

Debreceni Egyetem Informatikai Kar

**Gyártósor programozása programozható logikai
vezérlővel**

Témavezető:
Dr. Husi Géza
tanszékvezető
főiskolai docens

Készítette:
Papp Gábor
Mérnök informatikus
hallgató

Debrecen

2010

Tartalomjegyzék

1. Bevezetés	3
2. Fischertechnik.....	6
3. Programozható Logikai Vezérlők.....	6
3.1 A PLC-k rövid története	6
3.2 A PLC-k általános jellemzése.....	7
3.3 TWIDO PLC	8
3.3.1 TWIDO kompakt vezérlők bemutatása:	8
3.3.2 TWD LCDE 40DRF vezérlő bemutatása:	10
3.3.3 Diszkrét Input/Output bővítőmodulok	12
3.3.4 Tápegység.....	13
3.3.5 TWIDO Suit	13
4. A Gyártósor	21
4.1 A Gyártósor felépítése	21
4.1.1 Rövid futószalag	21
4.1.2 Hosszú futószalag	22
4.1.3 Hegesztőrobot.....	23
4.1.4 Forgó tálca	24
4.1.5 Robotkar	24
4.2 Folyamat	26
5. Az elektronika bekötése.....	26
5.1 Robotkar bekötése	26
5.2 A rövid és a hosszú futószalag bekötése	28
5.3 A hegesztő és a forgó tálca bekötése	29
6. Programozás	30
7. Fejlesztési lehetőségek	33
8. Összegzés	34
Irodalomjegyzék	36

Plágium - Nyilatkozat

Szakedolgozat készítésére vonatkozó szabályok betartásáról nyilatkozat

Alulírott Papp Gábor (Neptunkód: FTU3GY) jelen nyilatkozat aláírásával kijelentem, hogy a Gyártósor programozása programozható logikai vezérlővel

című szakdolgozat/diplomamunka

(a továbbiakban: dolgozat) önálló munkám, a dolgozat készítése során betartottam a szerzői jogról szóló 1999. évi LXXVI. tv. szabályait, valamint az egyetem által előírt, a dolgozat készítésére vonatkozó szabályokat, különösen a hivatkozások és idézések tekintetében.

Kijelentem továbbá, hogy a dolgozat készítése során az önálló munka kitétel tekintetében a konzulenszt, illetve a feladatot kiadó oktatót nem tévesztettem meg.

Jelen nyilatkozat aláírásával tudomásul veszem, hogy amennyiben bizonyítható, hogy a dolgozatot nem magam készítettem vagy a dolgozattal kapcsolatban szerzői jogsértés ténye merül fel, a Debreceni Egyetem megtagadja a dolgozat befogadását és ellenem fegyelmi eljárást indíthat.

A dolgozat befogadásának megtagadása és a fegyelmi eljárás indítása nem érinti a szerzői jogsértés miatti egyéb (polgári jogi, szabálysértési jogi, büntetőjogi) jogkövetkezményeket.

Papp Gábor

hallgató

Debrecen, 2010.11.16.

1. Bevezetés

Mérnök informatikus hallgatóként az informatikának sok területére betekintést nyertük elméleti szinten, de gyakorlati síkon kevésbé. A választott szakirányunkon a Vállalati Információs rendszereken tartott tantárgyak között több olyan óra is volt, amin az informatika gyakorlati alkalmazásait láthattuk, ez közelebb állt hozzánk, mint az elméleti sík.

Az ötlet hogy ezt a feladatot válaszuk szakdolgozati feladatnak egy Termelés informatika órán keletkezett, amikor a tanár úr bemutatta a tavalyi csoport által készített működő gyártósor modelljét és javasolta egy csoportos témának. Megtetszet nekünk és elhatároztuk, hogy mi is elkészítjük közösen a saját modellünket.

Munkánk során egy olyan gyártósor modelljét készítettük el, amely egy terméket állít össze két részből.

A modellünket olyan eszközzel szeretnénk volna vezérelni, amelyeket manapság alkalmaznak az ipari automatizálásnál is. Választásunk ezért a programozható logikai vezérlőkre (PLC) eset, így könnyen tudtuk hasznosítani azon ismereteinket, amelyeket egy szakmai szabadon választott tantárgy keretében elsajátítottunk.

Tudtuk hogy nem egyszerű feladatra vállalkozunk, de reméltük hogy a csapatmunka előnyünkre válik. Munkánk során számos kisebb nagyobb nehézségekbe ütköztünk ezért problémákat több részre felosztva külön-külön, de egymás munkáját segítve oldottuk meg. Ez hasznosnak bizonyult, mivel a csapatmunka által szerzett ismeretek megkönnyítik a jövőbeni beilleszkedést egy munkahelyi csoportba. Napjainkban a legtöbb vállalatnál előtérbe került a csapatmunka, mivel rájöttek hogy az alkalmazottak hatékonyabban dolgoznak együtt kisebb csoportokban. Tudnunk kell a másikkal együttműködni és alkalmazkodni hozzá a munkavégzés során.

A szakdolgozat célja, hogy bemutassa ennek a modellnek a tervezését, az elkészítését és működését.

Tervezésnél szem előtt tartottuk az órán megszerzett ismereteinket és tanár úr által ajánlottakat. Szakmai napokon tartott kiállításon hosszabb ideig működtettük így fény derült

az apróbb problémákra, valamint a tapasztaltabb szakemberek hozzászólásait is figyelembe vettük így lehelőségünk volt ezen hibáink korrigálására is.

Gyártósorunk alapjául a Műszaki kar által szolgáltatott Fischertechnik modelljei szolgáltak, amelyek a következők: egy prés gép, fűrőgép, egy hegesztőrobot és egy RTT robotkar, valamint futószalagok és különböző érzékelők. Mivel összeállítási útmutatót nem találtunk csak a robotkarhoz és a hegesztőrobothoz ezért a többi elemet saját fantáziánk alapján építettük meg és helyeztük el a gyártósoron.

A vezérlést a Schneider Electric cég TWIDO termékcsaládjába tartozó vezérlővel és annak bővítőmoduljaival oldottuk meg.

Dolgozatomban kifejtem a gyártósor működését, az egyes részegységek felépítését és elektronikai bekötésüket, valamint a vezérlő általános felépítést és jellemzését

A csapatmunkában a társaim Sóvágó Imre (SMJT9H) és Hajdu Zoltán (CYT3JS).

2. Fischertechnik

A gyártósor felépítéséhez a Fischertechnik modelljeit használtuk fel. A céget Németországban alapították. 1965-ben Artur Fischer professzor alkotta meg az első elemeket. A rendszer alap eleme egy olyan kocka, amit mind a hat oldalán lehet bővíteni, különböző elektronikai eszközökkel pl.: infravörös távirányítóval vagy fényérzékelőkkel még tovább lehet bővíteni a lehetőségeket. 20 év óta már számítógéppel is lehet ezeket a modelleket vezérelni. A cég modelljei annyira elterjedtek, hogy ma már egy több mint 30000 tagot számláló rajongói klub is létrejött. Sok egyetem és iskola is csatlakozott, hogy a diákjaikkal is megismertessék ezeket az eszközöket és sok mérnök kezdett karriert egy ilyen szettel. Reméljük nekünk is hasznunkra lesz a későbbiekben, hogy egy ilyen modellt elkészítettünk majd beprogramoztuk.

3. Programozható Logikai Vezérlők

3.1 A PLC-k rövid története

„A PLC első definíciója a General-Motorsánál született meg, 1968-ban, az alábbi kritériumokkal:

1. A vezérlőnek egyszerűnek kell lennie, és igény szerint bármikor új programot lehet rá tölteni.
2. A vezérlőnek biztonságosabbnak kell lennie, mint egy elektromechanikus megoldás, és ár - érték arányban meg kell előznie ezeket a megoldásokkal.”[9]

Az első PLC-t Richard Morley és Odo J. Struger építette meg 1969-ben. Mivel elektrotechnikai szakembereknek készítették így kifejlesztették a hozzá a létra nyelvet, amely felépítésében az áramútrajzok logikáját tükrözte.

A programozható vezérlőket a gépjárműgyártásban alkalmazták kezdetben, amikor a gyártási modell váltásakor a huzalozott vezérlőpanelek újrahuzalozása helyett csak a szoftvert módosították. A PLC elterjedése előtt a gépjármű vezérlők, sorrendezések, és biztonsági záruk logikájának gyártása során több száz relét, időzítőt, és dobvezérlőt használtak, illetve

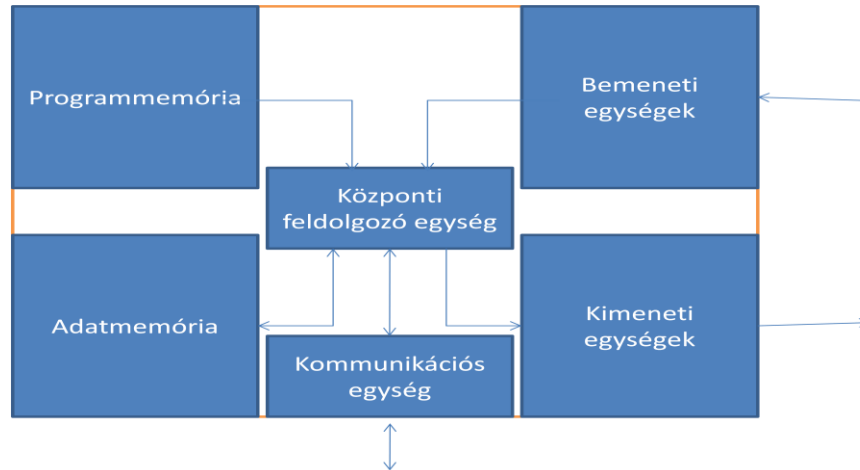
szabályozóüzemet alkalmaztak. Az éves modellváltások során ezek nagyon időigényesek és költségesek voltak, mivel a szerelőknek újra kellett huzalozniuk minden egyes relét. Ma is a gépjárműipar az egyik legnagyobb felhasználója ezeknek az eszközöknek.

3.2 A PLC-k általános jellemzése

Elsősorban abban különböznek a személyi számítógépektől, hogy ellenállnak a szélsőséges környezeti hatásoknak (pl.: por, nedvesség, rázkódás, hideg, meleg) és kiterjedt bemenet/kimenet lehetőségekkel rendelkeznek. A PLC kapcsolja össze az érzékelőket a végrehajtókkal, leolvassa az analóg értékeket (pl.: nyomás, hőmérséklet) és a helymeghatározó rendszer pozíciói, egyes modernebb eszközök képesek képek feldolgozására is. A vezérléshez hidraulikus vagy pneumatikus hengereket, villanymotorokat, reléket, elektromágneseket, vagy analóg kimeneteket használnak. A PLC- k működőképessége ma már odáig fejlődött, hogy magába foglalja a mozgás irányítását, a szekvenciális relé vezérlést, a forgalomirányítást, az elosztott irányítórendszereket és a hálózatokat. A legmodernebb eszközök adatkezelése és tárolása, teljesítmény átviteli és kommunikációs képességei szinte egyenértékűek egy személyi számítógéppel.

Funkcionális felépítésük:

- Központi logikai feldolgozó egység
- Adatmemória
- Programmemória
- Kimeneti egységek
- Bemeneti egységek
- Kommunikációs egység



3.1 ábra A PLC funkcionális felépítése

A központi egység a bemenetek és a kimenetek közötti logikai kapcsolatokat sorosan és ciklikusan hajtja végre a programmemóriában tárolt program alapján. A soros feldolgozás miatt a ciklikus feldolgozást nagy sebességgel kell végrehajtani, hogy kifelé valós idejűnek látszódjon. A programozható vezérlők előnyei: a szabad strukturálhatóság, a gyakorlatilag végtelen kapcsolási szám és a rendszerbe szervezés lehetősége. A szabad strukturálhatóság azt jelenti, hogy a felhasználó az egyedi programmal az univerzális hardvert az adott feladatra alkalmassá teszi.

3.3 TWIDO PLC

Munkánk során a Schneider Electric cég által gyártott TWIDO PLC-k közül a TWD LCDE 40DRF típusú vezérlőt és annak két bővítő modulját, a TWD DRA 16RT és a TWD DDI 16DT. Ez a modell relés kimenetű ezért könnyen megvalósítható vele a forgásirányváltás az elektromos motoroknál.

3.3.1 TWIDO kompakt vezérlők bemutatása:

„A Twido kompakt felépítésű programozható vezérlőinek sora 80-tól 157 x 90 x 70 mm-es, kompakt méretben mindent magában foglaló megoldást kínál. A kompakt vezérlők tízféle típusa kapható, ezek egymástól feldolgozókéességükben és 24 V- os egyenfeszültségű bemeneteik, valamint relé- és tranzisztorkimeneteik számában különböznek (10, 16, 24 és 40

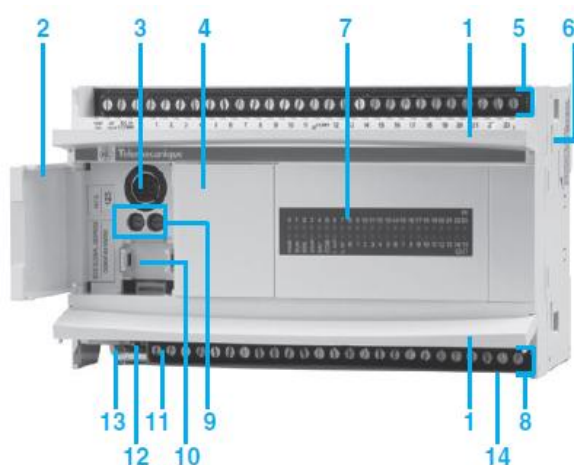
db I/O lehetséges). Tápfeszültségként ezek a vezérlők használhatnak: 100 és 240 V közötti váltakozó feszültséget (biztosítva az érzékelők 24 V- os tápfeszültséget), vagy 19,2 és 30 V közötti egyenfeszültséget (ekkor az érzékelők számára külső segéd táplálást kell biztosítani).

Ez a fajta kompakt vezérlő a következő előnyöket kínálja:

- jelentős számú (akár 40) I/O lehetőséget kis befoglaló méretben, csökkentve ezáltal a panel és konzol méreteket olyan alkalmazásoknál, ahol a hely fontos tényező
- 24 és 40 I/O kiépítésű típusoknál a többféle bővítési lehetőség és termékopció révén a felhasználó olyan mértékű rugalmasságot elvezhet, amely általában csak a nagyobb platformoknál áll rendelkezésre:
 - a 24 I/O lehetőséget nyújtó TWD LC*A 24DRF rendelési számú kompakt vezérlőkkel akár 4 diszkrét és/vagy analóg I/O bővítő és/vagy kommunikációs modult.
 - 40 I/O lehetőséget nyújtó TWD LC** 40DRF rendelési számú kompakt vezérlőkkel akár 7 (diszkrét és/vagy analóg I/O és/vagy kommunikációs) bővítő modult, opcionális modulokat, mint a digitális kijelző, a memóriabővítő kazetta, a valós idejű óra és a további RS 485 vagy RS 232C kommunikációs port.
- A kompakt vezérlős megoldás a huzalozásban is nagy rugalmasságot enged meg. A diszkrét I/O bővítőmodulokhoz (a TWD LC*A 24DRF és TWD LC** 40DRF típusú vezérlőknél) egy sor csatlakozási lehetőség áll rendelkezésre, mint például kivehető csavaros sorkapcsok és a rugós típusú csatlakozások, amelyek gyors, könnyű és megbízható huzalozást biztosítanak. Az Advantys Telefast ABE 7 előhuzalozott rendszer lehetővé teszi a modulok HE 10 típusú csatlakozókkal történő csatlakozását:
 - előszerelt, egyik végen szabad vezetékű kábelek közvetlen csatlakoztatását az érzékelőkhöz illetve működtetőszervekhez,
 - a Twido vezérlő Advantys Telefast ABE 7 előhuzalozott rendszeréhez (csatlakozókábel és ABE 7 közdarabszerelvény).
- A kijelző, illetve a dugaszolható memória lehetősége lehetővé teszi az alkalmazások könnyű beállítását, átvitelét és biztonsági mentését:
 - A digitális kijelző helyi kijelzőként és beállítóeszközként használható,

- a memóriakazetták EEPROM- technológiája lehetővé teszi a programok biztonsági mentését és átvitelét bármilyen kompakt vagy moduláris felépítésű Twido vezérlőre.
- A TwidoSuite szoftver könnyű programozást tesz lehetővé, felhasználva az utasításlista-nyelvű utasításokat és a létranyelv grafikus objektumait.” [1]

3.3.2 TWD LCDE 40DRF vezérlő bemutatása:



3.2 ábra TWD LCDE 40DRF [1]

1. „nyitható sorkapocs védőfedelelet, az 5 jelű kivezetésekhez való hozzáféréshez,
2. egy nyitható védőfedelelet, a hozzáféréshez.
3. egy RS485 típusú, mini- DIN csatlakozót a soros porthoz (amely lehetővé teszi a programozó terminál csatlakoztatását).

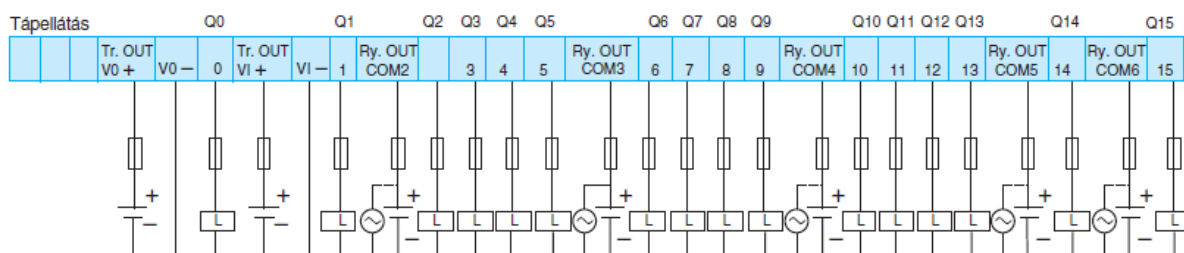
4. egy levehető fedél által védett modulhelyet, a TWD XCP ODC típusú, digitális diagnosztikai és karbantartási kijelző modul részére.
5. egy csavaros sorkapcsot, az érzékelők 24 V (1) tápfeszültséggel való ellátásához és a bemeneti érzékelők csatlakoztatásához.
6. egy csatlakozót, a TWD D**, TWD A** I/O-bővítő- és a TWD NOI10M3/ NCO1M kommunikációs modulok számára (legfeljebb 7 db modul lehetséges).
7. egy kijelzőblokkot, amely a következőket mutatja: a vezérlő állapotát 7 db jelzőfény segítségével (PWR, RUN, ERR, BAT, COM, LACT és L ST)
 - bemenetek és a kimenetek állapotát (INPUT és OUTPUT)
 - egy felhasználói jelzőfényt (STAT), amelyet az alkalmazási program vezérel a felhasználó követelményei szerint
8. egy csavaros sorkapcsot a kimeneti működtetőelemek csatlakoztatásához

9. két analóg beállítópontot,
10. egy bővítőcsatlakozót, második RS232C/RS485 soros port hozzáadásához, egy TWD NAC *** típusú adapter segítségével,
11. egy csavaros sorkapcsot, a 100-240 V feszültségű hálózat vagy a 19,2-30 V- os táplálás csatlakoztatásához.

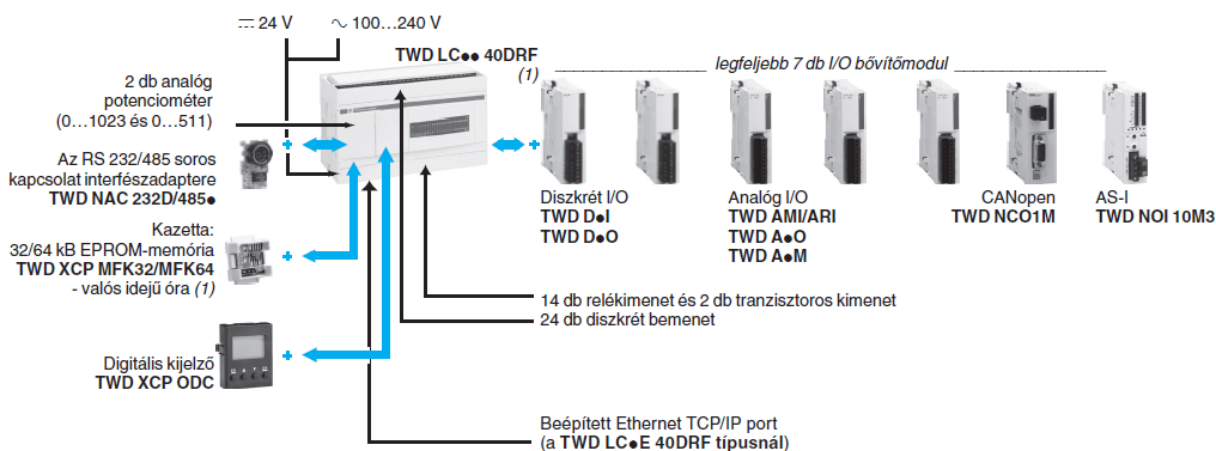
A vezérlő alján keresztül hozzáférhető:

12. egy csatlakozó, a 32 vagy 64 kB kapacitású, TWD XCP MFK32/MFK64 típusú memóriakártya számára,
13. egy, a vezérlő alján keresztül hozzáférhető RJ45 csatlakozó, az Ethernet TCP/IP hálózatra történő csatlakozáshoz,
14. egy modulhely, a vezérlő belső RAM- memóriájának biztonsági táplálására szolgáló, opcionális akkumulátor befogadására.

A kompakt vezérlőket alapesetben szimmetrikus sínre, szerelvénylapra, vagy panelre lehet felszerelni (2 db 4,3 mm átmérőjű furat segítségével).” [1]



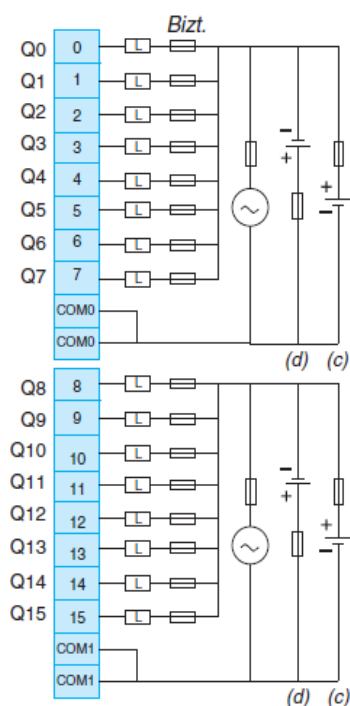
3.3 ábra TWD LCDE 40DRF kompakt vezérlő bekötési rajza [1]



3.4 ábra Csatlakoztatható bővítőmodulok és kiegészítők [1]

3.3.3 Diszkrét Input/Output bővítőmodulok

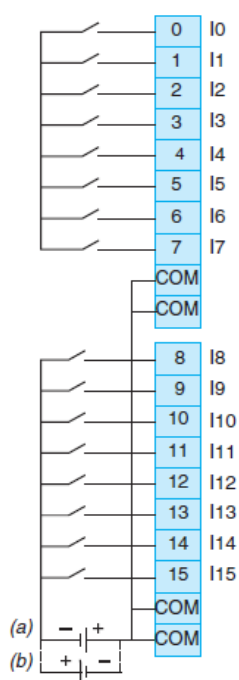
TWD DRA 16RT kimeneti bővítőmodul:



- 16 db relékimenettel rendelkezik
- Kimenetei kivethető, csavaros sor kapoccsal vannak ellátva
- „Kivezetések:
 - COM0 belsőleg összekötve
 - COM1 belsőleg összekötve
 - COM0 es COM1 egymástól független
- (c) Nyelőkimenet (negatív logika)
- (d) Forráskimenet (pozitív logika)” [1]

3.5 ábra Kimeneti bővítőegység és bekötési rajza [1]

TWD DDI 16DT bemeneti bővítőmodul:



- 16 - 24 V- os egyenáramú bemenetekkel rendelkezik
- 8ms válaszidő
- „A COM kivezetések belsőleg össze vannak kötve
- (a) Forrásbemenet (negatív logika).
- (b) Nyelőbemenet (pozitív logika).”[1]

3.6 ábra Bemeneti bővítőegység és bekötési rajza [1]

3.3.4 Tápegység

A gyártósorunkon 24 V- os motorokat használtunk ezért szükség volt egy olyan tápegységre, ami a hálózati feszültségből előállítja azt. Erre a célra a Schneider cég által gyártott ABL8REM24050 eszközt használtuk.



1. „2,5 mm² -es zárt, csavaros kivezetések a bemeneti feszültség csatlakoztatásához (egyfázisú N-L1, vonalfeszültségű L1-L2)
2. Üvegből készült, védő zárófedél
3. A bemeneti feszültség állapotát jelző LED (narancssárga).
4. A kimeneti DC- feszültség állapotát jelző LED (zöld).
5. Az üvegből készült zárófedél retesze (tömíthető)
6. Rácsíptethető jelölécímke.
7. Kimeneti feszültséget beállító potenciométer
8. 2,5 mm² -es, zárt, csavaros sorkapocs a kimeneti DC- feszültség csatlakoztatásához” [1]

3.7 ábra ABL8REM24050 tápegység [1]

3.3.5 TWIDO Suit

„A TwidoSuit egy grafikus fejlesztő környezet a Twido programozható vezérlők alkalmazásainak elkészítéséhez, konfigurálásához és karbantartásához.

ATwidoSuit teszi lehetővé a vezérlő programok beírását a TwidoSuit létra vagy lista szerkesztőinek segítségével, valamint a program átvitelét a vezérlőre és futtatását is biztosítja.

TwidoSuit fő jellemzői:

- Szabványos Windows felhasználói interfész
- A Twido vezérlők programozása és konfigurálása
- Vezérlő kommunikáció és vezérlés

A programozható vezérlő egy vezérlőprogram alapján olvassa a bemeneteket, logikai döntéseket hoz, és kiválasztja a kimeneteket. A Twido vezérlőt irányító vezérlőprogram a Twido programozási nyelvek egyikén írt utasításokból áll.

A Twido vezérlőprogramok létrehozásához az alábbi nyelveket lehet használni:

- Utasítás Lista nyelv

Az Utasítás Listaprogramozási nyelv (IL vagy Lista) egy sor alapú, szöveges, Boole-algebrai nyelv, amelyet logikai és numerikus műveletek írására használnak.

- Létra nyelv

A Létraprogramozási nyelv egy létrafok alapú, grafikai Boole-algebrai nyelv.

- Grafcet nyelv

Twido támogatja a Grafcet Lista utasítások használatát, de nem támogatja a grafikai Grafcet nyelvet. Ezen programozási nyelvek használatával PC-n (személyi számítógépen) is létrehozhatóak és szerkeszthetőek a Twido vezérlőprogramok. A Lista/Létra reverzibilitás lehetővé teszi, hogy tetszés szerint átfordítsunk egy programot Létra nyelvből Lista nyelvbe, illetve Lista nyelvből Létra nyelvbe.

Utasítás Lista nyelv:

Az Utasítás Lista nyelven írt program egy sor utasításból áll, amelyeket a vezérlő szekvenciálisan hajt végre. A következőkben egy példát láthatunk a Lista programokra.

```
0  BLK  %C8
1  LDF  %I0.1
2  R
3  LD   %I0.2
4  AND  %M0
5  CU
6  OUT_BLK
7  LD   D
8  AND  %M1
9  ST   %Q0.4
10 END_BLK
```

3.8 ábra Utasítás lista példa [2]

Grafcet nyelv:

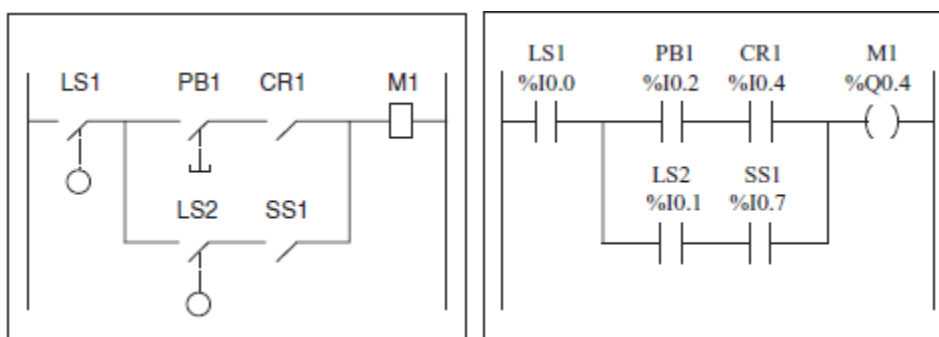
A Grafcet egy analitikus módszer, amely minden szekvenciális vezérlőrendszer lépések sorára bont le, amelyekhez különböző műveletek, átvitelek és feltételek tartoznak.

Létra nyelv:

A Létra diagramok hasonlóak a relé logikai diagramokhoz, amelyeket relé vezérlőkörök reprezentálására használnak. A kettő közötti fő különbséget a létraprogramozás alábbi tulajdonságai képezik, (ezek nincsenek jelen a relé logikai diagramokban):

- Minden bemenetet kontaktusszimbólumok reprezentálnak ($-||-$).
- Minden kimenetet tekercsszimbólumok reprezentálnak ($-()-$).
- A grafikus létrautasítások készlete numerikus műveleteket tartalmaz.

A következő kép egy relé logikai áramkör, valamint az azzal ekvivalens létra diagram egyszerűsített huzalozási diagramját mutatja be.



3.9 ábra Relé logikai áramkör

Létra diagram [2]

Létrafokok:

Egy létra nyelven írt program létrafokokból tevődik össze, amelyek két függőleges tápsín közé rajzolt grafikai utasítási készleteknek felelnek meg. A vezérlő szekvenciálisan hajtja végre a létrafokokat.

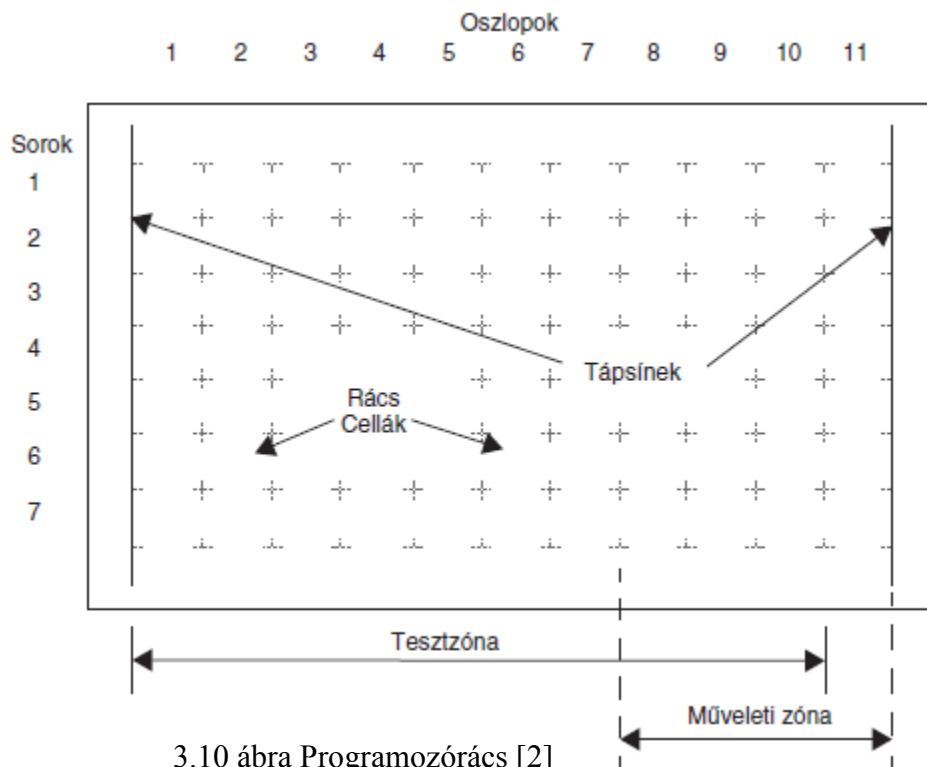
A grafikai utasítások készlete a következő funkciókat tartalmazza:

- Vezérlő bemenetei/kimenetei (nyomógombok, érzékelők, relék, ...)
- Vezérlő funkciói (időzítők, számlálók, ...)
- Matematikai és logikai műveletek (összeadás, osztás, AND, XOR, ...)
- Összehasonlítás műveletek, illetve egyéb numerikus műveletek ($A < B$, $A = B$, léptetés, forgatás, ...)
- Belső változók a vezérlőben (bitek, szavak, ...)

Ezek a grafikai utasítások vízszintes és függőleges csatlakozásokkal vannak ellátva, amelyek végül egy vagy több kimenethez és/vagy művelethez vezetnek. Egy létrafok nem képes egynél több kapcsolt utasításcsoport támogatására.

Programozórács:

Minden létrafok egy hét sort, illetve tizenegy oszlopot tartalmazó rácsból áll, amely két zónára van felosztva, az alábbi ábra szerint.



3.10 ábra Programozórács [2]

Rácszónák:

A létra diagram programozórács két zónára oszlik:

- Egy tesztzóna, amely tartalmazza azokat a feltételeket, amelyeknek igaznak kell lenniük egy művelet létrejöttéhez.
- Egy műveleti zóna, amely tartalmazza azt a kimenetet vagy műveletet, amely a társított tesztfeltételek következményének eredményét képezi.

Utasítások bevitele a rácsba:

Egy létrafok egy hétszer tizenegyszeres programozórácscsal rendelkezik, amely a rács bal felső sarkában lévő cellában kezdődik. A programozás az utasításoknak a rács celláiba történő beviteléből áll. A tesztutasítások, összehasonlítások illetve funkciók a tesztzóna celláiba kerülnek bevitelre, melyek balra zártak. A tesztlogika kontinuitást biztosít a műveleti zóna számára, amelybe tekercek, numerikus műveletek, illetve programlefoylás-vezérlési utasítások kerülnek bevitelre, melyek jobbra zártak. A rácsban a létrafok fentről lefelé, illetve

balról jobbra kerül megoldásra vagy végrehajtásra (a tesztek elvégzésre, a kimenetek hozzárendelésre).


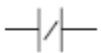
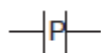
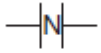
Létrafok fejlécek: A létrafok mellett egy létrafok fejléc jelenik meg közvetlenül a létrafok felett. A létrafok fejléc segítségével dokumentálja a létrafok logikai célját. A létrafok fejléc a következő információkat tartalmazhatja:

- Létrafok száma
- Címkék (%Li)
- Szubrutin nyilatkozatok (SRi:)
- Létrafok címe
- Létrafok megjegyzések

Létranyelv grafikai elemei:

Kontaktusok:

A kontaktus grafikai elemei a tesztzónában kerülnek programozásra és egy cellát foglalnak el (egy sor magas és egy oszlop széles).

Név	Grafikai elem	Utasítás	Funkciók
Alapállapotban nyitott kontaktus		LD	Átmenő kontaktus, amikor az azt vezérlő bitobjektum 1-es állapotban van.
Alapállapotban zárt kontaktus		LDN	Átmenő kontaktus, amikor az azt vezérlő bitobjektum 0 állapotban van.
Kontaktus felfutó él észlelése		LDR	Felfutó él: észleli az azt vezérlő bitobjektum változását 0-ról 1-re.
Kontaktus egy lefutó él észlelése		LDF	Lefutó él: észleli az azt vezérlő bitobjektum változását 1-ről 0-ra.

Kapcsolati elemek:

A grafikai kapcsolati elemek segítségével lehet csatlakoztatni a teszt, illetve a művelet grafikai elemeit.

Név	Grafikai elem	Funkciók
Vízszintes csatlakozó	—	Sorban összekapcsolja a teszt, illetve művelet grafikai elemeit a két lehetséges sor között.
Alsó csatlakozó		Párhuzamosan összeköti a teszt, illetve művelet grafikai elemeit.

Tekercek:


A tekercs grafikai elemek a tesztzónában kerülnek programozásra, és egy cellát foglalnak el (egy sor magas, illetve egy oszlop széles).

Név	Grafikai elem	Utasítás	Funkciók
Direkt tekercs	-()-	ST	A kapcsolt bitobjektum a tesztzóna eredményének értékét kapja.
Negált tekercs	-(∩)-	STN	A kapcsolt bitobjektum a tesztzóna eredményének negált eredményét kapja.
SET tekercs	-(S)-	S	A kapcsolt bitobjektum 1-re van beállítva, amikor a tesztzóna eredménye 1.
RESET tekercs	-(R)-	R	A kapcsolt bitobjektum 0-ra van beállítva, amikor a tesztzóna eredménye 1.
Ugrás és szubrutinhívás	->>%Li ->>%SRi	JMP SR	Csatlakoztatás címkézett utasításhoz, a lefutással megegyező vagy azzal ellentétes irányban.
Átmeneti feltétel tekercs	-(#)-		Grafcet nyelvben megadva, akkor kerül használatra, amikor az átmenetekhez kapcsolt átmeneti feltételek programozása átváltást eredményez a következő lépésre.

Visszatérés egy szubrutinból	<RET>	RET	Lehetővé teszi a hívómodul visszatérését, amikor a tesztzóna eredménye 1.
Leállító program	<END>	END	Leállítja a program futását, amikor a tesztzóna eredménye 1.

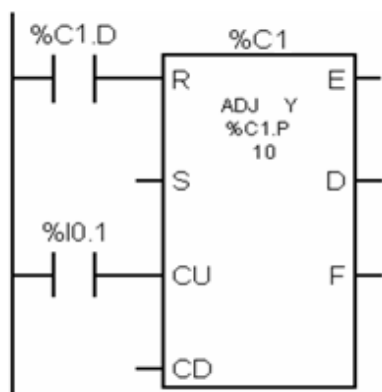
Funkcióblokkok:

A funkcióblokkok grafikai elemei a tesztzónában kerülnek programozásra, és négy sor illetve két oszlop méretű cellát igényelnek (kivéve a nagysebességű számlálókat, amelyekhez öt sorra és két oszlopra van szükség).” [2]

Név	Grafikai elem	Funkciók
Időzítők, számlálók, stb.		Mindegyik funkcióblokk bemeneteket és kimeneteket használ, amelyek lehetővé teszik a kapcsolatot más grafikai elemekkel. Megjegyzés: a funkcióblokkok kimenetei nem lehetnek egymáshoz kapcsolva (függőleges csatlakozás).

A Twido Suit sokféle funkcióblokkal rendelkezik, de mi a programunkban csak számlálókat és időzítőket használtunk.

Számláló jellemzése:



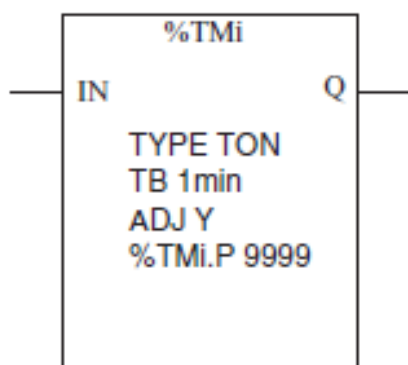
3.11 ábra Számláló funkcióblokkja

A funkcióblokkja így néz ki a programban. Lehet vele felfele és lefele számlálást végezni.

%C0	A számláló száma
%C0.P	A beállított érték
R	Reset bemenet, ha itt 1 a bemenet akkor nullázza a számláló értékét
S	Set bemenet, ha itt 1 a bemenet akkor a kezdeti értékre állítja vissza a számláló értékét
CU	A felfele számláló bemenet. Egy felfutó élre egyet számol felfele.
CD	A lefele számláló bemenet. Egy felfutó élre csökkent a számláló értékét
E	Alulcsordulási kimenet
D	Az előre beállított kimenet elérve
F	Túlcsordulási kimenet

A programban használható számlálók 0-9999 közötti tartományban tudnak számlálni. Ennél nagyobb számokig történő számlálás esetén kettő vagy több számláló segítségével lehet megoldani. Ehhez használhatjuk az E és F kimeneteket.

Időzítők jellemzése:



Az időzítő funkcióblokkja így néz ki. Háromfajta időzítő használata lehetséges:

- TON Bekapcsolás késleltetés
- TOF Kikapcsolás késleltetés
- TP Meghatározott hosszúságú impulzus létrehozása

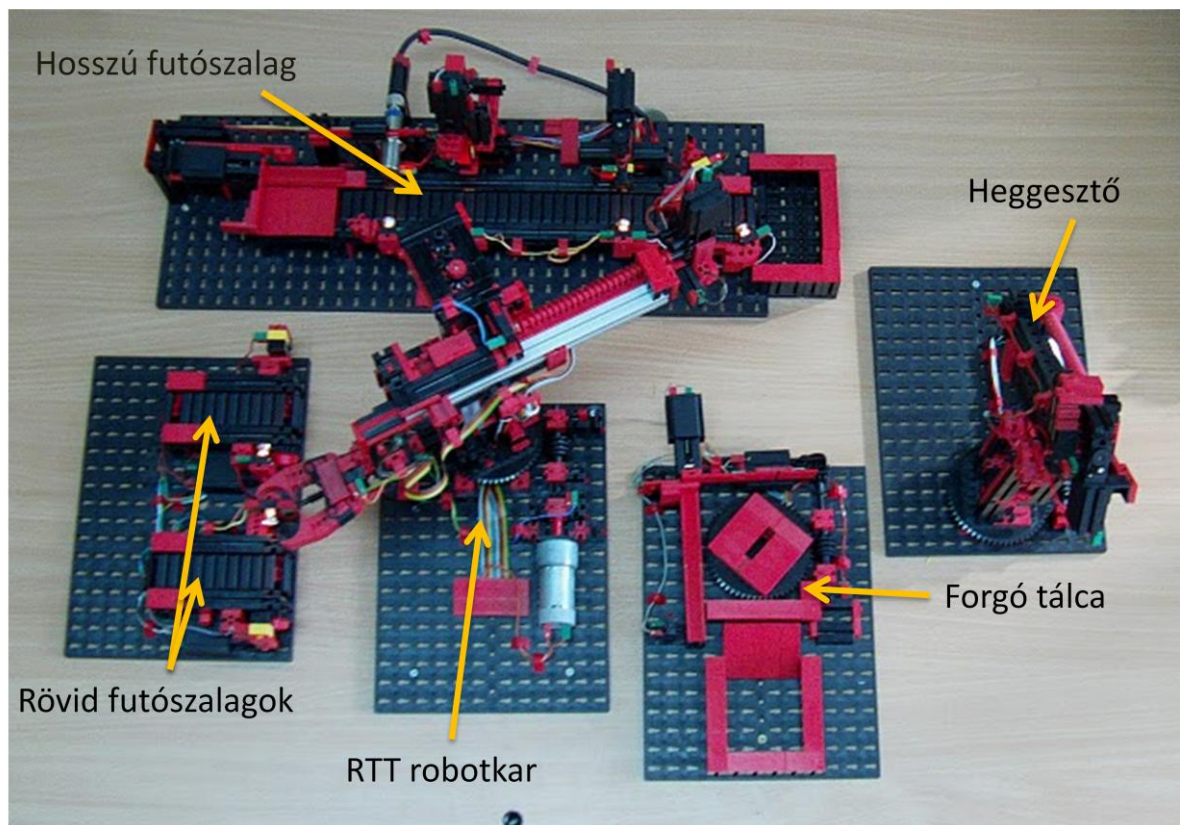
3.12 ábra Időzítő funkcióblokkja

%Tmi	Az időzítő száma, kompakt vezérlőnél 0-127
TB	Az időegység (1perc, 1s, 100ms, 10ms, 1ms)
TYPE *	Az utána álló szöveg határozza meg az időzítő típusát
%Tmi.P	A beállított értéket mutatja az utána álló szám
IN	Az időzítőt elindító bemenet
Q	Az időzítő kimenete, a típusa határozza meg a működését.

4. A Gyártósor

4.1 A Gyártósor felépítése

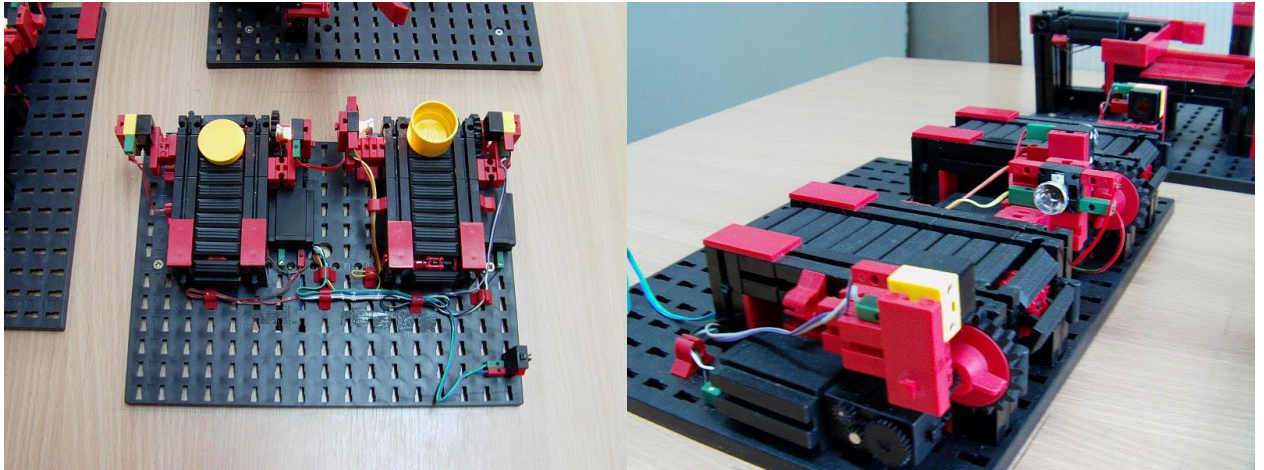
Ahogy a dolgozat elején említettem Fischertechnik által gyártott elemekből építettük föl a gyártósort. A robotkart és a hegesztőrobotot összeszerelve kaptuk meg, a többi részt pedig mi raktuk össze fel.



4.1 ábra Gyártósor felépítése

4.1.1 Rövid futószalagok

Két rövid futószalag, amelyek a robotkarhoz továbbítják a termék alsó és felső részét. A meghajtásról kettő darab 24V-os motor gondoskodik. Az érzékelés szintén 24V-os izzókkal és a hozzájuk tartozó fénykapukkal történik oly módon, hogy ha fénysugár megszakad az érzékelő jelet küld és megállítja a szalagot, mert a termék a megfelelő pozícióba ért.

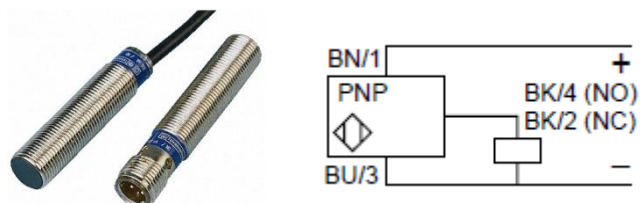


4.2 ábra Rövid futószalagok

4.1.2 Hosszú futószalag

Az elején található egy síkfelület ahova a robotkar elhelyezi a munkadarabokat, majd egy motorikus betolóegység amely a szalagra helyezi azokat. Egy induktív érzékelő, amely a selejtkezeléshez szükséges, valamint egy présgép és egy fűrőgép helyezkedik itt el. A szalag végén található egy tároló, amibe a selejtterméket gyűjtjük. A meghajtásról itt is 24V-os motorok gondoskodnak, melyek közül kettő mind a két irányba képes mozogni, ezek a prést és a betoló egységet működtetik. Ezeket motorokat végállás kapcsolók segítségével mozgatjuk az egyik állapotból a másikba. Mindegyik munkafolyamat alatt található egy fénykapu, ezek segítségével állítjuk meg a megfelelő helyen a terméket, valamint egy a szalag végén ahonnan könnyen át tudja helyezni a robotkar a munkadarabot a következő munkafolyamathoz.

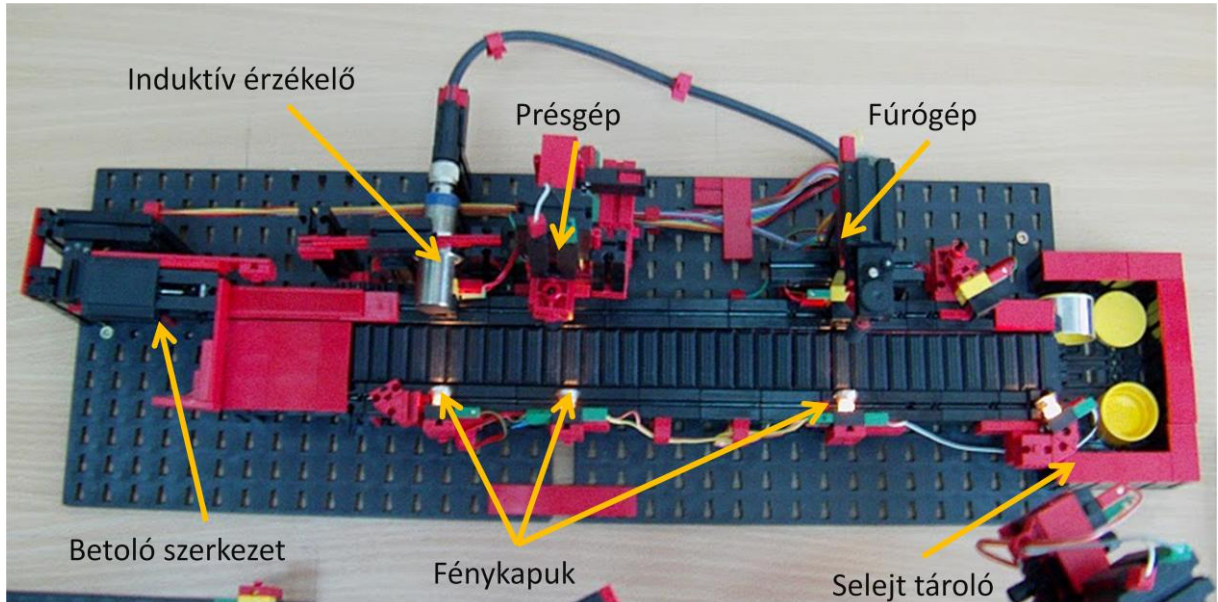
Az induktív érzékelőnek egy XS1-M18PA370D három vezetékes PNP tranzisztoros Schneider által gyártott eszközt használtunk.



4.3 ábra Az induktív érzékelő és bekötési módja

Az induktív érzékelők működési elve a következő: azt a fizikai elvet használják ki, hogy minden vezetőképes anyagban fellépő örvényáramok által okozott veszteségek

megváltoztatják a rezonancia-áramkör jósági tényezőjét. Az LC rezgőkör nagyfrekvenciás váltakozó mezőt hoz létre, ha ide egy elektromosan vezetőképes anyag kerül, akkor az energiát von el a rezgőkörből és ez a változás egy jelet hoz létre, amelyet az érzékeléshez használunk. A működési elvből adódóan bármilyen fémet képes érzékelni.



4.4 ábra Hosszú futószalag

4.1.3 Hegesztőrobot

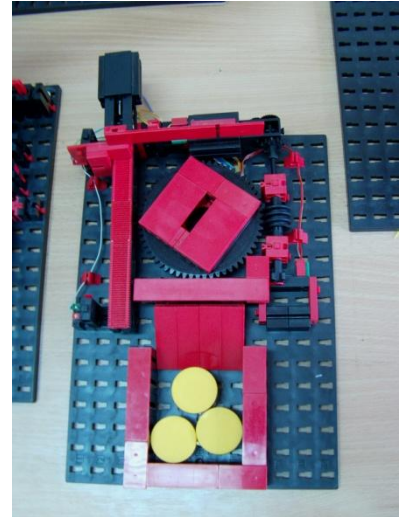
A hegesztőrobot, ami forgatható talpazattal rendelkezik, valamint a tetején lévő kart tudja mozgatni. Aminek a végén egy LED található, amely a hegesztés imitálja. Két darab 24V-os motor található rajta, az egyik nagyobb teljesítményű, amelyet a forgatáshoz használtunk fel. Található rajta 4 érzékelő, amelyek közül kettő számlálóként funkcionál, a másik kettő a végállás kapcsolóként. A számlálók segítségével tudjuk meghatározni, hogy mennyit forduljon a robot és a fenti kart mennyire tolja ki.



4.5 ábra Hegesztőrobot

4.1.4 Forgó tálcá

Az általunk kitalált eszköz, amelyet forgó tálcának nevezünk el. A munkadarabot forgatja hegesztés közben a robot előtt, valamint található rajta egy olyan eszköz, amely a készterméket a raktárba tolja. A meghajtásról két darab szintén 24V-os motor gondoskodik, amelyik a raktárba továbbítja a terméket mind a két irányba képes működni. Három érzékelőt használtunk fel ezen az egységen kettő végállás érzékelőként az egyik pedig számlálóként működik, melynek segítségével tudunk egy teljes fordulatot megtenni a tálcával.



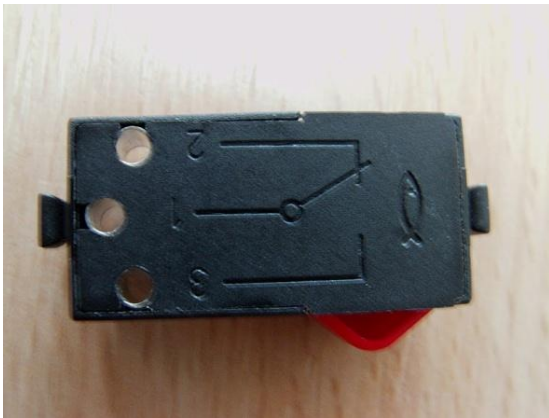
4.6 ábra Forgó tálcá

4.1.5 Robotkar

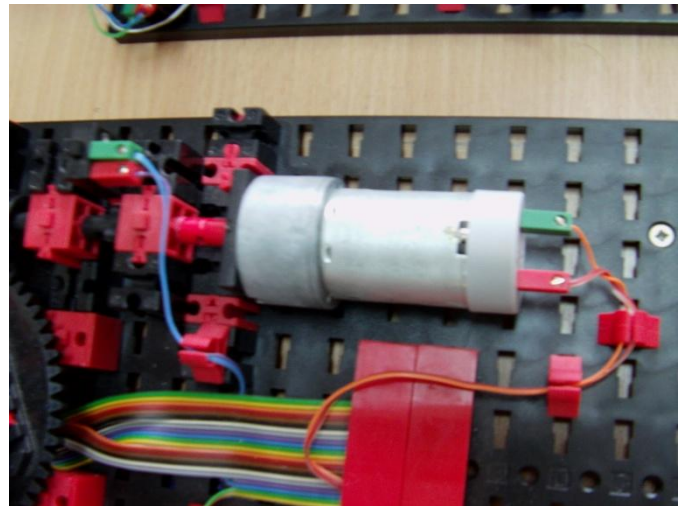
A gyártósor egyik fő alkotóeleme, melynek segítségével mozgathatjuk a munkadarabokat az egyik munkafázisból a másikba. A robotkart más néven manipulátornak is nevezik. Ez egy RTT kar, ez azt jelenti hogy képes a saját tengelye körül elfordulni, valamint függőleges és vízszintes irányban mozogni. Az ilyen mozgásokra képes robotok egy úgynevezett henger koordinátarendszerben mozognak. Az ilyen manipulátorok sokféle feladatot elláthatnak, attól függően, hogy milyen eszköz van végükre szerelve (pl.: festékszóró, fúró, hegesztő, stb.). A mi modellünkben lévő karnak a végén egy úgynevezett kétpofás megfogó van, melynek segítségével mozgatja a munkadarabokat egy adott helyről egy másik megadott pozícióba.

Az általunk használt eszköz 4 darab 24V-os egyenfeszültségű motorral rendelkezik, amelyeknek a forgásirányát az áram irányának megváltoztatásával befolyásolhatjuk. Így lehet a kart felemelni és leengedni, balra illetve jobbra forgatni, kitolni és behúzni a kart, valamint a megfogóval megfogni és elengedni valamit. A függőleges és forgó mozgáshoz nagyobb teljesítményű, a vízszintes és a megfogó működéséhez kisebb teljesítményű motorok vannak felszerelve. A manipulátoron található nyolc darab érzékelő, ezek az érzékelők lehetnek nyitó

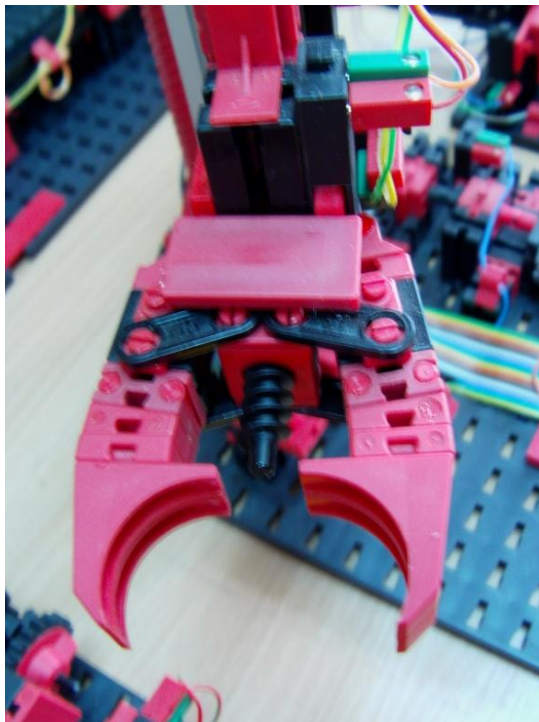
illetve záró érzékelők. Mi záró érzékelőként használtuk ez azt jelenti, hogy az 1-es és a 3-as lábakat kötöttük össze. Az érzékelőknek a felét végállás kapcsolóként használtuk, amelyek akkor érzékelnek amikor a kar teljesen balra van forgatva és felemelve valamint be van húzva illetve a megfogó elengedve van. A másik négy érzékelőt számlálóként használjuk, amelyben egy tengelyfordulat négy számlálást jelent. A robotkarnak a forgás végállását eltávolítottuk a gyári pozícióból és a másik oldalra helyeztük, erre a kar alaphelyzete miatt volt szükség.



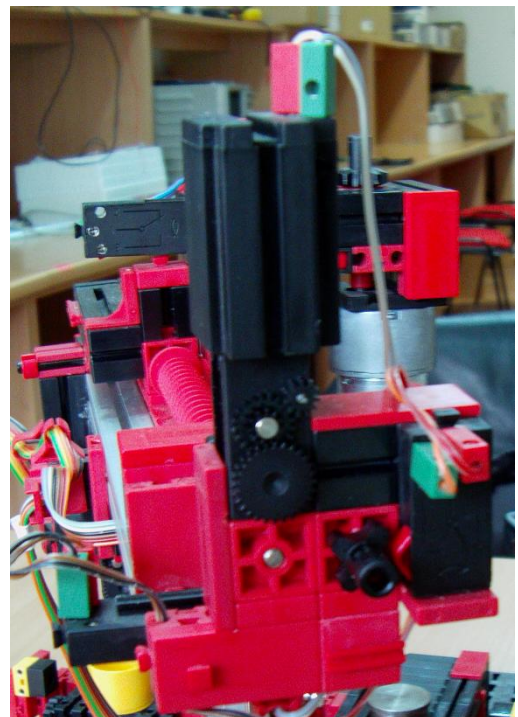
4.7 ábra Érzékelő



4.8 ábra Nagyobb teljesítményű motor



4.9 ábra A kétfás megfogó



4.10 ábra Kisebb teljesítményű motor és a vízszintes számláló

4.2 Folyamat

A két rövid futószalag végére helyezzük a munkadarab alsó és felső részeit, az ott elhelyezett fénykapuk érzékelik, hogy nincsen a robotkar mozgási tartományában ezért oda mozgatják azokat. Ezután a robotkar először felemeli az alsó rész és azt a hosszú futószalag végén lévő sík részre helyezi, majd ugyanez megismétli a felső résszel is, de azt az alsó rész tetejére helyezi el. Itt a betoló a futószalagra tolja a munkadarabot, ahol történik időrendi sorrendben egy selejtkezelés, egy préselés és egy fúrás, ezek után tovább halad az utolsó érzékelőig. Ha a termék selejt, akkor a szalag végén lévő tárolóba kerül és a gyártási folyamat kezdődik előlről, a selejt terméken nem végzünk préselést és fúrást. Nem selejtes termék esetén a robotkar azt áthelyezi a hegesztő előtt található forgó tálcára, ahol a hegesztő robot elvégzi a hegesztést miközben a tálca teljesen körbefordul, ha ez megvan akkor a késztermék továbbítódik a raktárt szimbolizáló tárolóba, és a gyártás kezdődik előlről.



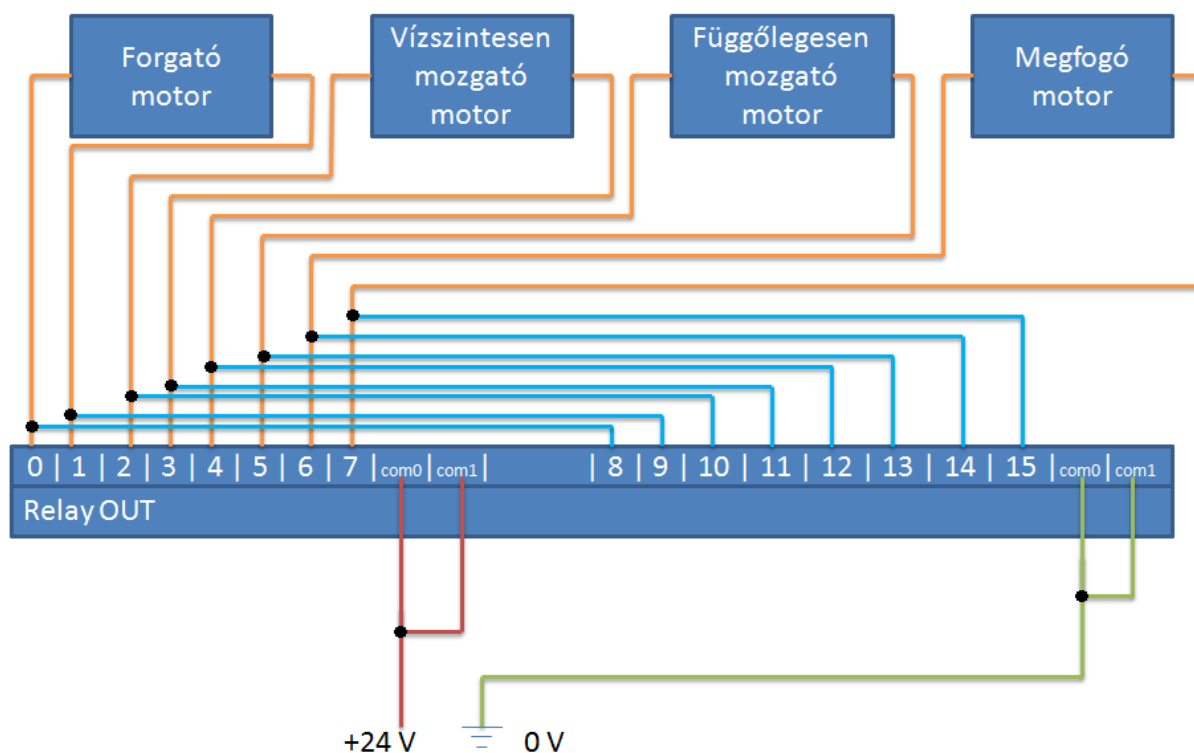
4.11-12 ábra Selejtes és Jó termékek

5. Az elektronika bekötése

5.1 Robotkar bekötése

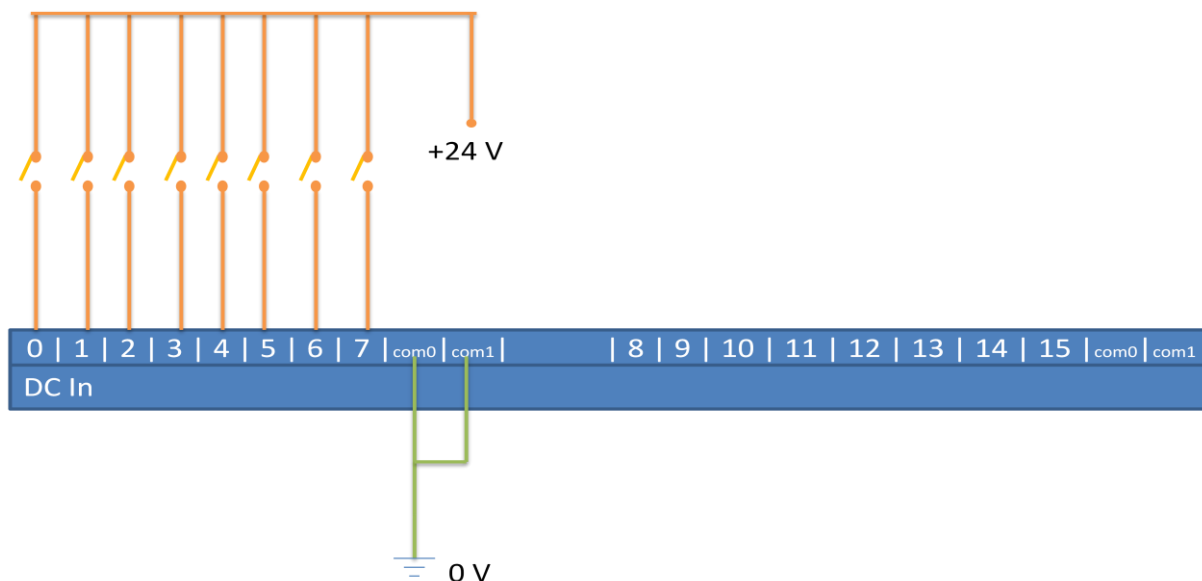
A gyártósor felépítése után az összes motort és érzékelőt össze kellett kötni a PLC be és kimeneteivel és csak ezután kezdhettünk hozzá a programozáshoz. Mivel a robotkart összeszerelve kaptuk kézhez így annak a bekábelezése nem volt szükséges, mivel az összes érzékelő és motor vezetéke egy 26 pólusú csatlakozóba ki volt vezetve. Ezekből összesen 24 van használatban mivel a karon 4 motor és 8 érzékelő található. Az 1. és 2. pólus a forgató motor, a 7. és 8. a vízszintes mozgásért felelős motor, a 13. és 14. a függőleges mozgást végző motor, a 19. és 20. pólus a megfogó nyitását és zárását végző motor kivezetései. Mivel ezek a motorok mind a két irányba forgathatóak, így az egyik relés bővítőmodulra vannak kötve. Egy pólusra két kábel van kötve, egy +24V és egy 0V, tehát egy motorhoz összesen 4 kábel megy. A relés kimentek az alábbiak szerint vannak bekötve: a forgatáshoz jobbra 0-ás és a 8-as,

balra 1-es és 9-es, a felemeléshez 4-es és 12-es, a leengedéshez 5-ös 13-as, a kitoláshoz 2-es és 10-es, a behúzáshoz 3-as 11-es, a megfogáshoz 6-os 14-es és az elengedéshez 7-es 15-ös.



5.1 ábra A robotkar bekötése a kimeneti bővítőmodulba

A 3. és 4. pólus a forgás végérzékelő, az 5. és 6. a forgás számlálója, 9. és 10. a vízszintes végálláskapcsoló, a 11. és 12. a vízszintes mozgás számlálója, a 15. és 16. a függőleges mozgás végérzékelője, a 17. és 18. a függőleges mozgás számlálója, 21. és 22. a megfogó végérzékelője, amely akkor érzékel ha teljesen nyitott állapotban van és végül a 23. és 24. a megfogó számlálója. Az érzékelők pólusai páronként fölcserélhetőek, az egyik pólust a +24V-ra kell kötni a másikat a PLC bemenetére. A robotkar érzékelőit a bemeneti bővítőmodulra kötöttük, az alábbiak szerint: a 0-ás a forgás végkapcsolója az 1-es pedig a számlálója, 2-es a vízszintes mozgás végállása a 3-as pedig a számlálója, a 4-es függőleges mozgás végérzékelője az 5-ös pedig a számlálója, 6-os a megfogó végállása a 7 pedig a számlálója.



5.2 ábra A robotkar érzékelőinek bekötése

5.2 A rövid és a hosszú futószalag bekötése

Mivel ezeket a részeket mi állítottuk össze így ezek kábelezését is nekünk kellett megoldani, hogy könnyebben kezelhető legyen az összes kábelt érvéghüvelyekkel láttuk el és sorkapcsokba kötöttük.

A rövid futószalagoknál két darab egy irányba forgó motor van, ezért ezeknek az egyik kivezetésüket a 0V-ra kell kötni, a másikat pedig PLC kimenetére. Az alsó részt szállító szalag motorja a harmadik bővítőmodul 10-es, a felső részt szállító pedig a 11-es kivezetésre van kötve. Található itt két fénykapu, amelyek a PLC 0-ás és 2-es bemeneteire vannak kötve, a másik kivezetésüket szintén a +24V-ra kellett kötni, mint a nyomógombos érzékelőknél.

A hosszú futószalagon négy motor és számos érzékelő található, ezért a bekötése nem bizonyult egyszerű feladatnak. Minden egyes munkafázisnál valamint a szalag végénél is található egy fénykapu, amelyek következőképpen vannak bekötve: a fémérzékelő alatti az a PLC 10-es bemenetére, a prés gép alatt található a bemeneti bővítőmodul 9-es bemenetére, a fűrőgép alatt található szintén a PLC 1-es bemenetére, valamint a szalag végénél található is a PLC 11-es bemenetére. A motorok közül kettő csak egy irányba mozog, ezek pedig a szalagot

és a fűrőgépet mozgató motorok. Ezeknek az egyik lábuk a PLC 0-ás illetve 1-es kimeneteire vannak kötve a másik kettő pedig a 0V-ra. A prés gép motorja a harmadik kimeneti bővítőmodul 4-es és 12-es kimenetére a lefele mozgáshoz, a felfele mozgáshoz pedig az 5-ös és 13-as kivezetéshez. A prés alsó végérzékelője a PLC- n a 7-es a felső pedig 6-os bemenetre van csatlakoztatva. A szalag elején található betoló eszköz motorja a szintén a harmadik kimeneti bővítőmodulra van kötve, az alábbi módon: előrefele a 7-es és 14-es, hátrafele pedig a 6-os és 15-ös kimenetet használtuk. Az ide tartozó végállaskapcsolók a PLC bemeneteire vannak kötve, a hátsó az 5-ösre az első pedig a 4-re. Az induktív érzékelőt szintén itt a 9-es bemenetre kötöttük.

5.3 A hegesztő és a forgó tálca bekötése

A hegesztőn két motor található és négy darab érzékelő, amelyek a következő képen vannak bekötve: a forgató motor jobbra forgásához a 8-as és 13-as balra forgatásához pedig a 9-es 12-es, a felső kar kitolásához a 6-os 11-es behúzásához pedig a 7-es 10-es kimeneteket használjuk a PLC- n, valamint a kar végén található lámpa felkapcsolásához 14-es kimenetet. Az itt található négy érzékelőből kettő számlálóként van használatban. Ezek a PLC bemeneteire vannak kötve az alábbi módon: a forgás végérzékelője a 18-as az itt lévő számláló a 19-es, a felső kar végállása a 16-os a számlálója pedig a 17 –es.

A hegesztő előtt található forgó tálcán két motor van, amelyek közül az egyik két irányba tud mozogni, valamint három darab érzékelő. A forgató motor a vezérlőn az 5-ös kimenetre van kötve a számlálója pedig 14-es bemenetre. A készterméket a raktárba továbbító kar motorja a harmadik kimeneti bővítőmodulon az előre menethez a 0-ás és 14-es a hátramenethez pedig az 1-es 15-ös kivezetésekhez. Az itt található hátsó végérzékelő a 13-as az első pedig a 15-ös bemenetre van kötve a PLC- n.

6. Programozás

Mivel a feladat csoportmunka volt így elosztottuk egymás között a programozási feladatokat, de egymást is segítettük, ha elakadtunk valahol. Az én feladatomban a robotkar programozása volt. Először csapatmunkában egy alaphelyezési rutin megírásával kezdtük, erre azért volt szükség, mert úgy állítottuk össze a gyártósort hogy alaphelyzetből kezdődjön a gyártás. A alaphelyzetet egy szubrutin segítségével valósítottuk meg. Amelyben minden motort, aminek nincs a végérzékelője bekapcsolt állapotban addig mozgatunk amíg ez a feltétel nem teljesül. Amint ez igazgá válik a motorok mozgását leállítjuk. Minden motorhoz páronként egy set és egy reset tekerccs tartozik a programban. A set tekerccsel elindítjuk a reset- tel pedig megállítjuk. Valamint elhelyeztünk a programban egy run/stop és egy RESET utasítást is. A run/stop utasítást a PLC- n a 20 bemeneten lévő gombbal lehet használni. Ennek segítségével lehet a vezérlőn lévő programot elindítani illetve megállítani, ha az utóbbi műveletet választjuk és utána újból elindítjuk onnan folytatja a program futtatását ahol abba hagyta. A RESET utasítás egy vészleállítóként működik, ha ezt használjuk a programot nem tudja tovább folytatni, az elejétől kell elkezdni, természetesen miután alaphelyzetbe állítottunk mindent.

A mielőtt a robotkar programozásához hozzákezdtem volna, a két rövid futószalagot kellett beprogramozni. Az algoritmusuk a következő mielőtt megnyomnánk a start gombot ezek a szalagok működésbe lépnek és addig forognak, amíg egy munkadarabot nem szállítanak az elől található fénykapu elé, ekkor érzékelik hogy a szalag másik végére ért a munkadarab ezért leállítják a motorok forgását. Ezután indíthatjuk el a gyártási folyamatot.

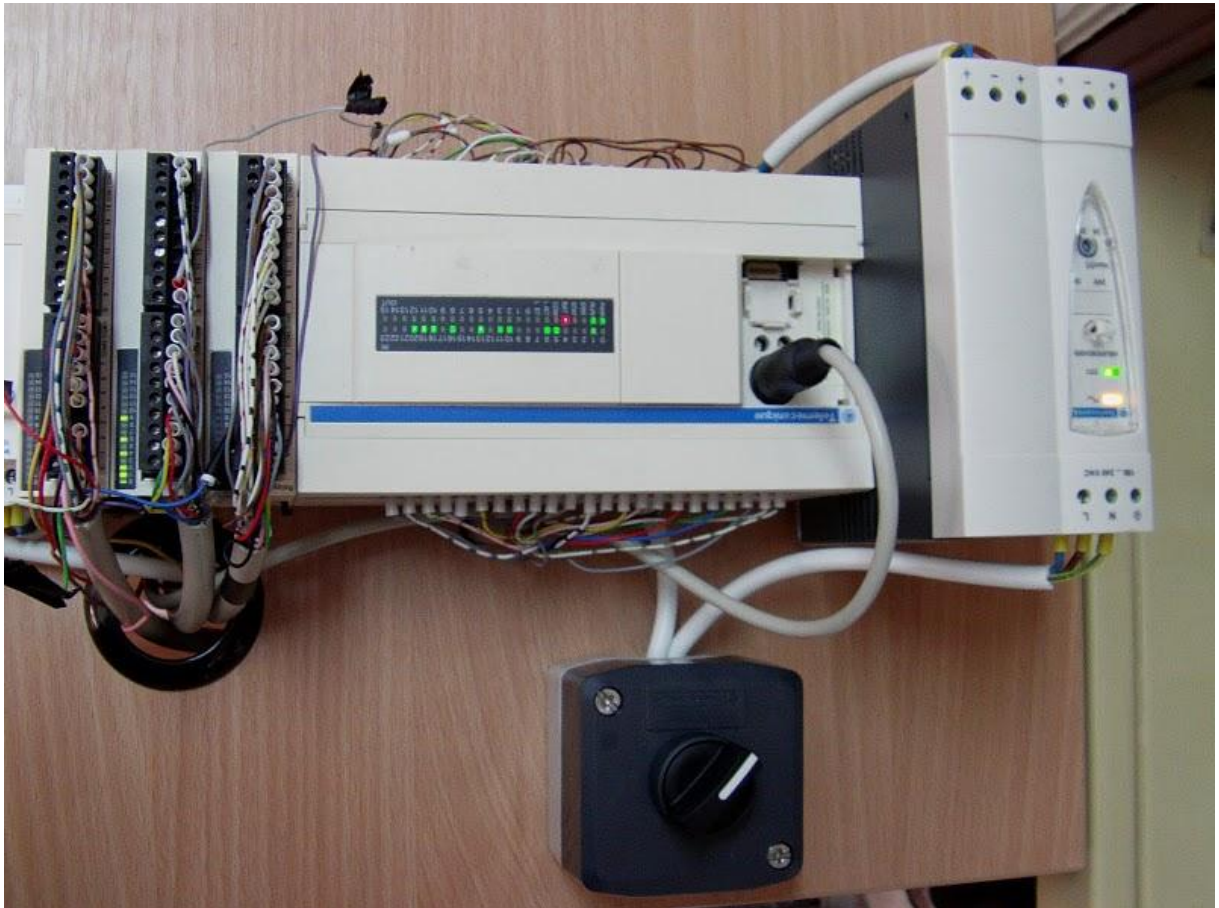
A robotkar minden olyan mozgását, amely nem végérzékelőig történik a számlálók segítségével valósítunk meg. Ezek a számlálók a programban egy adott számig elszámolnak és ez alapján áll a kar a megfelelő pozícióba. Első lépésben a robotkar leereszkedik egy adott mélységig majd kitolja a kart és a megfogó segítségével megfogja majd felemeli a munkadarabot és a hosszú futószalag elejére helyezi. Ezután felemelkedik, de csak egy adott magasságig és balra fordul addig amíg el nem éri a felsőrészt szállító futószalagot, ahol leereszti a kart majd megfogja a munkadarab felső részét és felemeli, majd már a szalag végénél lévő alsó részre helyezi. Ez a feladat nem volt egyszerű mivel a robotkaron található számlálók és a mozgató végző műanyag elemek eléggé pontatlanok. Miután ezeket a műveletek végrehajtották a kar teljesen felemelkedik és behúzza magát, erre azért volt

szükség, mert a szalagon található felépítmények elég közel vannak. A hosszú futószalagon megtörténnek a különböző műveletek, ha a termék selejt akkor az a selejttárolóba kerül időközben a robotkar felveszi a kiindulási pozíciót, majd késleltetve újraindul a gyártási folyamat. Ha a termék megfelelő, akkor a szalag végén megáll, majd a kar odafordul majdnem teljesen kitolja a felső részét és leereszti azt, majd megfogja a munkadarabot. Itt is szükség volt a pontos pozicionálásra, mert kevés hely áll rendelkezésre. Ezután felemeli majd a hegesztő előtt található forgó tálcáig forog jobbra, majd elengedi a munkadarabot. Ezután ismét felveszi a kiindulási helyzetet oly módon hogy először a teljesen behúzza a végállásig a kart, majd az összes többi mozgást egyszerre elvégzi szintén addig amíg a végérzékelők nem kerülnek bekapcsolt állapotba. Miután a hegesztés megtörtén előről kezdődik az egész folyamat.

Az alábbi táblázatban szereplő számlálók a robotkar mozgását vezérlik időrendi sorrendben:

Számláló száma	Folyamatban lévő helye	Számláló értéke
%C0	Első lefele: az alsó részt szállító szalag irányába	82
%C1	Első kifele: a szalagon lévő munkadarab irányába	30
%C2	Első megfogás: a munkadarab megfogása	21
%C3	Első felfele: munkadarab felemelése	50
%C4	Első jobbra: hosszú szalag felé való fordulás	54
%C5	Második kifele: a szalag végének elérése	30
%C6	Második lefele: a munkadarab lehelyezése	45
%C25	Második fel: a kar felemelése	40
%C7	Első balra: felsőrészt szállító szalag felé való pozicionálás	28
%C8	Harmadik lefele: a munkadarab irányába	45
%C9	Második megfogás: a felső rész megfogása	21
%C10	Harmadik felfele: a munkadarab felemelése	52
%C11	Második jobbra: a hosszú futószalag felé fordulás	28
%C12	Harmadik kifele: az alsó rész felé való pozicionálás	63
%C13	Negyedik lefele: munkadarab elhelyezése	21
%C14	Harmadik jobbra: a szalag végéhez való fordulás	55
%C15	Negyedik kifele: a munkadarab felé pozicionálás	73
%C16	Ötödik lefele: a kar leeresztése a munkadarabhoz	70
%C17	Harmadik megfog: munkadarab megfogása	21
%C18	Hatodik fel: majd annak felemelése	20
%C19	Negyedik jobbra: a forgó tálca felé való fordulás	46
%C20	Harmadik be: a tálca közepére való pozicionálás	20
%C21	Hatodik le: a munkadarab forgótálcára való elhelyezése	12

Mivel minden motort mind a két irányba lehet mozgatni ezért ügyelni kellett arra, hogy nehogyszerre akarjuk mind a két irányba mozgatni. Továbbá mivel ezek a motorok sokszor vannak használatban, így a program elején el kellett helyezni az minden egyes motort. Erre a PLC működési elve miatt volt szükség. Ezeket a továbbiakban markelekkel vezéreltünk. Ez egy olyan programozási eszköz, amelynek segítségével a programon belül akárhol elérhetjük ezeket a motorokat.



6.1 ábra A PLC teljesen bekábelezve

7. Fejlesztési lehetőségek

Már menetközben is jöttek fejlesztési ötletek, de ezek közül csak a selejtkezelést sikerült megvalósítani, a többi fejlesztés sajnos elmaradt idő vagy eszköz hiány miatt.

Az egyik ilyen például a pontosabb pozicionálás lett volna a robotkaroknál. Ezt például a olyan számlálók segítségével lehetett volna megoldani amelyen nem négyet számolnak egy tengelyfordulatonként, hanem jóval többet. Így nem adódtak volna olyan problémák, hogy ami még előzőnap megfelelő helyre állította a kart másnap már jóval messzebb vagy közelebb.

Másik ilyen lehetőség a modelleken lévő műanyag kopóalkatrészek és fogaskerekek fém alkatrészre való cseréje.

Az időhiány miatt elmaradt fejlesztés például a gyártási folyamat felgyorsítása. Az eredeti változat egyszerre csak egy terméket gyárt, amíg ez a termék végig megy a különböző munkafolyamatokon és a raktárba kerül, addig a többi része a gyártósornak nem csinál semmit. Ezt úgy kellett volna megoldani, hogy ne teljen el túl sok idő két késztermék között.

A programban néhány helyen időzítéssel oldottuk meg a felmerülő problémákat. Ez nem a legjobb megoldás, ha ezeket sikerült volna valahogy kiküszöbölni, akkor az egész gyártást eseményvezérelté tehetjük volna.

8. Összegzés

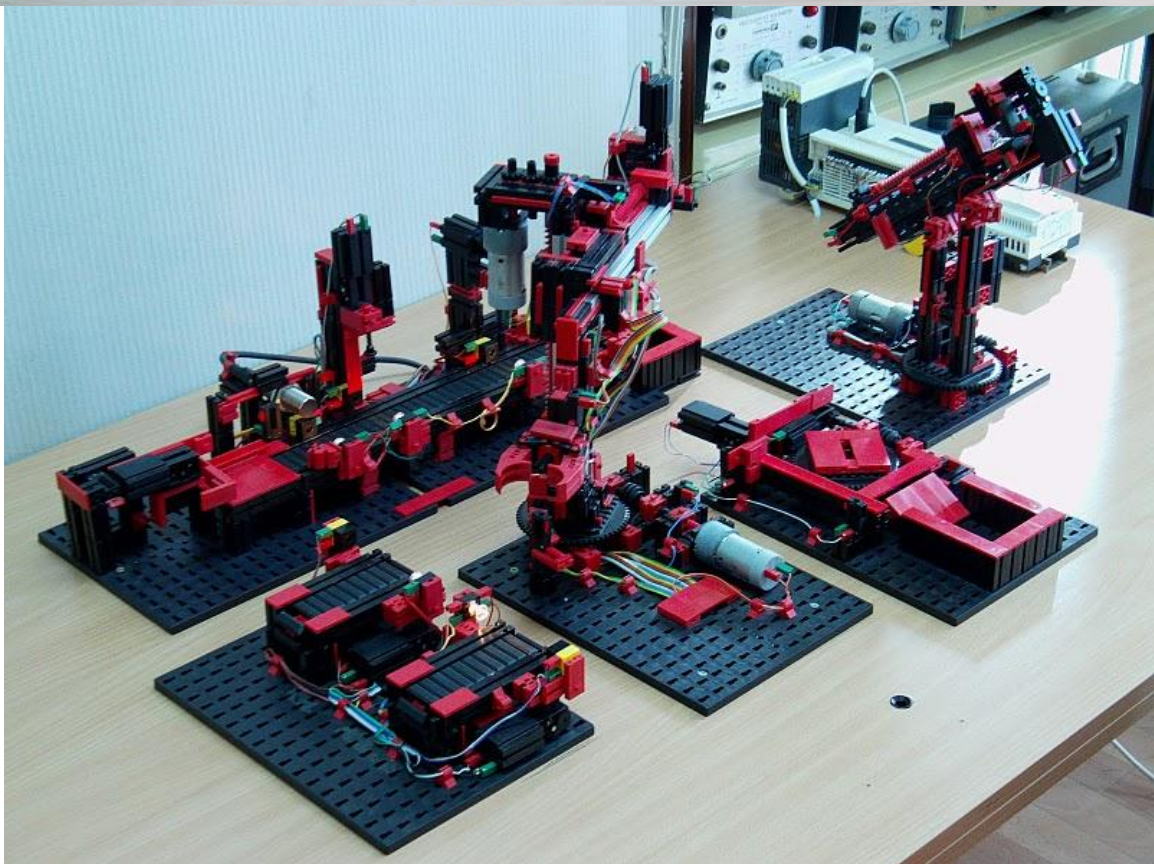
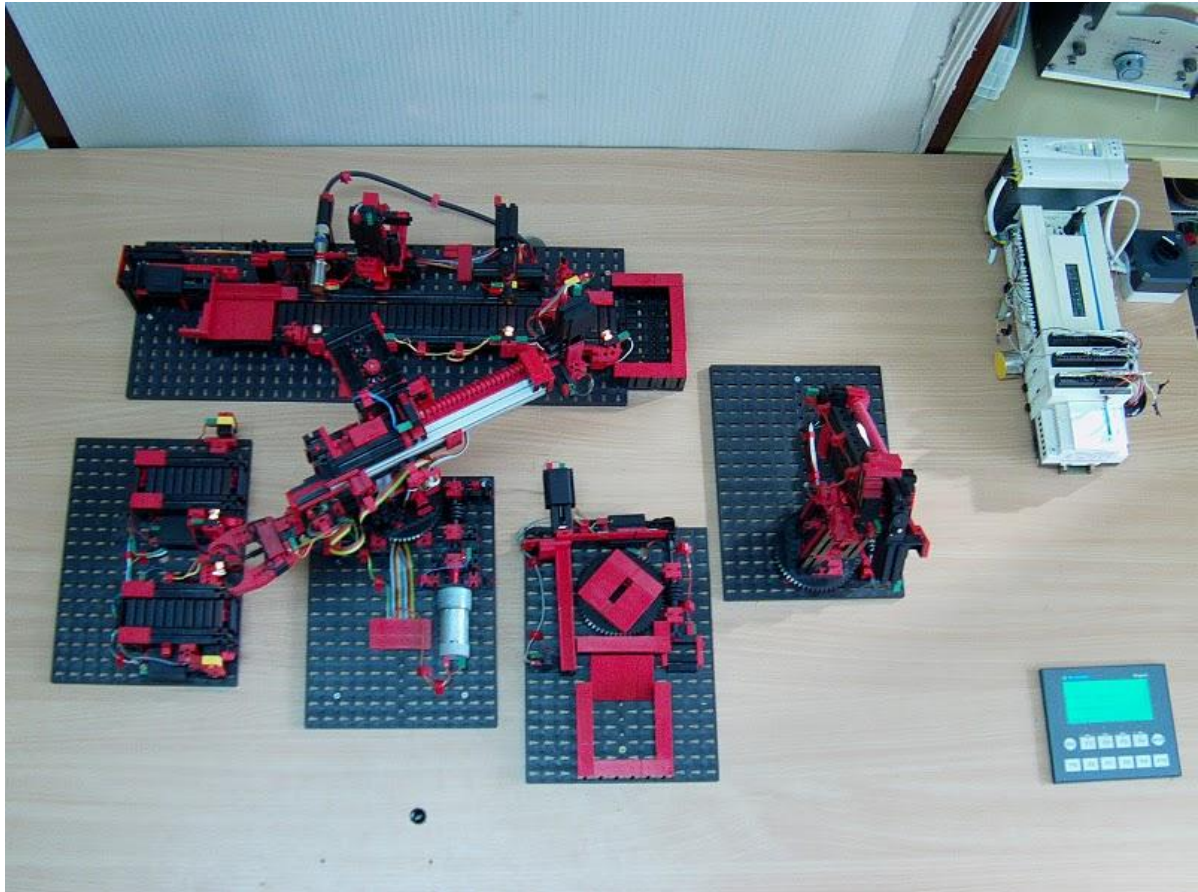
Célunk egy olyan gyártósor elkészítése volt, amely két részből összeállít egy terméket. Tervünk megvalósítása előtt tanulmányaink során betekintést nyerhettünk a gyártás elméleti részébe. Miután ezeket áttanulmányoztuk meg kellett ismerkednünk a gyártás fizikai részével is, és csak ezután kezdhettünk neki modellünk megépítésének.

A gyártósor megépítése és programozása közben nagyon sok apróbb hibába és nem várt problémába ütköztünk, de ezeket sikerült kijavítani. Így modellünk megfelelően el tudja látni feladatát.

Mivel ez a szakdolgozat csapatmunka volt, így nagyon hasznos és fontos tapasztalatokat szereztünk az együttműködés területén. Mindannyian külön feladatokat láttunk el, de ha valamelyikünk elakadt közösen oldottuk meg a problémát.

Ebben a munkában megszerzett ismereteinket máshol is fel tudnánk használni, mivel az itt használt vezérlési eszközökkel és programozási algoritmusokkal ugyanígy működtetnek egy valódi automatizált gyártósort.

Az elkészített modellt kiállítottuk a 2010 Szakmai napokon.



8.1- 2 ábra A kész gyártósor

Köszönetnyilvánítás

Ezúton szeretnénk Dr. Husi Gézának megköszönni, amiért elvállalta a témavezetői szerepet, és az ezzel járó fáradalmakat. Valamint szeretnénk megköszönni, hogy részt vehettünk a szakmai napokon, és hogy sok megoldásnál szabad kezet kaptunk.

Köszönettel tartozunk Bartha István és Dr. Tóth János tanár uraknak is. Az ő kiváló szakmai felkészültségük és háttérük nélkül mindez nem jöhetett volna létre.

Szeretnénk még megköszönni Szabó Tibor csoporttársunknak a segítséget, aki a termék acélból esztergált felső részét biztosította.

Irodalomjegyzék

- [1] Automatizálás és vezérlés TWIDO Programozható vezérlő Katalógus 2008 (6, 7, 9, 45, 47, 120 oldalak)
- [2] Twido Szoftver referencia kézikönyv (18, 19, 20, 140, 141, 142, 143, 147, 148, 149 oldalak)
- [3] www.schneider-electric.hu oldaláról letölthető könyvek
- [4] Mechanics of Microelectromechanical Systems - N.Lobontiu and E.Garcia 2005 Springer Science + Business Media, Inc.
- [5] Robot Manipulator Control Theory and Practice - Frank L.Lewis, Marcel Dekker, Inc., 2004
- [6] Dr. habil Kulcsár Béla, LSI Oktatóközpont, Robottechnika, 1998
- [7] Dr Sieglér András: Robot irányítási modellek, LSI, 1987, Bp

Linkek:

- [8] http://www.ifm.com/ifmhu/web/pinfo1_10_10_40.htm 2010.11.03.
- [9] http://ob121.com/plc_basic.html 2010.10.31.
- [10] <http://e-oktat.pmmf.hu/robotir2> 2010.11.10.
- [11] <http://www.fischertechnik.de/de/> (cégtörténet) 2010.11.02.
- [12] <http://www.plcprogramozas.hu/a-plc-tortenete> 2010.11.05.
- [13] <http://users.atw.hu/plcweb/plc/plc.htm> 2010.11.06.