

SZAKDOLGOZAT

Nagy István

Debrecen

2010

Debreceni Egyetem

Informatika Kar

Műszaki kar

MANIPULÁTOR PROGRAMOZÁSA

Témavezető:

Dr. Husi Géza

Tanszékvezető főiskolai docens

Készítette:

Nagy István

Mérnök informatikus hallgató

Debrecen

2010

TARTALOMJEGYZÉK

1. TÉMAVÁLASZTÁS.....	1
2. A FELADATRÓL.....	2
3. ROBOTOKRÓL ÁLTALÁBAN.....	3
3.1.1. TÖRTÉNETI HÁTTÉR	3
3.1.2. ROBOTOK FOGALMA.....	3
3.1.3. A ROBOTIKA HÁROM ALAPTÖRVÉNYE.....	4
3.1.4. ROBOTGENERÁCIÓK.....	4
3.1.5. ROBOTOK OSZTÁLYOZÁSA	5
4. FISCHERTECHNIK 3D ROBOT.....	9
4.1.1. FISCHERTECHNIK KÉSZLETEK ÉS ROBOTOK.....	9
4.1.2. KÉSZLETEK.....	10
4.1.3. IPARI ROBOT FELÉPÍTÉSE.....	10
4.1.4. PROGRAMOZÁSI LEHETŐSÉGEI.....	11
5. LABVIEW-RÓL ÁLTALÁBAN.....	12
5.1.1. NI VISION.....	14
5.1.2. LABVIEW ELŐNYEI.....	14
6. PÁRHUZAMOS ÉS USB PORT KOMMUNIKÁCIÓ.....	15
6.1.1. KOMMUNIKÁCIÓ A 3D ROBOT ÉS A LABVIEW KÖZÖTT.....	15
6.1.2. KOMMUNIKÁCIÓ A LABVIEW ÉS A KAMERA KÖZÖTT.....	17
7. VEZÉRLŐ PANEL	17
7.1.1. ELEKTRONIKAI MEGVALÓSÍTÁS.....	17
7.2.1 FELHASZNÁLT ESZKÖZÖK.....	18
7.2.2 MOTORVEZÉRLÉS KAPCSOLÁSI RAJZA.....	21
7.2.3 AZ ÁRAMKÖR PARAMÉTEREI.....	23
7.2.4 VÉGÁLLÁS KAPCSOLÓK.....	23
7.2.5 ENKÓDEREK.....	25

8. VEZÉRLÉS LABVIEW-AL.....	26
8.1.1. ENKODEREK A LABVIEWBAN.....	26
8.1.2. MOTORVEZÉRLÉS A LABVIEWBAN.....	27
8.1.3. KÉZI VEZÉRLÉS.....	31
9. MUNKATERÜLET.....	32
10. DÖNTÉSEK, KALKULÁCIÓK.....	33
11. MEGSZAKÍTÁSOK.....	36
12. ÁLTALÁNOS TAPASZTALATOK	39
13. ÖSSZEFOGLALÁS	40
14. IRODALOMJEGYZÉK	42
15. FELHASZNÁLT IRODALOM	43
16. FÜGGELÉK	44
17. KÖSZÖNETNYILVÁNÍTÁS	46

1. TÉMAVÁLASZTÁS

Eddigi tanulmányaim során lehetőségem nyílt különböző programozási nyelvekkel megismerkednem és azok különböző területen való felhasználásaival. Ezen felhasználási területek egyike a robotok programozása, aminek köszönhetően betekintést nyertem a robotok világába. Az egyetemeken való oktatásból tükröződni látszik, hogy milyen mértékben megnőtt világunk kapcsolata a robotokkal, éppen ezért döntöttem úgy, hogy szakdolgozatom is témához kapcsolódóan egy robot felprogramozása, és vezérlési elektronikájának megtervezése lesz.

Választásom egy olyan iparban is felhasználható robotnak a modelljére esett mely széles körben felhasználható és különféle tesztelési lehetőségeket biztosít, egyedül a pozicionálási pontosság hagy maga után kívánni valót. Munkámhoz a Fischertechnik egyik 3 dimenzióban mozogni képes robotját fogom használni.

Működésének megvalósításához, általam még nem ismert programozási nyelvet választottam, tanulmányaim bővítése céljából. Remélhetőleg a dolgozat végére ez teljesülni is fog.

2. A FELADATRÓL

A feladat, mint ahogyan a témaválasztásban is említést nyert egy olyan iparban is használható manipulátor modelljének programozása ami jelentős segítséget nyújthat az ipar egyes területein. A projekt megvalósítása két főt igényelt, melyben Berna Attila mérnök informatikus csoporttársam volt a projektársam is egyben. A szakdolgozatom a manipulátor mechanikai felépítése, az elektronikai panel megtervezése és megépítése, valamint a vezérlésből készítem, míg a kameravezérlést és képfeldolgozást Attila.

Ezen részek összességének megvalósításából egy olyan manipulátort szeretnénk összeállítani ami egy kamera által vezérelt robotkar, mely képes a hozzátartozó munkaterületén munkadarabokat pozícionálni majd elmozgatni. Részleteiben azt szeretnénk megvalósítani, hogy a kamera által felmért munkadarab munkaterületi elhelyezkedését meg tudjuk állapítani átszámolni valós távolságba és paramétereit megadva a robotkart egy általam épített vezérlő elektronika segítségével a megfelelő helyre pozícionálni. Ennek a gyakorlati hasznosságáról beszélve említhetünk olyan ipari tevékenységet, ahol egy futószalagon sorba érkező munkadarabok vagy bármilyen más tárgyak közül a hibásat felismerve kiveszi azt a sorból. Egy másik felhasználási lehetőség olyan munkaterületről történő munkadarab kiválasztása ahol esetleg a munkadarabok a futószalagon nem arányosan érkeznek, vagy annak megállásakor elmozdulhatnak, elgurulhatnak.

Szeretnék egy olyan motorvezérlő elektronikát amin keresztül a megírt program vezérelni tudja a robotkart a kamera által megadott paraméterek alapján.

Remélhetőleg minden sikeresen meg is valósul, az összegzésből kiderül.

3. ROBOTOKRÓL ÁLTALÁBAN

3.1.1 TÖRTÉNETI HÁTTÉR

Politika I. című művében Arisztotelész már megfogalmazta koncepcióját olyan eszközökről, amelyek képesek automatikusan, a saját erejükből tevékenykedni. Az ókor első működő automatáját valószínűleg az ókori Alexandriában építette Archimédész kortársa, Kteszibiosz, az időszámításunk előtti III. században, igaz, pusztán szórakoztatás céljából. Az i.sz. I. században a szintén alexandriai Héró ötletes szerkezeteiről részletes és pontos leírások maradtak fenn. Többek között árusító automatákat, valamint egy automata színházat tervezett és épített meg - ez utóbbi, a *Dionüszosz apoteózisa*, egy önjáró miniatűr templom tetején játszódott. [1.]

Alexandria arab bevétele után is tovább élt az automaták építésének hagyománya, bár főleg vallásos célokat használt.

Az automaták kulturális objektumokból a 20. század második felében válnak inkább technológiai objektumokká. Igazolja ezt az 1928-ban a londoni gépipari kiállítást megnyitó „mesterséges ember”, aki rövid beszéddel és udvarias gesztusokkal hajtotta végre a megnyitást.

Az első mai szemmel is iparinak nevezhető robot az amerikai Unimation cég "PUMÁ"-ja volt a 60-as évek közepén. 1987-ben Nyugatnémet országban több, mint száz cég 250 különböző robotot kínált. A nagyvilágban 1990-ben már kb. 200 000 robot működött.[1.]

3.1.2 ROBOTOK FOGALMA

A robot újraprogramozható, sokfunkciójú manipulátor, amely anyagokat és szerszámokat változó program szerint mozgat, technológiai feladatokat ellát, összeszerelést végez. A programozható ipari robotok a programozásukkal meghatározott ismétlődő mozgássorozatok elvégzésére alkalmasak, általában szerszámokat vagy munkadarabok

mozgatnak szakaszosan vagy folyamatosan vezérlőegységük memóriájába tárolt programnak megfelelően. A munkaciklus befejezése után visszaállnak alaphelyzetükbe, és a következő indító jelre a programozott mozgássorozatot megismétlik. A robotokat, fokozatos, fejlesztésüknek megfelelően generációba soroljuk.[2.]

3.1.3 A ROBOTIKA HÁROM ALAPTÖRVÉNYE

I, Robot (Asimov, 1950) tudományos, fantasztikus könyvéből [3.]:

- 1. „A robotnak nem szabad kárt okoznia emberi lényben vagy tétlenül túrnie, hogy emberi lény bármilyen kárt szenvedjen.”*
- 2. „A robot engedelmeskedni tartozik az emberi lények utasításainak, kivéve, ha ezek az utasítások az első törvény előírásaiba ütköznének.”*
- 3. „A robot tartozik saját védelméről gondoskodni, amennyiben ez nem ütközik az első és a második törvény előírásaiba.”*

3.1.4 ROBOTGENERÁCIÓK

Első generációs robotok:

Kötött program szerint működnek, pontvezérlésűek. E típusoknál a környezetből érkező jelek nem befolyásolják a robot mozgásprogramját.

Ezeknél az úgynevezett “vak” robotoknál pl.: anyagmozgatás esetén külön gondoskodni kell a felveendő tárgy megfelelő pontosságú tájolásáról.

Második generációs robotok:

Ezek már érzékelőkkel vannak ellátva, és az ezekből kapott jelek befolyásolják a mozgást. (Általában helyzet, erő, nyomaték, hőmérséklet, stb. érzékelőket alkalmaznak)

Az érzékelők által szolgáltatott jelek alapján az információk gyors feldolgozására képes - többnyire mikroprocesszoros - vezérlőegység a mindenkori helyzet figyelembevételével célorientált utasításokat ad a végrehajtó szervek részére. Ezek a robotok

már alkalmazkodni képesek a környezetükben történő bizonyos változásokhoz. Vezérlésük pont, vagy pályavezérlés.

Harmadik generációs robotok:

Komplex feladatok elvégzésér (pl.: automatizált szerelés ellenőrzése stb.) alkalmasak. E robotok érzékelőkkel környezetükből sokféle, információt gyűjtenek, ill. dolgoznak fel, amelyek segítségével önálló döntésekre, bonyolult folyamatokban való részvételre is alkalmasak.

Alak és helyzetfelismerő képességük, mesterséges intelligenciájuk magas fokú. Az egyes robotgenerációk között, természetesen, nem lehet éles határt vonni. Napjainkban is egyre többféle új robot-modell jelenik meg.[2.]

3.1.5 ROBOTOK OSZTÁLYOZÁSA

Robotok osztályozása többféle szempont szerint történhet:

- *Szolgáltatásaik szerint lehetnek:*

Egyszerű robotok

Pozicionálásuk ütközők segítségével történik, általában pneumatikus működésűek, vezérlésük elektropneumatikus. Programozásuk többnyire a programhengeren történő beállítással valósul meg. Főleg munkadarabok, szerszámok mozgatására használják.

Univerzális robotok

Többnyire hidraulikus vagy villamos hajtásúak. Mágneses memóriájuk és mikroprocesszoros központi egységük van. Ponttól - pontig vezérléssel, vagy pályavezérléssel működnek.

Alkalmazási területük elsősorban technológiai jellegű. pl.: hegesztés, felületkezelés, szerelőszalagok kiszolgálása

- *Alkalmazási területük szerint:*

Mozgást végző robotok

Jellegzetes alkalmazási területük:

- fémforgácsoló szerszámgépek kiszolgálása
- munkadarabok átrakása, átfordítása
- kovácsolóműhelyek kiszolgálása
- anyagmozgatás
- stb.



1. ábra: Anyagmozgató robot (forrás:[4.]).

Technológiai műveleteket végző robotok

Jellegzetes alkalmazási területük:

- festékszórás
- pont és vonal hegesztés

- öntvénytisztítás
- revételenítés
- sorjázás
- köszörülés
- szerelés végzése



2. ábra: Hegesztő robot (forrás:[4.]).

- **Ellenőrző robotok**

Termékek automatikus vizsgálatát, mérését, osztályozását végzik. Ellenőrzik a gyártástechnológiai folyamatot, és saját működésüket.



3. ábra: Ellenőrző robot (forrás[5.]).

Jellemzőik:

- mechanikai felépítés
- mozgástér
- karrendszer
- koordináta rendszer
- szabadságfokok
- terhelhetőség
- ismétlési pontosság
- dinamikus pontosság

[2.]

4. FISCHERTECHNIK 3D ROBOT

4.1.1 FISCHERTECHNIK KÉSZLETEK ÉS ROBOTOK

A Fischertechnik készletek, egy az oktatásban nagy szerepet magáénak tudható rugalmas építési rendszer, mely nagyban elősegíti a tudomány, a matematika, a technológia és a mérnöki ismeretek elsajátítását a tantermeken belül.

A Fischertechnik termékei, olyan több száz alkatrészből álló készletek, melyek rugalmasan felhasználhatóak különböző eszközök, robotok megépítéséhez. Az egyik ilyen már a Fischertechnik által összeállított ipari 3 dimenziós robotkar, mely egy adott munkaterületen elérhető munkadarabok mozgatására lett tervezve. 3 szabadsági fokkal rendelkezik minek köszönhetően a munkaterülete elég nagy. A 3 dimenziós robotkaron kívül más ipari robotok is megépíthetőek a készletből. A Fischertechnik meg is tervezett, és épített is néhányat.(4. ábra) és (5. ábra).



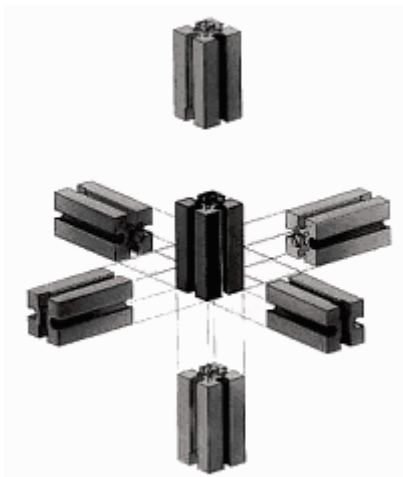
4. ábra Fischertechnik Targonca(forrás:[6.]).



5. ábra Fischertechnik hegesztő robot (forrás:[6.]).

4.1.2 KÉSZLETEK

A Fischertechnik ipari robot készletének tartalma lehetővé teszi 3 iparban használatos robot megépítését, mint ahogy azt a 4. ábrán valamint 5. ábrán is láthattuk, de most csak a 3D robotkar megépítéséhez szükséges elemekből mutatok be párat. (6. ábra) és (7. ábra)



6. ábra Fischertechnik építőelemek (forrás: [7.]).

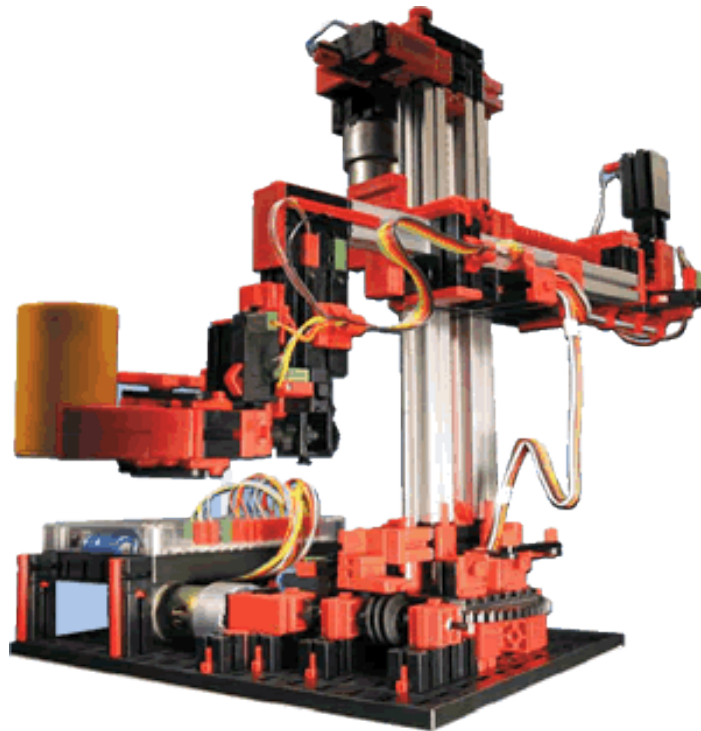


7. ábra Fischertechnik építőelemek (forrás: [7.]).

A készlet tartalmából felhasznált elemek kimagasló része különböző, maga a kart alkotó és tartó vázból, valamint a mozgást lehetővé tevő tengelyekből és forgó alkatrészekből áll. A készletben felhasználásra került még 4 DC motor a hozzá tartozó 4 enkóderrel, valamint 4 végállás kapcsoló, illetve megfelelő mennyiségű kábel és csatlakozó.

4.1.3 IPARI ROBOT FELÉPÍTÉSE

Felépítését tekintve, mint már említettem 3 szabadsági fokkal rendelkezik, így képes a jobbra-balra, előre-hátra, fel-le mozgásokra, valamint a kar végén található fogókar nyitás-zárás mozgásokat tud végrehajtani.



8. ábra Fischertechnik ipari 3D robotkar (forrás:[8.]).

Az irányoknak megfelelően minden tengely végére végállás kapcsolók vannak szerelve illetve minden motor el van látva egy-egy enkóderrel, lévén DC motorokról van szó, így szükség van a számlálókra.

4.1.4 PROGRAMOZÁSI LEHETŐSÉGEI

A Fischertechnik 3D robotkar programozási lehetőségeit tekintve több különböző egymástól eléggé eltérő megoldást is találhatunk. Az egyik ilyen megoldás a PLC azaz programozható logikai egység használata, mely módszerhez volt szerencsém tanulmányaim során. Egy másik megoldás, a gyártó által kiadott programozási egységhez készített Fischertechnik ROBO Pro [6.] nevű szoftver. Ez a megoldás a legegyszerűbb a robotkar funkcióit tekintve. Egy harmadik módszer a szabad felhasználás, köszönhetően, belső, a

robotkarhoz tartozó programozói egység hiányának. A feladatot tekintve a felhasznált készleten kívül még felhasználásra került egy kamera is, ez kissé megváltoztatja a programozási feltételeket.

Éppen ezért esett a választás a LabVIEW 8.6 szoftverre, mely lehetővé teszi a képfeldolgozást amire szükségünk van a feladat megvalósításához. Ez a választás egy újabb problémát tárt fel mégpedig a vezérlés problémáját. Szükség van egy nyomtatott áramköri panelre ami a LabVIEW-től kapott információk alapján vezérli a robotkart.

5. LABVIEW-RÓL ÁLTALÁBAN

A LabVIEW egy grafikus programozói nyelv melynek fő alkalmazási területe a virtuális mérés technika és automatizálás. A LabVIEW a National Instruments terméke amit az Apple Macintosh-ra fejlesztettek ki 1986-ban.

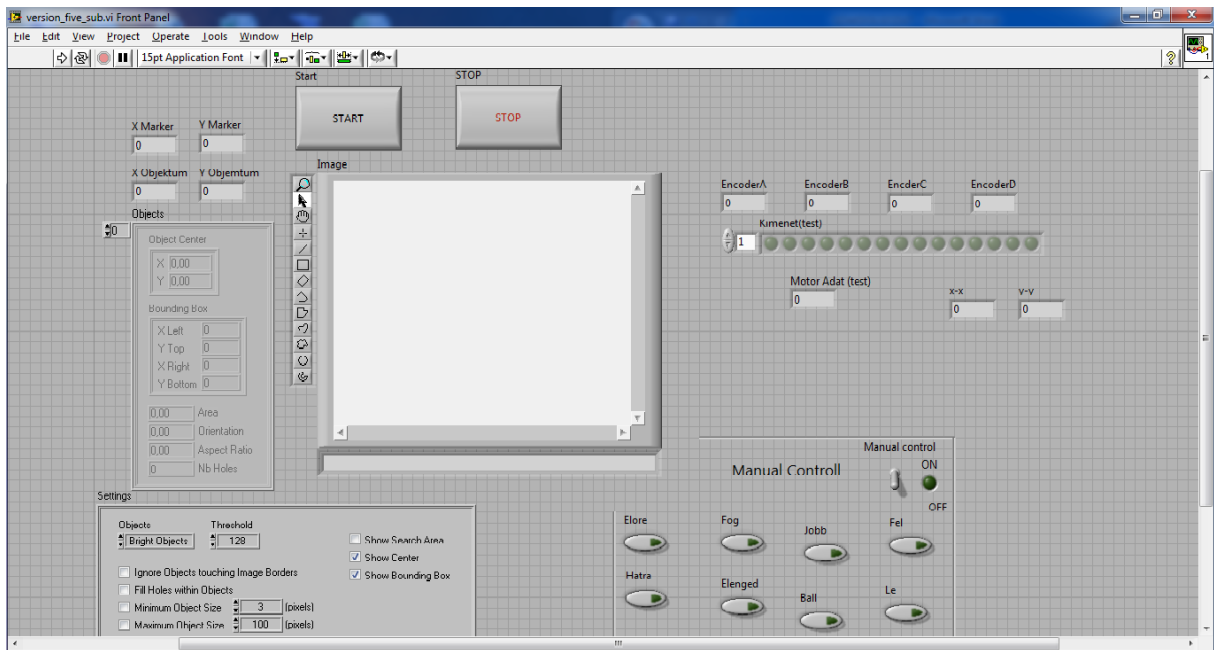
Ezen program segítségével adatgyűjtő műszervezrlő tesztelő és mérő, valamint adatnaplózó és mérés elemző alkalmazásokat készíthetünk.

Használható még általános célú grafikus programozási nyelvként is.

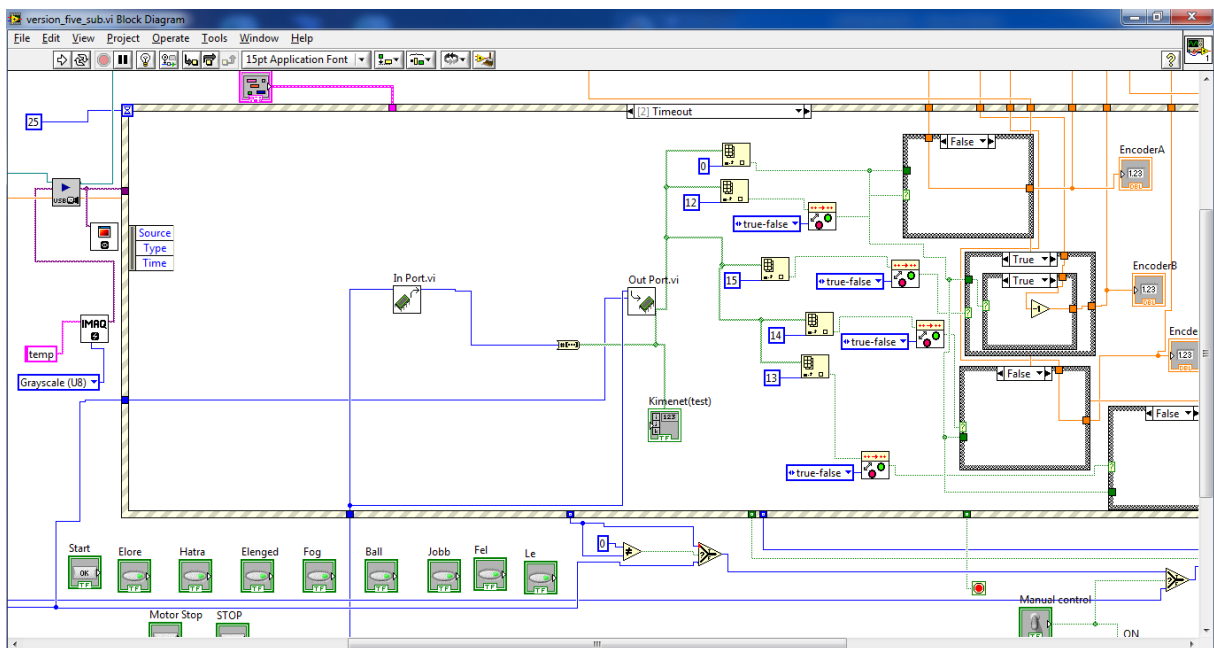
Virtuálisan építhetők különböző hardverelemek és tesztelhetők is, valamint lehetőség van hardverrel történő kommunikációra úgymint:

- adatgyűjtés,
- gépi látás,
- mozgásvezérlő eszközökkel való kommunikáció.

A LabVIEW programokat virtuális műszereknek, vagy röviden VI – oknak nevezzük, mert megjelenésükben és működésükben a fizikai műszereket utánozzák. A LabVIEW VI-ok három komponensből állnak: az előlapból(9. ábra), blokk diagramból(10. ábra) és az ikon és csatlakozó panelből.



9. ábra LabVIEW előlap (forrás: [Általam készített képkivágás.]).



10. ábra LabVIEW blokk diagramm (forrás: [Általam készített képkivágás.]).

A LabVIEW lévén egy 32 bites fordítóprogram, lehetőségünk van, hogy egységes és önállóan fordítható programokat valamint DLL fájlokhoz hasonló könyvtárakat hozzunk létre.

5.1.1 NI VISION

Az NI Vision fejlesztő modul egy olyan átfogó könyvtár, amely több száz képfeldolgozó algoritmust tartalmaz. Ennek a kiegészítő modulnak köszönhetően a LabVIEW segítségével olyan alkalmazásokat is létre tudunk hozni amiben egy vagy akár több kamerát is használunk. A Vision-t használhatjuk képek videók szerkesztésére, feldolgozására, javítására, tárgyak jelenlétének felismerésére, azonosítására, ellenőrzésére. Mindezen funkciók igénybe vételéhez és egy kamera felhasználásához szükségünk van maga az eszköz és a LabVIEW közti kommunikációra. Ez a kommunikáció különböző I/O perifériákon valósul meg.

5.1.2 LABVIEW ELŐNYEI

Más programozási nyelvekkel szemben a LabVIEW használata a feladat megvalósításához sokkal célszerűbb lévén, képfeldolgozás is szerepel benne és kamera, azaz a Vision által küldött adatok amelyeket a LabVIEW feldolgoz majd továbbít a vezérlőpanelnek sokkal egyszerűbb megvalósítani. C programozási nyelven ez akár több tíz oldalnyi kódot is maga után vonna. Ezen programozási nyelv mellett szól még a jól kezelhető eseményvezérlés, melyet ügyesen használva nagy mértékben kímélhetjük a processzor munkáját. Segítséget jelent továbbá a már említett előlap, hiszen a blokk diagram felületen készített alkalmazás nyomon követhető ezen a felületen. Ezen a felületen történik az adatok bevitele és megjelenítése.

6. PÁRHUZAMOS ÉS USB PORT KOMMUNIKÁCIÓ

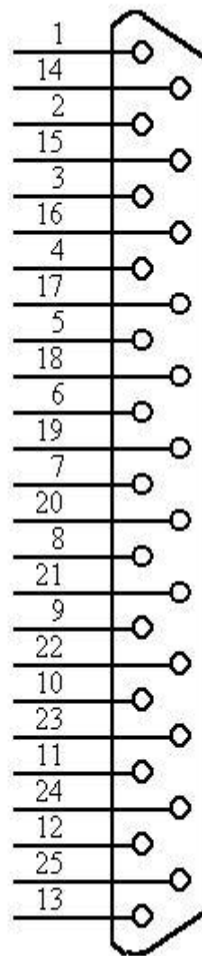
6.1.1 KOMMUNIKÁCIÓ A 3D ROBOT ÉS A LABVIEW KÖZÖTT

A manipulátor, pontosabban a manipulátort vezérlő egység a LabVIEW-al párhuzamos porton keresztül kommunikál. Azért választottam ezt a megoldást, mert a feladat során kiderült, hogy a kapcsolók és számlálók sebességéhez a párhuzamos port által kínált adatátviteli sebesség a legmegfelelőbb, mert az adatátviteli sebesség 9600 bit/s. A LabVIEW szemszögéből nézve ez a hardveres megoldás a megfelelő, mert nem kell külön szoftver a használatához így direkt módon címezhetjük. A port kezelését illetően a párhuzamos port címe általában 378, ami egy hexadecimális szám. Ez az érték általában átírható a BIOS-ban. Felhasználáskor át kell alakítani decimális számmá így 888- at kapunk amit már a LabVIEW portkezelője is lát.

IEEE 1284 (Institute of Electrical and Electronic Engineers Standards) interfész működése TTL tranzisztorok segítségével történik. Ezek az úgynevezett logikai kapuk 0-5 V közötti jelszinten 0 értéket adnak 0V esetén 1-et 5V esetén. A párhuzamos port az említett IEEE 1284 szabvány szerint működik.

A párhuzamos porton van összesen 4 darab állapotbit és 8 adatbit ami lehet kimeneti vagy bemeneti láb, amiből 5 került felhasználásra A projekt során felváltva használjuk ki és bemenetként, mert egyszerre nem használható, a program váltogatja a bemenetet kimenetre és fordítva.

1. vezérlő.bit0
2. adat. bit 0
3. adat. bit 1
4. adat. bit 2
5. adat. bit 3
6. adat. bit 4
7. adat. bit 5
8. adat. bit 6
9. adat. bit 7
10. állapot.bit6
11. állapot.bit7
12. állapot.bit5
13. állapot.bit4
14. vezérlő.bit1
15. állapot.bit3
16. vezérlő.bit2
17. vezérlő.bit3
18. GND
19. GND
20. GND
21. GND
22. GND
23. GND
24. GND
25. GND



11. ábra Párhuzamos port lábkiosztás(forrás: [Saját kép.]).

A 0 számú adatbit lesz az irány bit ami a forgásirányt fogja megadni.

A 1,2,3,4 bitek fogják a motorvezérléshez szükséges adatot biztosítani.

0 bitérték esetén a motor előre 1 bitérték esetén a motor visszafelé, azaz ellentétes irányban fog forogni.

A 10,11,12,13 – as bemenetek pedig a számlálókra vannak bekötve.

A 18- 25 lábak pedig a földelés.

Ezen kapcsolások a vezérlés fejezetben részletesen kifejtésre kerülnek.

6.1.2 KOMMUNIKÁCIÓ A LABVIEW ÉS A KAMERA KÖZÖTT

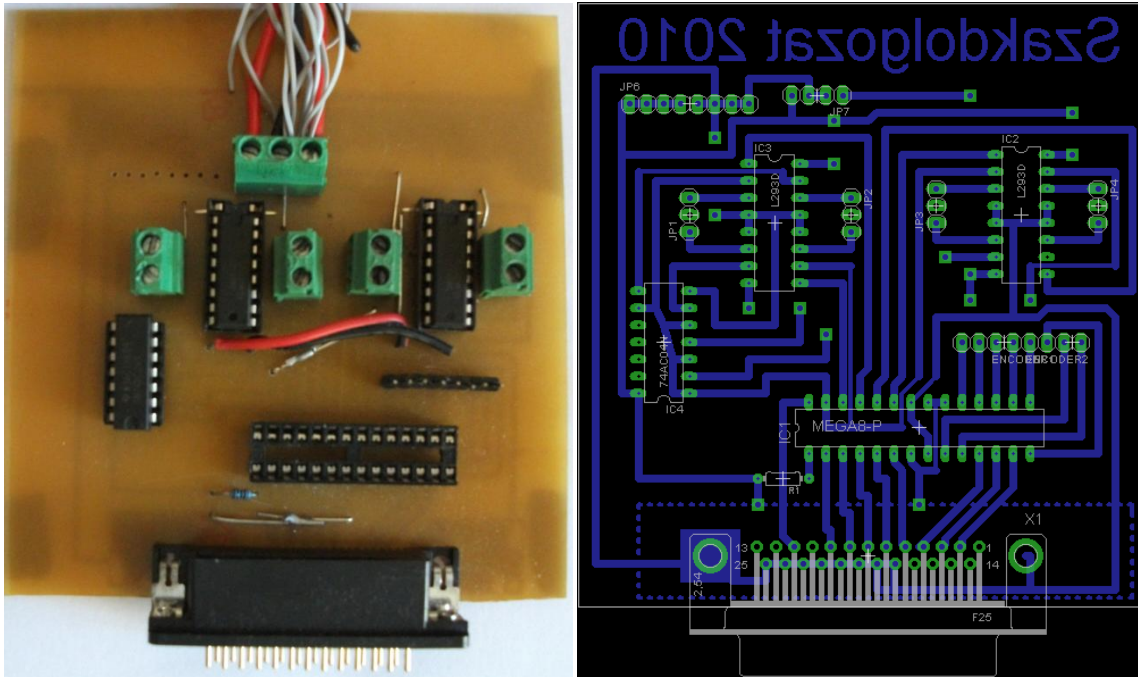
A képfeldolgozáshoz szükséges kamera egy számítógéphez csatlakozik USB porton keresztül. A képfeldolgozást és az USB kommunikációt tekintve a LabVIEW esetén szintén egyszerűbb dolgunk van, mint egy másik programozási nyelvet használva, mivel a LabVIEW lehetőséget ad a kamerák automatikus felismerésére.

7. VEZÉRLŐ PANEL

7.1.1 ELEKTRONIKAI MEGVALÓSÍTÁS

A robotkar vezérlése, mint előbbieken már említettem megkövetelte egy saját vezérlés megépítését. A vezérlés megtervezésekor végig kellett gondolni a teljes működést, mint egy folyamatot. Az adat útját végigkövetve a megfelelő elektronikai eszközöket felhasználva terveztem meg a vezérlést. Elsősorban az egyik fő szempont a külső eszközök csatlakozása, amint azt a "Kapcsolatok" részben már említettem.

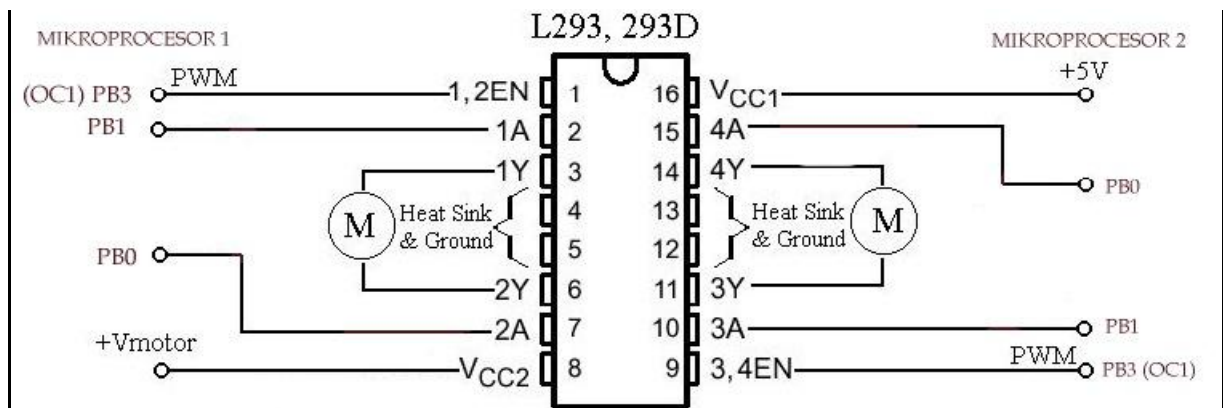
A feladata tehát összekapcsolni és adatot továbbítani a számítógéptől a robot felé és vissza. Tehát a panel tartalmaz egy párhuzamos port csatlakozási felületet, ezen keresztül kommunikál majd a számítógéppel. A porton keresztül érkező adatok továbbítódnak egy L293D nevű IC- be ami a motorvezérlésért felelős. A párhuzamos port kimeneteleivel spórolva egy inverterre is szükség van. A motorvezérlő IC motorvezérléshez szükséges lábai pedig a motorok csatlakozóinak megfelelő csatoló felületébe vannak kivezelve. A panelen található lesz még két nyomógomb, az egyik a START másik pedig a STOP funkciógomboknak felelnek meg.



12. ábra Vezérlő elektronika(Tesztáramkör) (forrás: [Saját készítésű kép.]).

7.2.1 FELHASZNÁLT ESZKÖZÖKRŐL

L239D motorvezérlő IC



13. ábra L239D Motorvezérlő IC (forrás: [9.]).

Az L293D motorvezérlő egy 4 csatornás meghajtó IC mely kiválóan használható motor vezérlésekhez, léptető vagy akár DC motorokhoz is. A független engedélyező bemeneteinek köszönhetően egyszerűen kettős hídba köthető. Tápfeszültsége 4,5V-tól -36 V-ig terjed így a feladat megvalósításához tökéletes, hiszen 12 V-t használok a DC motorok meghajtásához.

Mint már említettem, az engedélyező lábak segítségével praktikus össze lehet kapcsolni egy másik L293D IC-vel, így már mind a 4 motor vezérelhető lesz.

Elektromos paraméterek és igazságtáblák:

ABSOLUTE MAXIMUM RATINGS

Symbol	Parameter	Value	Unit
V _S	Supply Voltage	36	V
V _{SS}	Logic Supply Voltage	36	V
V _i	Input Voltage	7	V
V _{en}	Enable Voltage	7	V
I _o	Peak Output Current (100 μs non repetitive)	1.2	A
P _{tot}	Total Power Dissipation at T _{pins} = 90 °C	4	W
T _{stg} , T _j	Storage and Junction Temperature	- 40 to 150	°C

TRUTH TABLE (one channel)

Input	Enable (*)	Output
H	H	H
L	H	L
H	L	Z
L	L	Z

EN	1A	2A	FUNCTION
H	L	H	Turn right
H	H	L	Turn left
H	L	L	Fast motor stop
H	H	H	Fast motor stop
L	X	X	Fast motor stop

L = low, H = high, X = don't care

14. ábra L293D Motorvezérlő IC adatok(forrás:[10.])

74HCT04 INVERTER

A párhuzamos port I/O erőforrásainak jobb kihasználásának érdekében szükség van egy inverter alkalmazására is mely segítségével kevesebb kimeneti lábat kell felhasználnunk a párhuzamos portról. Ennek a gyakorlati megvalósítása és használati felépítése a későbbiekben bemutatott kapcsolási rajznál lesz részletesebben bemutatva.

Elektromos paraméterek és igazságtáblák:

QUICK REFERENCE DATA

GND = 0 V; $T_{amb} = 25\text{ }^{\circ}\text{C}$; $t_r = t_f \leq 6.0\text{ ns}$.

SYMBOL	PARAMETER	CONDITIONS	TYPICAL		UNIT
			HC04	HCT04	
t_{PHL}/t_{PLH}	propagation delay nA to nY	$C_L = 15\text{ pF}$; $V_{CC} = 5\text{ V}$	7	8	ns
C_i	input capacitance		3.5	3.5	pF
C_{PD}	power dissipation capacitance per gate	notes 1 and 2	21	24	pF

FUNCTION TABLE

See note 1.

INPUT	OUTPUT
nA	nY
L	H
H	L

Note

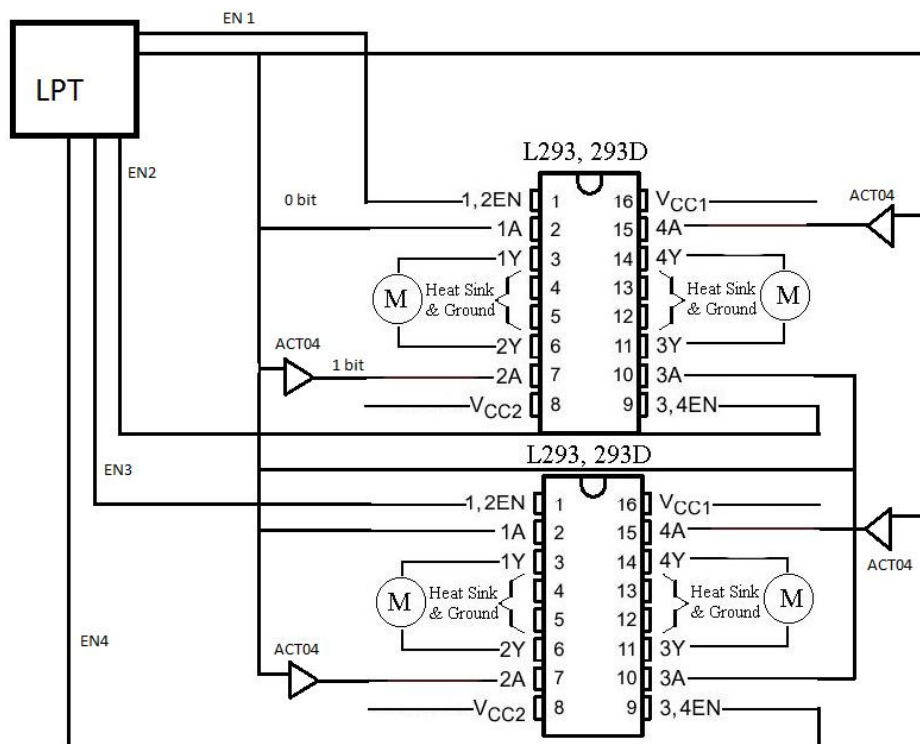
- H = HIGH voltage level;
L = LOW voltage level.

RECOMMENDED OPERATING CONDITIONS

SYMBOL	PARAMETER	CONDITIONS	74HC04			74HCT04			UNIT
			MIN.	TYP.	MAX.	MIN.	TYP.	MAX.	
V_{CC}	supply voltage		2.0	5.0	6.0	4.5	5.0	5.5	V
V_i	input voltage		0	–	V_{CC}	0	–	V_{CC}	V
V_o	output voltage		0	–	V_{CC}	0	–	V_{CC}	V
T_{amb}	ambient temperature	see DC and AC characteristics per device	–40	+25	+125	–40	+25	+125	$^{\circ}\text{C}$
t_r, t_f	input rise and fall times	$V_{CC} = 2.0\text{ V}$	–	–	1000	–	–	–	ns
		$V_{CC} = 4.5\text{ V}$	–	6.0	500	–	6.0	500	ns
		$V_{CC} = 6.0\text{ V}$	–	–	400	–	–	–	ns

15. ábra 74HCT04 inverter adatok(forrás:[11.]

7.2.2 MOTORVEZÉRLÉS KAPCSOLÁSI RAJZA



16. ábra Motorvezérlés (forrás: [Saját készítésű kép.]).

A 0 számú adatbit lesz az irány bit ami a forgásirányt fogja megadni.

A 1,2,3,4 bitek fogják a motorvezérléshez szükséges adatot biztosítani

0 bitérték esetén a motor előre 1 bitérték esetén a motor visszafelé, azaz ellentétes irányba fog forogni.

0- Tiltott

1- Engedélyezett

EN- Engedélyező bit

(0-1): Motor számozás

1 Motor esetén: 10-előre forgás (EN1, irány bit)

11-hátrafelé forgás (EN1, irány bit)

2 Motor esetén: 100-előre forgás (EN1, EN2, irány bit)

101-hátrafelé forgás (EN1, EN2, irány bit)

3 Motor esetén: 1000-előre forgás (EN1, EN2, EN3, irány bit)

1001-hátrafelé forgás (EN1, EN2, EN3, irány bit)

4 Motor esetén: 10000-előre forgás (EN1, EN2, EN3, EN4, irány bit)

10001-hátrafelé forgás (EN1, EN2, EN3, EN4, irány bit)

7.2.3 AZ ÁRAMKÖR PARAMÉTEREI

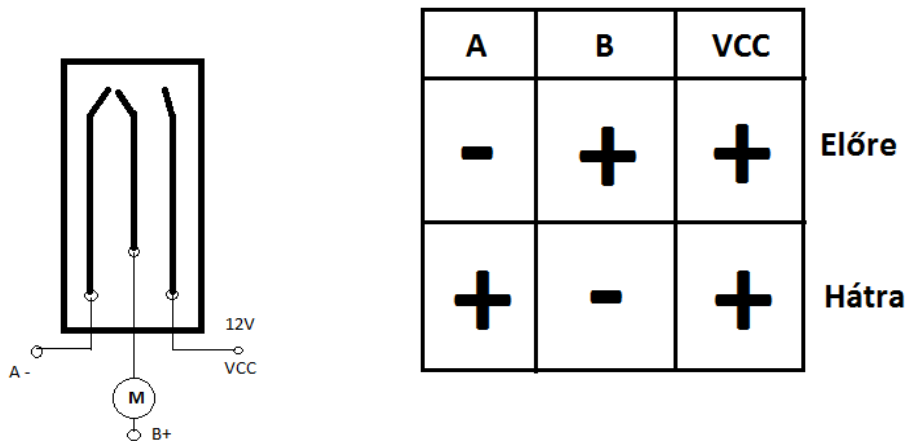
Paraméterek\Motorok	Mértékegység	Motor 1		Motor 2		Motor 3		Motor 4	
		Min	Max	Min	Max	Min	Max	Min	Max
VCC	V	12	12	12	12	12	12	12	12
I Előre	mA	44	50	55	60	30	35	30	50
I Hátra	mA	44	50	90	105	25	32	30	40
Ellenállás	Ohm	32		32		82		82	
Mechanikus paraméterek									
Osztás		1:20		1:20		1:01		1:01	

17. ábra Motorvezérlés (forrás: [Saját készítésű táblázat méréseim alapján.]).

A 17. ábrán feltüntetett mérési adatokra a fejlesztés során volt szükségem. Az áramkör megtervezésekor mértem le. Ezeknek az adatoknak megfelelően tudtam a feladathoz leginkább megfelelő eszközöket kiválasztani. Mint a mérésekből is látszik, vannak kisebb nagyobb eltérések a motorok előre és hátra forgásakor, főleg az emelőkart mozgató motor esetében mikor fölfelé emeli a kart.

7.2.4 VÉGÁLLÁS KAPCSOLÓK

A végállás kapcsolók a robotkar forgás és elmozdulás irányainak a végpontjaiban vannak elhelyezve, megakadályozva a kar olyan pozíciókba való jutását amire szerkezeti felépítése nem ad lehetőséget. A 18. ábrán látható a kapcsoló felépítése majd a 19. ábrán a hozzá tartozó igazságtábla ami szemlélteti működését.



18. ábra Végálláskapcsoló (forrás: [Saját rajz.]). 19. ábra Végálláskapcsoló működése (forrás: [Saját rajz.]).

Az A jelzésű ábra negatív, a B jelzésű ábra pozitív, majd a VCC lábra pedig állandó 12 V-ot adunk. Alap forgás esetén, mikor a kapcsoló nincs benyomva, az AB kört zárja ami azt jelenti, hogy a motor előre forog, viszont ha a kapcsoló zár benyomódik, akkor a B VCC kört zárja megszakítva az AB kört, minek következtében a motor forgása leáll.

Mikor forgásirány váltás következik be, akkor a B és a VCC láb között potenciál különbség lesz. Ezért a motor ellentétes irányba indul el így a kapcsoló újból nyit. A kapcsoló újbóli nyitásakor, vagyis mikor a BA kört zárja a motor továbbra is visszafelé fog forogni.

Igazságtábla magyarázat, folyamat leírás:

A-B kör zárt: a motor előre forog.

B-VCC kör zárt: a motor áll.

Forgásirányváltás.

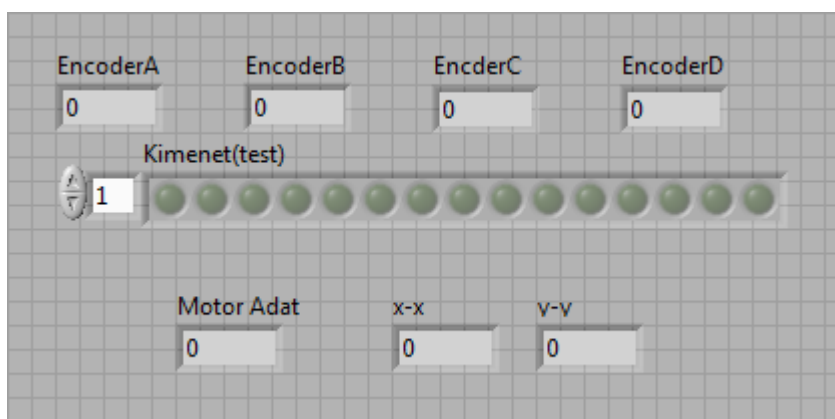
B-VCC kör zár: a motor visszafelé forog.

B-A kör zárt: a motor továbbra is visszafelé forog.

7.2.5 ENKÓDEREK

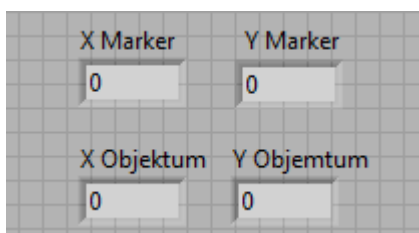
A vezérlés egyik legfontosabb eleme, hiszen DC motorokról lévén szó a számlálók használata elengedhetetlen. Szükség van minden motor mellé egy számlálóra aminek segítségével számon tudjuk tartani, hogy a motor éppen hol jár, azaz egy alap beállított értékhez viszonyítva megtudjuk mondani a pontos helyzetét a karnak. Ennek segítségével lehetőség van vezérelni a kart is. Ha tudjuk hova szeretnénk állítani, akkor ismernünk kell a megadott pozíció paramétereit is amit két koordináta segítségével tudunk megadni.

Ahhoz, hogy távolságot tudjunk számolni, kell egy viszonyítási alap, ezért a munkaterületen van egy fix pont ami ezt a célt szolgálja. Ennek segítségével már pontosan tudunk a munkaterületen különböző helyeket meghatározni. Ezek a távolságok, hogy felhasználhatóak legyenek valahogy mérnünk kell a fordulatokat és azt is tudnunk kell, hogy mikor kezdjen el számolni. A számlálók a párhuzamos port 10,11,12,13 kimeneteleire vannak kötve így a kódban azt kell figyelni, hogy mikor van ezeken a portokon aktivitás. Ekkor egyel növeljük a számláló értékét. A számláló leolvasásával már lehetőség van adott pozícióra állítani a robotkart vezérelve a motorokat a megfelelő enkóder értékekre, de a munkaterületen való elmozdulását még nem ismerjük, ezért egy mérésre van szükség.



20. ábra Számláló állását jelző panel (forrás: [Általam készített képkivágás.]).

Az enkóderek adott értékeit a LabVIEW segítségével állandóan nyomon követhetjük egy panelen. Ezen az ablakon belül van még két kijelző amin a már említett számolás eredménye látható. A kamera által készített képen a viszonyítási pont és a munkadarab közti távolságot a kamera pixelben adja meg. A pixelből történő számláló, értékben való átváltására van szükség X és Y tengelyen ami a tengely körüli, és előre illetve hátra történő mozgást figyel. A jobbra – balra történő mozgás pixelben megadott érték egy hetede ad egy számláló elmozdulás értéket. A 18. ábrán látható x-x, y-y ablakokban az x tengelyen történő elhelyezkedését és az y tengelyen történő elhelyezkedést mutatja az adott munkadarabnak. Erre a két értékre kell álljanak az A és B enkóderek értékei. Az értékek egyezésénél a robotkar a megfelelő pozícióba állítja a tengely körüli és előre hátra mozgó karokat. A kimeneti panelen pedig figyelemmel tudjuk kísérni a párhuzamos porton történő adatmozgást.



21. ábra Paraméterjelző ablak (forrás:[Általam készített képkivágás.]).

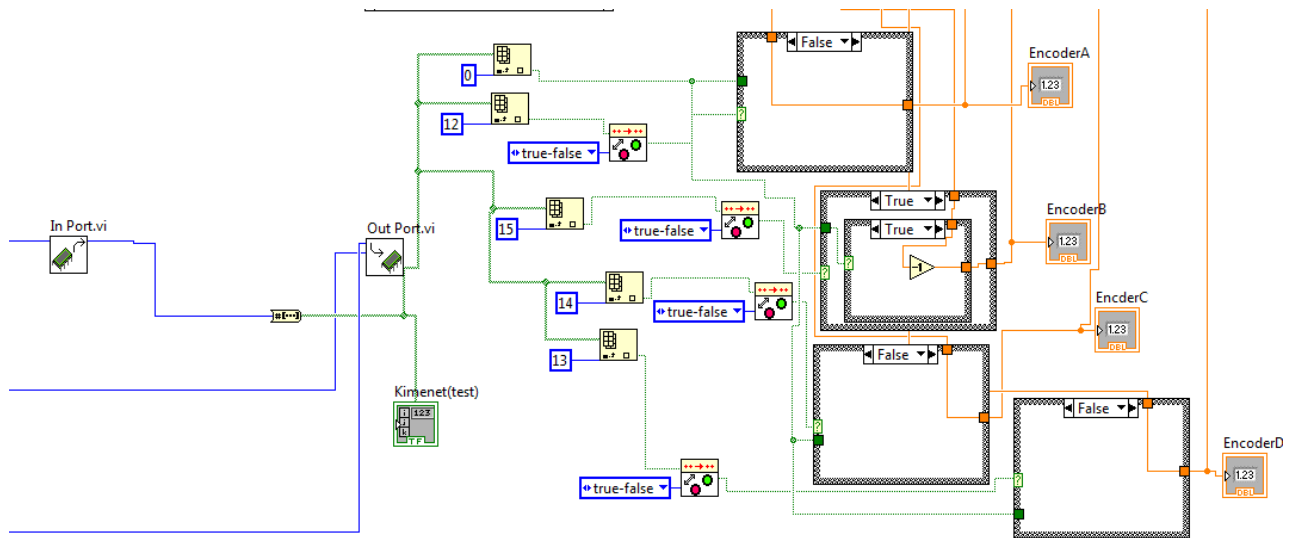
Az 21. ábrán a még át nem számolt pixelben megadott érték kerül kiíratásra. A felső sorban a viszonyítási alapul szolgáló pont pixel értékeinek x és y tengelyen való elhelyezkedésének pontos koordinátái a alsó sorban pedig a munkadarab koordinátáinak értékei.

8. VEZÉRLÉS LABVIEW-VAL

8.1.1 ENKÓDEREK A LABVIEWBAN

A párhuzamos porton keresztül beérkezett adatokat a LabVIEW tömbként kezeli, és ezt a tömböt egy tömbleválogató segítségével megsűrjük. Olyan értékeket engedünk tovább amire éppen szükségünk van, minden egyes tömbből a neki megfelelő adatbiten lévő értéket nézzük csak. Megvizsgáljuk, hogy ezen az adatbiten történt e változás igazából hamisra(lefutó élre). Abban az esetben ha történt változás azt jelenti a motor forog tehát a számláló aktivitását figyelni kell, és a forgásiránynak megfelelően az enkóder értékét eggyel növeljük

vagy csökkentjük. A motor forgás irányát hasonlóan a 1-es párhuzamos port adatbiten lévő értékből egy tömbleválogatót használva figyeljük és ennek segítségével tudja eldönteni, hogy hozzá kell 1-et adnia vagy ki kell vonnia 1-et.



22. ábra Enkóderek vezérlése (forrás: [Általam készített képkivágás.]).

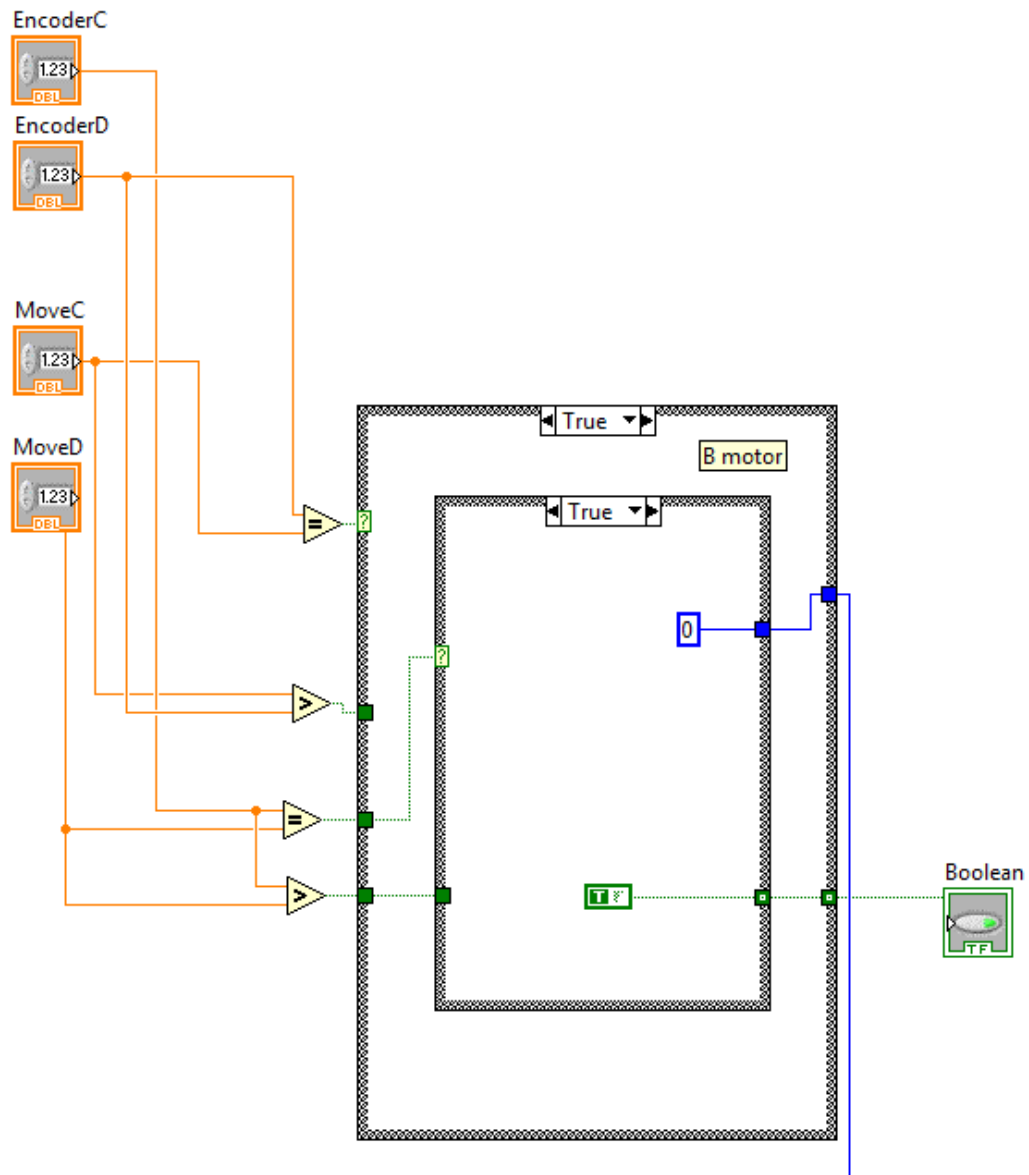
Ezek az értékek, azaz maga a számolás így tarthatna egészen addig amíg a végállás kapcsolók nem állítanak le a motorokat. Ezért szükség van egy vizsgálatra ami folyamatosan nézi, hogy az egyes motorok mikor vannak a megfelelő pozícióban, és ha ott vannak akkor mi történjen, hogyan folytassa a működését. Szükség van tehát egy olyan motorvezérlő részre ami az enkóderek adatai alapján irányítani tudja a motorokat és ezáltal maga a kart.

8.1.2 MOTORVEZÉRLÉS A LABVIEWBAN

Mint ahogyan az enkóderek a LabVIEW-ban részben említettem szükség van tehát egy olyan rendszer felépítésére ami megfelelően vezérli a robotkart. A motorvezérlés több részből tevődik össze. Ha egy folyamatként tekintünk rá, akkor az első lépés az enkóderek adatainak feldolgozása. Ez a folyamat sok döntést igényel, amit a későbbiekben részletesen tárgyalok majd. A számlálók adatait regiszterekben tároljuk és összehasonlítjuk az első, azaz

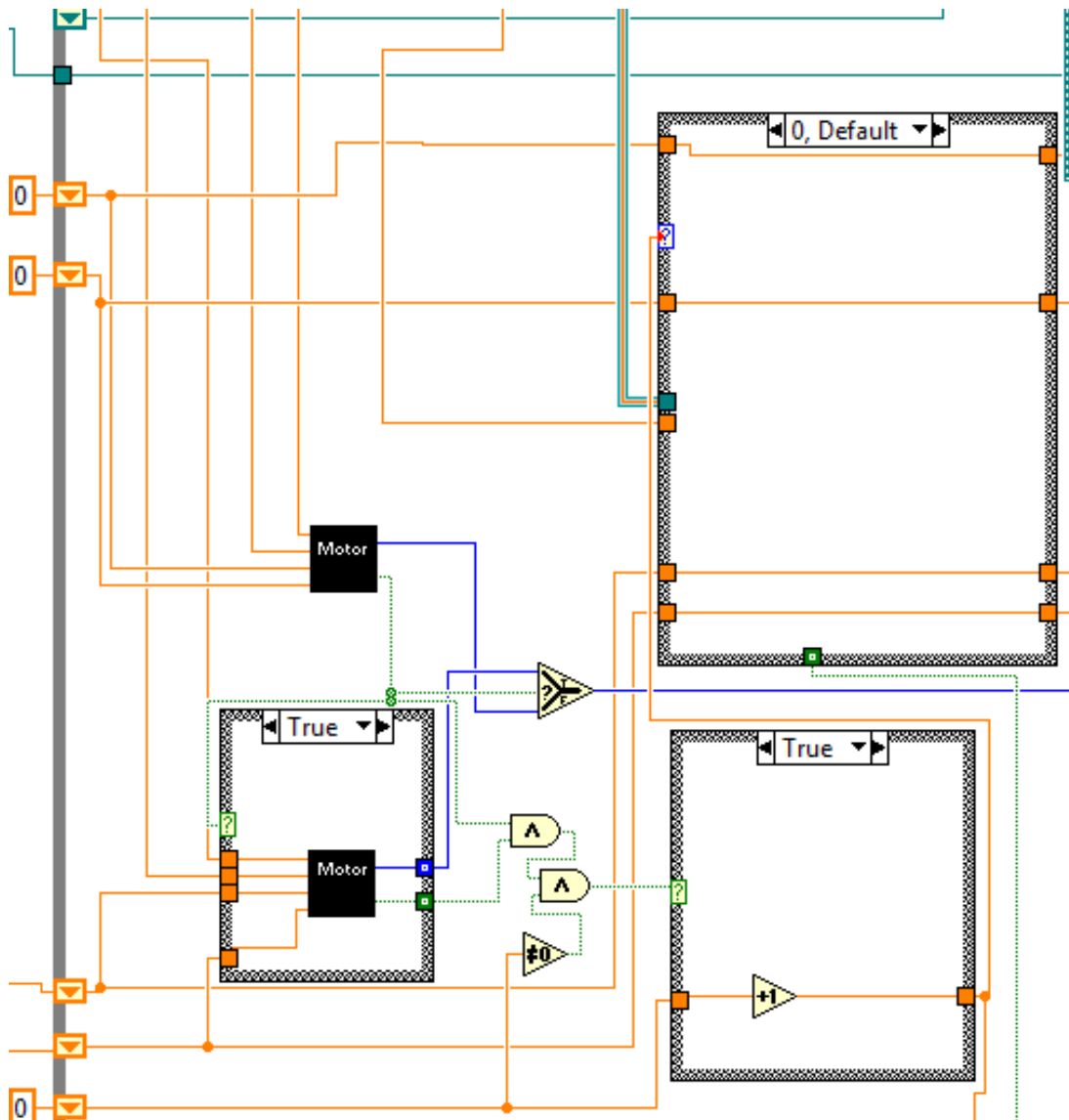
a tengely körüli motor számlálójának értékével amennyiben ez a két érték nem egyezik, akkor a motort a megfelelő irányba meghajtja. Az irány megfelelő felismerését egy másik döntés adja, hiszen a tárolt enkóder adat, ami leírja a munkadarab pozícióját. Összehasonlításkor el tudjuk dönteni, hogy nagyobb vagy kisebb-e a motor számlálójának értékénél. Ennek megfelelően tudjuk, hogy előre vagy hátra kell a motort meghajtani. Mikor a motor számlálójának értéke megegyezik a munkadarab számolt pozíciójának enkóder értékével, a motor a megfelelő irányban van akkor meghívódik egy belső esemény ami a kettes motor vezérléséért felelős. Akárcsak az előző motor esetében a folyamat ugyan az. Amíg a motor enkóder értéke nem egyezik meg a munkadarab számolt enkóder értékével, addig azt a megfelelő irányban hajtja. A harmadik és negyedik motor vezérlése eltérő az előző kettőjéhez képest, hiszen a kar mélységi mozgásához és a fogókar összecukódásához valamint kinyitódásához nincs szükség döntésekre annak pozíciójának meghatározásához. Ezek az értékek állandóak, mivel a munkadarab mindig azonos magasságban van és a megfogás elengedés értéke ennek a robotnak az esetében konstans.

A következő folyamat ami megkapja a vezérlést a már megfogott munkadarabbal a megfelelő pozícióba menni,ami szintén egy adott konstans paraméterű hely. A valóságban ez lehet egy futószalag vagy bármilyen más terület. A következő esemény a munkadarab megfelelő helyre való letevése majd a kar újabb munkavégzéshez szükséges pozíció felvétele. Minden egyes folyamat lefutása után meghívódik a következő esemény, mindaddig amíg az utolsó esemény végezetével egy megszakítás nem hívódik meg. Ekkor átadódik a vezérlés a képfeldolgozó eseménynek, ahol egy újabb kép készül feldolgozásra és kezdődik az egész folyamat előlről. Ez a folyamatos eseményhívásos rendszer mindig egy előbbi esemény befejezésével lép a következőre.



23. ábra Motorvezérlés (forrás:[Általam készített képkivágás.]).

A 23. ábrán látható a fentiekben leírt motorvezérlés. Az EncoderC és EncoderD a kamera által kiszámolt munkadarab pozíciójának enkóder értéke amit összehasonlít a MoveC és MoveD motorok enkóder értékeivel. Négy vizsgálat megy végbe, két egyenlőséget ellenőrző, hogy a motorok a megfelelő helyen állnak-e, és két logikai vizsgálat ami eldönti a forgásirányt.



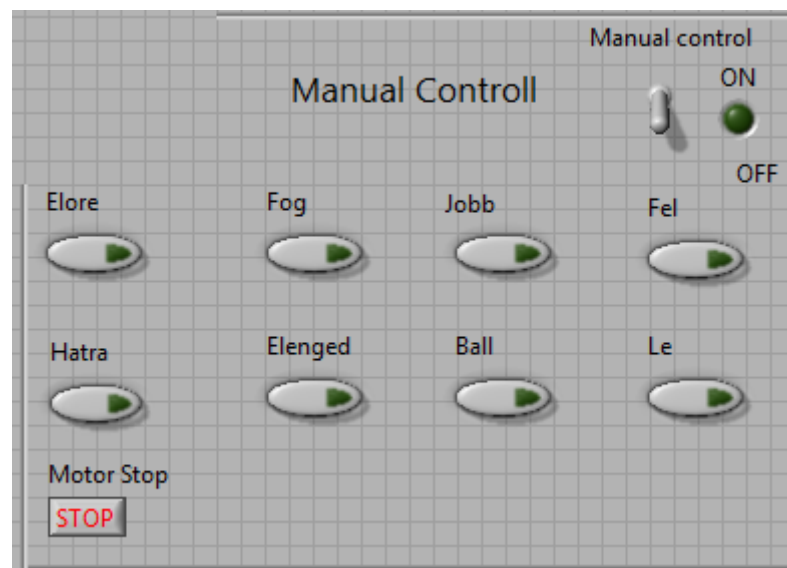
24. ábra Motorvezérlési folyamat (forrás:[Általam készített képkivágás.]).

Az 24. ábrán látható a motorvezérlés felépítése az eseményhívásokkal, valamint egy olyan esemény ami ezt a lépegetést végzi a folyamatban. Minden egyes végrehajtott esemény után egyel növeli a folyamatban az események sorszámát így jön létre az események 10 eseményei közötti lépegetés.

8.1.3 KÉZI VEZÉRLÉS

A robotkar vezérlésében van még egy teljesen különálló rész, a manuális vezérlés. Ez a programrész lehetővé teszi a manipulátor vezérlését nyomógombok által. Egy kapcsoló segítségével tudunk váltani, kézi vezérlés és automata vezérlés között. Ez a lehetőség megoldást jelent olyan munkadarabok elérése esetén, amelyek esetleg nem azonos síkban vannak a beállított szinttel amin a munkadarabok vannak.

Ennél a vezérlés formánál elektronikai szempontból minden ugyan úgy működik, mint a kamera által vezérelt önműködő rendszer esetén, kivéve, hogy itt a vezérlő utasítást nem a kamera adja, hanem az felhasználó. Hasznos felhasználási funkció, esetleges meghibásodás vagy leállás esetén a kar vezérelhető.

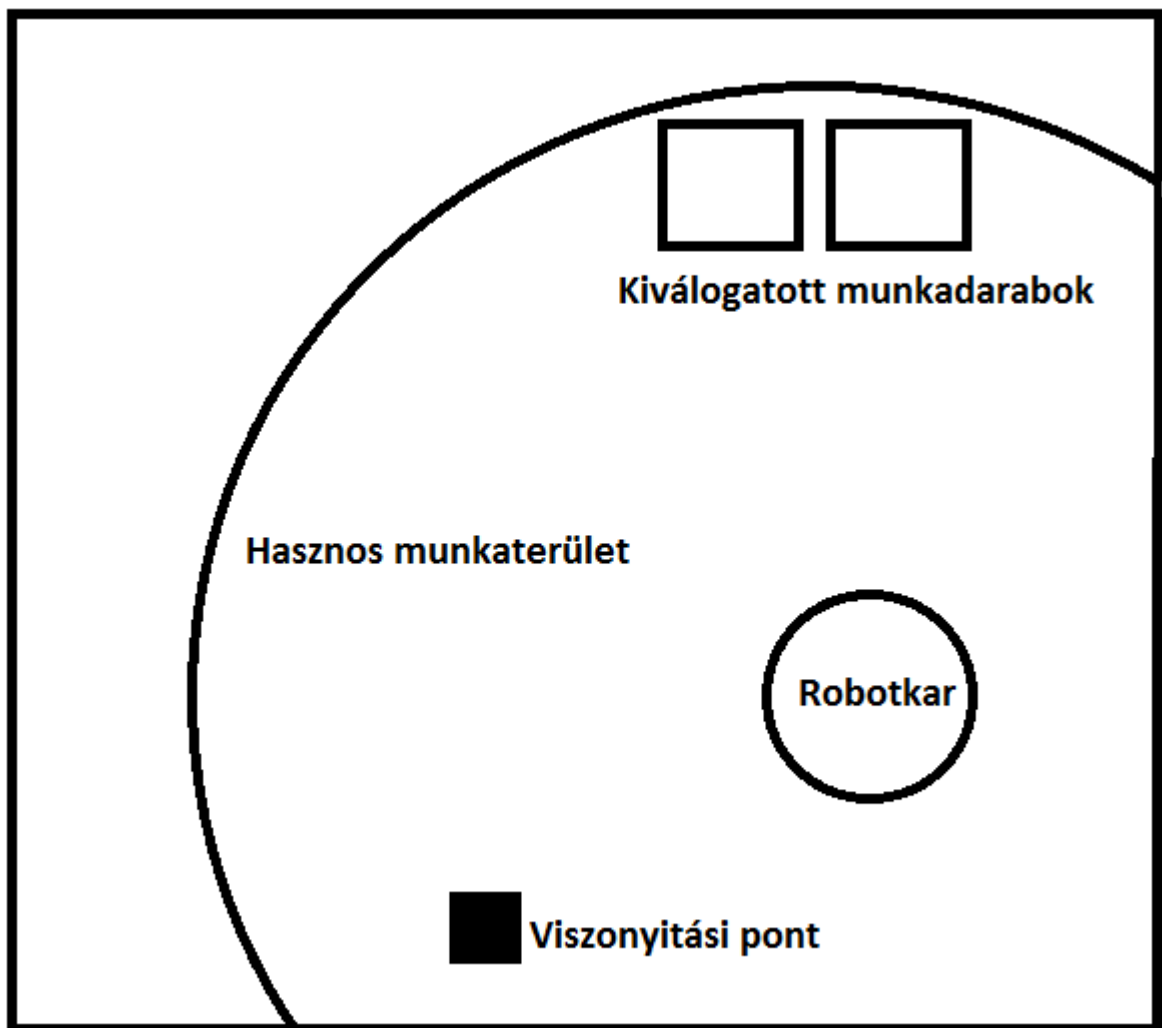


25. ábra Kézi vezérlési felület (forrás:[Általam készített képkivágás.]).

A 25. ábrán látható az a vezérlő felület ami tartalmazza a funkciógombokat minek segítségével vezérelni tudjuk a kart. Továbbá egy kapcsoló segítségével átválthatunk kézi vagy önműködő rendszerre, ami generál egy megszakítást és akár egy aktív folyamat közben is átvehetjük a vezérlést a kar felett. Ki vagy bekapcsolt állapotáról egy indikátor ad jelzést. A

panel tartalmaz még egy STOP gombot is ami megszakít minden aktív munkamenetet, leállít minden motort.

9. MUNKATERÜLET



26. ábra Munkaterület (forrás:[Saját rajz.]).

A 26. ábrán látható a robotkar és a hozzá tartozó hasznos munkaterület, ami egy íves pályát ír le. A robotkarnak vannak korlátozásai, szerkezeti felépítéséből adódóan. Látható

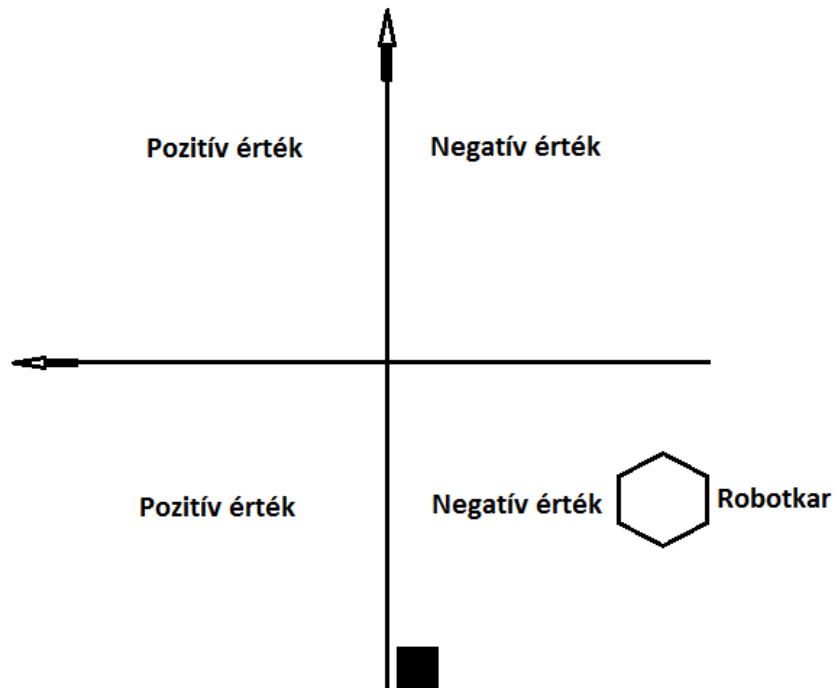
továbbá a viszonyítási pont amihez mérten tudja a kamera meghatározni a munkadarab helyzetét. A kamera a hasznos munkaterület közepére van fókuszálva, kb. 60 cm magasságban.

Továbbá láthatóak azon területek ahova a már úgynevezett feldolgozott munkadarab kerül.

10. DÖNTÉSEK, KALKULÁCIÓK

A programozás során, lévén ez egy olyan feladat, melynek eredménye egy olyan eszköz, ami változó tényezőktől függ, rengeteg döntést kellett alkalmazni. Nem egy olyan eszköz, mely mindig adott pozícióról adott pozícióra mozog, hanem adott helyzettől függően mindig az aktuálisra. Legtöbb felhasznált döntésforma a motor elmozdulásának irányából, valamint a számlálók működéséből fakadt. A döntések nagyon fontos szerepet játszanak a robotkar megfelelő működtetésében, hiszen hiányukkal vagy rossz működésükkel nem tudnánk megfelelő és biztonságos működést biztosítani.

Első ilyen fontosabb terület a kamera által felmért munkadarab pozíciójának pontos kiszámítása, valamint a tükrözés problémájának megoldása. A viszonyítási alapul szolgáló pont a koordinátarendszer 0 pontja majd x és y koordináták pozitív síkban történő kezelése nem okozott gondot, viszont ha a munkadarab ettől a ponttól hátrább esik, akkor az már negatív síkban van mint ahogyan az 27. ábrán is látszik. Itt az érték abszolút értékével kellett számolni máskülönben nem a munkadarab pozíciójára menne a kar, hanem a pozitív síkjában lévő tükrözött pozícióra.

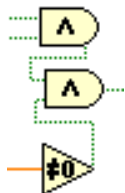


27. ábra Munkaterület elérhetőségei (forrás:[Saját rajz.]).

Egy másik fontos számolást maga után vonó döntés, a motorok megfelelő irányba való elmozdulását vezérlő döntéshozatal. A kamera által meghatározott munkadarab pozíciója és a motorok enkóder pozícióinak az összevetését követően, a motorokat a megfelelő irányban kell elindítani. Itt történik egy értékegyeztetés, nem egyezés esetén egy logikai vizsgálat szükséges, ami eldönti az irányt, mint ahogy ez az 28. ábrán látható.



Továbbá kell még egy logikai döntés is annak meghatározására, hogy egy folyamat befejeződött-e már vagy még folyamatban van, hiszen ha még egy motor aktív, addig más motor nem kaphatja meg a vezérlést. Ez a kapu, ha igaz értéket ad eredményül, akkor megkapja a vezérlést a következő motor. Miután az is keresztülment ugyanezekben a lépésekben újabb igaz érték után szintén továbblépünk az eseményhívásokban.

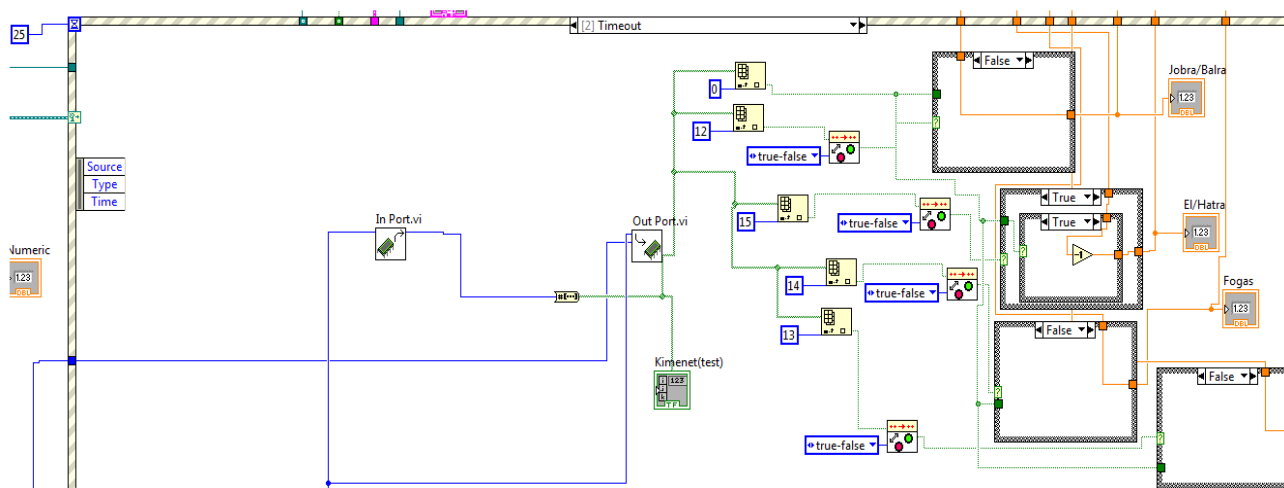


28. ábra Motorvezérlési döntések (forrás:[: Általam készített képkivágás.]).

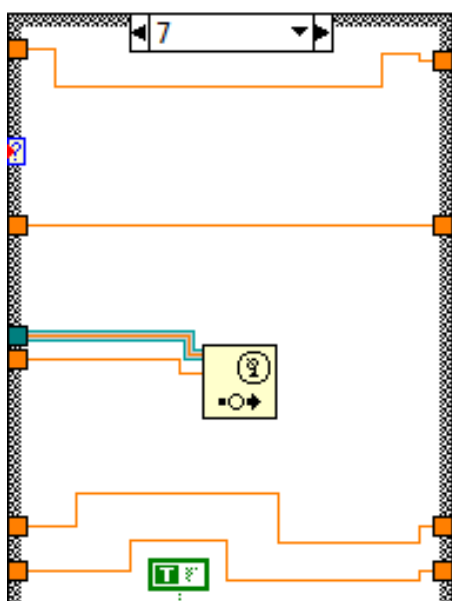
Egy másik nagyon fontos döntési egység a kézi irányításnál, a motorvezérlés. Ebben az esetben több logikai és egyeztető lépéseken kell

11. MEGSZAKÍTÁSOK

Megszakításokról beszélve az első legfontosabb megszakítás az amelyik az eseményvezérlést szakítja meg, amiben a párhuzamos porton történő adatátvitel történik. A párhuzamos portról az adatokat pontosabban a számláló adatait 25 milliszekundumonként nézzük, azaz 25 milliszekundumonként egy megszakítás történik, ez elegendő ahhoz, hogy a számlálók adatai átmenjenek a LabVIEW-ba de elég rövid ahhoz, hogy ne számoljon két fordulatot egynek.



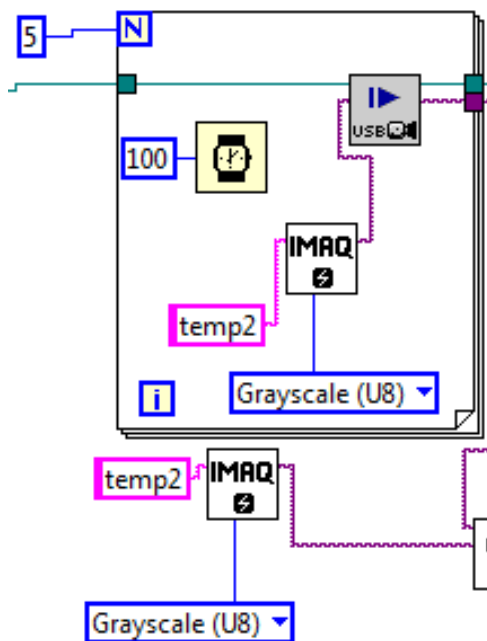
30. ábra 25 milliszekundumonként meghívódó folyamat (forrás:[Általam készített képkivágás.]).



31. ábra Motorvezérlés megszakítása és vezérlés átadása a képfeldolgozó rendszernek (forrás:[Általam készített képkivágás.]).

Egy másik megszakítás a motorvezérlésnél lévő eseményhívási lánc végén van, mikor minden motor a megfelelő pozícióba érkezett és elvégezte a munkadarabbal aktuálisan megadott feladatát. Ekkor egy megszakítás hívódik meg minek eredménye, hogy újból

aktiválódik a képfeldolgozó rész és kezdődik az egész folyamat előlről.



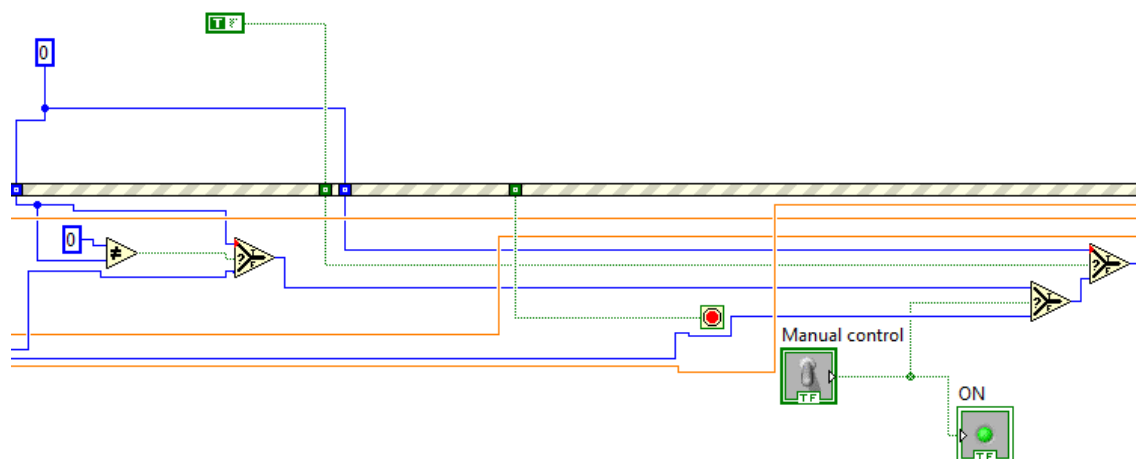
32. ábra Képfeldolgozó rendszer megszakításai (forrás:[Általam készített képkivágás.]).

A képfeldolgozó rendszer készít egy újabb sorozat képet, pontosan 5 darabot, ahogyan a 32. ábrán is látszik, majd itt is meghívódik egy megszakítás és az utolsó kép kerül feldolgozásra. Itt egy 100 milli szekundumos megszakításra volt szükség a kamera fényegyensúly állítása miatt, hogy 100% valós képet kapjunk a munkaterületről és a rajta elhelyezkedő munkadarabokról.

Mint minden robot esetében és ez itt sincs másként, ahogy a 33. ábrán is látszik van egy teljes megszakítás tulajdonsággal bíró gomb. Ennek a STOP gombnak a megnyomásával megszakad az éppen aktuális és ezzel együtt a teljes munkamenet is, legyen bármilyen helyzetben is a kar. Ez a megszakítás automatikus és kézi vezérlés esetén is működik.

A gomb megnyomásakor a motorvezérlő vizsgálatokhoz becsatlakozó stop gomb aktív vizsgálatához, az eddigi hamis érték helyére, igazat vezetünk. Ekkor a vizsgálat eredménye az igaz részben található értéket veszi figyelembe, ami ebben az esetben nem más mint a 0, tehát

motorok megállnak.



33. ábra Teljes megszakítás vizsgálata (forrás:[Általam készített képkivágás.]).

12. ÁLTALÁNOS TAPASZTALATOK

Az alábbi felsorolás a feladat megvalósítása során összegyűlt tapasztalataimnak összegzése a Fishertechnik ipari robot és egyéb általam épített eszközöket használva.

- A Fishertechnik készletek árban és minőségben is megfelelőek mindazon célokra, amire szánták fejlesztőik.
- Ugyan csak ezen készlet mellett szól, hogy könnyedén átalakítható más ipari robotná köszönhető jól megtervezett aprólékos alkatrészeinek, melyek mindegyike műanyag borítású acéllapka. Ennek köszönhetően teherbíróbb és flexibilisebb.
- Sok alkatrésznek van kettős felhasználhatósága.
- Többfajta motort tartalmaz, aminek hasznát vettem, mert az emelő kar súlyának emelésekor szükség van a nagyobb teljesítményre.
- A példaként megépített ipari robotok nagyon jó struktúrájúak és jól használhatóak. A feladatok során egyedül a több részből összeállított csigánál lépett fel egy apróbb probléma, melynek következtében bizonyos idő elteltével akár 6 számláló egységet is téved.
- A számlálók és végállás kapcsolók elhelyezése nagyon jól van megoldva akárcsak az ezekhez tartozó jelentős mennyiségű kábelek elvezetése a beépíthető kábeltartóknak köszönhetően.
- A számlálók mérési pontosságát tekintve hagy maga után kívánni valót mivel egy teljes fordulat csak 4 részre van felosztva. Ennek eredménye 3 motor esetén ami 3 irányt határoz meg akár több cm eltérést is eredményezhet.

13. ÖSSZEFOGLALÁS

Szakdolgozatom célkitűzését megfogalmaztam a Feladatról pontban de röviden ismertetve annak lényegi tartalmát, egy olyan manipulátor programozása, amit egy kamera vezérel és a megépített vezérlés segítségével irányítva a robotkart, tudunk, különböző pozíciókban lévő anyagot mozgatni.

A célkitűzés rám eső részét, véleményem szerint teljes mértékben sikerült megvalósítanom.

A rendelkezésre álló Fischertechnik ipari 3D robotkart vezérlő elektronikát építettem, melynek segítségével megvalósult a motorok, érzékelők és számlálók vezérlése, miközben ez a vezérlő panel végig kapcsolatban áll a számítógéppel. A kameravezérlés és képfeldolgozás segítségével olyan programot sikerült írni, mely automatikusan a kamerával együttműködve ki tudta számolni a munkadarabok helyét adott munkaterületen, majd a megfelelő adatokat továbbítva a vezérlőpanelnek a kart is a megfelelő helyre tudja irányítani. A munkadarab elszállítását követően egy vizsgálat hajtódik végre ami megnézi, hogy van e feldolgozandó elem. Abban az esetben, ha van akkor újraindul a folyamat és megkeresi a pozíciót, majd úgymond feldolgozza, viszont ha nem talál feldolgozandó elemet, akkor visszaáll a kezdeti pozícióba. Sikerült még egy olyan programot is írni aminek segítségével manuálisan is vezérelni lehet a kart különböző feladatok elvégzéséhez.

Összességében sikerült megvalósítani a teljesen automatikusan működő ipari rendszert, amit elterveztem.

Dolgozatomban próbáltam általános információkkal szolgálni a robotok világáról, valamint a konkrét feladatról olyan adatokat leírásokat készíteni, melyből kitűnnek a felmerülő problémák és azok megoldásai.

Elmondhatom, hogy a Fischertechnik készletek és megépített ipari robotok széles körben felhasználhatóak és érdemes is kihasználni az általa nyújtott lehetőségeket mivel korunk meghatározó részét képezik a robotok. A világ számos pontján különböző fontos

feladatokat látnak el, az egyre bővülő elvárásoknak eleget tevően. Egyre több cég foglalkozik a robotok készítésével, főleg háztartási gépek forgalmazásával.

Személy szerint azt tudom elmondani, hogy mindenképp élményektől telített egy szakterület és biztos vagyok benne, hogy minden érdeklődőt magával ragad a robotok világa.

Remélem, sikerült felhívni minden érdeklődő figyelmét, aki hasonló témában szeretne elmerülni és kamatoztatni tudását. Meglepően sok lehetőséget tár egy programozó elé egy ilyen jellegű feladat megvalósítása.

14. IRODALOMJEGYZÉK

Minden felhasznált és letöltött anyag 2010.11.15 én még felhasznált formában és akkori helyén volt található.

[1.] Ismeretlen szerző: Robotokról

<http://itl7.elte.hu/hlabdb/robot/robkie.html>

[2.] Ismeretlen szerző: Robottechnika

<http://www.oveges.hu/technikus/lacik/automatika/robottechnika.doc>

[3.] Asimov Isac. (1950). *Én a robot*. Gnome Press.

[4.] Ismeretlen szerző: Ipari robotok.

<http://www.mestersegesintelligencia.hu/doc/ipari%20robotok.php>

[5.] NI:NI CompactRIO a robotika: Hands-On tanulás

<http://zone.ni.com/devzone/cda/tut/p/id/10455>

[6.] Ismeretlen szerző: Személyes és Oktatási robotika

<http://www.robotica-personal.es/2010/02/fischertechnik-en-ro-botica-entendiendo.html>

[7.] Fischertechnik ROBOT KIT

<http://www.robotmatrix.org/fischertechnik-robot.htm>

[8.] Fischertechnik 3D Robot

<http://www.robotmatrix.org/Fischertechnik-3D-robot.htm>

[9.] Ismeretlen szerző: L239D Motorvezérlő IC.

<http://wiki.kn.vutbr.cz/robot/index.cgi?AVR%20regul%C3%A1tor%20dvou%20motork%C5%AF%20-%20hw>

[10.] Hivatalos adatlap: L293D Motorvezérlő IC

<http://www.st.com/stonline/books/pdf/docs/1330.pdf>

[11.] Hivatalos adatlap: 74HC04 inverter

http://www.nxp.com/documents/data_sheet/74HC_HCT04.pdf

15. FELHASZNÁLT IRODALOM

Minden felhasznált és letöltött anyag 2010.11.15 én még felhasznált formában és akkori helyén volt található.

- **Bevezetés a robotok világába**

<http://itl7.elte.hu/hlabdb/robot/robkie.html>

- **Industry Robots II készlet információk**

http://www.res.hu/roboshop/FischerTechnik/Computing_IndustryRobots2_h.htm

- **LabVIEW Tutoriál**

http://e-oktat.pmmf.hu/jelek_iii_fej_2_2

- **LabVIEW információk**

<http://cnx.org/content/m13455/latest/>

<http://www.ni.com/vision/vdm.htm>

- **LabVIEW információk**

<http://cnx.org/content/m13455/latest/>

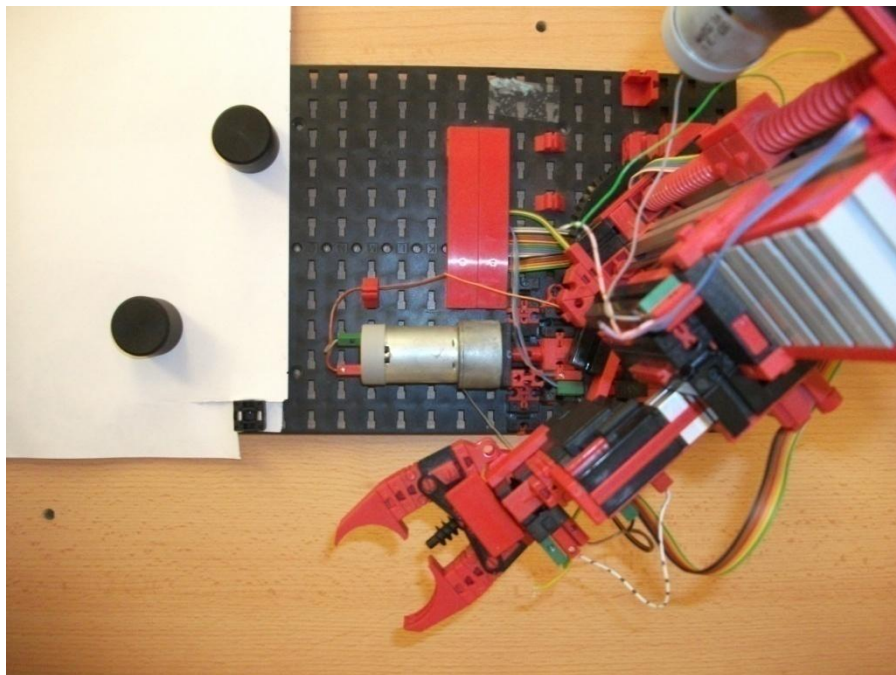
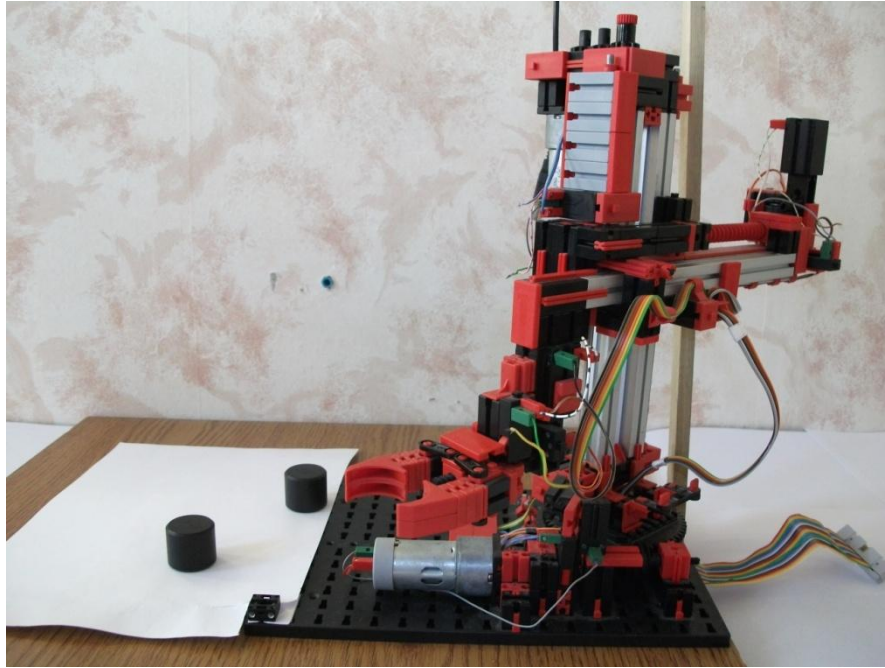
- **Ipari robotok**

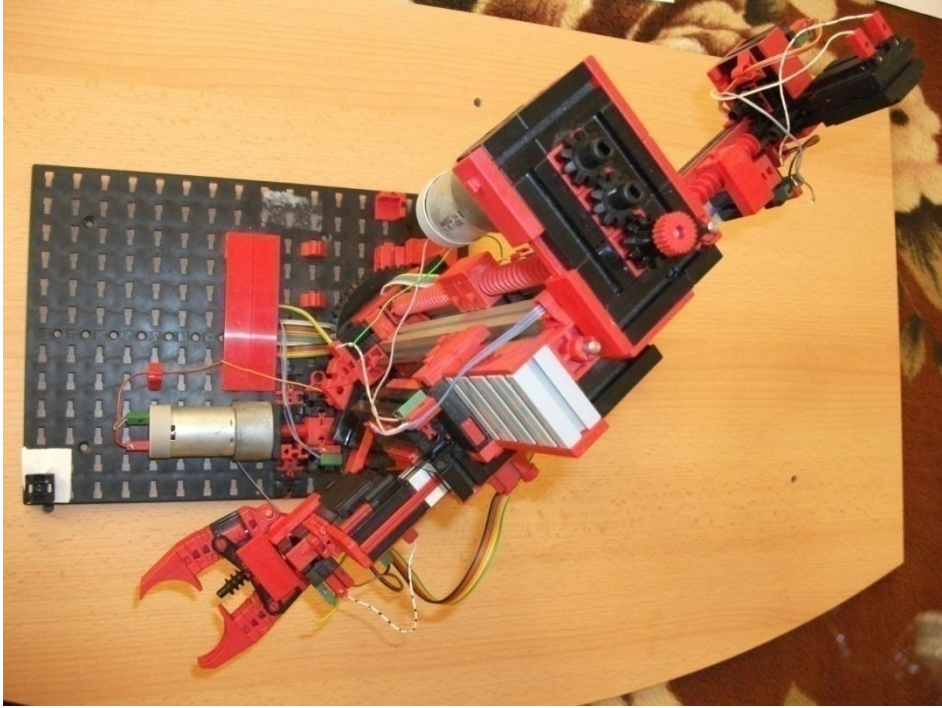
<http://www.mestersegesintelligencia.hu/doc/ipari%20robotok.php>

- **Ismeretlen szerző: Robottechnika**

<http://www.oveges.hu/technikus/lacik/automatika/robottechnika.doc>

16. FÜGGELÉK





17. KÖSZÖNETNYILVÁNÍTÁS

Köszönetet szeretnék mondani Dr. Husi Gézának, a Debreceni Egyetem Műszaki Főiskolai Kar tanszékvezetői docensének a témaválasztásban nyújtott segítségével és tanácsaiért valamint a munkám során felmerülő problémák megoldásához vezető ötletekért, és a dolgozatom készítése során adott útmutatásokért.