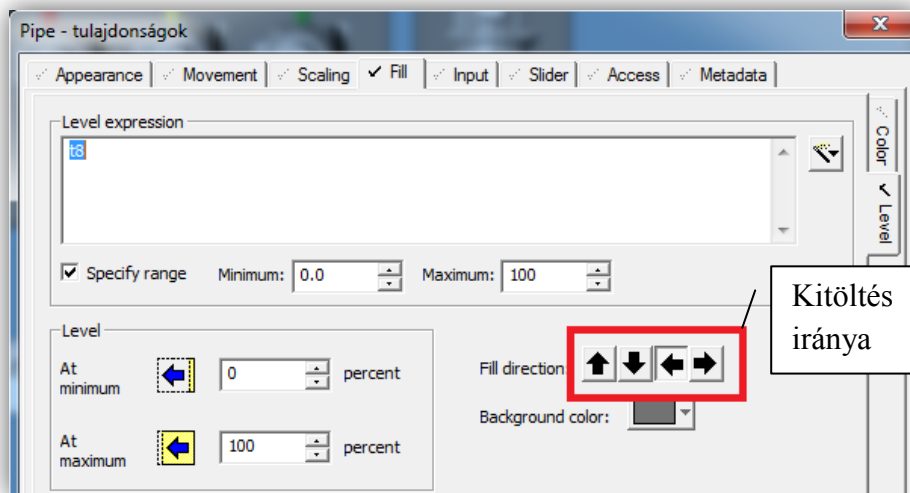
A zöld festék értékének növekedése a zöld tartály szintjének csökkenését eredményezi, aminek a beállítása hasonló a piros tartályéhoz (lásd piros tartálynál).

A zöld tartályban lévő zöld festék mennyiség kiszámítása: $100 - \text{Zold_Festek}$.

Zöld tartály feltöltése: A zöld tartály akkor töltődik fel, ha megnyomjuk a Zöld tele gombot vagy ha a zöld festék kifogyott a tartályból. A zöld festék 0 szintje esetén szükséges a Zold_tele változó nullázása, mert a továbbiakban is megtörténhet a Zöld tele gomb megnyomása.

```
IF Zold_Tele=1 OR Zold_Festek=0 THEN
Zold_Festek=0
END
IF Zold_Festek=0 THEN
Zold_Tele=0
END
```

Csövek festési irányának beállítása: A csövek elhelyezkedését figyelembe véve különböző irányból kell lefesteni azokat. A t1, t2, t4, t5, t6 és t7 csöveket felülről lefelé, a t3 csövet balról jobbra, míg a t8 csövet pedig jobbról balra.



A képen a t8 cső festési beállítása látható.

A festő időzítő és az elfogyasztott festék mennyiség közötti kapcsolat:

Az anyag a festő érzékelőhöz érve 10 időegységre megáll. Ez idő alatt történik a festés.

A megállási idő és az elfogyasztott festék mennyisége ugyanannyi, mivel a szelep csak addig van nyitva amíg a festő érzékelő érzékel, a festék pedig addig fogy, amíg a szelep nyitva van. Tehát a 10 időegységnyi festési időszak alatt 10 egységnyi festék fogy el a kiválasztott festék tartályából.

Megjegyzés:

A festésnél lehetőleg úgy kell időzíteni a festési gombok megnyomását, hogy mire az anyag a szelep alá ér, addigra a kiválasztott szín csövei fel legyenek töltődve. Ha a csövek az anyag szelep alatt tartózkodása közben töltődnek fel, akkor ez azt eredményezi, hogy csak az anyag egy része festődik le. Tehát érdemes a start gomb megnyomása előtt a kívánt szín kiválasztása és a hozzá tartozó gomb megnyomása. Így, mire az anyag a szelep alá ér a kiválasztott szín csövei biztosan fel lesznek töltődve.

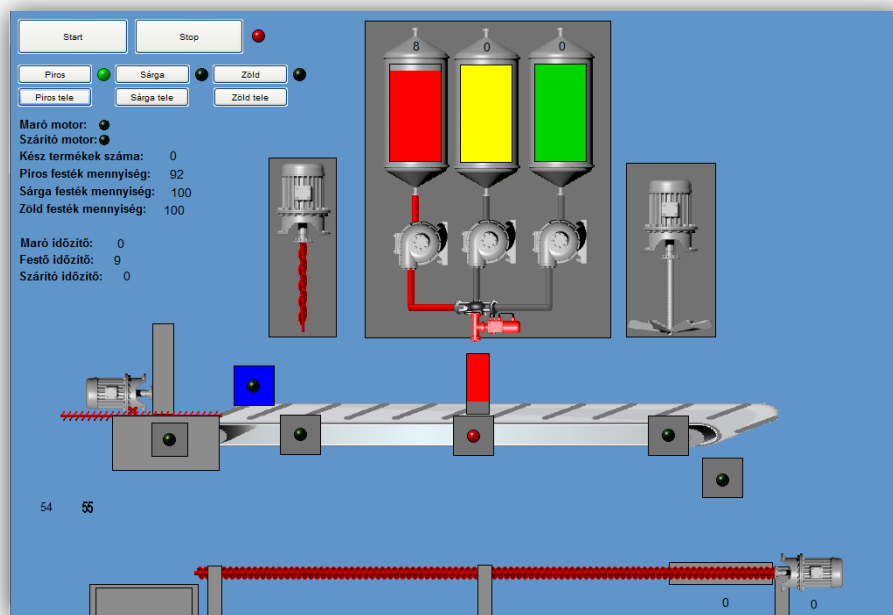
Resetelés:

A gyártási folyamat végén néhány változót alaphelyzetre szükséges állítani.

Ha az anyagmozg elérte a 100-as értéket, ha a kitoló egység visszaért kiinduló állapotába (Kitolo_Motor_Kint=0) és ha a transzport egység végigért mozgási pályáján, akkor az anyagmozg, szinez, pirosszinez, sargaszinez, zoldszinez változók álljanak vissza kezdőállapotba, vagyis 0-ra.

```
IF anyagmozg=100 AND Kitolo_Motor_Kint=0 AND Transzport_Motor=100 THEN
  anyagmozg=0
  szinez=0
  pirosszinez=0
  zoldszinez=0
  sargaszinez=0
END
```

A gyártósor a szimulációban:



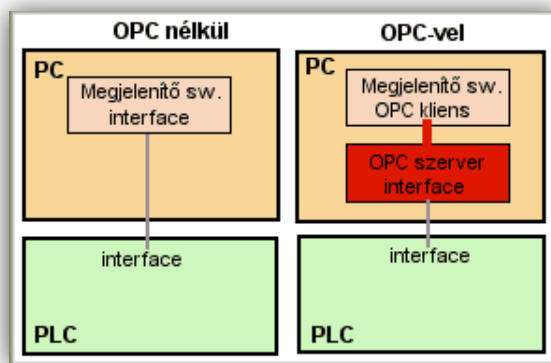
Fischertechnik gyártósor vezérlése és szimulációja Vijeo Citect 7.20 segítségével OPC serveren keresztül

Az OPC server bemutatása

Az OPC egy szoftver interfész szabvány, amely lehetővé teszi, hogy a Windows programok kommunikálni tudjanak ipari hardver eszközökkel.

Az OPC egy server/kliens páron keresztül valósul meg. Az OPC server egy olyan szoftver, amely átalakítja a PLC által használt hardver kommunikációs protokollt OPC protokollá. Az OPC kliens az OPC servert használja arra, hogy adatot kapjon ill. parancsokat küldjön a hardvernek.

A PLC és a PC között kétféleképpen jöhet létre kapcsolat:



Az első esetben a megjelenítő szoftver rendelkezik egy PLC interfész programmal (DLL), amin keresztül közvetlenül kommunikál a PLC-vel.

A második esetben az OPC használatakor a PC-n - a megjelenítő szoftvertől teljesen elkülönülten - fut egy OPC server alkalmazás, ami átveszi a PLC-vel, annak

anyanyelvén való kommunikálás feladatát.

Az OPC server egy definiált szoftveres felületen bármely olyan szoftvernek kiadja az adatokat, ill. fogad utasításokat, amely szakszerűen tud kérdezni ill. parancsolni. Ezek az OPC kliensek.

Az általam létrehozott kapcsolat is hasonló elven működik.

Megjelenítő szoftver, OPC kliens: Vijeo Citect 7.20

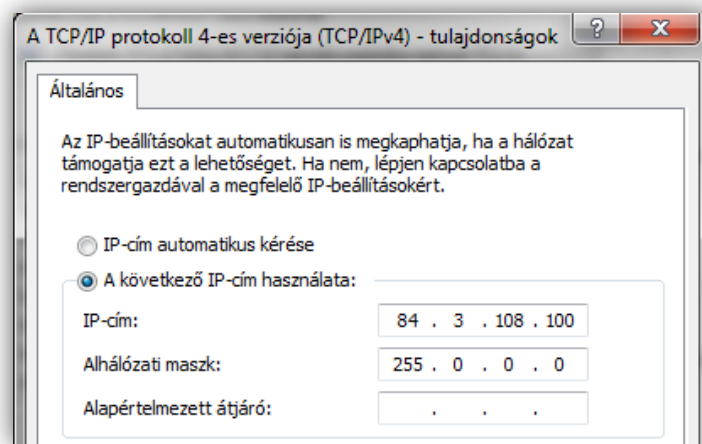
OPC server: Schneider OPC Factory Server

Schneider OPC Factory Server konfigurálása a Vijeo Citect 7.20-hoz

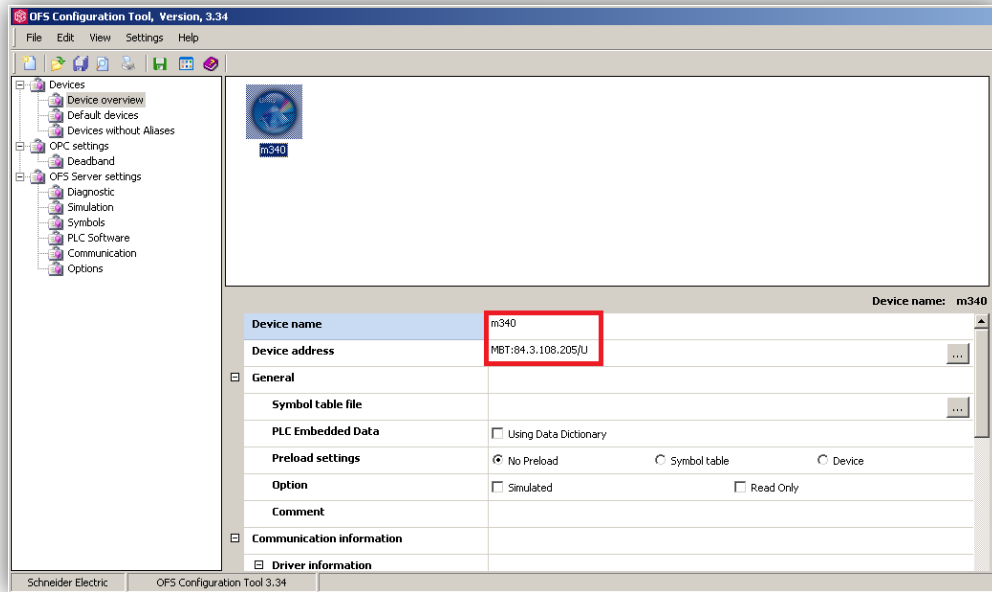
A Schneider M340-es PLC-hez egy KTI Network switch segítségével csatlakoztattam a laptopom UTP kábelek összeköttetésével.



A PLC IP címe: 84.3.108.205, ezért a számítógépünkön létrejött új hálózati kapcsolat IP címét azonos hálózatba kell konfigurálni ezen IP címmel. A PC IP címe: 84.3.108.100

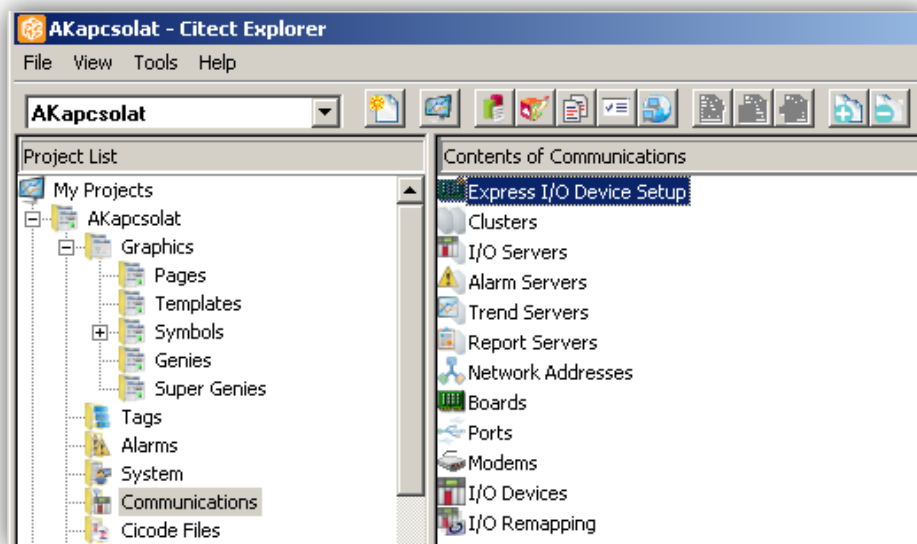


Következő lépésként az OPC Server OSF Configuration Tool alkalmazásban létrehozzuk és beállítjuk a kapcsolathoz szükséges eszköz nevét és címét.



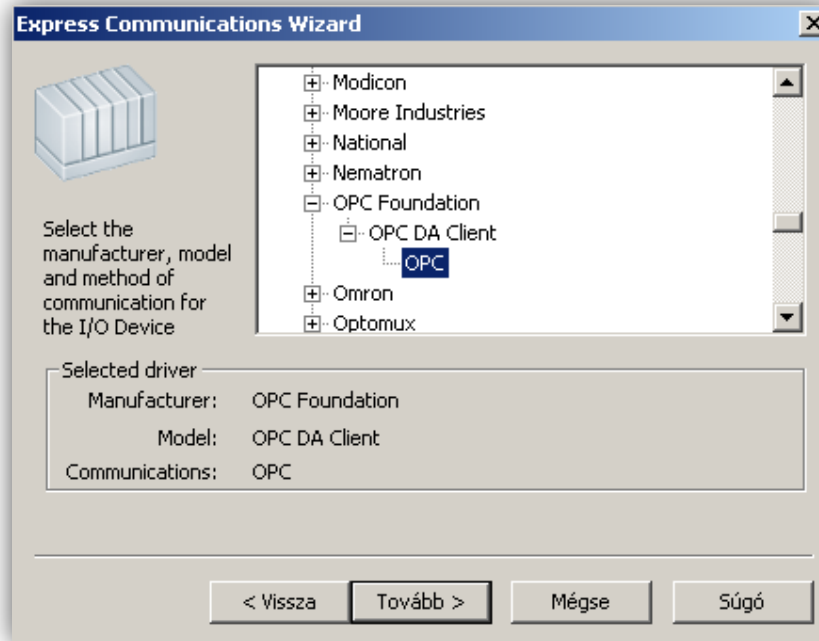
Vijeo Citect 7.20 konfigurációja

I/O eszköz készítése az Express I/O Device Setup segítségével a communications menüpontban.

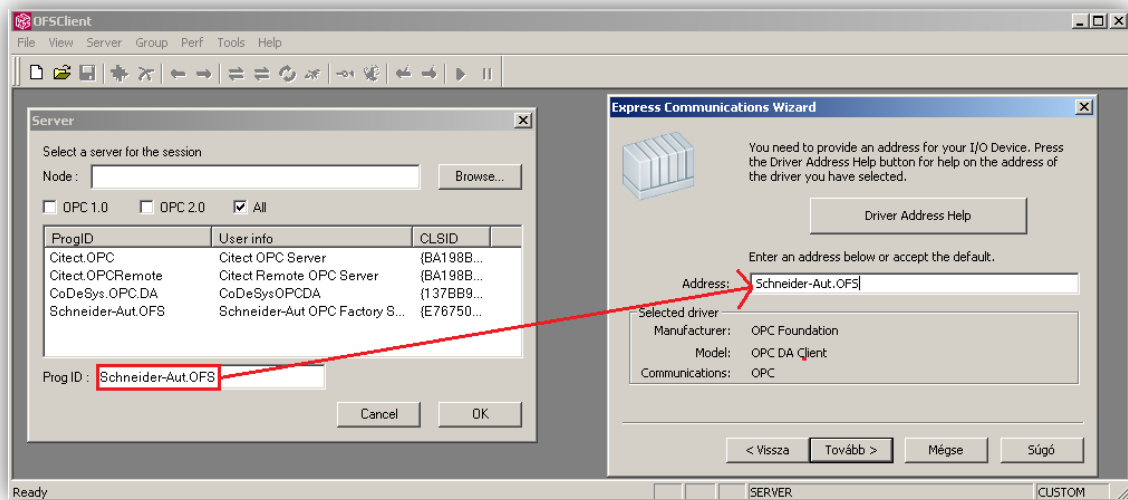


A következőekben konfigurálhatjuk az I/O eszközt.

A létező I/O Server és a külső eszköz nevének beállítása után válasszuk az OPC kommunikációs modellt.

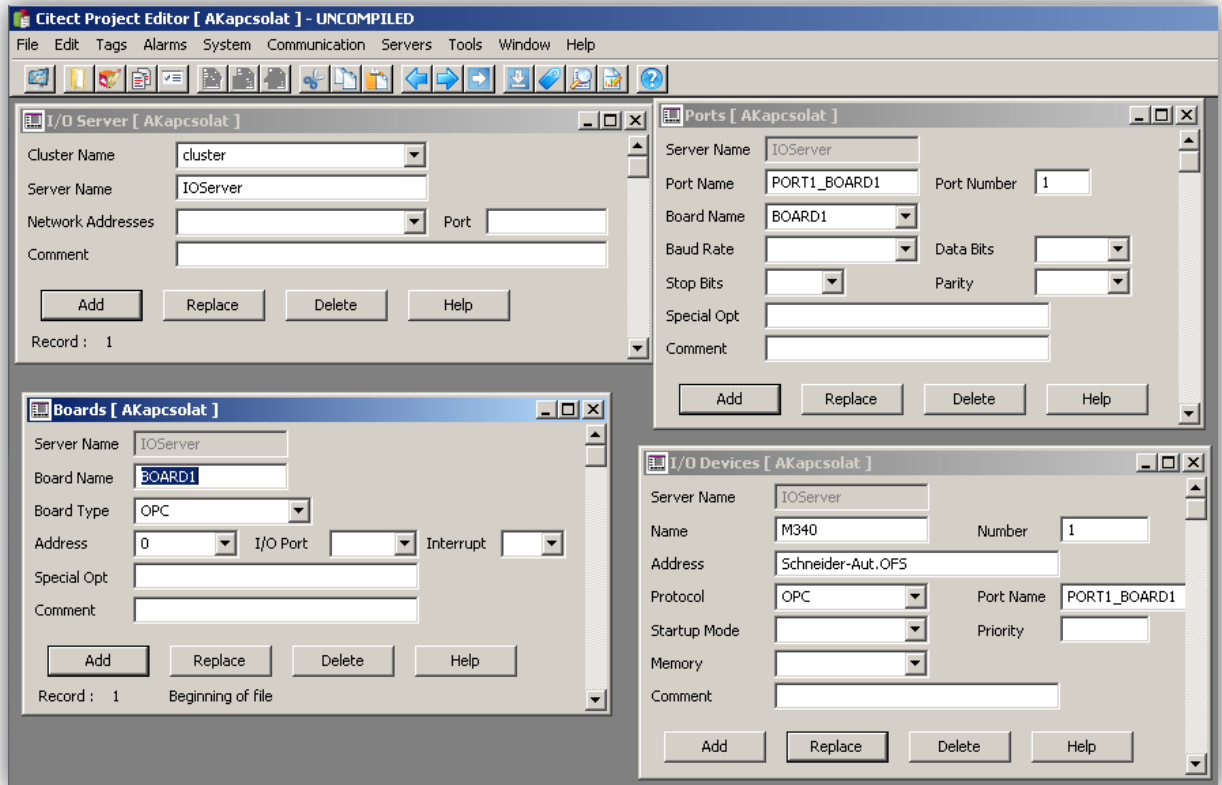


Adjuk meg az I/O eszköz címét:

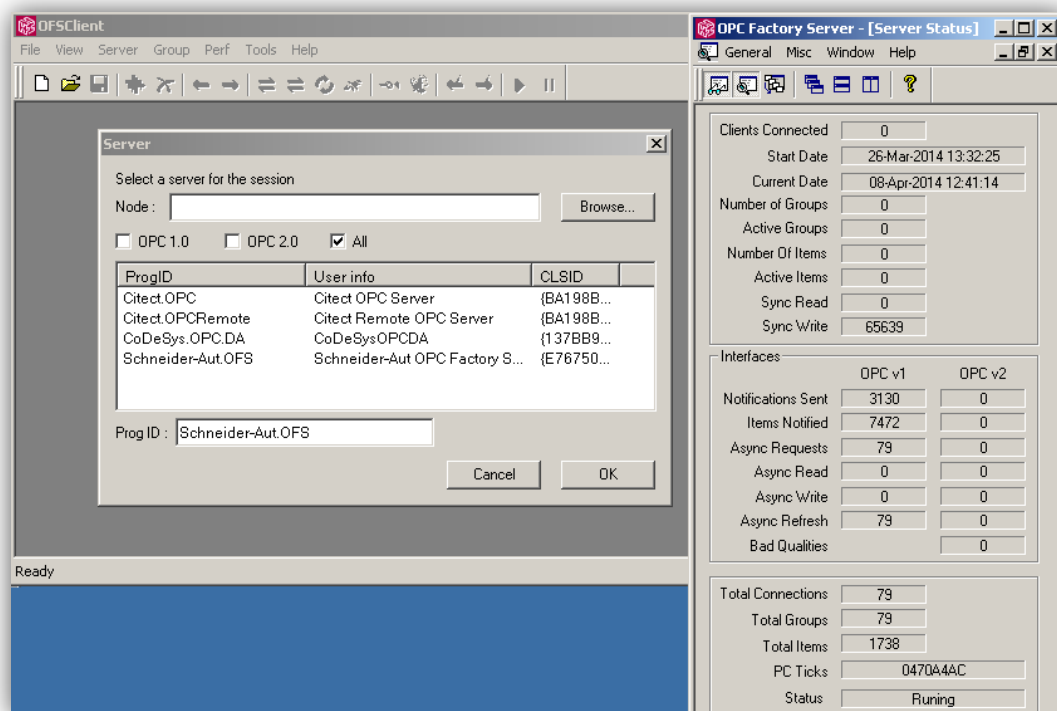


Majd a befejezés gombra kattintva létrejön a kapcsolat.

Vijeo Citect 7.20 konfigurációja összességében



OPC szervert és kliens

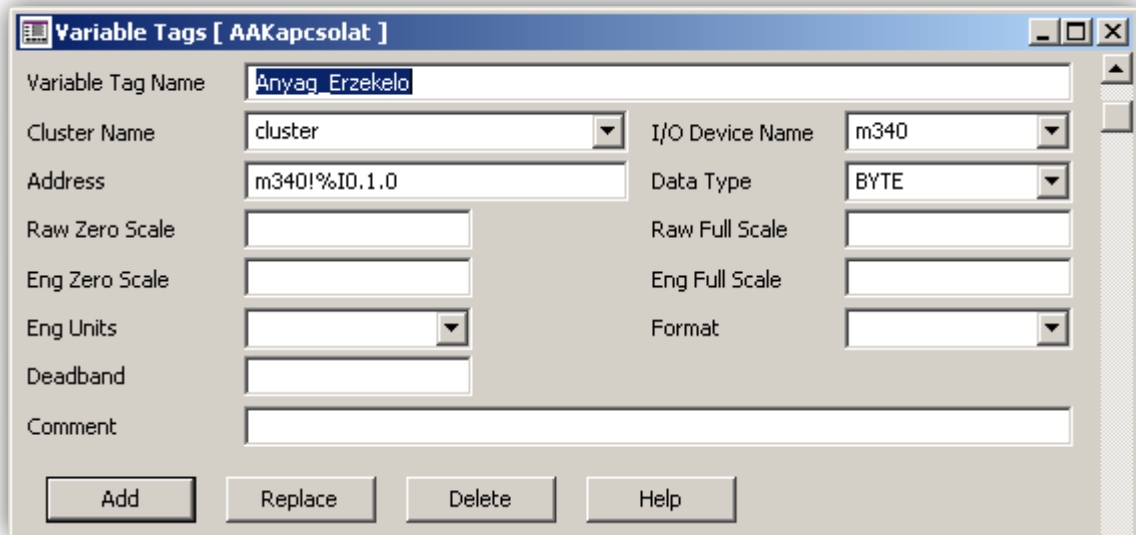


Fischertechnik gyártósor vezérlése és szimulációja Vijeo Citect 7.20 segítségével.

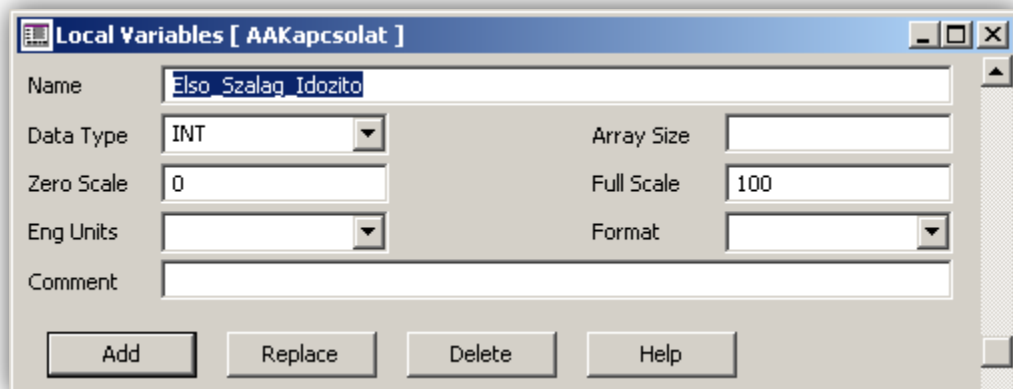
A célom az volt, hogy a valódi gyártósor vezérlése és működése közben a Vijeo Citect szimulációs programból is tudjam vezérelni illetve látni, hogy éppen hol jár a gyártási folyamat.

A PLC-hez való kapcsolódás után első lépésként a PLC változók listáját vettem fel.

A változók felvétele során meg kellett adni azok neveit, a kapcsolódási I/O eszköz nevét, címüket és típusukat.



A helyi változók listája a szimuláció kialakításának előre-haladásával folyamatosan bővült. Ezen változók esetében elég volt megadni a változó nevét, típusát, kezdő és végértékét.



Gyártás indítás

A gyártási folyamatot a szimulációs Start gomb illetve a PLC-s Start gomb megnyomásával is el tudjuk indítani.

PLC-s Start/Stop gomb

A PLC-s Start és Stop gomb vezérlésének megvalósításához létrehoztam egy StartPLC nevű lokális változót, aminek az értéke 1, ha Start gombot nyomtunk, 0, ha Stop gombot nyomtunk.

```
IF Start_Gomb<>0 THEN
  StartPLC=1
END
IF Stop_Gomb<>0 THEN
  StartPLC=0
END
```

Szimulációs Start/ Stop gomb

A szimulációs Start és Stop gombok a Start lokális változót vezérlik. Ha Start gombot nyomtunk, a Start változó értéke 1 lesz. Ha Stop gombot nyomtunk, akkor a Start változó értéke 0 lesz.

Továbbá létrehoztam egy LED-et a szimulációban, ami a Start vagy a StartPLC változók 1-es értékénél villan fel.

Szimulációs és PLC-s Start/Stop gombok:



Kitoló egység

A Kitoló egység programozása összetett és időigényes feladat volt. Első lépésként 4 nyomógombot hoztam létre: Kitoló ki, Kitoló ki állj, Kitoló be, Kitoló be állj.

A "Kitoló ki" nyomógomb a kitoló egység kifelé mozgását valósította meg, tehát a Kitolo_Ki változót 1-es értékre állította, míg a "Kitoló ki állj" a Kitolo_Ki-t 0-ra állította, ami pedig azt eredményezte, hogy a kitoló egység az aktuális helyzetében megállt.

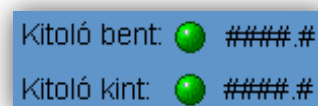
A "Kitoló be" nyomógomb a kitoló egység befelé mozgásáért volt felelős, tehát a Kitolo_Be változót 1-re állította, míg a "Kitoló be állj" gomb a kitoló befelé mozgásának megállítását eredményezte (Kitolo_Be=0).



Következő lépésben létrehoztam 2 LED és szám változót, ami a Kitoló egység kinti illetve benti állapotát jelezték. A "Kitoló bent" nevezetű LED felgyulladt, ha a kitoló motor a kitoló egység végén lévő végállás-kapcsolót benyomta, tehát amikor Kitolo_Bent változó 0 volt.

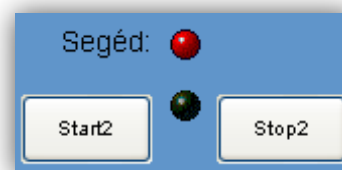
Az ehhez tartozó szám változó pedig 1-es értékre változott.

A "Kitoló kint" nevezetű LED akkor gyulladt fel, amikor a kitoló a kitoló egység elején lévő végállás-kapcsolót benyomta. Ekkor a Kitolo_Kint változó 0 értéket vett fel, a hozzá tartozó szám változó pedig 1-re váltott.



A továbbiakban létrehoztam egy segéd változót és egy hozzá tartozó LED-et, ami akkor váltott át zöld színűre, amikor a seged változó értéke 1 lett.

A kitoló egység működésének tesztelése céljából létrehoztam egy Start2 változót is, amit a Start2 ill. Stop2 nyomógomb működtetett. Ezen változóra és nyomógombokra a későbbiekben nem volt szükség.



A tesztelés folyamán sok probléma adódott abból, hogy a kitoló motor visszafele mozgása esetén is érzékelt az anyag érzékelő.

A sok tesztelés és próbálkozás után sikerült kiküszöbölnöm ezt a problémát és elkészítettem a kitoló egység végleges működését.

A kitoló egység és egyben az egész gyártási folyamat a szimulációs Start gomb (Start=1). vagy a PLC-s Start gomb lenyomása után kezdődik (StartPLC=1).

Ha a Start változó értéke 1 és, ha az anyag érzékelő érzékel (Anyag_Erzekelo=0) vagy ha a StartPLC változó értéke 1 és, ha az anyag érzékelő érzékel akkor, ha a segéd változó értéke 1, akkor a kitoló motor kifelé mozog (Kitolo_Ki=1), egyébként a Kitolo_Ki változó értéke 0. Ha a kitoló egység elérte a kinti végállás-kapcsolót (Kitolo_Kint<>0), akkor a segéd változó 0-ra vált.

Ha a segéd változó 0, akkor a kitoló motor befelé mozog (Kitolo_Be=1), egyébként a Kitolo_Be változó értéke 0. Ha a kitoló egység elérte a benti végállás-kapcsolót (Kitolo_Bent<>0), akkor a segéd változó 1-re vált és kezdődik az egész folyamat előlről.

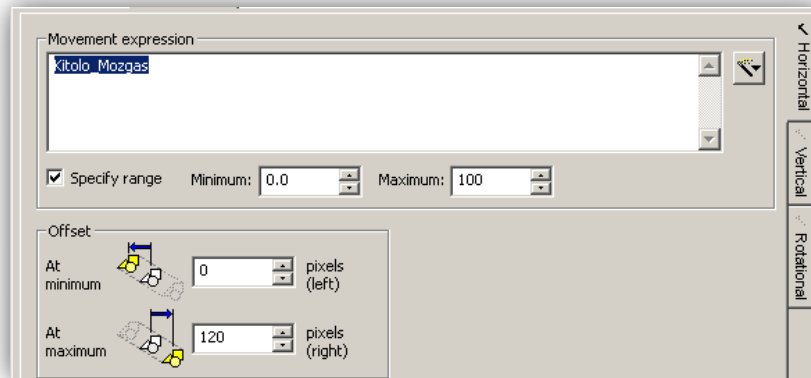
```
IF (Start=1 AND Anyag_Erzekelo=0) OR (StartPLC=1 AND Anyag_Erzekelo=0) THEN
  IF seged=1 THEN
    Kitolo_Ki=1
  ELSE
    Kitolo_Ki=0
  END
  IF Kitolo_Kint<>0 THEN
    seged=0
    Elso_Szalag=1
  END
ELSE
  Kitolo_Ki=0
END

IF Start=1 OR StartPLC=1 THEN
  IF seged=0 THEN
    Kitolo_Be=1
  ELSE
    Kitolo_Be=0
  END
  IF Kitolo_Bent<>0 THEN
    seged=1
  END
ELSE
  Kitolo_Be=0
END
```

Kitoló egység és az anyag érzékelő a szimulációban

A szimulációban az anyag érzékelőt egy LED valósítja meg, ami az anyag érzékelő érzékelése esetén villan fel.

A kitoló egység szimulációs mozgását a Kitolo_Mozgas lokális változó valósítja meg, aminek mozgási tartománya 120 pixel jobbra.

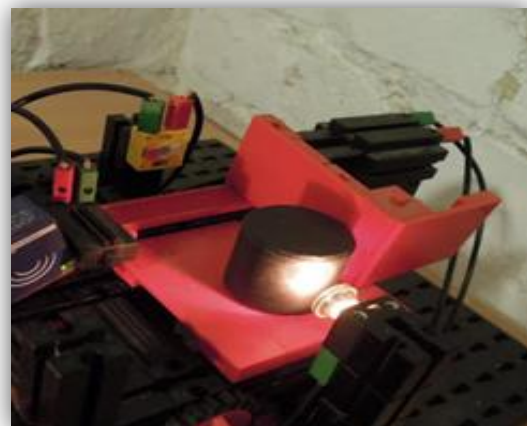
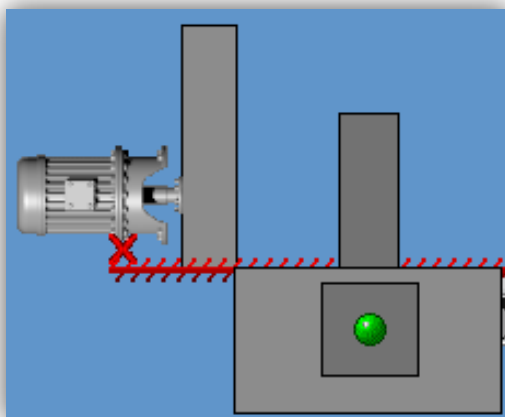


A kitoló 0 kezdőállapotú pozícióból indul. Értéke 10 egységenként növekszik, ha valóságban a kitoló kifelé mozog, tehát ha a Kitolo_Ki PLC változó értéke 1.

A kitoló mozgása 10 egységenként csökken, tehát visszafele mozog, ha a Kitolo_Be PLC változó értéke 1.

```
IF Kitolo_Ki=1 THEN
  Kitolo_Mozgas=Kitolo_Mozgas+10
END
IF Kitolo_Be=1 THEN
  Kitolo_Mozgas=Kitolo_Mozgas-10
END
```

Kitoló egység a szimulációban és a valóságban:



Első futószalag

Az első szalag mozgása a kitoló kinti állapotának elérésénél kezdődik:

```
IF Kitolo_Kint<>0 THEN
  seged=0
  Elso_Szalag=1
END
```

Ezen a részen található a maró és a festő részleg.

Kezdetben az első futószalag akkor állt meg, ha a maró vagy a festő érzékelő érzékelt. Ezzel a megoldással az volt a probléma, hogy a szalagon lévő anyag mindig túlfutott az érzékelőkön és csak azután állt meg. A futószalag gyors mozgását a 24V-os tápfeszültség eredményezi.

A cél az volt, hogy a szalagon lévő anyag pontosan az érzékelőknél álljon meg, különben a marás és festési folyamat nem tud megvalósulni.

A probléma megoldására időzítőt használtam: Amikor a kitoló eléri kinti állapotát és a szalag elindul (Elso_Szalag=1), elindul egy időzítő is (Elso_Szalag_Idozito).

Az időzítő csak akkor számol, ha az első szalag mozog.

A feladat az volt, hogy megfigyeljem, hogy hányadik időpillanatban halad el illetve állna meg a szalagon lévő anyag a maró illetve festő érzékelőknél. Így az időzítő 8. (marás) illetve 17. (festés) időegységében az első szalag megáll. Ezzel azt sikerült elérnem, hogy az anyag pontosan a maró illetve festő érzékelőnél áll meg. A marás és festés után a szalag ismét elindul.

```
IF Elso_Szalag=1 THEN
  Elso_Szalag_Idozito=Elso_Szalag_Idozito+1
  IF Elso_Szalag_Idozito=8 THEN
    Elso_Szalag=0
  END
  IF Elso_Szalag_Idozito=17 THEN
    Elso_Szalag=0
  END
END
```

Maró egység

A szalagon lévő anyag a maró érzékelőhöz érve megáll az "Elso_Szalag_Idozito=8" hatására.

Ha a maró érzékelő érzékel, elindul a maró egység időzítője. Ha az időzítő átlépi a 10. időegységet a szalag és vele az anyag ismét elindul.

```
IF Maro_Erzekelo=0 THEN
  Maro_Idozito=Maro_Idozito+1
  IF Maro_Idozito>10 THEN
    Elso_Szalag=1
  END
END
END
```

Míg a maró érzékelő érzékel, elindul a maró egység motorja is (Maro_Motor=1). Miután az időzítő lejárt, a szalag elindult, és az érzékelő már nem érzékeli az anyagot, a maró motor leáll.

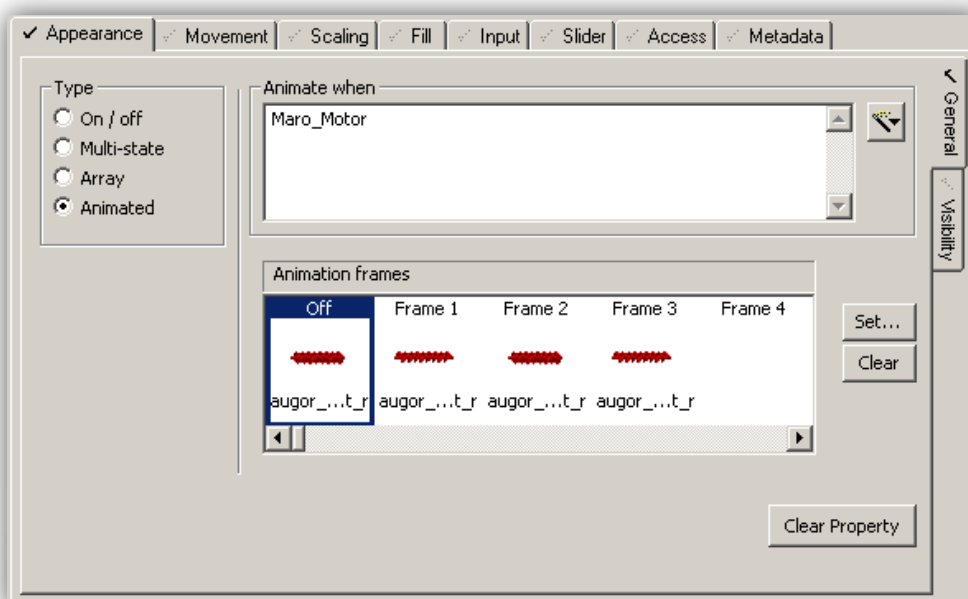
A maró időzítőt a festő érzékelő nullázza le.

```
IF Maro_Erzekelo=0 THEN
  Maro_Motor=1
ELSE
  Maro_Motor=0
END
```

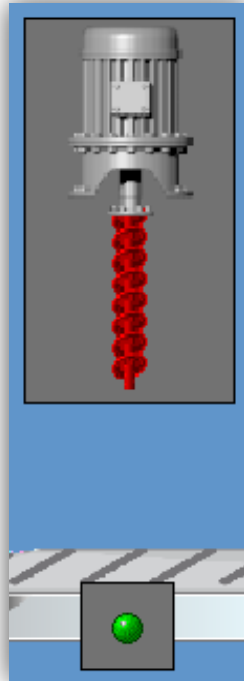
Maró egység szimulációja:

A szimulációban a maró érzékelőt egy LED valósítja meg, ami a maró érzékelő érzékelése esetén villan fel.

A maró motor animálását a Maro_Motor PLC változó határozza meg. Ha a valóságban a maró motor forog, akkor a szimulációban is.



Maró egység a szimulációban és a valóságban:



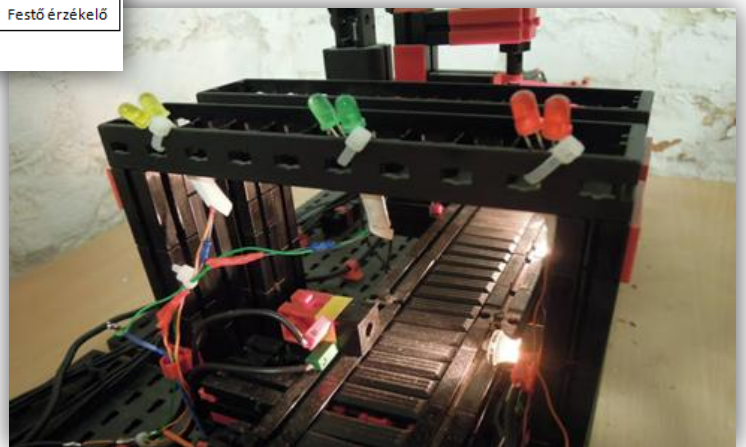
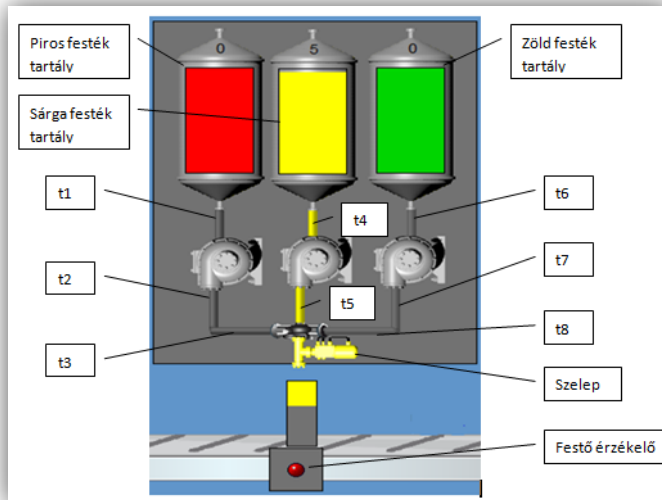
Festési részleg

A festési folyamat megvalósítása hasonló a régebben létrehozott, gyártósortól független szimulációhoz. A szalagon lévő anyag a festő érzékelőhöz érve megáll az "Elso_Szalag_Idozito=17" hatására.

Ha a festő érzékelő érzékel, elindul a festő egység időzítője. Ha az időzítő átlépi a 10. időegységet a szalag és vele az anyag ismét elindul.

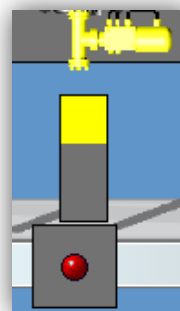
```
IF Festo_Erzekelo=0 THEN
  Maro_Idozito=0
  Festo_Idozito=Festo_Idozito+1
  IF Festo_Idozito>10 THEN
    Elso_Szalag=1
  END
END
END
```

Festési részleg a szimulációban és valóságban



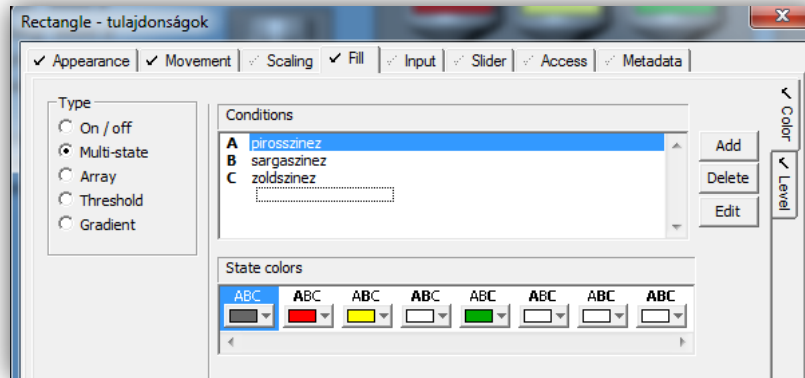
Szelep: A festési részlegnél történő várakozás során történik az anyag festése a festékszóró szelep segítségével. Ha a festő érzékelő érzékel, akkor a szelep változó értéke 1, egyébként 0.

```
IF Festo_Erzekelo=0 THEN
  szelep=1
ELSE
  szelep=0
END
```

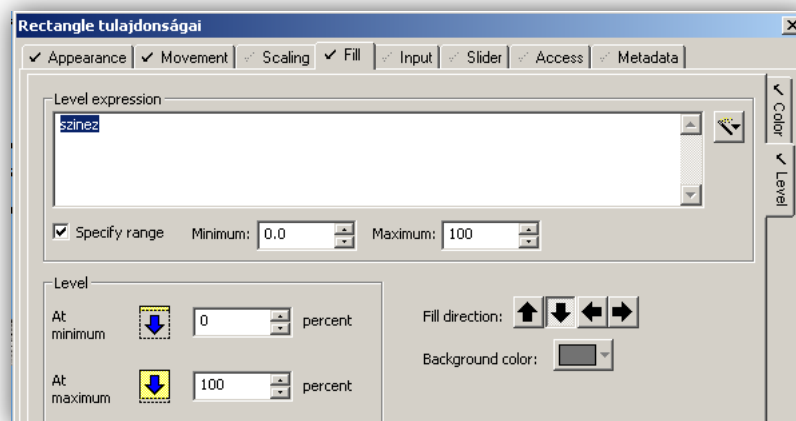


Első lépésként az anyag kitöltési tulajdonságait szükséges beállítanunk a szimulációban.

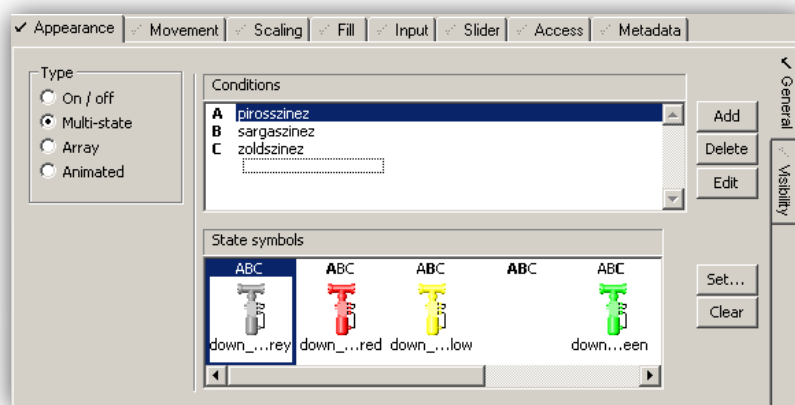
Az anyagot a szelep alatt való tartózkodása idején 3 féle színre festhetjük le (piros, sárga, zöld), ezért állapotát Multi-state-re kell állítani. Ezt követően meg kell adnunk a 3 lehetséges színezést, amihez 3 változó hozzárendelése szükséges (pirosszinez, sargaszinez, zoldszinez).



Az alábbi beállítás az anyag színezésének irányát és maximum értékét mutatja. Az egész színezés folyamatát a szinez változó irányítja.



A szelep is - hasonlóan az anyaghoz - beszíneződik az általunk kiválasztott színre.



A különböző színezéseket a szimulációs nyomógombok és a PLC nyomógombok segítségével is kiválaszthatjuk.

Pirosra színezés

A pirosra színezést C_Piros_Gomb és a Piros_Gomb változók vezérlik.

A C_Piros_Gomb a Vijeo Citect-beli szimulációs nyomógombot valósítja meg, míg a Piros_Gomb pedig a PLC-hez kötött valós nyomógombot.

A szimulációs piros gomb lenyomása, illetve a PLC-s valós, piros nyomógomb folyamatos nyomva tartása esetén a t5 (sárga festék csöve), a t7 és t8 (zöld festék csöve) 0-ra állítódik, ami a szimuláció folyamán azt eredményezi, hogy azok üresek lesznek, nem lesznek feltöltve festékkel. Ezt követően a t1, t2 (felülről lefelé) és t3 (balról jobbra) csövek feltöltődnek piros festékkel egymás után 10 egységenként, jelezve a

festék áramlását a csövek belsejében. Ha a t3-as cső is feltöltődött és ha a szelep értéke 1, ami akkor történik ha a festő érzékelő érzékel, a pirosszinez változó átadódik a szinez változónak, amely 10 egységenként növekedni kezd. Ez azt eredményezi, hogy a szelep alatt álló anyag felülről lefelé 10 egységenként pirosra színeződik.

Továbbá a gyártósor festő részlegénél elhelyezett piros LED lámpa felgyullad, így jelezve, hogy az itt álló anyag pirosra színeződik a szimulációban.

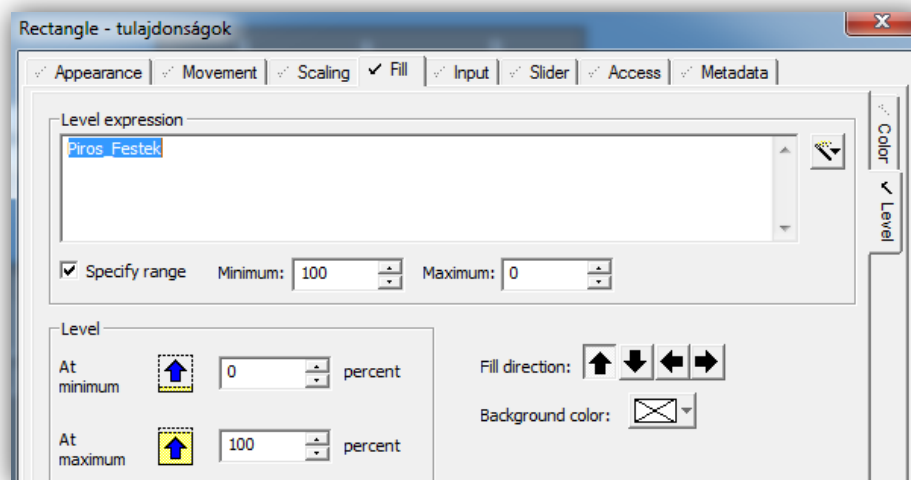
```
IF C_Piros_Gomb=1 OR Piros_Gomb<>0 THEN
  t5=0
  t8=0
  t7=0
  t1=t1+10
  IF t1=100 THEN
    t2=t2+10
    IF t2=100 THEN
      t3=t3+10
    END
  END
END
IF t3=100 AND szelep=1 THEN
  pirosszinez=szinez
  szinez=szinez+10
  Piros_Lampa=1
ELSE
  Piros_Lampa=0
END
```

Piros festék tartály: A piros festék tartály szintjét a Piros_Festek változó befolyásolja.

Ha a t3-as cső feltöltődött és ha a szelep nyitva van (értéke 1), akkor a piros festék értéke nő (kezdőállapota 0).

```
IF t3=100 AND szelep=1 THEN
  Piros_Festek=Piros_Festek+1
END
```

A piros festék értékének növekedése a hozzá tartozó tartály szintjének csökkenését eredményezi, amihez a tartály képen látható beállítása szükséges:



Ezek alapján a tartályban lévő piros festék mennyiség kiszámítása: $100 - \text{Piros_Festek}$.

Piros tartály feltöltése: A piros tartály akkor töltődik fel, ha megnyomjuk a Piros tele gombot vagy ha a piros festék kifogyott a tartályból. A piros festék 0 szintje esetén szükséges a Piros_tele változó nullázása, mert a továbbiakban is megtörténhet a Piros tele gomb megnyomása.

```
IF Piros_Festek=100 OR Piros_Tele=1 THEN
  Piros_Festek=0
END
IF Piros_Festek=0 THEN
  Piros_Tele=0
END
```

Sárgára színezés:

A sárgára színezést a C_Sarga_Gomb és a Sarga_Gomb változók vezérlik.

A C_Sarga_Gomb a Vijeo Citect-beli szimulációs nyomógombot valósítja meg, míg a Sarga_Gomb pedig a PLC-hez kötött valós nyomógombot.

A szimulációs sárga gomb megnyomása illetve a PLC-s valós, sárga nyomógomb folyamatos nyomva tartása közben a t2, t3 (piros festék csövei) és a t7, t8 (zöld festék csövei) 0-ra állítódnak, így üresek lesznek. Ezután a t4-es cső 10 egységenként felülről lefelé befestődik, majd ugyanígy a t5-ös cső is. Ha a t5-ös cső feltöltődött és ha a szelep nyitva van,

akkor a sargaszinez változó átadódik a szinez változónak, ami 10 egységenként növekszik, melynek eredményeképp a festő érzékelőnél álló anyag felülről lefelé 10 egységenként sárgára színeződik.

Továbbá a gyártósor festő részlegénél elhelyezett sárga LED lámpa felgyullad, így jelezve, hogy az itt álló anyag sárgára színeződik a szimulációban.

Sárga festék tartály: A sárga festék tartály szintjét a Sarga_Festek változó befolyásolja.

Ha a t5-ös cső feltöltődött és ha a szelep nyitva van (értéke 1), akkor a sárga festék értéke nő.

A sárga festék értékének növekedése a sárga tartály szintjének csökkenését eredményezi, aminek a beállítása hasonló a piros tartályéhoz (lásd a piros tartálynál).

A sárga tartályban lévő sárga festék mennyiség kiszámítása: 100-Sarga_Festek.

```
IF C_Sarga_Gomb =1 OR Sarga_Gomb<>0 THEN
  t2=0
  t3=0
  t8=0
  t7=0
  t4=t4+10
  IF t4=100 THEN
    t5=t5+10
  END
END
IF t5=100 AND szelep=1 THEN
  sargaszinez=szinez
  szinez=szinez+10
  Sarga_Lampa=1
ELSE
  Sarga_Lampa=0
END
```

```
IF t5=100 AND szelep=1 THEN
  Sarga_Festek=Sarga_Festek+1
END
```

Sárga tartály feltöltése: A sárga tartály akkor töltődik fel, ha megnyomjuk a Sárga tele gombot vagy ha a sárga festék kifogyott a tartályból. A sárga festék 0 szintje esetén szükséges a Sárga_tele változó nullázása, mert a továbbiakban is megtörténhet a Sárga tele gomb megnyomása.

```
IF Sarga_Festek=100 OR Sarga_Tele=1 THEN
    Sarga_Festek=0
END
IF Sarga_Festek=0 THEN
    Sarga_Tele=0
END
```

Zöldre színezés

A zöldre színezést a C_Zold_Gomb és a Zold_Gomb változók vezérlik.

A C_Zold_Gomb a Vijeo Citect-beli szimulációs nyomógombot valósítja meg, míg a Zold_Gomb pedig a PLC-hez kötött valós nyomógombot.

A szimulációs zöld gomb megnyomása illetve a PLC-s valós, zöld nyomógomb folyamatos nyomva tartása közben a t5 (sárga festék csöve) és a t2, t3 (piros festék csövei) üresek lesznek, majd a zöld festék csövei feltöltődnek 10 egységenként. A t6 és t7 felülről lefelé a t8 pedig jobbról balra. Ha a t8-as cső tele van és ha a szelep nyitva van, a zoldszinez változó átadódik a szinez változónak, ami 10 egységenként

```
IF C_Zold_Gomb =1 OR Zold_Gomb<>0 THEN
    t5=0
    t3=0
    t2=0
    t6=t6+10
    IF t6=100 THEN
        t7=t7+10
        IF t7=100 THEN
            t8=t8+10
        END
    END
END
IF t8=100 AND szelep=1 THEN
    zoldszinez=szinez
    szinez=szinez+10
    Zold_Lampa=1
ELSE
    Zold_Lampa=0
END
```

növekszik. Így a szelep alatt lévő anyag felülről lefelé 10 egységenként zöldre színeződik.

Továbbá a gyártósor festő részlegénél elhelyezett zöld LED lámpa felgyullad, így jelezve, hogy az itt álló anyag zöldre színeződik a szimulációban.

Zöld festék tartály: A zöld festék tartály szintjét a Zold_Festek változó befolyásolja.

Ha a t8-as cső feltöltődött és ha a szelep nyitva van (értéke 1), akkor a zöld festék értéke nő.

A zöld festék értékének növekedése a zöld tartály szintjének csökkenését eredményezi, aminek a beállítása hasonló a piros tartályéhoz (lásd piros tartálynál).

```
IF t8=100 AND szelep=1 THEN  
  Zold_Festek=Zold_Festek+1  
END
```

A zöld tartályban lévő zöld festék mennyiség kiszámítása: 100-Zold_Festek.

Zöld tartály feltöltése: A zöld tartály akkor töltődik fel, ha megnyomjuk a Zöld tele gombot vagy ha a zöld festék kifogyott a tartályból. A zöld festék 0 szintje esetén szükséges a Zold_tele változó nullázása, mert a továbbiakban is megtörténhet a Zöld tele gomb megnyomása.

```
IF Zold_Festek=100 OR Zold_Tele=1 THEN  
  Zold_Festek=0  
END  
IF Zold_Festek=0 THEN  
  Zold_Tele=0  
END
```

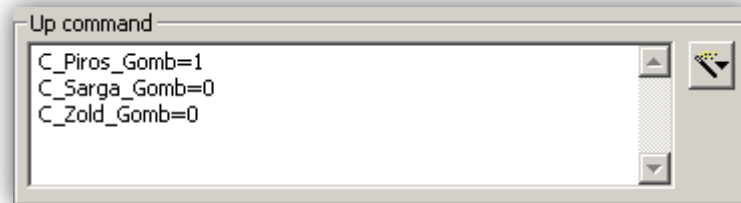
Megjegyzés:

A festésnél lehetőleg úgy kell időzítanünk a festési gombok megnyomását, hogy mire az anyag a szelep alá ér, addigra a kiválasztott szín csövei fel legyenek töltődve. Ha a csövek az anyag szelep alatt tartózkodása közben töltődnek fel, akkor ez azt eredményezi, hogy csak az anyag egy része festődik le. Tehát érdemes a start gomb megnyomása előtt a kívánt szín kiválasztása és a hozzá tartozó gomb megnyomása. Így mire az anyag a szelep alá ér, a kiválasztott szín csövei biztosan fel lesznek töltődve.

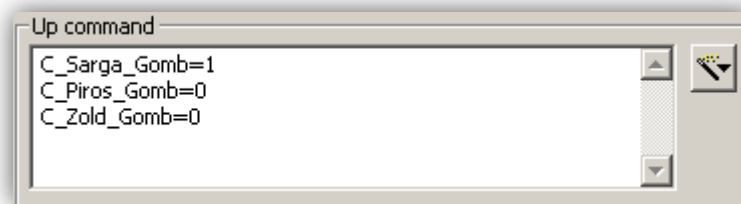
A valódi PLC-s nyomógombok esetében a gombok folyamatos nyomása szükséges ahhoz, hogy a hozzájuk tartozó festék csövei feltöltődjenek. Amint ezek a csövek feltöltődtek a kívánt színnel, a nyomógombot el lehet engedni és a festés megvalósulhat.

A szimulációban LED-ek jelzik, hogy éppen melyik festékhez tartozó gombot nyomtuk le. Egyszerre csak egy színhez tartozó LED világíthat. Ezt az alábbi módon érhetjük el:

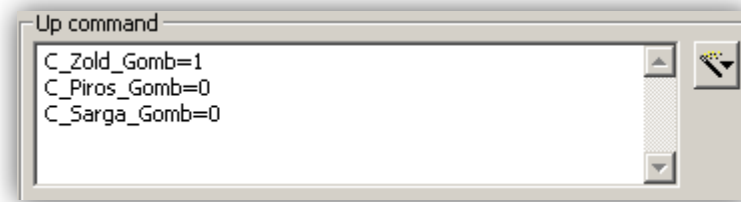
Piros gomb konfigurációja: Ha a piros gomb nyomva van, a másik kettő nullázásra kerül.



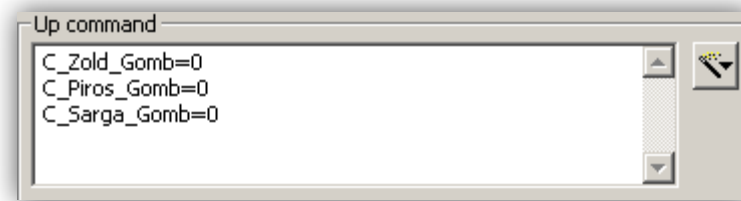
Sárga gomb konfigurációja: Ha a sárga gomb nyomva van, a másik kettő nullázásra kerül.



Zöld gomb konfigurációja: Ha a zöld gomb nyomva van, a másik kettő nullázásra kerül.



PLC nyomógomb: Ez a nyomógomb a szimulációs piros, sárga és zöld nyomógombokat nullázza ki, ezzel átadva a vezérlést a PLC-s festő nyomógomboknak.



Erre azért volt szükség, mert ha kezdetben a szimulációból irányítottuk a festést, viszont a későbbiekben a PLC-s nyomógombokkal szeretnénk vezérelni, akkor a PLC-s nyomógombokat hiába tartjuk nyomva, a csövek nem fognak feltöltődni, mert valamelyik szimulációs festő gomb még aktív. Így ezen gomb megnyomásával mindhárom szimulációs festő gombot nullázhatjuk.

PLC-s és szimulációs festést irányító nyomógombok:



Második futószalag

A második futószalag elindul, ha a festő érzékelő érzékel, vagyis ha az anyag a festő részleghez ért.

```
IF Festo_Erzekelo=0 THEN  
  Masodik_Szalag=1  
END
```

A gyártósor ezen részén található a szárító egység.

Szárítás

A marási és festi résztől eltérően, szárításnál nem egy időzítő állítja meg a szalagot, hanem a szárító érzékelő érzékelése. Az időzítés megoldás ebben az esetben azért nem volt lehetséges, mert a két futószalag közötti kis hézagban az anyag felakad több vagy kevesebb időre. Így azt meghatározni, hogy az anyag hányadik időpillanatban érne a szárító érzékelőhöz nem lehetséges. Ezzel viszont azt a problémát nem tudtam kiküszöbölni, hogy az anyag ne fusson túl a szárító érzékelőn.

A második futószalag megáll, ha a rajta lévő anyag a szárító érzékelőhöz ér. Ekkor az első futószalag is megáll és a hozzá tartozó időzítő (Elso_Szalag_Idozito), valamint a festő időzítő is nullázódik.

Közben elindul a szárító egység időzítője. Ha az időzítő átlépi a 10. időegységet a szalag és vele a megszáritott anyag ismét elindul.

```
IF Szarito_Erzekelo=0 THEN  
  Elso_Szalag_Idozito=0  
  Elso_Szalag=0  
  Masodik_Szalag=0  
  Festo_Idozito=0  
  Szarito_Idozito=Szarito_Idozito+1  
  IF Szarito_Idozito>10 THEN  
    Masodik_Szalag=1  
  END  
END  
END
```

A szárítást a szárító motor (ventilátor) végzi, ami addig forog, míg a szárító érzékelő érzékel.

```
IF Szarito_Erzekelo=0 THEN
  Szarito_Motor=1
ELSE
  Szarito_Motor=0
END
```

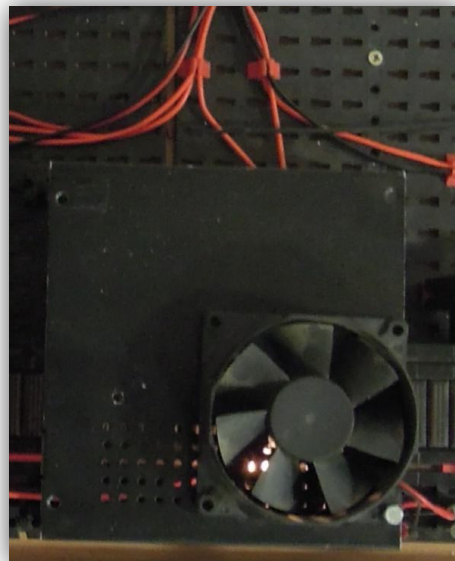
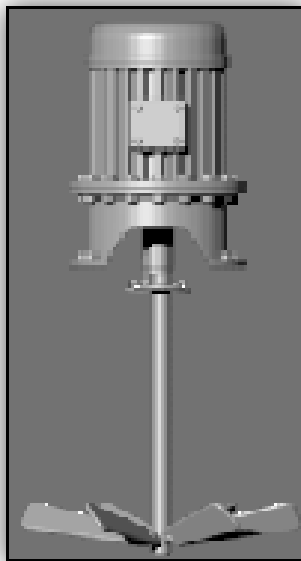
A szárítás után, a második szalag időzítője határozza meg, hogy mennyi ideig menjen még a szalag. Ez az időzítő a szárító időzítő 10. időegysége után indul. Ha eléri a 20. időegységét a szalag megáll, a szárító időzítő és vele együtt a második szalag időzítője is nullázódik.

```
IF Szarito_Idozito>10 THEN
  Masodik_Szalag_Idozito=Masodik_Szalag_Idozito+1
  IF Masodik_Szalag_Idozito>20 THEN
    Masodik_Szalag=0
    Szarito_Idozito=0
    Masodik_Szalag_Idozito=0
    Kesz_Termek=Kesz_Termek+1
  END
END
```

Ezzel véget ér egy termék gyártása.

A kész termékek számát a Kesz_Termek változó határozza meg. Értéke 1-el nő a gyártás végén.

Szárító egység a szimulációban és a valóságban



Induktív érzékelő működése és a fémek megszámlálása

Az induktív érzékelő a kitoló egységet követően helyezkedik el. Ha fém anyagot érzékel, akkor az ebből a célból létrehozott Fem lokális változó értéke növekedik, mindaddig amíg az érzékelés tart. Az így előállt Fem változó értéke számunkra nem fontos. Miután a fém anyag végighaladt a

```
IF Induktiv_Erzekelo<>0 THEN
  Fem=Fem+1
END

IF Fem<>0 AND Masodik_Szalag_Idozito=15 THEN
  Femes_Termek=Femes_Termek+1
  Fem=0
END
```

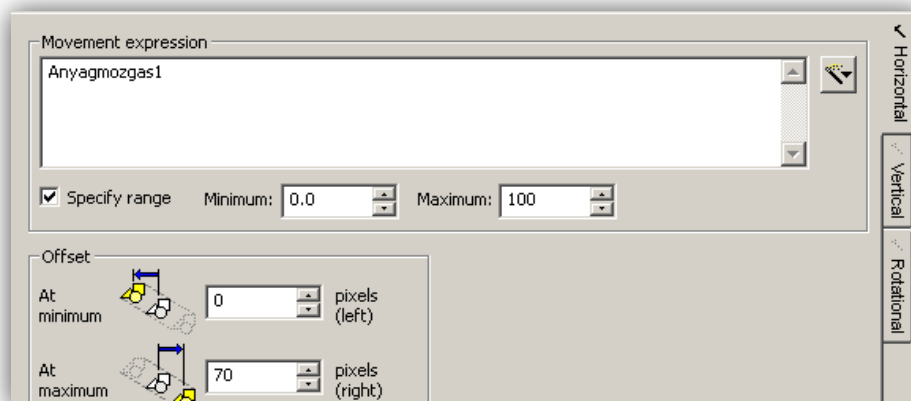
gyártósoron és a második szalag időzítője átlépte a 15. egységet, a Femes_Termek lokális változó értéke 1-el nő a Fem<>0 feltétel miatt. Femes_Termek változó jelenti a fém termékek számát a kész termékek számából. Ha az anyag nincs fém bevonattal ellátva, akkor az induktív érzékelő nem érzékel és a Fem változó 0 marad. Így gyártás végén csak a kész termékek száma nő.

Anyagmozgás megvalósítása a szimulációban

Az anyagmozgást 3 változó valósítja meg: Anyagmozgas1, Anyagmozgas2, Anyagmozgas3. Az Anyagmozgas1 a kitoló egységen, az Anyagmozgas2 a kitoló egység végétől a szárító egységig, míg az Anyagmozgas3 a szárító egységtől a gyártósor végéig valósítja meg az anyagmozgást.

Első lépésként a mozgások távolságát szükséges meghatározni.

Az Anyagmozgas1 a kitoló egység közepétől indul a kitoló egység végéig. Mozgástartomány 70 pixel jobbra.



Az Anyagmozgas2 a kitoló egység végétől a szárító egységig terjed. Mozgástartománya 490 pixel jobbra.

Az Anyagmozgas3 a szárító egységtől a gyártósor végéig terjed. Mozgástartománya 200 pixel jobbra.

A következő lépésben megadtam, hogy a különböző anyagmozgások mikor és milyen nagyságú lépésekben valósuljanak meg.

Amikor a valóságban a kitoló motor kifelé mozog, akkor valósul meg a szimulációban az Anyagmozgas1 mozgása, 7 egységenként lépkedve.

```
IF Kitolo_Ki=1 THEN  
  Anyagmozgas1=Anyagmozgas1+7  
END
```

Ezzel közel azt a hatást sikerült elérnem a szimulációban, mintha a szimulációs kitoló tolná ki az anyagot, közben pedig az egészet a Kitolo_Ki PLC változó vezérli.

Mivel ez a mozgás 7 egységenként lépked, ezért a maximális értékét 98-ra állítottam, mivel ez a 100-hoz legközelebbi 7 többszöröse.

A valóságban az első szalag elindulásakor valósul meg az Anyagmozgas2 mozgása. Az Anyagmozgas2 csak akkor mozog, ha az első futószalag is mozog, tehát ha a valóságban az anyag megáll a marási ill.

```
IF Elso_Szalag=1 THEN  
  Anyagmozgas2=Anyagmozgas2+3  
END
```

festési részlegnél, akkor az a szimulációban is megáll. Haladása 3 lépésegységenként történik, így a maráshoz, festéshez és szárításhoz is akkor érkezik az anyag amikor a valóságban.

Az Anyagmozgas3, az előző anyagmozgásokkal ellentétben, nem PLC-s változóhoz van kötve, hanem egy szimulációs lokális változóhoz, a Szarito_Idozito-höz.

```
IF Szarito_Idozito>10 THEN  
  Anyagmozgas3=Anyagmozgas3+5  
END
```

Ez az anyagmozgás akkor kezdődik el, amikor a szárító időzítő lejárt (Szarito_Idozito>10).

A valós gyártósorhoz viszonyítottan, a szimulációban 5 egységenként halad.

A következő célom az volt, hogy a gyártás egymás után többször is meg tudjon valósulni, ezzel elérve a gyártás monotonitását. Ehhez az anyagmozgásokat a szimulációban nullázni kell.

Az Anyagmozgas1 az Anyagmozgas2 9. lépésénél, az Anyagmozgas2 az Anyagmozgas3 20. lépésénél, míg az Anyagmozgas3 a Masodik_Szalag_Idozito 20. időegységében nullázódik.

```
IF Anyagmozgas2=9 THEN
  Anyagmozgas1=0
END

IF Anyagmozgas3=20 THEN
  Anyagmozgas2=0
END

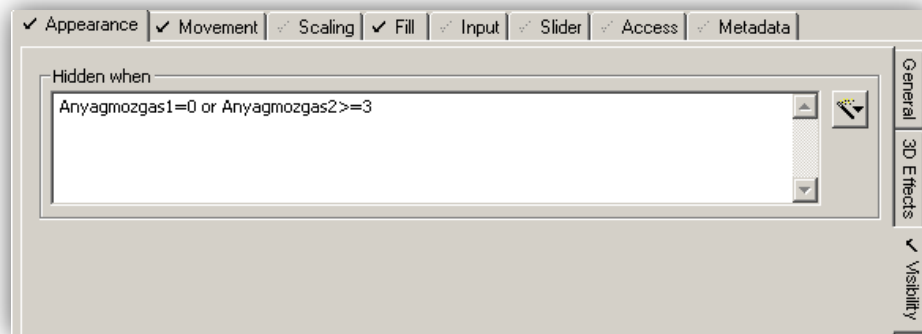
IF Masodik_Szalag_Idozito=20 THEN
  Anyagmozgas3=0
END
```

Anyagláthatóság

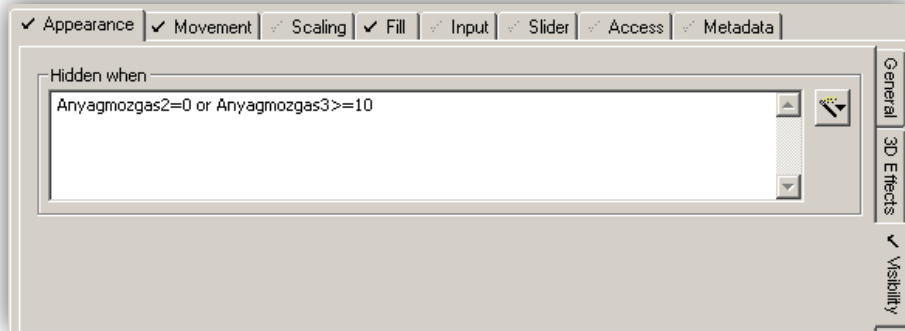
A szimulációban a gyártás folyamán a futószalagon 3 anyag illetve azok mozgása valósul meg. A cél az, hogy egyszerre csak egy anyag legyen látható, azt a hatást keltve, hogy az egész gyártósoron egy és ugyanaz az anyag halad végig.

Ennek megvalósítása érdekében szükséges beállítani a 3 anyag láthatóságát.

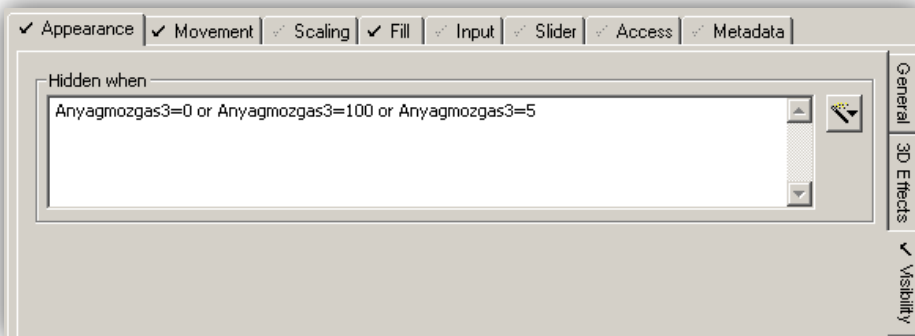
Az Anyagmozgas1 a 0. állapotában vagy az Anyagmozgas2 első lépése (3 egység) után lesz láthatatlan.



Az Anyagmozgas2 a 0. állapotában vagy az Anyagmozgas3 második lépése (10 egység) után lesz láthatatlan.



Az Anyagmozgas3 a 0. ,5. illetve 100. állapotában láthatatlan.

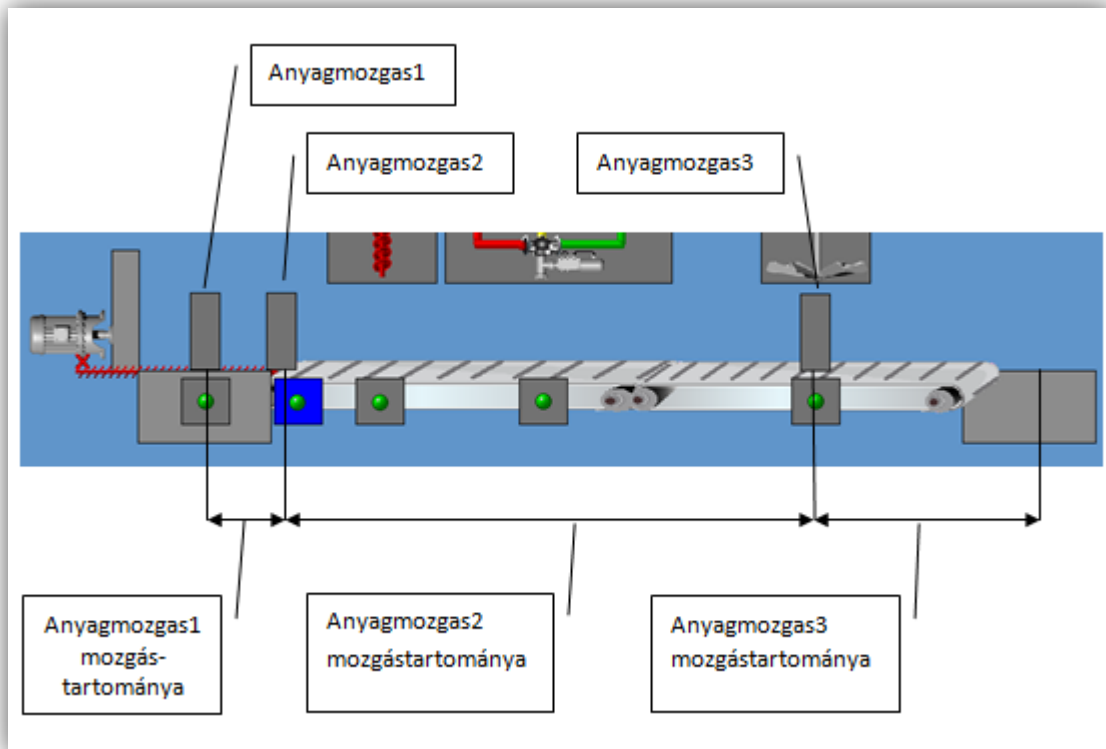


Az első anyag gyártásakor az Anyagmozgas3 a 0. állapotában van. Ezt követően viszont az 5. állapotára ugrik vissza, mivel a szárító időzítő csak később nullázódik:

```
IF Szarito_Idozito>10 THEN
  Anyagmozgas3=Anyagmozgas3+5
END
```

Ezért volt szükséges az Anyagmozgas3 5.lépését is láthatatlanná tenni.

Anyagmozgások:



Vészleállító

A vészleállítást a szimulációs vészleállító gombbal illetve a PLC-s vészleállító gombbal is megvalósíthatjuk.

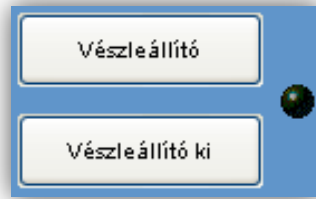
Működését a Leall lokális változó irányítja, miszerint értéke 0 vagy 1. Állapotának szemléltetésére a szimulációban egy LED-et hoztam létre.

A gyártósorhoz tartozó valós vészleállító (beragadó gomb) lenyomásakor (Veszleall_Gomb<>0) vagy a szimulációs Vészleállító gomb megnyomásakor (Veszleallito=1) a Leall változó értéke 1 lesz, egyébként 0. Ebben az esetben az összes PLC-s változó, amelyek a gyártósoron valamilyen mozgást valósítanak meg, 0-ra állítódnak, vagyis minden egység megáll.

```
IF Veszleall_Gomb<>0 OR Veszleallito=1 THEN
    Leall=1
ELSE
    Leall=0
END
```

```
IF Leall=1 THEN
    Start=0
    StartPLC=0
    Kitolo_Ki=0
    Kitolo_Be=0
    Elso_Szalag=0
    Masodik_Szalag=0
    Maro_Motor=0
    Szarito_Motor=0
    Zold_Lampa=0
    Piros_Lampa=0
    Sarga_Lampa=0
    Szelep=0
END
```

Szimulációs és PLC-s vészleállító gombok:



Stop és vészleállító gomb közötti működési különbségek

A szimulációs és PLC-s vészleállító gombok lenyomása esetén az egész gyártási folyamat megáll. A gyártást csak a vészleállítók kioldásával lehet folytatni.

Ezzel ellentétben a szimulációs és PLC-s Stop gombok csak a kitoló egységet állítják meg. Ha a kitoló már elérte a kinti végállás-kapcsolót és közben Stop gombnyomás történik, akkor a már futószalagon lévő anyag végighalad a gyártósoron, a kitoló egység pedig az aktuális helyzetében megáll. A szimulációs illetve a PLC-s Start gombok valamelyikének lenyomása után a kitoló egység visszaáll kezdeti helyzetébe és új anyag érkezésével folytatódhat a gyártás.

Információs panelek

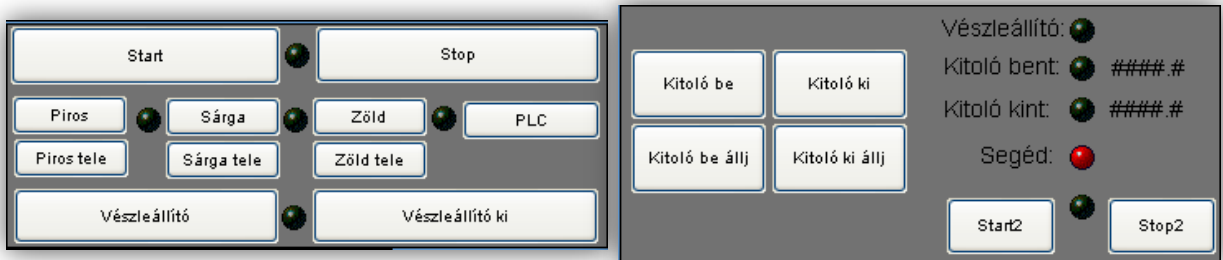
A gyártás folyamán az információs panelek segítségével nyomon tudjuk követni a gyártással kapcsolatos információkat: időzítőket, mozgásokat, festék mennyiséget, kész és fémes anyagok számát.

Maró időzítő: #####.#
Festő időzítő: #####.#
Szárító időzítő: #####.#
Második szalag időzítő: #####.#
Első szalag időzítő: #####.#
Anyagmozgás1: #####.#
Anyagmozgás2: #####.#
Anyagmozgás3: #####.#
Kitoló mozgás: #####.#

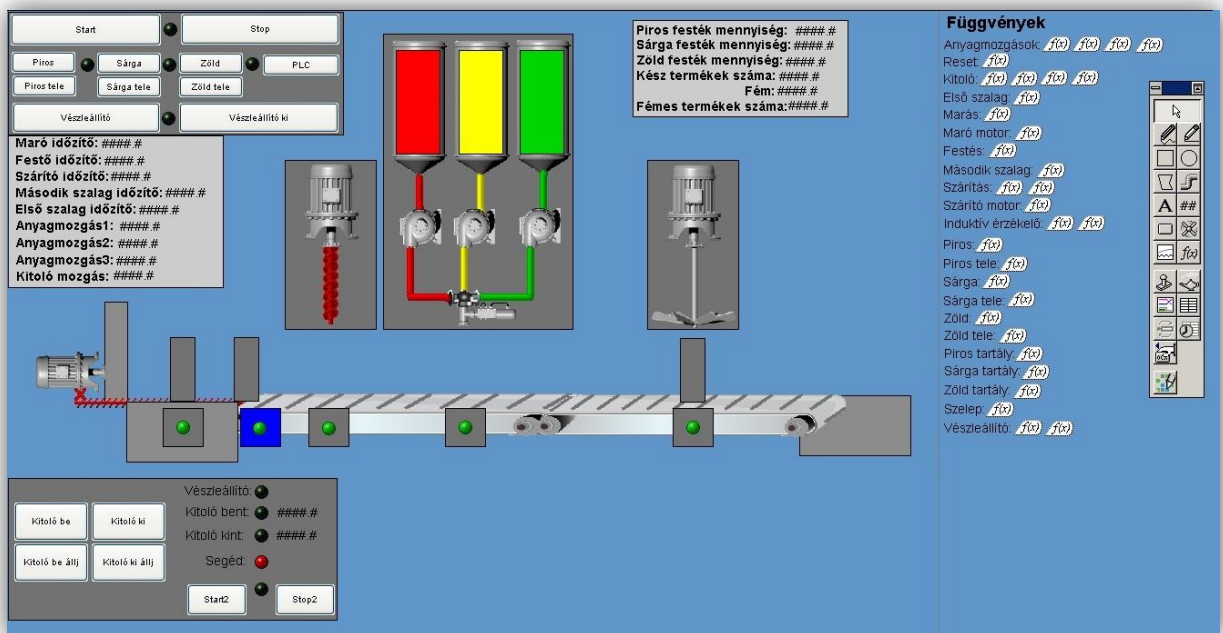
Piros festék mennyiség: #####.#
Sárga festék mennyiség: #####.#
Zöld festék mennyiség: #####.#
Kész termékek száma: #####.#
Fém: #####.#
Fémes termékek száma: #####.#

Vezérlő panelek

A vezérlőpaneleken elhelyezkedő gombok felelősek a gyártás irányításáért: indító és vészleállító gombok, festést vezérlő gombok. A gombok részletes működését fentebb bemutattam.



A szimulációs gyártósor egészében:



Összefoglalás

Célunk egy automatikusan működő gyártósor létrehozása volt, amelynek a vezérlése és szimulációs megvalósítása az Én feladatomban volt. A projekt megkezdése előtt a tanáraink segítségével megvitattuk, hogy milyen működési szabályoknak kell megfelelnie elkészülése során. Az elektronika bekötését követően - melyet a két társam hozott létre - kezdtem meg munkámat. A gyártósortól független szimuláció kialakítása során rengeteg szimulációs problémába ütköztem, de a Vijeo Citect nevű program folyamatos megismerésével sikerült kiküszöbölnöm a problémákat. Kezdetben a főbb gyártósori mozgásokat valósítottam meg és haladtam az egyre kisebb, kevésbé fontosabb problémák megoldása felé. A gyártósortól független szimuláció megvalósítása során elég tapasztalatot szereztem a programmal és a szimulációs problémákkal kapcsolatban. Ezt követően kezdtem el a gyártósor vezérlésének és összhangjának kialakítását a szimulációval. Ez esetben is a bonyolultabb problémákkal kezdtem a munkámat és haladtam a kisebb szimulációs és vezérlési feladatok megvalósítása felé.

A vezérlés és szimuláció megvalósítása során munkámat egyedül végeztem, az ezzel kapcsolatban felmerülő problémákat egyedül oldottam meg. Így a problémamegoldó képességem nagy mértékben fejlődött.

A különböző részfeladatok megvalósítására képzeletbeli időpontokat tűztem ki, ezzel is ösztönözve magam a határidőszerű munkavégzésre, ami egy mérnök számára nagyon fontos munkája során.

A megszerzett ismereteket egy valódi gyártósor vezérlésénél és szimulációjánál is fel tudnék használni, mivel ezek irányítása és szimulációja is megoldható lenne az általam használt szimulációs programmal.

Remélem, hogy a jövőbeli munkahelyemen lehetőség adódik hasonló projektek megvalósítására, vagy valamilyen más szimulációs feladat megoldására.

Függelék

A gyártósortól független szimulációhoz szükséges változók listája:

Neve	Típusa	Értéke
start	Digital	0-1
Kitolo_Motor	Int	0-100
anyagmozg	Int	0-100
t1	Int	0-100
t2	Int	0-100
t3	Int	0-100
t4	Int	0-100
t5	Int	0-100
t6	Int	0-100
t7	Int	0-100
t8	Int	0-100
Piros_Gomb	Digital	0-1
Sarga_Gomb	Digital	0-1
Zold_Gomb	Digital	0-1
szinez	Int	0-100
Anyag_Erzekelo	Digital	0-1
Maro_Erzekelo	Digital	0-1
Festo_Erzekelo	Digital	0-1
Induktiv_Erzekelo	Digital	0-1
Szarito_Erzekelo	Digital	0-1
Piros_Festek	Int	0-100
Sarga_Festek	Int	0-100
Zold_Festek	Int	0-100
sargaszinez	Digital	0-1
pirosszinez	Digital	0-1
zoldszinez	Digital	0-1
Kesz_Termek	Int	0-100
Piros_Tele	Digital	0-1
szelep	Digital	0-1
Sarga_Tele	Digital	0-1
Zold_Tele	Digital	0-1
Maro_Motor	Digital	0-1
Szarito_Motor	Digital	0-1
Kitolo_Motor_Kint	Digital	0-1
anyagmozg2	Int	0-100
Transzport_Erzekelo	Digital	0-1
seged	Digital	0-1
Transzport_Motor	Int	0-100
Maro_Idozito	Int	0-10
Festo_Idozito	Int	0-10
Szarito_Idozito	Int	0-20

Fischertechnik gyártósor vezérléséhez és szimulációjához szükséges változók listái:

PLC változók listája:

Változó neve:	Címe:	Típusa:	Használata:
Anyag_Erzekelo	%I0.1.0	Byte	Kitolóegységénél lévő érzékelő
Maro_Erzekelo	%I0.1.1	Byte	Maró érzékelő (fényzorompó)
Festo_Erzekelo	%I0.1.2	Byte	Festő rész érzékelője (fényzorompó)
Induktiv_Erzekelo	%I0.1.3	Byte	Induktív érzékelő Schneider Osiprox E
Szarito_Erzekelo	%I0.1.4	Byte	Szárító egység érzékelője
Kitolo_Kint	%I0.1.5	Byte	Kitoló kinti helyzetét jelző érzékelő (végkapcs)
Kitolo_Bent	%I0.1.6	Byte	Kitoló benti helyzetét jelző érzékelő (végkapcs)
Start_Gomb	%I0.1.7	Byte	Start gomb
Veszleall_Gomb	%I0.1.8	Byte	Vészleállító gomb
Piros_Gomb	%I0.1.9	Byte	Piros színt kiválasztó gomb
Sarga_Gomb	%I0.1.10	Byte	Sárga színt kiválasztó gomb
Zold_Gomb	%I0.1.11	Byte	Zöld színt kiválasztó gomb
Stop_Gomb	%I0.1.12	Byte	Stop gomb
Kitolo_Be	%Q0.3.0	Byte	Kitolót befelé mozgatja
Kitolo_Ki	%Q0.3.1	Byte	Kitolót kifelé mozgatja
Elso_Szalag	%Q0.3.2	Byte	Első futószalag motorja
Maro_Motor	%Q0.3.3	Byte	Maró motor
Masodik_Szalag	%Q0.3.4	Byte	Második futószalag motorja
Szarito_Motor	%Q0.3.5	Byte	Ventilátor motor
Zold_Lampa	%Q0.3.6	Byte	Zöld LED
Sarga_Lampa	%Q0.3.7	Byte	Sárga LED
Piros_Lampa	%Q0.3.8	Byte	Piros LED

Helyi változók listája:

Változó neve:	Típusa:	Értéke:
Kitolo_Mozgas	Int	0-100
Maro_Idozito	Int	0-20
C_Piros_Gomb	Byte	0-1
C_Sarga_Gomb	Byte	0-1
C_Zold_Gomb	Byte	0-1
Piros_Festek	Int	0-100
Sarga_Festek	Int	0-100
Zold_Festek	Int	0-100
t1	Int	0-100
t2	Int	0-100
t3	Int	0-100
t4	Int	0-100
t5	Int	0-100
t6	Int	0-100
t7	Int	0-100
t8	Int	0-100
szinez	Int	0-100
pirosszinez	Digital	0-1
sargaszinez	Digital	0-1
zoldszinez	Digital	0-1
szelep	Byte	0-1
Piros_Tele	Byte	0-1
Sarga_Tele	Byte	0-1
Zold_Tele	Byte	0-1
Festo_Idozito	Int	0-20
Szarito_Idozito	Int	0-20
Masodik_Szalag_Idozito	Int	0-40
Elso_Szalag_Idozito	Int	0-100
seged	Digital	0-1
Start	Digital	0-1
Start2	Digital	0-1
Anyagmozgas1	Int	0-98
Anyagmozgas2	Int	0-100
Anyagmozgas3	Int	0-100
Kesz_Termek	Int	0-100
Femes_Termek	Int	0-100
Fem	Int	0-100
Veszleallito	Digital	0-1
StartPLC	Digital	0-1
Leall	Digital	0-1

Irodalomjegyzék

Linkek:

1. <http://www.controlshop.hu/szakcikk/opcsuli1/index.html>
2. <http://www.controlshop.hu/szakcikk/opcsuli2/index.html>
3. <http://www.controlshop.hu/szakcikk/opcsuli3/index.html>
4. <http://www.opcdatahub.com/WhatIsOPC.html#note1>
5. <http://www.schneider-electric.hu>
6. <http://www.control.com/thread/1276592440>
7. <http://szirty.uw.hu/>
8. http://sirkan.iit.bme.hu/dokeos/courses/BMEV8IA349/document/Eloadas_PLC.pdf?cidReq=BMEV8IA349
9. http://www.electrocentr.com.ua/doc/SE/automation/plc/m340/M340_Ethernet_Communications_Modules_and_Processors_2009_eng.pdf
10. <http://www.slideshare.net/simonrooke/introduction-to-vijeo-citect>

Könyvek:

1. http://www.controsys.hu/download/Citect/CitectSCADA_Tech_Overview_lores.pdf
2. http://www.scada.ru/CitectSCADA_User_Guide.pdf
3. <http://vius.ru/files/VijeoCitectv7.0Gettingstarted.pdf>
4. http://www.schneider-electric.hu/documents/schneider-magazin/Schneider_Magazin_2007_december.pdf

Felhasznált oktató videók:

1. <http://www.youtube.com/watch?v=5zZHBvbx-f-Q>
2. <http://www.youtube.com/watch?v=t3MZFWyY1Gw>
3. <http://www.youtube.com/watch?v=aW9YhcX1goU>
4. <http://www.youtube.com/watch?v=t3IsZsbS9VM>
5. <http://www.youtube.com/watch?v=Cso3nHWj9dk>
6. <http://www.youtube.com/watch?v=SbxqJxZfeLk>
7. <http://www.youtube.com/watch?v=ZUS8HXrqdYA>
8. <http://www.youtube.com/watch?v=nB89AJD-XjA>

Köszönetnyilvánítás

Köszönöm témavezetőmnek, Vitéz Attilának, hogy elvállalta a témavezetéssel járó feladatokat, valamint a szakdolgozatom elkészítése során nyújtott segítségét és útmutatását. Külön köszönet jár a Debreceni Egyetem Műszaki Karának, hogy biztosította számunkra a Fischertechnik-es alkatrészeket, valamint a helyet, ahol megépítettük és teszteltük a gyártósorunk. Továbbá szeretnék köszönetet mondani a Debreceni Egyetem Műszaki kar minden olyan munkatársának, aki valamilyen módon segítséget nyújtott.