

Debreceni Egyetem

Informatikai Kar

Gyártósor programozása programozható logikai vezérlővel

Témavezető:

Dr. Husi Géza

tanszékvezető főiskolai docens

Szerző:

Hajdu Zoltán

mérnök informatikus hallgató

Debrecen 2010.

Plágium – Nyilatkozat

Szakedolgozat készítésére vonatkozó szabályok betartásáról nyilatkozat

Alulírott Hajdu Zoltán (Neptunkód: CYT3JS) nyilatkozat aláírásával kijelentem, hogy a

Gyártósor programozása és automatizálása programozható logikai vezérlővel

című szakedolgozat/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.

Debrecen, 2010

Hajdu Zoltán

Tartalomjegyzék

| | |
|---|----|
| Bevezetés | 4 |
| 2.1 Fischertechnik | 6 |
| 2.1.1 Két kis futószalag | 7 |
| 2.1.2 Hosszú futószalag | 7 |
| 2.1.3 RTT Robotkar | 8 |
| 2.1.4 Hegesztő tálca | 9 |
| 2.1.5 Hegesztő | 10 |
| 2.1.6 Tároló rekeszek | 10 |
| 2.2 Induktív érzékelő | 11 |
| 2.3 Programozható logikai vezérlő | 11 |
| 2.4 A Tápegység | 15 |
| 2.5 A programozói felület | 17 |
| 3. A Gyártósor | 22 |
| 3.1 A gyártási folyamat | 22 |
| 3.2 Elektronika és bekötés | 23 |
| 3.2.1 Robotkar bekötése | 23 |
| 3.2.2 A hegesztő tálca és a hegesztő bekötése | 26 |
| 4. Programozás | 28 |
| 4.1 Alaphelyzet algoritmus, nullpont felvétel | 28 |
| 4.2 A hegesztő tálca és a hegesztő kar algoritmus | 29 |
| 4.3 A vezérlő egység programozása | 31 |

Bevezetés

Vajon meddig robotog még az informatika megállíthatatlan mozdonya? Tesszük fel a kérdést nap, mint nap felgyorsult világunkban. Napjainkban az informatika térnyerése mindennapi életünkben és az iparágak széles skáláján megállíthatatlan. Ez a végeláthatatlan állandó exponenciális fejlődési ütem ösztönzött arra, hogy az informatika területén folytassam tanulmányaimat.

Napjainkban a növekvő igény a magas színvonalú és magas számban előállított termékekre, a gyárakban informatikai technológia bevonását igényli. Ugyanakkor mivel egy gyártósort nem lehet egyedül megépíteni és felprogramozni fontos volt még számomra a csapat munka. A csapatban dolgozás a majdani munkahelyen való beilleszkedést is segíti. Mivel egy gyártósor működése különböző részfeladatokra épül így könnyű volt megosztani a munkát a csapaton belül.

Témaválasztásomat ez a három fő tényező nagyban befolyásolta, az informatika gyors ütemű fejlődése, a gyártási folyamatok informatikai és robotikai támogatása a termelési színvonal növelésének érdekében, és a csapatmunka fejlesztése. Ezen kívül szerettem volna a Vállalati információs rendszerek szakirány által biztosított eszközökkel foglalkozni, mindig is érdekelték az automatizált robotok, és szerettem volna valami maradandót illetve kézzel foghatót alkotni.

Önálló labor 1. keretein belül már foglalkoztam különböző automatizált folyamatok szimulációjával, ekkor már tudtam, hogy szakdolgozatom témájául is valami hasonló feladatot fogok végezni, csak gyakorlatban. A gyakorlatban elvégzett munka több időráfordítást igényel ugyan, de a jövőben megtérül. Nagy hatással volt témaválasztásomra az is, hogy Bartha István tanár úr említette, hogy a Biogal is programozható logikai vezérlőket használ az automatizáláshoz.

A szakdolgozatom célja egy automatizált gyártósor szimulálása, ami tulajdonképpen egy termék előállítását teszi lehetővé emberi beavatkozás nélkül. Az elkészült gyártósor nagy sikernek örvendett a szakmai napokon. Ezen a rendezvényen hasznos tanácsokat kaptunk melyeket beépítve optimalizálni tudtuk a gyártósort.

Gyártósor programozása programozható logikai vezérlővel

A gyártósor a Fischertechnik modelljeiből épül fel, amely tartalmaz három futószalagot egy robotkart egy présgépet, egy fúrót egy hegesztő gépet egy általunk kitalált hegesztő tálcát, és két tároló rekeszt. Valamint tartalmazza még a programozható logikai vezérlőt (PLC-t), annak bővítő moduljait, és egy monitoros vezérlő egységet.

Az automatizálást a Schneider Electric cég által készített PLC-vel oldottuk meg, amelyet a Twido Suite nevű programban programoztunk.

Dolgozatomban ki fogom fejteni a különböző egységek részletes működését, bekötését, a PLC működését, az általam írt program egységek algoritmikus elgondolását és részletes működését a gyakorlatban.

Csapatmunkában a társaim: Papp Gábor FTU3GY, Sóvágó Imre SMJT9H

2. Alkotórészek

2.1 Fischertechnik

A projektünkhöz az egyetem a Fischertechnik által gyártott eszközöket biztosította. A céget Arthur Fischer alapította 1966-ban, Németországban. A kezdetekben gyártott termékek hamarosan akkora sikert arattak, hogy a cég termékeit 1970-től Amerikában is elkezdték forgalmazni. A termékekbe az 1970-es évek végén építették be a Bináris logika kezelését. Ennek eredményeként a termékek tudtak kezelni AND, OR, NAND, NOR műveleteket.

A gyártósorhoz használt elemeket 24 voltos motorral mozgatjuk, kompakt PLC-vel és annak két bővítő egységével vezéreljük.

A gyártósor az alábbi Fischertechnikből épített elemeket tartalmazza:

2.1.1 Két kis futószalag

2.1.2 Hosszú futószalag

2.1.3 RTT Robotkar

2.1.4 Hegesztő tálca

2.1.5 Hegesztő

2.1.6 Tároló rekeszek

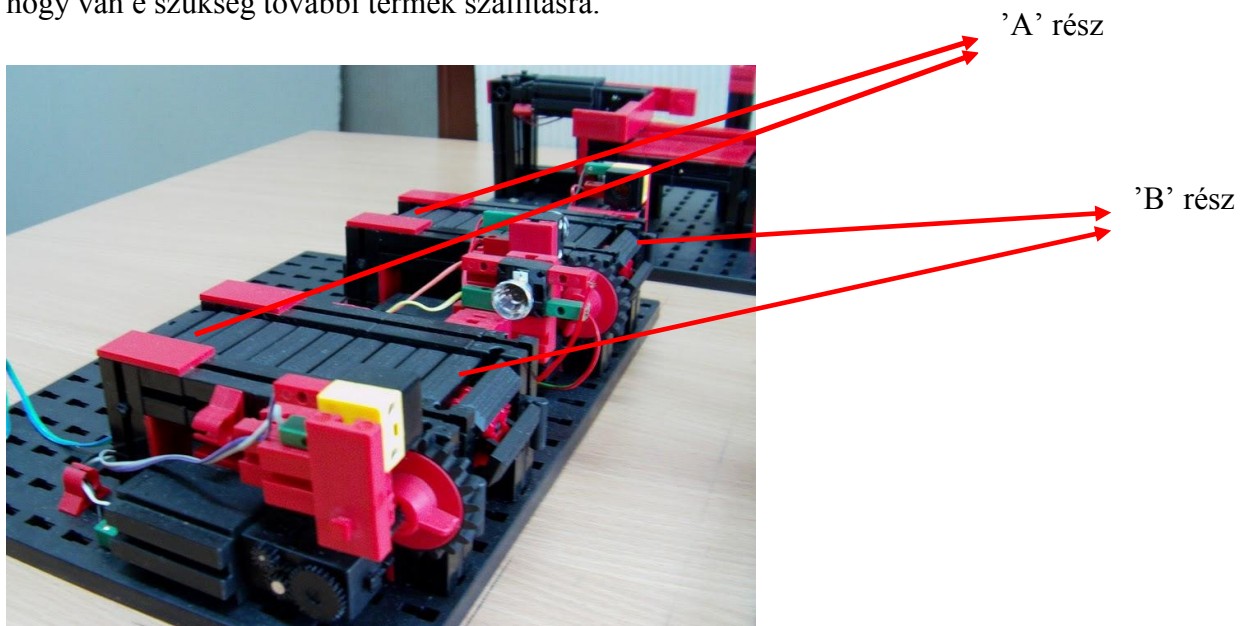
Valamint tartalmaz más gyártó által biztosított elemeket is:

2.2 Induktív érzékelő

2.1.1 Két kis futószalag

Az 1. ábrán látható a két futószalag. A futószalagok funkciójukat tekintve a nyersanyagok folyamatos áramlását biztosítják. Meghajtásuk 24 voltos motorral történik.

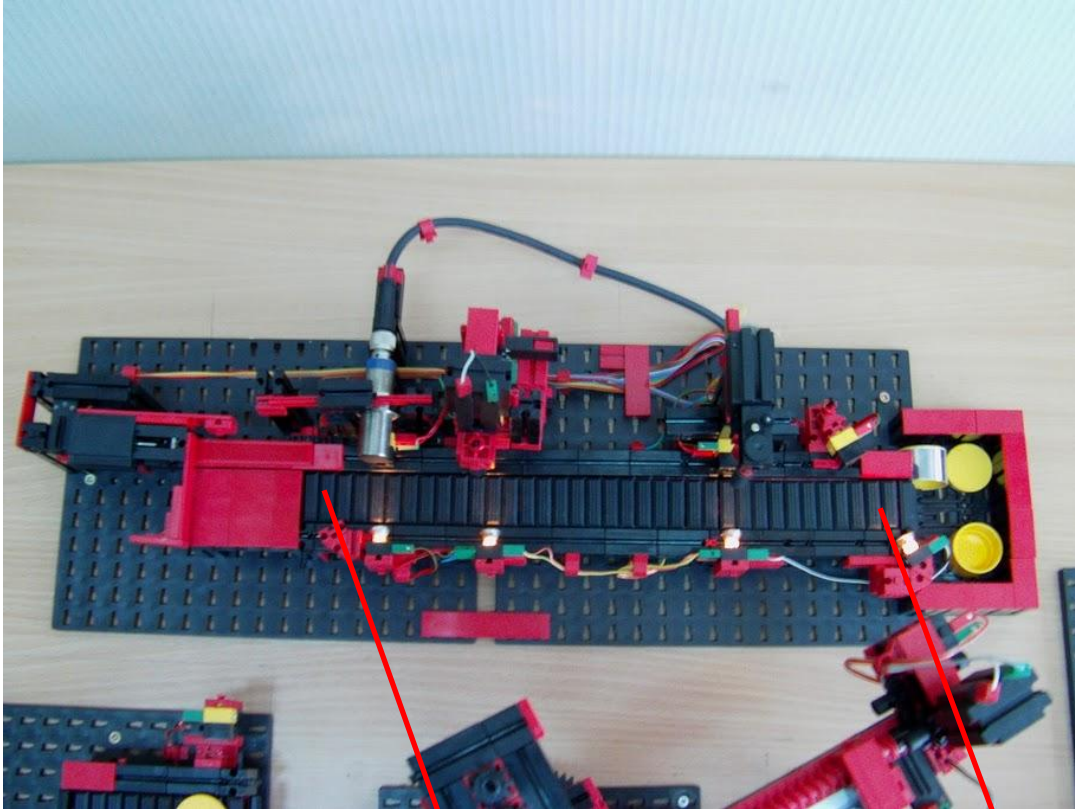
Az egyes számú futószalag a termék testét, a kettes számú futószalag pedig a termék tetejét szállítja a futó szalag 'A' részéről a 'B' részére. 'B' résznél fénykapu segítségével dönti el, hogy van-e szükség további termék szállításra.



1. ábra Két kis futószalag

2.1.2 Hosszú futószalag

Ezen a futószalagon végzi el a gyártósor a terméken a legtöbb műveletet. Az első elvégzett művelet egy selejtérzékelés, a fémérzékelő segítségével. Továbbá, mire a termék eljut 'A'-ból 'B'-be, végrehajtódik rajta a préselés és a fűrés. Minden művelet elvégzése fénykapuhoz van kötve. Ha a termék a prés alatti fénykapuhoz ér, eltakarja a fényforrást, ez azt eredményezi, hogy a futószalag megáll, amíg a préselés végbe megy. A fűrő művelet ugyanígy megy végbe. Miután a termék a szalag végén elhelyezkedő fénykapuhoz ért, a szalag megáll, és a robotkar a terméket leemeli és áthelyezi a hegesztő tálcára.



2. ábra Hosszú futószalag

'A' rész

'B' rész

2.1.3 RTT Robotkar

A gyártósor szíve a Rotation-Translation-Translation robotkar, más néven hengerkoordinátás robot modell. Praktikusan ez azt jelenti, hogy a kar képes a tengelye körül forogni, lefele illetve felfele irányuló mozgást végezni, és kitoló illetve behúzó mozgást végezni. A kiinduló nyerstermék áthelyezését biztosítja a különböző műveletek, illetve futószalagok között. A gyártás meggyorsítása érdekében a robotkar munkáját ahol lehetett futószalagokkal váltottuk fel.

A robotkar négy fajta mozgást tud végezni. Ezeket a mozgásokat négy különböző 24 Voltos motor hajtja végre. Tud előre-hátra, fel-le egyenes vonalú mozgást végezni. Jobbra-balra

forgást, és a kar egy megfogó mozdulatot képes végezni. A kétfás megfogó pozícionálását számlálók és végállás kapcsolók segítségével végezzük.

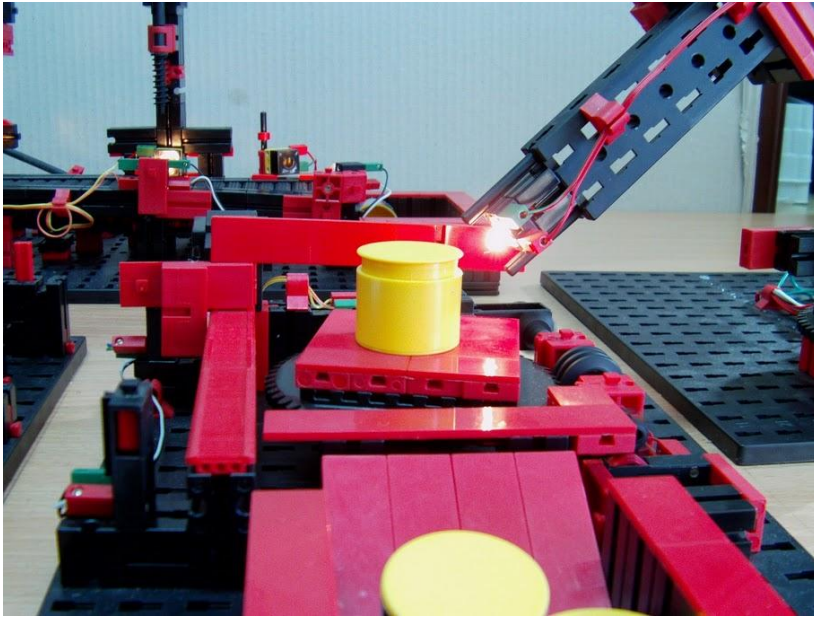


3. ábra RTT Robotkar

2.1.4 Hegesztő tálca

Általunk kitalált eszköz, amely a termék forgását teszi lehetővé a hegesztő robot alatt, ezzel elősegítve a folyamat meggyorsítását és pontosítását. Amint a hegesztés elkészült, a késztermék mozgó kar a terméket a tárolóba tolja.

Két motort tartalmaz az eszköz, az egyik a forgást teszi lehetővé, a másik pedig a kart mozgatja. A tálcára történő elhelyezés pontos pozícionálása még nem megoldott.



4. ábra Hegesztő tálca

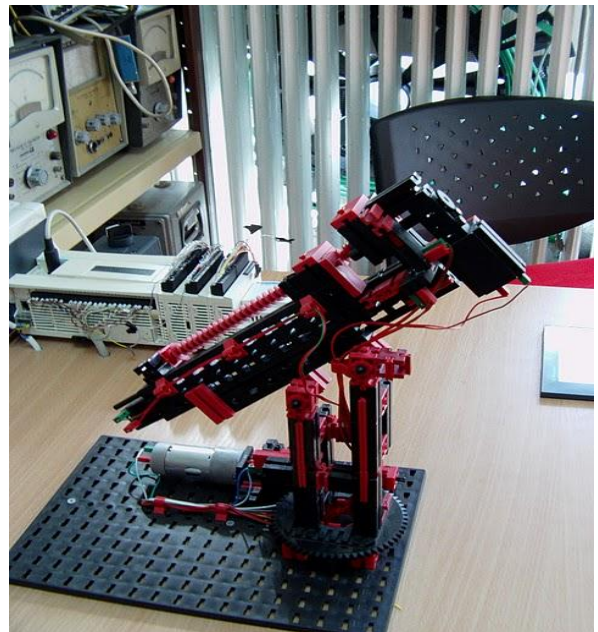
2.1.5 Hegesztő

A hegesztő robot a termék két részét hegeszti egybe, a felső részt és az alsó részt. A művelet elvégzéséhez a robot alapállapotából a termék felé fordul. Ha a robot végzett, visszafordul.

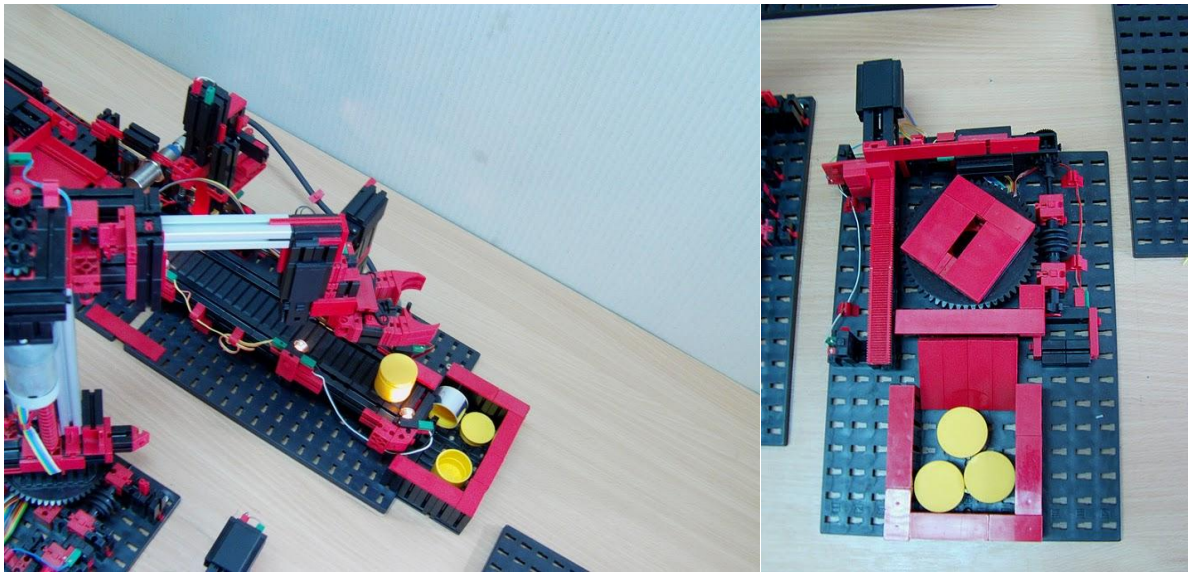
2.1.6 Tároló rekeszek

A hosszú futószalag végén található a 5. ábra Hegesztő kar

selejttermék tároló, közvetlenül a hegesztő tálca mellett pedig a késztermék tároló rekesz, azaz a „raktár”. Az előbbibe kerülnek a felső rész nélküli, illetve sérült termékek. Az utóbbi tartalmazza a kész, hibátlan terméket.



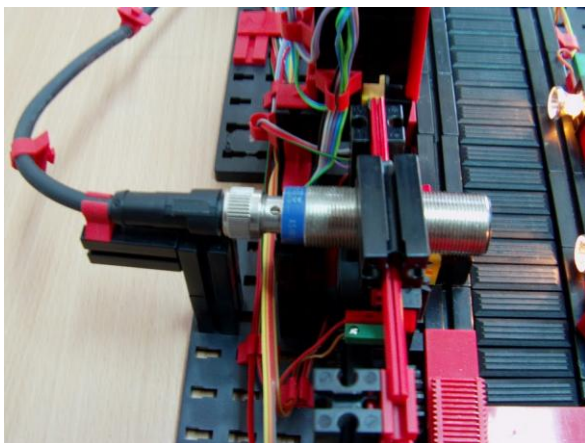
5. ábra Hegesztő kar



6. ábra Tároló rekeszek

2.2 Induktív érzékelő

A fémerzékelőt a selejt termék felismeréséhez használjuk. A termék felső része fémből készült, abból a célból, hogy tudjunk beépíteni selejtkezelést. Ha az érzékelő nem érzékel fémet, a terméket selejtnek minősíti és az a selejt raktárba kerül.



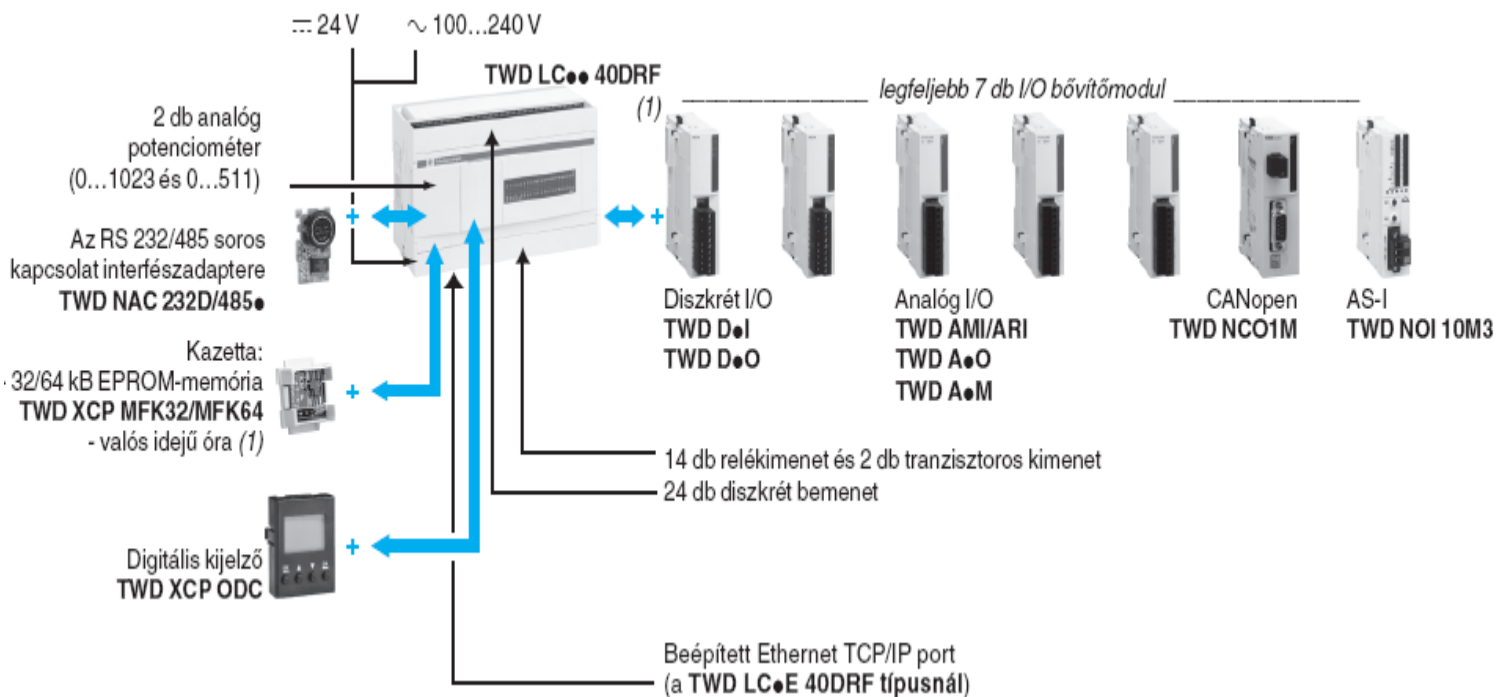
7. ábra Induktív érzékelő

2.3 Programozható logikai vezérlő

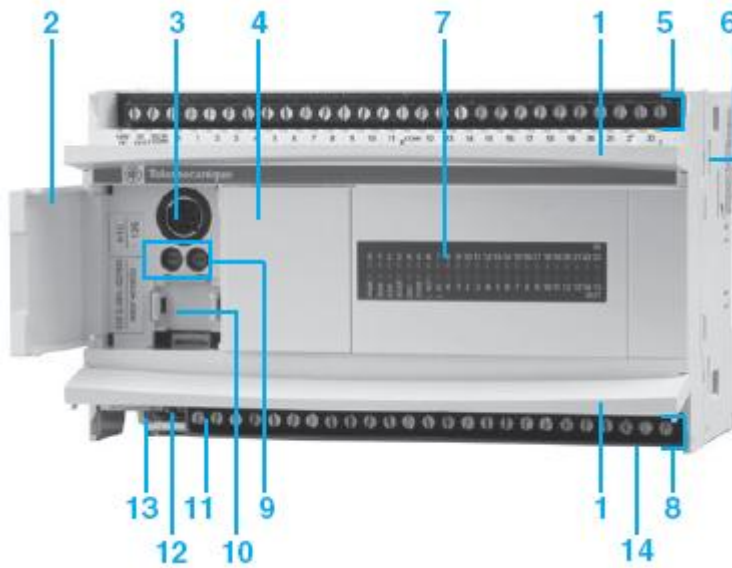
A gyártósor vezérlését és automatizálását programozható logikai vezérlővel oldottuk meg. Szükségünk volt két relés bővítő egységre, hogy a robotok két irányba tudjanak mozogni, és egy analóg input bővítő modulra.

2.3.1 A PLC jellemzői

„ Típusa TWDLC 40 DRF. Azért választottuk ezt a PLC-t mert kiválóan lehet bővíteni, és a laborban volt hozzá felszerelés.



8. ábra TWDLC 40 DRF PLC[3.]



9. ábra PLC részletes jellemzés

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 kijelzőmodul részére.
5. Egy csavaros sorkapcsot, az érzékelők c 24 V (I) tápfeszültséggel való ellátásához és a bemeneti érzékelők csatlakoztatásához.
6. Egy csatlakozót, a **TWD Dpp**, **TWD App** 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,ERRBAT, COM,LACT és L ST),
- a bemenetek és a kimenetek állapotát (INp és OUTp),
- 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** ppp típusú adapter segítségével,
11. Egy csavaros sorkapcsot, a a 100...240 V feszültségű hálózat vagy a c 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

„[3.]

2.4 A Tápegység

„120 Watt teljesítmény, és 5 Amper leadására képes.

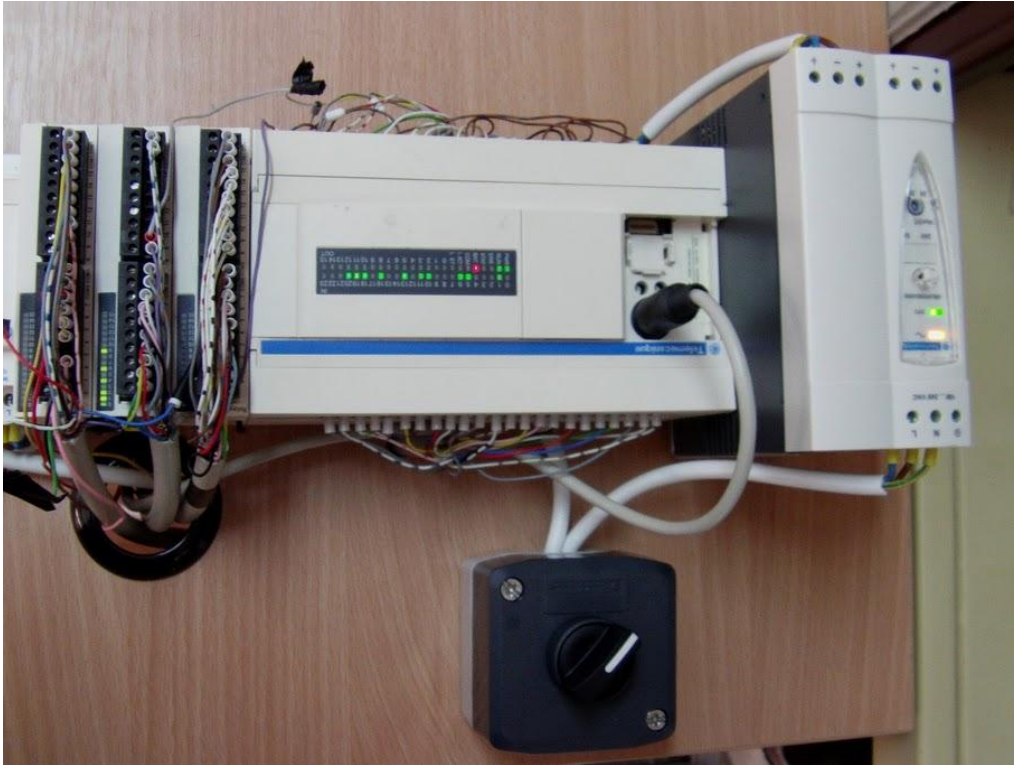


1. 2,5 mm² -es zárt, csavaros kivezetések a bemeneti feszültség
2. csatlakoztatásához (egyfázisú N-L1, vonalfeszültségű L1-L2)
3. Üvegből készült, védő zárófedél
4. A bemeneti feszültség állapotát jelző LED (narancssárga).
5. A kimeneti DC-feszültség állapotát jelző LED (zöld).
6. Az üvegből készült zárófedél retesze (tömíthető)
7. Rácsíptethető jelölécímke.
8. Kimeneti feszültséget beállító potenciométer
9. 2,5 mm² -es, zárt, csavaros sorkapocs a kimeneti DC-feszültség csatlakoztatásához”

„[3.]

10. ábra Tápegység [3.]

A valóságban:



11. ábra PLC a modulokkal, a főkapcsolóval és a tápegységgel

Az induktív érzékelő

„Az induktív érzékelők azt a fizikai jelenséget használják ki, hogy a vezetőképes anyagban fellépő örvényáramok általi veszteségek megváltoztatják a rezonancia-áramkör jósági tényezőjét. Az LC-rezgőkör nagyfrekvenciás váltakozó mezőt hoz létre. Ez a mező az érzékelő aktív felületén lép ki. Ha elektromosan vezetőképes anyag kerül a mezőbe, akkor az indukció törvénye alapján örvényáramok keletkeznek, amelyek energiát vonnak el a rezgőkörből. Ezáltal csökken a rezgés amplitúdója. Ez a változás kapcsolási jellé alakul. A működési elvből következik, hogy bármely fémét érzékel – függetlenül attól, hogy azok mozognak-e vagy sem.

Speciális típusorozatot képeznek az „efector m” alkalmazásérzékelők. Az „m” a moduláris technika jele. Ennek alapját egy új mechanikai koncepció, új szerelési technika és innovatív tömítésrendszer jelenti. Ezen érzékelők közös jellemzője a megnövelt kapcsolási távolság. Az univerzális csatlakoztatási technikának köszönhetően a készülékek választhatóan két vagy háromvezetékesként alkalmazhatók. A beépített LED-es beállítókijelző lerövidíti a szerelési időt és egyidejűleg garantálja a megnövelt kapcsolási távolság kihasználását. Minden készülék lézerrel beégetett típusablával van ellátva, amely így nem veszt el. Ezáltal a készülékek évek múlva is egyértelműen azonosíthatók.”

[8.]

A selejtkezelés lett megoldva ezzel az érzékelővel. A termék felső részét egy barátunk testvérével készítettük el, aki egy acél csőből esztergálta ki nekünk a részeket. Az induktív érzékelő a nagy futószalagon helyezkedik el, ha a terméken nem érzékel fémet, akkor a termék végig halad a futószalagon a selejt raktárba, majd a gyártási folyamat elindul előlről.

2.5 A programozói felület

„A Schneider-Electric által gyártott programozható logikai vezérlők programozása a TwidoSuit nevű szoftverrel történik. Ez egy grafikus fejlesztői környezet, amely az eszközök programozását, karbantartását és vezérlését teszi lehetővé. Ezen a programon keresztül tudunk kapcsolatot létrehozni a PC és a programozható logikai vezérlő között.

A TwidoSoft egy grafikus fejlesztői környezet a Twido programozható vezérlők alkalmazásainak elkészítéséhez, konfigurálásához és karbantartásához. A TwidoSoft teszi lehetővé a vezérlő programok beírását a TwidoSoft létra vagy lista szerkesztőinek segítségével, valamint a program átvitelét a vezérlőre és futtatását is biztosítja.

A TwidoSoft egy 32 bites Windows-alapú program olyan személyi számítógépek (PC) számára, melyeken Microsoft Windows 98 Second Edition vagy Microsoft Windows 2000 Professional operációs rendszer van.”

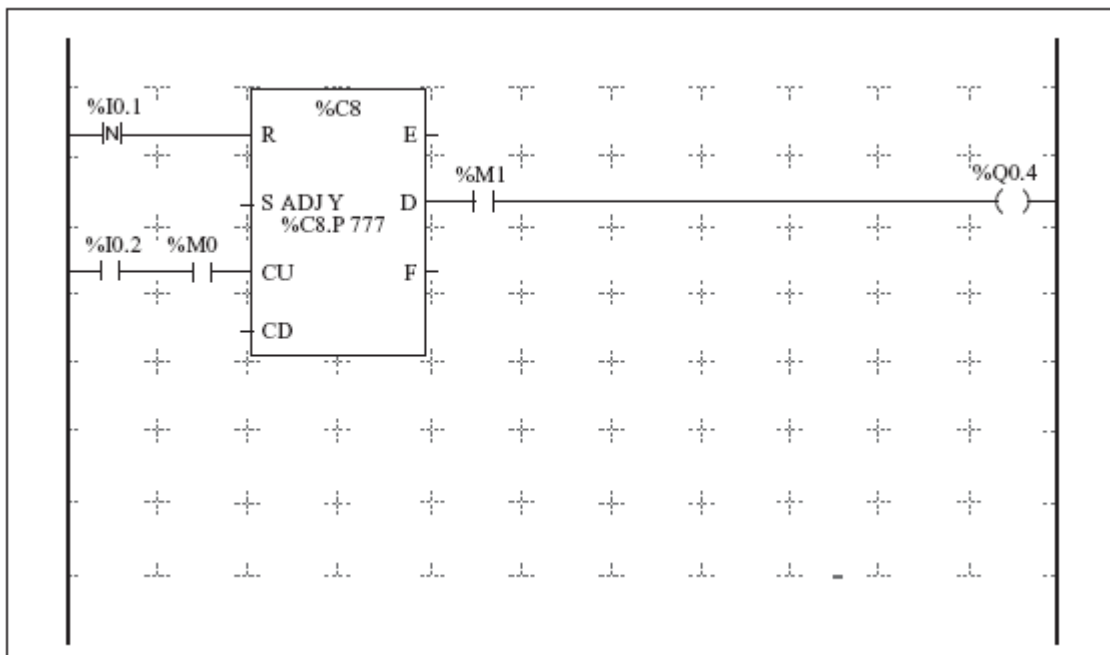
A TwidoSoft fő jellemzői:

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

Létra nyelv

„A Létra diagramok hasonlóak a relé logikai diagramokhoz, amelyek relévezérlő áramkörök reprezentálásához használatosak. Az olyan grafikai elemek, mint a tekercsek, kontaktusok és blokkok, mind utasításokat jelenítenek meg.





A következőkben egy példát láthatunk a Létra programokra.



12. ábra Létra nyelv [1.]






Létra nyelv grafikai elemei:

Gyártósor programozása programozható logikai vezérlővel

| 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. |

13. ábra Grafikai elemek [1.]

Tekercsek:

| 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 | A kapcsolt bitobjektum 1-re van beállítva, amikor a tesztzóna eredménye 1. |
| RESET tekercs |  | R | A kapcsolt bitobjektum 0-ra van beállítva, amikor a tesztzóna eredménye 1. |
| Ugrás és szubrutinhívás | <code>->>%Li</code> <code>->>%Sri</code> | 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. |

14. ábra Tekercsek [1.]

Grafcet nyelv

A Grafcet egy analitikus metódus, amely minden szekvenciális vezérlőrendszert lépések sorára bont le, amelyekhez különböző műveletek, átvitelek és feltételek tartoznak. Az alábbi illusztráció példával szolgál a Grafcet utasításokra mind Lista, mind Létra programok esetében.

| | | |
|----|-----|-------|
| 0 | _*_ | 3 |
| 1 | LD | %M10 |
| 2 | # | 4 |
| 3 | # | 5 |
| 4 | _*_ | 4 |
| 5 | LD | %I0.7 |
| 6 | # | 6 |
| 7 | _*_ | 5 |
| 8 | LD | %M15 |
| 9 | # | 7 |
| 10 | ... | |

15. ábra Utasítás példák [1.]

A fő bitobjektumok leírása

| Típus | Leírás | Cím vagy érték | Maximumszám | Írás hozzáféérés ¹ |
|------------------------------|---|--|--|-------------------------------|
| Azonnali értékek | 0 vagy 1 (Hibás vagy Helyes) | 0 vagy 1 | - | - |
| Bemenetek kimenetek | Ezek a bitek az I/O elektromos állapotainak „logikai képei”. Az adatmemóriában kerülnek tárolásra és a logikai program minden egyes ciklusa során frissülnek. | %Ix.y.z ² %Qx.y.z ² | Megjegyzés ⁴ | Nem Igen |
| Belső (Memory) | A belső bitek belső memória területek, amelyek közbenső értékek tárolására szolgálnak, míg fut egy program. Megjegyzés: A használaton kívüli I/O biteket nem lehet belső bitekként használni. | %Mi | 128 TWDLCAA10DRF, TWDLCAA16DRF 256 az összes többi vezérlőnél | Igen |
| Rendszer | A rendszerbitek %S0 és %S127 között a vezérlő helyes működését és az alkalmazott program helyes futását ellenőrzik. | %Si | 128 | i-nek megfelelően |
| Funkció-blokkok | A funkcióblokk bitek a funkcióblokkok kimeneteinek felelnek meg. Ezek a kimenetek lehetnek közvetlenül csatlakoztatva, illetve objektumként is lehet őket használni. | %Tmi.Q, %Ci.P, és így tovább. | Megjegyzés ⁴ | Nem ³ |
| Reverzibilis funkció-blokkok | Programozott funkcióblokkok, amelyek reverzibilis programozási utasításokat BLK, OUT_BLK és END_BLK használnak. | E, D, F, Q, TH0, TH1 | Megjegyzés ⁴ | Nem |
| Szóki-vonatok | Egyes szavakból ki lehet vonni a 16 bit egyikét, mint operandus bitet. | Variánsok | Variánsok | Variánsok |
| Grafcet lépések | A bitek %X1 és %X62 között a Grafcet lépésekhez vannak társítva. Az Xi Lépésbit 1-re van állítva, mikor a társított lépés aktív és 0-ra, mikor a lépés deaktivált állapotban van. | %X21 | 62 TWDLCAA10DRF, TWDLCAA16DRF 94 TWDLCAA24DRF, Moduláris vezérlők | Igen |

16. ábra A fő bitobjektumok leírása

„[1.]

3. A Gyártósor

A feladatunk egy több részből álló termék egyesítése volt, azonban a gyártósor elkészítéséhez teljes mértékben szabad kezet kaptunk. A modellt először papíron terveztük meg, majd mérőszalaggal lemértük a pontos méreteket, hogy a robotkar milyen távolságokban tud mozogni, ezt követte a beépítés.

3.1 A gyártási folyamat

A termék két részből áll, egy alsó és egy felsőből. Ezek az objektumok két kis futószalagon helyezkednek el. Az alsó részt a robotkar a hosszú futó szalag előtti tálcára helyezi. Ezután a robotkar a másik futószalagról felveszi a felső részt és az alsó rész tetejére helyezi.

Az így összeillesztett terméket a betoló kar rátolja az időközben elindult hosszú futószalagra. A következő lépésben egy fémérzékelő leellenőrzi a termék felső részét, hogy meg van e.

Ha nincs, akkor a termék a selejt raktárba kerül és nem hajtódik végre rajta feleslegesen a préselés és a fúrás művelete. Majd a robotkar alaphelyzetbe



17. ábra A gyártósor

áll és a gyártási folyamat előlről indul.

Ha a felső rész a helyén van, akkor mire a futószalag végére ér végrehajtodik rajta a préselés és a fűrés művelete. Ahogy az összepréselt és kifűrt termék a szalag végére ér, a robotkar leemeli a futószalagról és áthelyezi a hegesztő tálcára.

A hegesztő kar amint megkezd a hegesztést, a tálca egy teljes kört fordul vízszintesen, így biztosítva az egyenletes rögzítést a termék felső és alsó része között. Időközben a robotkar visszatér alaphelyzetbe, ezzel is időt megtakarítva, hogy a gyártási folyamat kevesebb időt vegyen igénybe.

A hegesztés ha végbe ment, akkor a kész terméket a kitoló kar a tároló rekeszbe helyezi. Amint a kitoló kar alaphelyzetbe kerül, a gyártási folyamat kezdődik előről. Ciklikus hiba nélküli üzemidő rekordnak körülbelül 100 percet mértünk.

3.2 Elektronika és bekötés

Miután a gyártósort megterveztük meg kellett építenünk. Az első lépés a robotok, futószalagok, és különböző egységek elhelyezése volt. Miután a PLC-t és a bővítő modulokat is a helyükre raktuk következett a bekábelezés.

3.2.1 Robotkar bekötése

A robotkar négy motort és nyolc érzékelőt tartalmaz. Minden egyes motorhoz párosul egy végállás érzékelő és egy számláló. Tehát összesen szükségünk van huszonnégy kábelre a robotkar vezérléséhez. Ahhoz, hogy megtudjuk, hogy melyik kábel melyik motorhoz, vagy

érzékelőhöz kapcsolódik, a motorok kivezetéseit a kábelek végével, a multiméter segítségével „összecsengengettük”.

A tengely körüli forgást végző motor az 1. és a 2. vezetéken kapja a jelet. A kitolást illetve a behúzást végző motor a 7. és a 8. vezetéken kapja a jelet. A függőleges elmozdulás jele a 13. és 14. vezetéken érkezik a motorhoz, a 19.-es és a 20.-as vezetéken pedig a megfogó rész összezárásáért illetve elengedésért felelős. A vezetékek végeit pólusok segítségével kötöttük hozzá a PLC kimeneteihez.

Az érzékelőket hasonlóképpen kerestük meg a vezetékek között. Ez alapján a 3.-as és 4.-es a tengely körüli forgás végállás kapcsolója, az 5.-ös és a 6.-os a számlálója. 9.-es és 10.-es pólus a kitolás-betolás művelet végállás kapcsolója, a 11.-es és 12.-es ennek a számlálója. 15.-ös és 16.-os vezeték a függőleges mozgás végállás kapcsolója, a 17.-es és 18.-as pedig ennek a számlálója. A megfogó végállás kapcsolója a 21.-es és 22.-es kábelben kapja a jelet, míg a számlálója a 23.-as és 24.-es póluson csatlakozik a PLC-hez.

A végállás kapcsolók a robotkaron minden esetben a robotkar alaphelyzetbe állítását teszik lehetővé. A számlalók segítségével pedig a robotkar ebből a helyzetéből pozícionálható, és vezérelhető volt különböző precíziós műveletek elvégzésére.

A robotkaron elhelyezett érzékelők Symbol táblája

| Used | Address | Symbol |
|-------------------------------------|---------|-----------------------|
| <input checked="" type="checkbox"/> | %I0.2.0 | ROBOTKAR_LENTI_VÉGÁLL |
| <input checked="" type="checkbox"/> | %I0.2.1 | ROBOTKAR_LENTI_SZÁML |
| <input checked="" type="checkbox"/> | %I0.2.2 | ROBOTKAR_VIZSZINTES_ |
| <input checked="" type="checkbox"/> | %I0.2.3 | ROBOTKAR_VIZSZINTES |
| <input checked="" type="checkbox"/> | %I0.2.4 | ROBOTKAR_FENTI_VÉGÁLL |
| <input checked="" type="checkbox"/> | %I0.2.5 | ROBOTKAR_FENTI_SZÁML |
| <input checked="" type="checkbox"/> | %I0.2.6 | ROBOTKAR_MAROK_VÉG |
| <input checked="" type="checkbox"/> | %I0.2.7 | ROBOTKAR_MAROK_SZÁ |

18. ábra A robotkar Symbol táblája

Gyártósor programozása programozható logikai vezérlővel

A bekötés könnyen szemléltethető a TwidoSuite Symbol táblájával. A képen jól látható, hogy a robotkar érzékelői hogyan lettek bekötve.

Például:

- % I.0.2.0 azt jelöli, hogy az érzékelő a 2-es számú bővítő modul 0-as bemenetére lett kötve. A mögötte található megnevezés, az nekünk programozóknak szól, úgynevezett beszédes név, hogy könnyen visszakereshető legyen az érzékelő neve alapján, hogy hányas bemenetre kötöttük.
- % IO.2.7 a marok, azaz a megfogó számlálója. A 2-es PLC bővítő modul hetes kimenetére van kötve. A számláló tehát a jeleket a 7-es bemeneten küldi a PLC-hez.

A motorok bekötése már nehezebbnek bizonyult. Ahhoz, hogy a motorokat két irányba tudjuk mozgatni egyszer az egyik pólusra kell kötni a feszültséget, és a másikra a 0 voltot, ha ellenkező irányba szeretnénk forgatni, akkor meg pont fordítva. Ezt a feladatot a PLC-be épített relé fogja ellátni nekünk.

Ahhoz, hogy két irányba tudjon mozogni egy motor négy kábellel kell bekötni, mivel a lábakon változik a 0 volt és a 24 volt. Tehát egy motorhoz négy kábel tartozik.

A robotkart mozgó motorok Symbol táblája

| Used | Address | Symbol |
|-------------------------------------|----------|-------------------|
| <input checked="" type="checkbox"/> | %Q0.1.0 | ROBOTKAR_JOBBRA1 |
| <input checked="" type="checkbox"/> | %Q0.1.1 | ROBOTKAR_BALRA1 |
| <input checked="" type="checkbox"/> | %Q0.1.2 | ROBOTKAR_KI1 |
| <input checked="" type="checkbox"/> | %Q0.1.3 | ROBOTKAR_BE1 |
| <input checked="" type="checkbox"/> | %Q0.1.4 | ROBOTKAR_FEL1 |
| <input checked="" type="checkbox"/> | %Q0.1.5 | ROBOTKAR_LE1 |
| <input checked="" type="checkbox"/> | %Q0.1.6 | ROBOTKAR_MEGFOG1 |
| <input checked="" type="checkbox"/> | %Q0.1.7 | ROBOTKAR_ELENGED1 |
| <input checked="" type="checkbox"/> | %Q0.1.8 | ROBOTKAR_JOBBRA2 |
| <input checked="" type="checkbox"/> | %Q0.1.9 | ROBOTKAR_BALRA2 |
| <input checked="" type="checkbox"/> | %Q0.1.10 | ROBOTKAR_KI2 |
| <input checked="" type="checkbox"/> | %Q0.1.11 | ROBOTKAR_BE2 |
| <input checked="" type="checkbox"/> | %Q0.1.12 | ROBOTKAR_FEL2 |
| <input checked="" type="checkbox"/> | %Q0.1.13 | ROBOTKAR_LE2 |
| <input checked="" type="checkbox"/> | %Q0.1.14 | ROBOTKAR_MEGFOG2 |
| <input checked="" type="checkbox"/> | %Q0.1.15 | ROBOTKAR_ELENGED2 |

19. ábra A motorok Symbol táblája

A bekötés könnyen szemléltethető a TwidoSuite Symbol táblájával. A képen jól látható, hogy a robotkart vezérlő motorok hogyan lettek bekötve.

Például:

- %Q 0.1.0 Azt jelöli, hogy a robotkar tengelyének jobb irányú forgása az 1-es számú bővítő modul 0-ás kimenetére van kötve, de ugyanakkor rá van kötve a %Q 0.1.8-ra is. Erre azért van szükség mert a nullás kimenetről megkapja a 0 voltot, a 8-asról pedig a +24 voltot. Bal irányú forgásnál pedig nem kap jelet erről a két kimenetről.
- Ezzel megegyezően működik a többi mozgás is. Egy motorhoz négy darab Q kimenet négy darab vezeték.

3.2.2 A hegesztő tálca és a hegesztő bekötése

A hegesztő és a tálca működése szorosan kapcsolódik egymáshoz, ezért bekötésük és programozásuk nem választható szét egymástól. A hegesztő tálca, vagy más néven tálca két motort tartalmaz. Az egyik motor a tálca forgásáért felelős, a másik pedig a forgás befejeztével a termék raktárban való elhelyezését elvégző kar mozgásáért. A tálca csak egy irányban foroghat, és nincs végállás érzékelője, csak egy számlálóval van ellátva. A kitoló kar motorja viszont kétirányú mozgást végez végállástól végállásig.

A hegesztő robot alaphelyzetből balra fordul a termék irányába, majd kitolja a hegesztőt és hegesztést szimulál, egy a végén elhelyezett égővel. Miután végzett visszatér alaphelyzetbe.

Ezek alapján a bekötés a következőképpen alakult:

Ugyanúgy mint a robotkarnál a két irányú mozgást végző motorokat több helyre kellett bekötnünk.

A hegesztő hegesztő tálca és a hegesztő Symbol táblája:

| Used | Address | Symbol |
|-------------------------------------|---------|-----------------|
| <input checked="" type="checkbox"/> | %Q0.0 | HOSSZÚ_FUTÓSZ |
| <input checked="" type="checkbox"/> | %Q0.1 | FÚRÓGÉP |
| <input type="checkbox"/> | %Q0.2 | |
| <input type="checkbox"/> | %Q0.3 | |
| <input type="checkbox"/> | %Q0.4 | |
| <input checked="" type="checkbox"/> | %Q0.5 | FORGÓ_TÁLCA |
| <input checked="" type="checkbox"/> | %Q0.6 | HEGGESZTŐ_KI1 |
| <input checked="" type="checkbox"/> | %Q0.7 | HEGGESZTŐ_BE1 |
| <input checked="" type="checkbox"/> | %Q0.8 | HEGGESZTŐ_JOB |
| <input checked="" type="checkbox"/> | %Q0.9 | HEGGESZTŐ_BAL |
| <input checked="" type="checkbox"/> | %Q0.10 | HEGGESZTŐ_BE2 |
| <input checked="" type="checkbox"/> | %Q0.11 | HEGGESZTŐ_KI2 |
| <input checked="" type="checkbox"/> | %Q0.12 | HEGGESZTŐ_BAL |
| <input checked="" type="checkbox"/> | %Q0.13 | HEGGESZTŐ_JOB |
| <input checked="" type="checkbox"/> | %Q0.14 | LÁMPA |
| <input type="checkbox"/> | %Q0.15 | LÁMP |
| <input checked="" type="checkbox"/> | %I0.15 | KILÖVÉS_ELSŐ_VÉ |
| <input checked="" type="checkbox"/> | %I0.16 | HEGGESZTŐ_FEL |
| <input checked="" type="checkbox"/> | %I0.17 | HEGGESZTŐ_FEL |
| <input checked="" type="checkbox"/> | %I0.18 | HEGGESZTŐ_ALS |
| <input checked="" type="checkbox"/> | %I0.19 | HEGGESZTŐ_ALS |
| <input type="checkbox"/> | %I0.20 | |
| <input type="checkbox"/> | %I0.21 | |
| <input type="checkbox"/> | %I0.22 | |
| <input type="checkbox"/> | %I0.23 | |

20. ábra A hegesztő Symbol táblája

A hegesztőt mozgató motorok a % Q0.6 - % Q0.13 között helyezkednek el.

Például:

- A kar kitolásáért a % Q0.6 és % Q0.11 kimenetek a felelősek. Az egyiket megkapja a 24 voltot a másikon pedig a nullát.

A hegesztő érzékelői a teljesség igénye nélkül láthatóak az alsó képen.

Például:

- A hegesztő balra fordulását két érzékelővel vizsgálhatjuk. az % I0.18 a végállás kapcsoló, az %I 0.19 pedig a számláló. Ezek az érzékelők közvetlenül a PLC-re vannak kötve, nem pedig a bővítő egységekre.

4. Programozás

Miután összeszereltük és bekötöttük a funkcionális részeket következhetett a programozás. Mielőtt még hozzá láttunk volna, elhatároztuk, hogy addig fogjuk csinálni, amíg minden célkitűzésünk meg nem valósul. Programozás során különös figyelmet kellett fordítani a motorokra adott jelekre. Ugyanis ha egyszerre a jobbra fordulási és balra fordulási jelet is megkapják, annak komoly következménye lehet.

A programozási résznél is igyekeztünk egymás munkáját segíteni, ezért maga a program egy közös szellemi termék, de vannak benne nagyjából elkülöníthető részek.

Maga a gyártósor programja három fő részből tevődik össze. Az első rész az alaphelyzetbe állító algoritmus, amely a robotkart, a futószalagokat, a karokat és a hegesztőt alaphelyzetbe állítja, amely egyben a kiinduló pozíciójuk. A második rész pedig maga a gyártási folyamat, a vezérlés, a START. A harmadik rész egy RESET, vagy más néven STOP amit akkor hívunk meg ha valami baj történne. Ez egy vészleállítóként üzemel, amely parancs bármikor kiadható.

Ezek közül az algoritmusok közül választhatunk az érintő képernyős vezérlő egységgel, amelyre a dolgozatom során később fogok kitérni.

4.1 Alaphelyzet algoritmus, nullpont felvétel

Az algoritmus a szerkezet kiinduló állapotba állítását végzi el, azaz a nullpont felvételt. Tulajdonképpen a végállás kapcsolókig való mozgást jelenti az algoritmus. A két irányba mozgó motorok addig kapják a jelet, amíg a vezérlés bemeneti oldalán meg nem jelenik a végállás kapcsolók által adott jel. Tehát például a robotkar a saját tengelye körül addig fordul balra, amíg el nem éri a végállását, ekkor a motor nem mozog tovább.

4.2 A hegesztő tálca és a hegesztő kar algoritmus

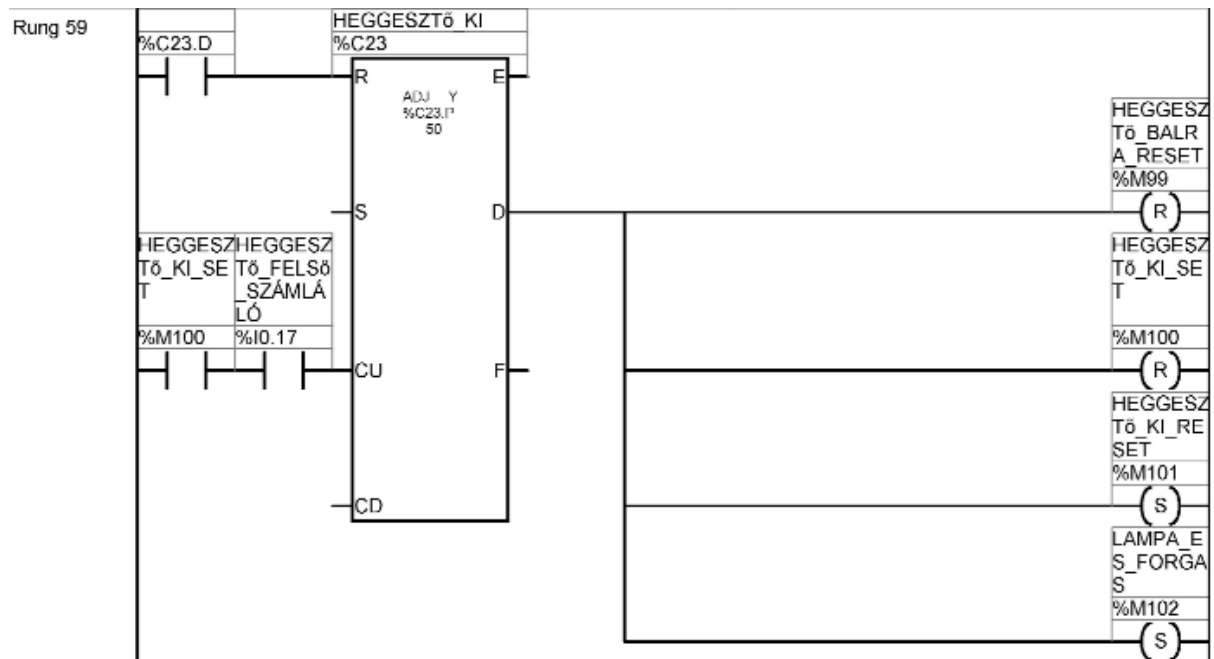
A robotkar miután a termék a hosszú futószalag végére ért, és a fényérzékelő kapu kiadta a jelet, felveszi a terméket, és a hegesztő felületre helyezi. Miután a robotkar vízszintesen teljesen visszahúzta magát, azaz végállásig a hegesztő kar balra fordul 90 fokot és ki tolja a kart. A hegesztést szimuláló izzó felgyulladás és a tálca forgása szinkronizálva van, egyszerre indulnak el.

Miután a tálca megtette a 360 fokos fordulatot, a hegesztő visszatér alaphelyzetébe. Alaphelyzetbe visszatértekor a végállás kapcsoló benyomódik, ami kiadja a jelet, a kitoló karnak, hogy a termék a raktárba kerüljön. Miután a kitoló kar is visszatért alaphelyzetbe, a komplett gyártási folyamat elkezdődik előlről.

Ez programozás technikailag úgy néz ki, hogy az 54-es rangban a %M98-as markel SET-elésre kerül, ami az 58-as rangban ÉS kapcsolatban áll a hegesztőn elhelyezett fordulatszámot számláló számlálóval. Miután 63-at számolt a %C22 jelű számláló, kiadja a jelet, amellyel a balra fordulás RESET-elődik és a %M100-as markel SET-elődik, amelyik a következő rangban kerül felhasználásra, a hegesztőkar kitolásához.

A %C23-as számláló a kar pozicionálását segíti. Bemenetére az %M100-as markel és az %IO.17-es érzékelő van kötve ÉS kapcsolattal. Miután 50-et számolt kiadja a jelet amely RESET-eli a kitolást és SET-eli a %M102-es markelt amely indítja a következő rangban elhelyezett műveleteket.

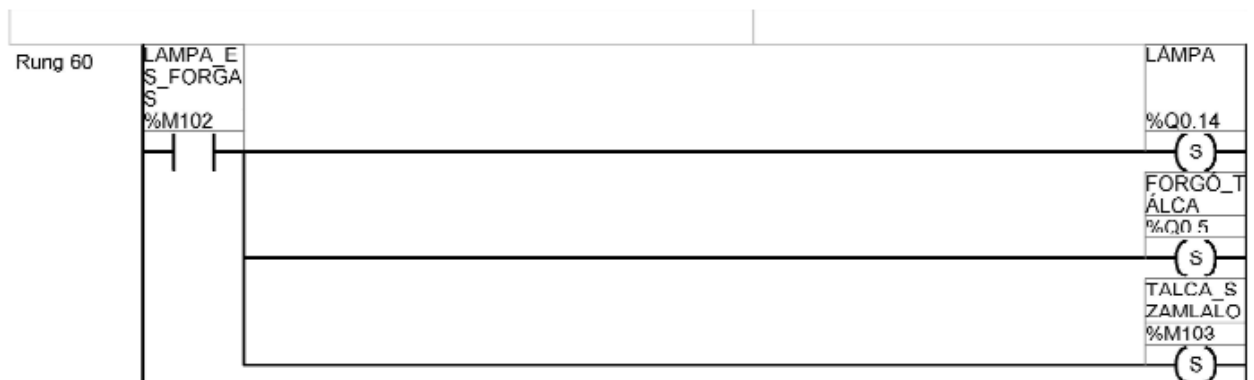
Az 59-es rang.



21. ábra 59. rung

A további műveletek is hasonlóan vannak megoldva, a gyártási művelet újratezdéséig.

Egy szemléletes példa a szinkronizációra:



22. ábra 60. rung

Jól látható, hogy a %M102 jelű markel, amely az előző rangban SET-elésre került elindítja a hegesztést, azaz a %Q0.14 jelű lámpát bekapcsolja, és a %Q0.5 jelű forgást.

4.3 A vezérlő egység programozása

A vezérlést egy Magelis XBT RT500-as egységgel oldottuk meg. Ez egy kis vezérlő panel, amely tartalmaz gombokat, és beállítható érintőkijelző funkciókat is. Az egységet Vijeo Designer Lite programban kellett beprogramozni. Az egység tulajdonképpen egy impulzus generátorként működik.

A kijelzőn a következő feliratokat helyeztük el:

- **Start (F1)**
- **Alaphelyzet (F2)**
- **Reset (F4)**

A gyártási folyamatot az F1 gomb lenyomásával érhetjük el. Előtte mindig célszerű lenyomni az F2-öt, hogy megbizonyosodjunk arról, hogy a robot kar, a szalagok és a többi egység alaphelyzetben legyen.

Az F3, vagyis a Reset gombbal a folyamat bármikor megállítható. Ezt az érzékelők Resetelése végett tettük be, illetve vészleállítóként is funkcionál. A funkciók az érintőkijelzőről is elérhetőek, de ezt nem ajánljuk, mert nem elég pontos.

Tulajdonképpen a vezérlés úgy működik, hogy a gombok lenyomásával jelet küldünk a PLC meghatározott bemeneteire, amely ennek megfelelően fog cselekedni.



23. ábra Magelis XBT RT[11.]

5. Fejlesztési lehetőségek

A szakdolgozat elkészítése során rengeteg új ötlettel álltunk elő. Ezek nagy része megvalósításra is került. Ilyen volt például a selejtkezelés megoldása, az induktív érzékelő beépítése által. A vezérlést is szeretnénk volna megoldani érintőképernyővel, de sajnos az eszköz nagyon kezdetleges érintőkijelzőt biztosít, így hát erről le tettünk.

Folyamatosan merültek fel ötletek az optimalizálással, és gyártással kapcsolatban, volt amire időhiány miatt nem került sor, volt amire anyag miatt. A gyártási folyamatban, ha több lenne az esemény vezérlés, a késleltetés helyett akkor több terméket tudnánk legyártani egységnyi idő alatt.

Felmerült az ötlete két robotkarnak, de ezt végképp elvetettük idő hiányában.

A jövőre nézve érdemes lenne a gyártási folyamatot felgyorsítani. Gondolkoztunk csomagoló funkció beépítésén is. A raktárba helyezéskor a kitoló kar a terméket egy automatikusan záródó fólián tolná át, és a fizikai mozgások a fóliát pont úgy gyűrnék össze, hogy az egy becsomagolási műveletet szimuláljon.

6. Összegzés

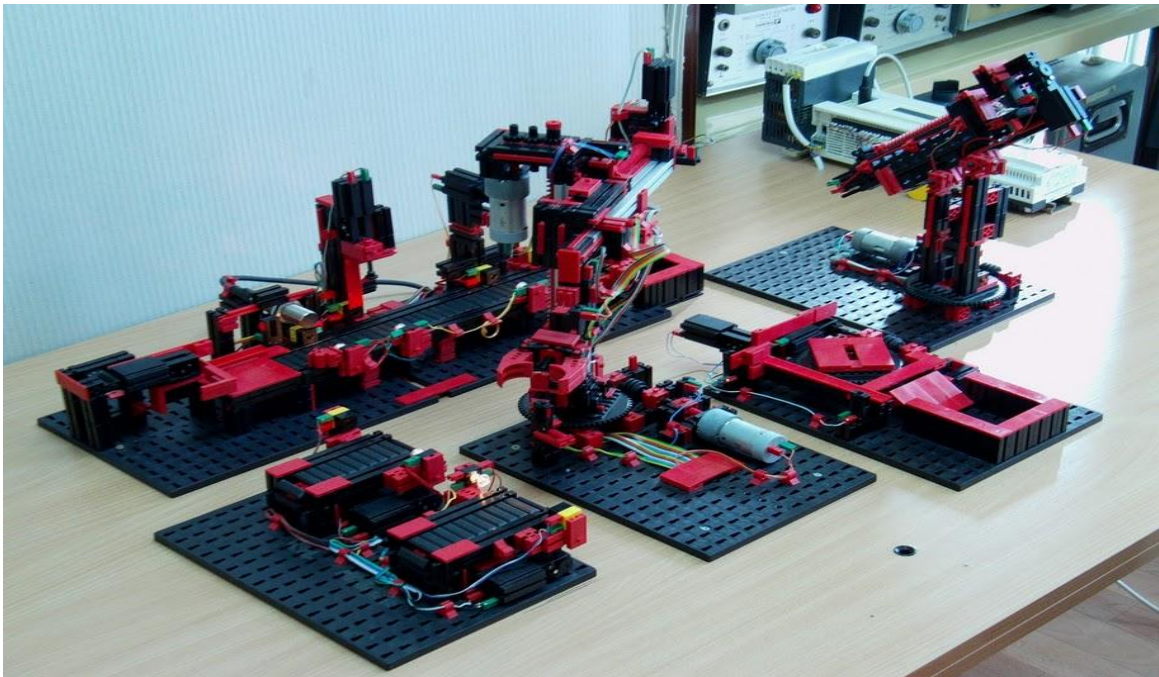
Szakdolgozatunk célja egy gyártósor elkészítése volt, amely egy terméket két különálló részből rak össze, valamint teljesen automatizáltan emberi beavatkozás nélkül működik. Két fő célunkat meg is valósítottuk.

Azért választottuk ezt feladatul, mert már korábbi tanulmányaink során megismerkedtünk a PLC-vel, és a gyártási folyamatokkal.

Nagyon sok és hasznos dolgot megtanultam a csapatmunkáról, melyeknek későbbi életem során nagy hasznát fogom venni. Úgy érzem, hogy problémamegoldó képességem is rengeteget fejlődött. Ugyanis a munka során rengeteg apró hiba, eltérés és nehézség merült fel. Amit így utólag úgy érzem sikerült áthidalnunk.

Külön öröm számunkra, hogy a 2010-es szakmai napokon kiállíthattuk a modellt a Debreceni Egyetem díszudvarában

Nagyon örülök, hogy ezt a témát választottam, élveztem a kihívást, a csapatmunkát és maga a téma is igen érdekes volt számomra, jelenleg sem döntenék másként.



24. 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

Könyvek

- [1.] Twido Szoftver referencia kézikönyv (18, 19, 20, 147, 148, 149)
- [2.] www.schneider-electric.hu oldalon található könyvek
- [3.] Automatizálás és vezérlés TWIDO Programozható vezérlő Katalógus 2008, (6, 7, 9, 45, 47, 120)
- [4.] Dr. habil Kulcsár Béla, LSI Oktatóközpont, Robottechnika, 1998
- [5.] Dr. Siegler András: Robot irányítási modellek, LSI, 1987, BP
- [6.] Handbook of Industrial Automation – Richard L. Shell and Ernest L. Hall 2000
Marcel Dekker
- [7.] Control of Redundant Robot Manipulators – R.V Patel and F. Shadpey, Springer –
Verlag Berlin Heidelberg, 2005

Linkek

- [8.] http://www.ifm.com/ifmhu/web/pinfo1_10_10_40.htm 2010.11.03
- [9.] <http://e-oktat.pmmf.hu/robotir2> 2010.11.05
- [10.] <http://en.wikipedia.org/wiki/Fischertechnik> 2010.10.19
- [11.] http://www.q-tech.hu/pdf/Schneider/magelis%20xbt_rt_broc_en_200807.pdf
2010.11.10
- [12.] <http://www.schneider-electric.hu/sites/hungary/hu/tamogatas/ipari-automatizalas-konyvtar/letoltes/download-documents.page> 2010.11.10
- [13.] http://www.schneider-electric.hu/hungary/hu/termekek-szolgalatasok/ipari-automatizalas/termekek/range-presentation.page?p_function_id=26&p_family_id=279&p_range_id=1054
2010.11.15