T.C.

KARABÜK ÜNİVERSİTESİ MÜHENDİSLİK FAKÜLTESİ
MEKATRONİK MÜHENDİSLİĞİ

3 EKSENLİ ROBOT KOL

LİSANS BİTİRME TEZİ

Hazırlayan

Fatih Can KAYA 2013010225033

Oğuzhan KURT 2012010225072

Tez Danışmanı

Dr. Öğr. Üyesi Can Bülent FİDAN

KARABÜK-2019

KABUL VE ONAY

Fatih Can Kaya.....tarafından hazırlanan "3 Eksenli Robot Kol....." başlıklı bu tezin Lisans Bitirme Tezi olarak uygun olduğunu onaylarım. 18../06../2019..

Tez Danışmanı

Dr. Öğr. Üyesi Can Bülent FİDAN

Bu çalışma, jürimiz tarafından oy birliği / oy çokluğu ile Mekatronik Mühendisliği Anabilim Dalında Lisans Bitirme Tezi olarak kabul edilmiştir. 18../06../2019.

Tez Jürisi

Başkan: Dr. Öğr. Üyesi Can Bülent FİDAN .....

Üye : Prof. Dr. Gökhan Göktaş .....

Üye : Arş. Gör. Ömer KARTALTEPE .....

KBÜ Mühendislik Fakültesi, Mekatronik Mühendisliği Mezuniyet Komisyonu ve Bölüm Başkanlığı bu tezi Lisans Bitirme Tezi olarak onamıştır. 18../06../2019.

Doç.Dr. İbrahim ÇAYIROĞLU

Mekatronik Müh. Bölüm Bşk.

TEŞEKKÜR

Bütün eğitim öğretim hayatımda yanımda olan aileme, mekatronik bölümünü bana sevdirip ufkumu genişleten hocalarıma ve lisans tez çalışmamın her aşamasında destek ve önerilerini eksik etmeyen kıymetli hocam Dr. Öğr. Üyesi Can Bülent FİDAN'a teşekkürlerimi borç bilirim.

Fatih Can KAYA

ÖZET

Günümüzde, robot kollar endüstriyel üretim tesislerinin temel taşı haline gelmiş olup hidrolik, pnömatik ve elektrikli, çeşitli motor sistemleriyle üretilen, endüstrinin birçok alanında kullanılmakta olan bir yapıdır. Robotik kol, programlanabilir, mekanik parçaların bütünü ya da kompleks bir robotun bir parçası olarak nitelendirebilir. İnsan gücünü en aza indirerek, hata payı oranını azaltıp üretim miktarını üst seviyelere çıkararak, günümüz teknolojisine önemli derecede katkı sağlamaktadır

Bu proje çalışmasında ADUC 842 kullanılarak 3 eksenli robot kol sistemi içinde kullanılan çeşitli motorların elle veya otomatik kontrolle aldığı bilgilerle iş yapabilmesi sağlanmıştır.

Anahtar Kelimeler: DC Motor, Aduc 842, Sensörler, Servo motor-Sürücü

ABSTRACT

Nowadays, robotic arms produced with hydraulic pneumatic and electric motor systems have become the cornerstone of industrial production facilities and are used in many areas of industry. The robotic arm may describe the whole of the programmable mechanical parts or as part of a complex robot. By minimizing human power, reducing human error in production and increasing the amount of mass production, it contributes significantly to today's technology.

In this project, using ADUC 842, various motors used in 3-axis robot arm system can work with information obtained by manual or automatic control.

Key Words: DC Motor, Aduc 842, Sensors, Servo Drive

İçindekiler

Sayfa

GİRİŞ ... 3
BÖLÜM 1. ... 4
1.1.ROBOTLARIN TARİHİ VE ÖNEMİ ... 4
1.2.ROBOT SİSTEMLERİ ... 5
1.2.1.Koordinat sistemlerine göre robot sistemleri: ... 5
1.2.2.Kullanılan tahrik sistemleri: ... 5
BÖLÜM 2. ... 6
2.1.MATLAB ... 6
2.1.1.İleri (Düz) Kinematik Hesaplama ... 6
2.1.2.Geri (Ters) Kinematik Hesaplama ... 6
2.2.SIMULINK VE GUIDE ... 7
2.3.8051 Serisi Mikrodenetleyici ... 9
2.3.1.8051 serisi mikrodenetleyicilerin genel özellikleri: ... 9
2.3.2.Temel Mimari Yapısı ... 10
2.3.3.Bacak Konfigürasyonu ... 10
BÖLÜM 3. ... 13
MATERYAL METOD ... 13
3.1.Kullanılan Malzemeler: ... 13
3.1.1.ADUC842 Mikrodenetleyici ... 13
3.1.2.Step Motor ... 14
3.1.3.ULN2003A Motor Sürücü ... 14
3.1.4.Mikro servo Sg90 (9G) ... 15
3.1.5.Fzr rulman (51117) ... 15
3.1.6.Tasarımı yapılan parçalar ... 16
3.1.7.Robotun Görünümü ... 17
3.2.Bağlantı ve Devre Şemaları ... 18
3.3.Kullanılan Kodlar ... 20
3.3.1.MAIN ... 20
3.3.1.1.Ctype ... 28
3.3.1.2.Intrins ... 28
3.3.1.3.Lib842 ... 30
3.3.1.4.Reg51 ... 32
3.3.1.5.Stdio ... 35

SONUÇLAR 37
KAYNAKLAR 38
ÖZGEÇMİŞ 39

GİRİŞ

Gelişen teknoloji ile bilgisayar kontrollü pnömatik devreler kurup, bunları yazılımlarla kontrol edebilir hale gelinmiştir [1]. Elektronik devreler ile mekanik sistemlerin kontrolünün sağlanabilmesi sıfır hata ilkesi ile üretim yapan fabrikalar için insan hatasını ortadan kaldırma konusunda fayda sağlamıştır. Teknoloji her zaman insanların hayatlarını kolaylaştırmak için gelişmiştir. Ağır sanayilerde zor şartlar altında çalışan insanların yerlerini alan bu robotlar sayesinde işçilerin ağır ve tehlikeli işleri bilgisayarlar ile kontrol edebilmesine imkan sağlamıştır [2]. Bu çalışmamda üç eksenli robot kolu tasarımı ve kodlaması ile ilgili yaptığım deneysel verilerimi paylaşacağım.

BÖLÜM 1.

1.1.ROBOTLARIN TARİHİ VE ÖNEMİ

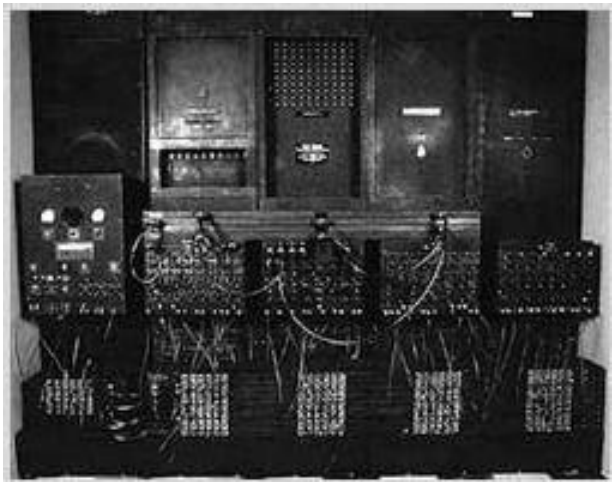
Robotlar denilince herkesin aklına elektronik olarak kontrol edilen makinalar gelmektedir. Ancak robotların çalışma prensipleri insanların hayatlarını kolaylaştırıp, ağır işleri daha az insan gücü harcayarak yapmak olarak da söyleyebiliriz. Bu anlamda çalışan milattan önce yapılmış yapılar ile günümüze gelene kadar "Robot" teriminin gelişimini inceleyebiliriz.

M.Ö. 270 yılında yunan matematikçi ktesibios ilk su saatini inşa etmiştir.

M.Ö. 100 yıllarında İskenderiye de otomatik açılan tapınak kapıları inşa edilmiştir.

1801: Joseph Jacquard delikli kart kullanarak çalıştırılan otomatik dokuma makinesi geliştirdi.

1946: J. Presper Eckert ve John Mauchly, Pennsylvania Üniversitesi'nde ilk elektronik bilgisayar olarak bilinen "ENIAC - Electrical Numerical Integrator and Computer" isimli (Şekil 2.1) bilgisayarı geliştirdi.



Şekil 2.1 ENIAC bilgisayarı

1966: Nokta kaynağı yapan ilk robotlar üretildi.

1980: Amerika Yapay Zeka Birliği (American Association For Artificial Intelligence-AAAI) kuruldu.

1993-1994: Önceki robotlara göre ucuz maliyetli ERRATIC ve PIONEER 1 isimli gezer robotlar üretildi

Robotların gelişimlerini yukarıda verilen kronolojik sıralamadan tahmin edebiliriz. Günümüzde seri üretim fabrikaları için vazgeçilmez noktada olan robotların en kötü şöhreti insan faktörünü devre dışı bırakıp kendi kendine düşünebilen robot üretilmesi fikridir. İnsanlık adına çok fazla faydası olan bu makinelerin günlük yaşantımıza sağladığı faydalar göz ardı edilemeyecek hale gelmiştir.

1.2.ROBOT SİSTEMLERİ

Otomasyon uygulamalarının en önemli unsurları robotlardır. Malzemelerin ve parçaların hareket ettirilebilmesi için tasarlanan çok fonksiyonlu ve programlanabilir manipülatör olarak kısaca robotun özetini yapabiliriz.

Günümüzde sanayi tipi robotlar kaynak yapabilme, cisim tutabilme, döküm yapabilme, kalite kontrol ve boya yapabilme gibi işlerde kullanılmaktadır. İnsan hatalarından kaynaklanabilecek hata oranlarını düşürüp, daha hızlı üretim ile maliyetlerin düşürülmesinde robotların kullanımı artık herkes için zorunlu bir hal almıştır [3].

1.2.1.Koordinat sistemlerine göre robot sistemleri:

- Kartezyen robot kolları: X, Y ve Z eksenlerinde doğrusal hareket eden robotlardır.
- Silindirik robot kolları: Kendi ekseni etrafında dönebilen mafsal üzerine yerleştirilen Kartezyen robotlardır.
- Küresel robot kolları: Bel omuz ve dirsek mafsallarından oluşan silindirik robotlarla aynı hareket eksenlerine sahip robotlardır.
- Scara robot kolları: İki eklem yerinde elektrik motoru ve aşağı yukarı hareket edebilen pnömatik koldan oluşan robotlardır.
- Mafsallı robot kolları: İnsan koluna en yakın robottur. Her ekleminde üç eksende hareket sağlayan motorlar ile kontrol edilirler.

1.2.2.Kullanılan tahrik sistemleri:

Pnömatik tahrik sistemleri: Hava basıncı ile tahrik sağlayan robotlardır.

Hidrolik tahrik sistemleri: Sıvı akışkan basıncı ile tahrik sağlayan robotlardır.

Elektrikli tahrik sistemleri: Step motor, DC servo veya AC servo motordan tahrik sağlayan robotlardır.

BÖLÜM 2.

2.1.MATLAB

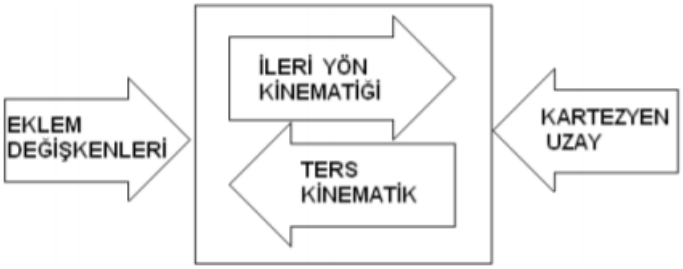
Matlab ilk olarak 1985 yılında Cleve Barry Moler tarafından geliştirilmiş bir paket programdır. Günümüzde mühendislik, matematik, fizik ve daha birçok alanda kullanıcılara kolaylıklar sunmaktadır. Sayısal hesaplamalar, denklem çözümleri, grafik işlemleri, istatistiksel incelemeler ve daha birçok matematiksel işlemler matlabda çok kolay bir Şekilde sonuçlandırılabilir. Bunların yanı sıra Matlab içerisinde bulundurduğu araçlar sayesinde, farklı dallarda da işlem yapabilmemize olanak tanır. Örneğin Simulink ile simülasyonlar yapabileceğiniz gibi yapay sinir ağları aracını kullanarak, yapar sinir ağları ile uygulamalar yapalabilir.

2.1.1.İleri (Düz) Kinematik Hesaplama

Bir robot ana çerçevesinden araç çerçevesine doğru birbirine prizmatik veya döner eklemlerle bağlanmış seri uzuvlardan oluşur. İki uzuv arasındaki ilişki bir homojen dönüşüm matrisiyle açıklanır. Eklem dönüşüm matrislerinin ard arda çarpılmasıyla ana çerçeve ile araç çerçeve arasındaki ilişki tanımlanır. Bu ilişki manipülatörün araç çerçevesinin konumunu ve yönelimini ana çerçeveye göre belirtir. Kısaca ileri yön kinematiği eklem değişkenleri ile uç işlevcisinin konumu ve yönelimini ana çerçeveye göre hesaplar diyebiliriz. Her bir ekleme bir koordinat istemi yerleştirilse komşu iki eklem arasındaki ilişki bir ${}^{i-1}_{i}T$ dönüşüm matrisiyle elde edilir.

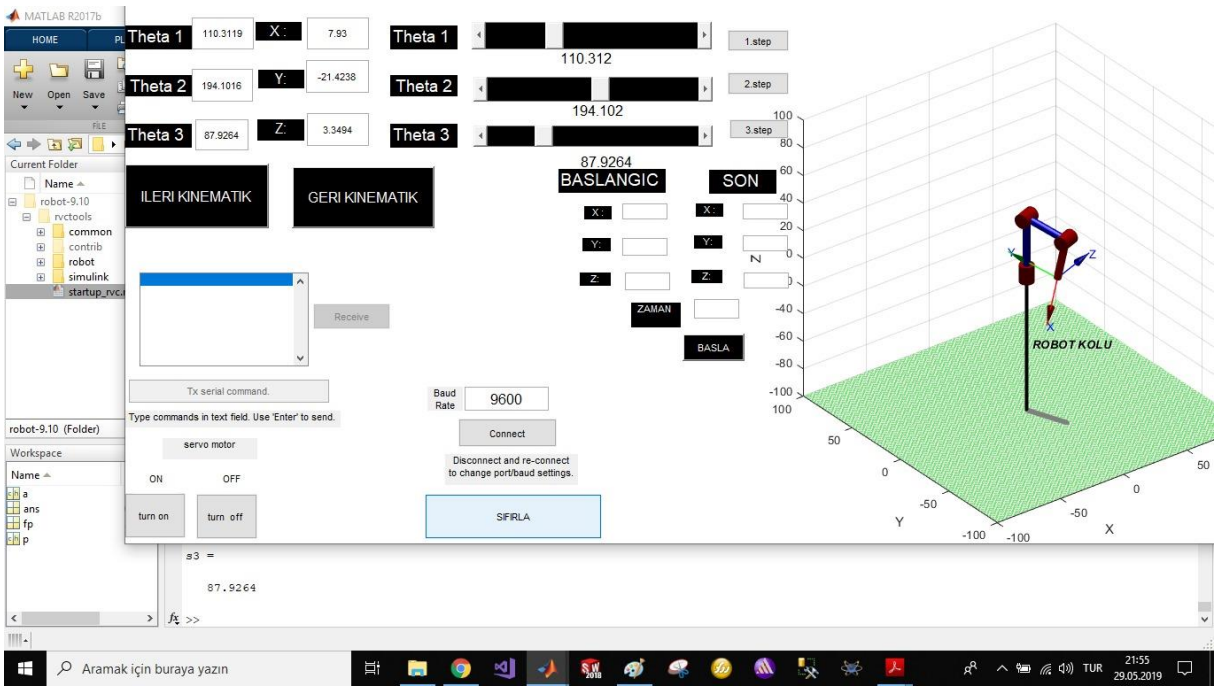
2.1.2.Geri (Ters) Kinematik Hesaplama

Ters Kinematik ileri kinematiğin tam tersidir ve robotun uç işlevcisinin ana çerçeveye göre konumu ve yönelimi verildiğinde manipülatörün bu konuma ve yönelime gelebilmesi için gerekli eklem değişkenlerinin bulunmasıdır. Başka bir değişle de uç işlevcisinin konum ve yönelimini kartezyen koordinat sisteminden eklem koordinat sistemine dönüştürme işlemi olarak da tanımlayabiliriz.



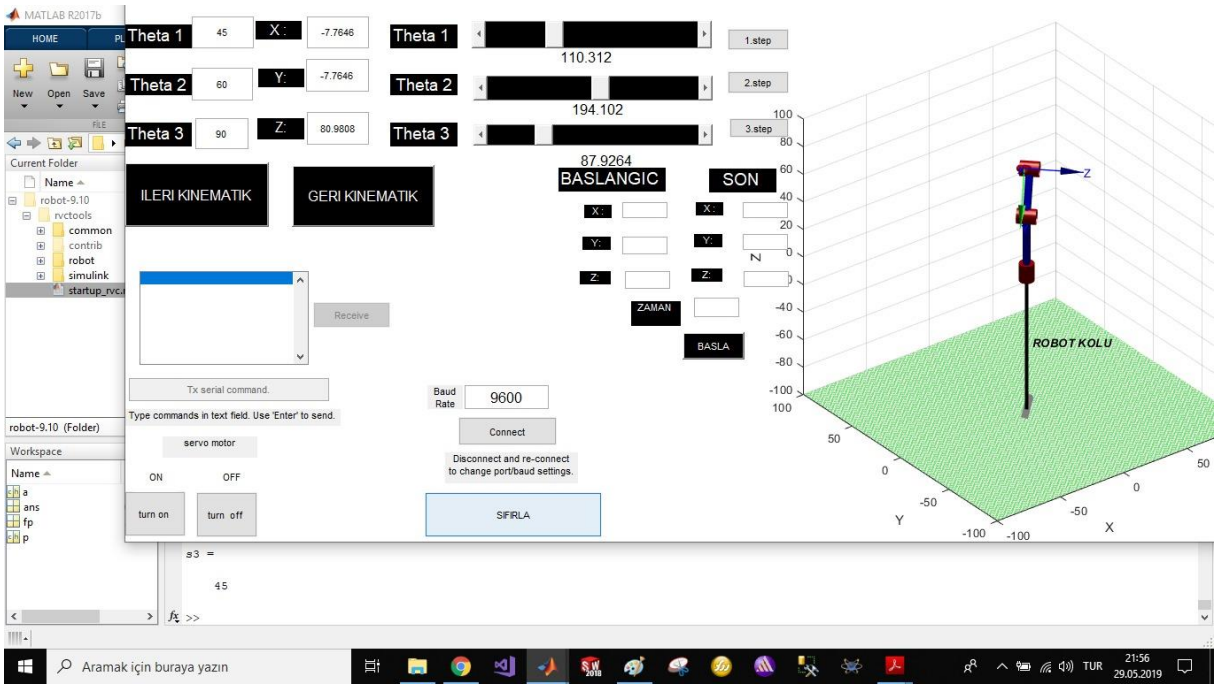
2.2.SIMULINK VE GUIDE

Simulink ve Guide Matlabın altında çalışan araçlardır. Her ne kadar Simulink aracı çok farklı şekillerde bizlere yardımcı olsa da en çok kullanıldığı alan, mevcut sistemlerin simülasyonunun yapılmasıdır. Bu çalışmada ise Simulinkte kurulmuş olan sistemin robot kolu üzerinde çalıştırılması özelliğinden yaralanılmıştır. Guide penceresi ise kullanıcı ile Matlab programı arasında bir ara yüzey sağlamaktadır. Guide penceresi sayesinde kullanıcı çalışan bir programa istediği bilgileri gönderebilir ya da istediği bilgileri alabilir.



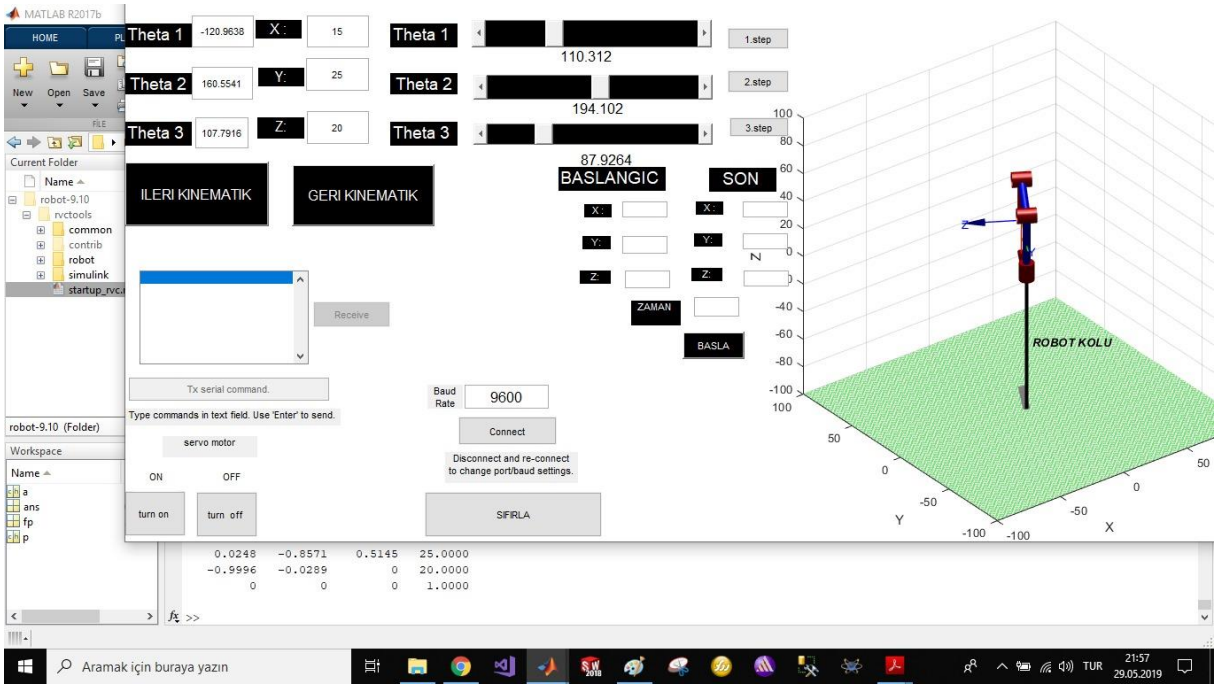
MatLab Simülasyon Örneği

Robot uzuvları, açı değerleri ya da konum bilgileri girilerek hareket ettirilebilir. Açı değerlerinin değişimi bar çubuk kullanımı ile kolaylaştırılmıştır. İşlemler gerçekleştirildikten sonra step motor butonlarına tıklayarak hareket sağlanır. Simülasyon görüntüsü robot kolun pozisyonunu göstermektedir. Turn on/off butonları robot kolun en uç kısmındaki servo motorla çalıştırılan kıskacın hareketini sağlar. Text kutucuğuna değer girerek band genişliği tekrardan düzenlenebilir. Sıfırlama butonu, robotun pozisyonunu sıfırlar.



MatLab İleri Kinematik Simülasyon Örneği

İleri kinematik hesaplama, girilen açı değerlerine göre robot uzuvlarının konum bilgisine ulaşabilmemizi sağlar.



MatLab Geri Kinematik Simülasyon Örneği

Geri kinematik hesaplama, girilen konum bilgisine göre robot kollarının kaç derecelik bir açıda duracağını belirtir.

2.3.8051 Serisi Mikrodenetleyici

2.3.1.8051 serisi mikrodenetleyicilerin genel özellikleri:

• Kontrol uygulamaları için optimize edilmiş 8 bitlik CPU.

• Genişletilmiş Boolean işleme komutları (tek bitlik lojik komutlar).

• On-chip program hafızası (Program ROM).

• On-chip veri hafızası (Data RAM).

• 4 adet 8 bitlik I/O portu.

• Çift yönlü kullanılabilen ve tek tek adreslenebilen I/O pinleri.

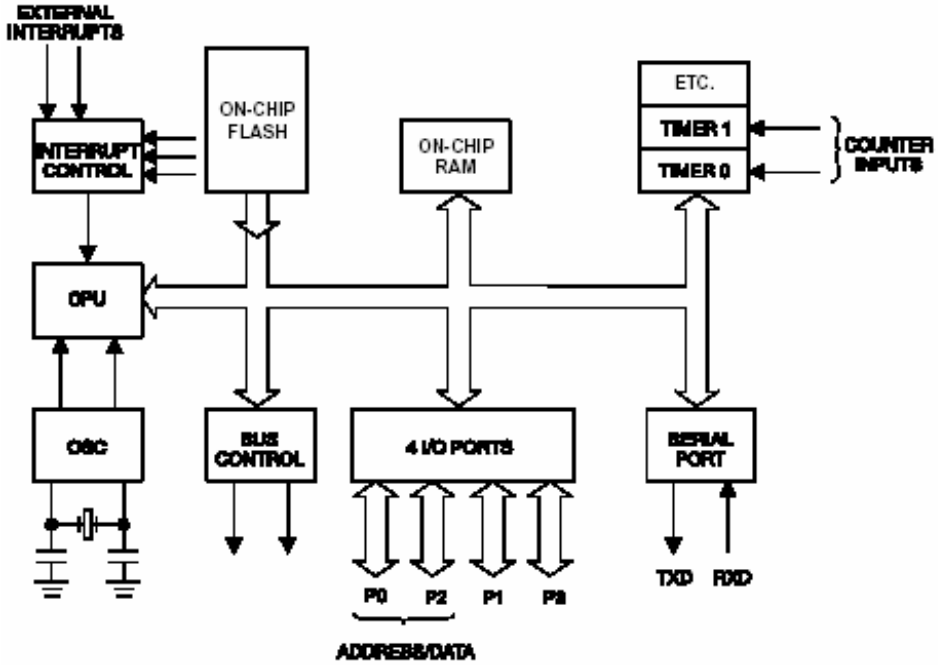
• 2 kanal 16 bitlik Timer/Counter (8052 de 3 kanal).

• Full Duplex UART (Seri haberleşme kanalı).

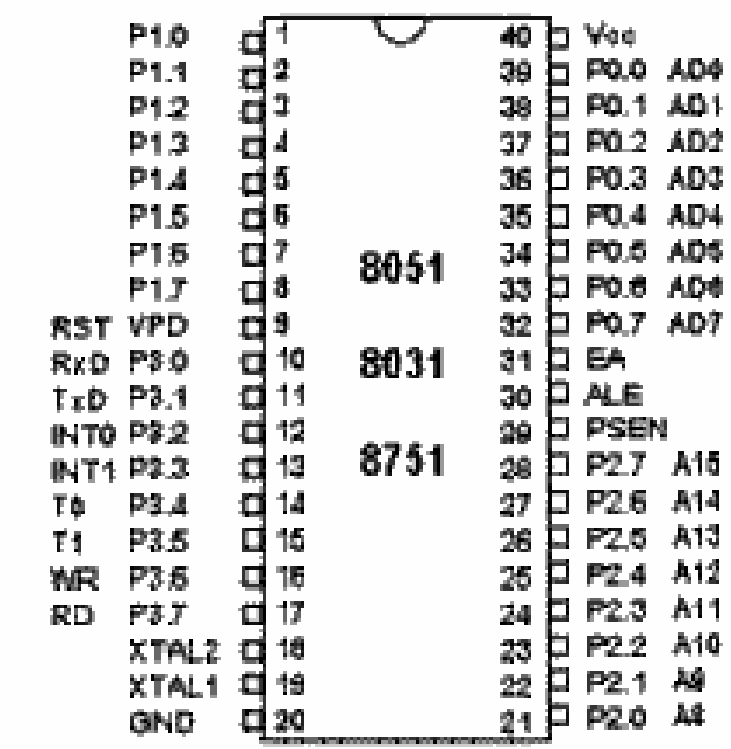
• Çok kaynaklı / vektörlü / öncelik seviyeli kesme yapısı (Interrupts).

• On-chip saat osilatörü.

2.3.2.Temel Mimari Yapısı



2.3.3.Bacak Konfigürasyonu



BACAK NO	SEMBOL	AÇIKLAMA
1,2,3,4,5,6,7,8	Port-1	Birden sekize kadar genel amaçlı giriş/çıkış portu pinleri paralel giriş veya çıkış olarak kullanılan port 1'e ait pinlerdir.
9	RST	Sıfırlama(Reset) pini Mikrodenetleyici resetlendiği anda program belleğinde kayıtlı olan sistem programları hafızasında çalışmaya başlar.
10,11,12,13,14,15,16,17	Port-3	Paralel giriş veya çıkış Port-3 'e ait port pinleridir. P3 'ün her pininin 2. bir fonksiyonu vardır.
10	P3.0 (RxD')	Asenkron seri haberleşmede seri bilginin alındığı port pinidir.
11	P3.1 (TxD')	Seri haberleşme esnasında bilginin gönderildiği port pinidir.
12	P3.2 (INT0)	0 nolu harici kesme girişi düşen kenar tetikleme ile aktif olur.
13	P3.3 (INT1)	Bir nolu harici kesme girişi düşen kenar tetikleme ile aktif olur.
14	P3.4 (T0)	Zamanlayıcı/Sayıcı - 0 Modunda çalışırken giriş olarak kulanılan bacaktır.
15	P3.5 (T1)	Zamanlayıcı/Sayıcı - 1 Modunda çalışırken giriş olarak kulanılan bacaktır.
16	P3.6 (WR')	Dış veri hafızasına veri yazılırken aktif olan kontrol işareti (WRITE) bu pinden üretilir.
17	P3.7 (RD')	Dış veri hafızasından veri okunurken aktif olan kontrol işareti (READ) bu pinden üretilir.
18	XTAL2	Clock Sinyal Giriş Ucu-2
19	XTAL1	Clock Sinyal Giriş Ucu-1
20	VSS	Mikrodenetleyicinin Referans Bacağı (Negatif besleme-Toprak) 0V uygulanır.
21,22,23,24,25,26,27,28	Port-2	Paralel giriş veya çıkış Port-2'nin uçları. Aynı zamanda dış çevre birimleri ile iletişimde adres yolunun yüksek anlamlı (most significant) 8-bitlik bilgisini taşır.
29	PSEN'	Dış program hafızasından bilgi okunurken,dış hafıza elemanı bu kontrol sinyali ile adres bilgisi geldikten sonra bilgiyi veri yoluna çıkarır.
30	ALE	Mikrodenetleyicinin dış dünya ile haberleşmesinde Port-0 hem veri, hem adres yolunun düşük anlamlı (least significant) sekiz biti için kullanılır.Bu iki ayrı işaret gurubu ALE sinyali ile birbirinden ayrılır.

31	EA'	Mikrodenetleyici içindeki ROM'un Program belleği olarak kullanılıp kullanılmayacağını belirler.Bu giriş ucu lojik 1 olursa program reset sonrası çalışması dahili ROM ‘dan aksi takdirde harici ROM ‘ dan gerçekleştirilir.
32,33,34,35,36,37,38,39	Port-0	Paralel giriş çıkış portu.Bu port dış elemanlar kullanılmadığı zaman normal bir port görevi yapar.Dış hafıza birimleri kullanıldığı zaman ise veri yolu ile adres yolunun düşük anlamlı sekiz biti için kullanılır.Bu kullanım zamnalama ile olur,bunuda ALE sinyali sağlar.
40	VCC	Mikrodeneleyicinin pozitif beslemesi – Standart yapıda 5V uygulanır.

[5]

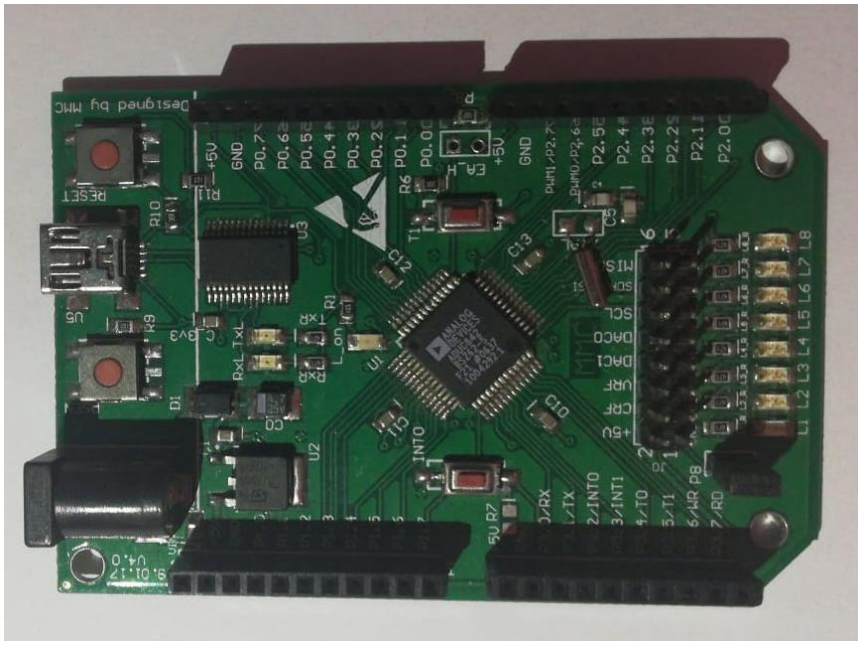
BÖLÜM 3.

MATERYAL METOD

3.1.Kullanılan Malzemeler:

- ADUC842 Mikrodenetleyici
- 4 adet Step Motor
- ULN2003A Motor Sürücü
- 1 adet 85X110X19 FZR Rulman
- 3D yazıcı ile basılan kol parçaları
- Mikro servo Sg90 (9G)

3.1.1.ADUC842 Mikrodenetleyici

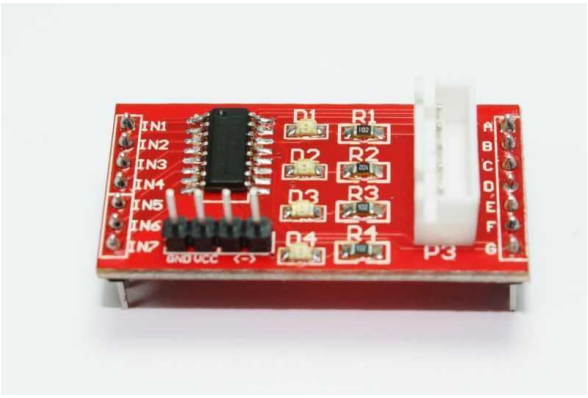


- 8-Kanal, 400 kSPS, 12-Bit ADC
- İki 12-Bit Rail-to-Rail Gerilim-Çıkış DAC'ları
- İki esnek PWM Çıkışı/16-Bit SD DACs
- 62K-Byte devre üzerinde yeniden programlanabilir Flash Program Belleği
- 4K-Byte Okuma
- 2K-Byte SRAM
- 32kHz dış kristal ve çip üzerinde
- Programlanabilir PLL

3.1.2.Step Motor

- Çalışma Gerilimi: 5V DC
- 4 Fazlı
- Adım Açısı 5.625°/64
- Frekans: 100 Hz
- Direnç: 130 Ohm

3.1.3.ULN2003A Motor Sürücü



- ULN2003 Step Motor Sürücü Kartı, 4-fazlı 5-telli step motor (5v-12v) sürmek için ULN2003 darlington dizilerini kullanmaktır.
- Arduino ve benzeri platformlar ile step motor sürmek için kolaylıkla kullanılabilecek küçük bir sürücü devresi modülüdür.
- Kart boyutları : 18 mm x 26 mm
- Lojik kontrol voltajı : 3 - 5.5V
- Motor voltaj beslemesi : 5 - 15V

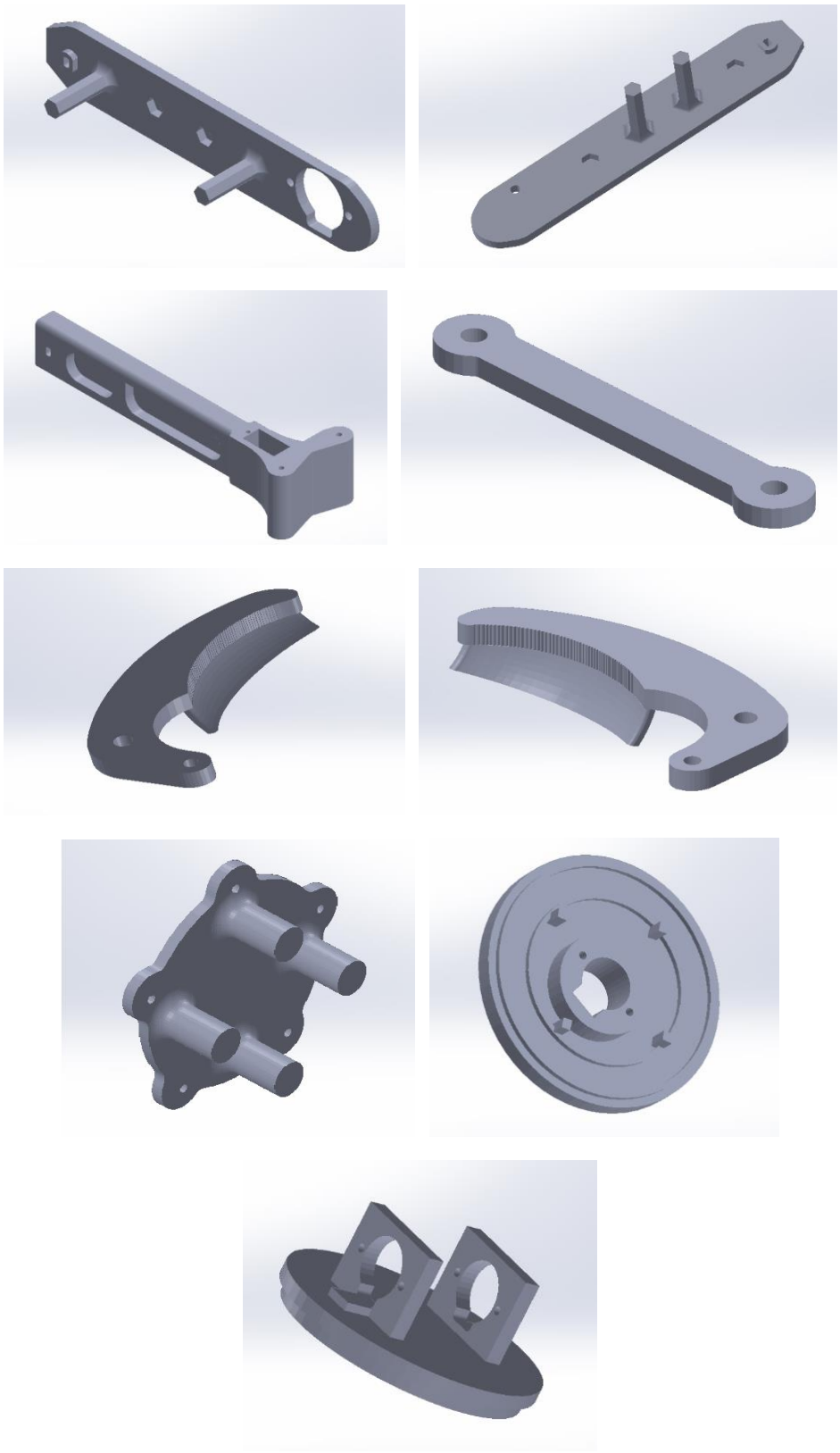
3.1.4.Mikro servo Sg90 (9G)

- Boyutlar: 23.1 x 12.2 x 29 mm
- Ağırlık: 9 g
- Çalışma gerilimi: 4.8 - 6.0 VDC
- Hız @4.8V: 0.1 sn/60°
- Zorlanma Torku @6V: 1.8 kg.cm
- Dişli kutusu: Plastik
- Dönüş açısı: 0-180°
- Çalışma PWM sinyali: 500-2400 µs
- Kablo Uzunluğu: 15 cm

3.1.5.Fzr rulman (51117)

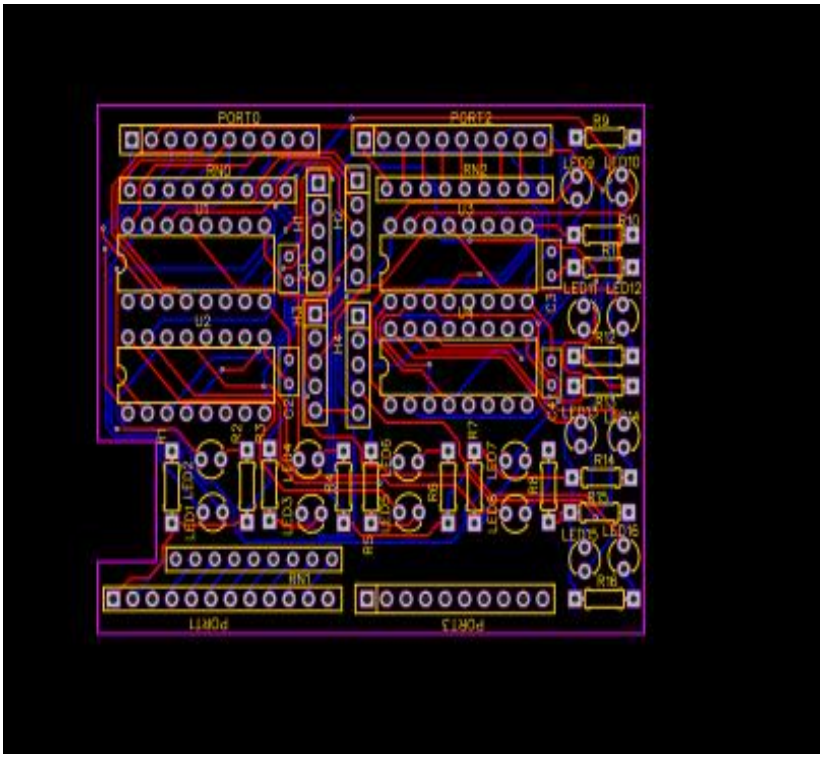
Boyutları: 85 X 110 X 19

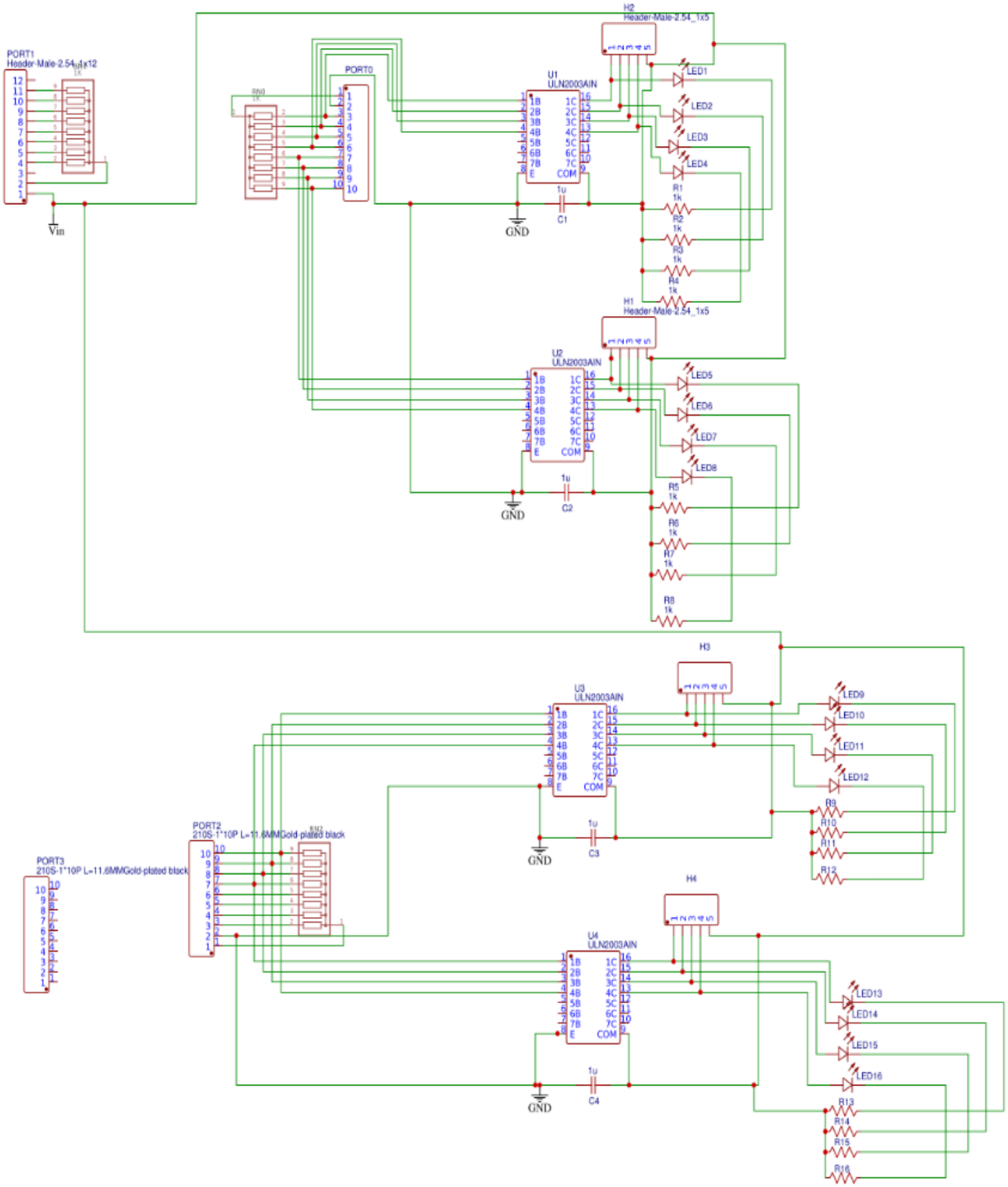
3.1.6.Tasarımı yapılan parçalar



3.1.7.Robotun Görünümü

3.2.Bağlantı ve Devre Şemaları




PORT1
Header-Male-2.54_1x12
PORT0
RN0
U1
ULN2003AIN
H2
Header-Male-2.54_1x5
LED1
LED2
LED3
LED4
R1
R2
R3
R4
C1
Vin
GND
H1
U2
LED5
LED6
LED7
LED8
R5
R6
R7
R8
C2
H3
U3
LED9
LED10
LED11
LED12
R9
R10
R11
R12
C3
PORT2
210S-1*10P L=11.6MMGold-plated black
RN2
PORT3
H4
U4
LED13
LED14
LED15
LED16
R13
R14
R15
R16
C4

3.3.Kullanılan Kodlar

3.3.1.MAIN

```
// Wave drive Mode
#include<reg51.h>

#include<lib842.h> //; Function and variable declaration file .
#include<stdio.h>       //"stdio.h"
#include<ctype.h>      //"ctype.h"
#include <intrins.h>
sfr SCONV = 0xDC ;
sfr T3FD = 0x9D;
sfr T3CON = 0x9E;
sfr PWM1L =0xB3;
sfr PWM1H =0xB4;
sfr PWM0L =0xB1;
sfr PWM0H =0xB2;
sfr PWMCON = 0xAE;



void UART_Init()
{
          // 9600 ICIN
          T3CON = 0x83;
          T3FD = 0x02D;
          SCON = 0x52;
}

void msdelay(unsigned int time)
   {
   unsigned i,j ;
   for(i=0;i<time;i++)
   for(j=0;j<400;j++);
   }
void motor1()
{
                    P2=0x01;          // 0001
       msdelay(1);
       P2=0x02;          //0010
       msdelay(1);
       P2=0x04;          //0100
       msdelay(1);
       P2=0x08;          //1000
       msdelay(1);

}
void motor1Ters()
{
                    P2=0x08;          // 0001
       msdelay(1);
```

```
        P2=0x04;        //0010
        msdelay(1);
        P2=0x02;        //0100
        msdelay(1);
        P2=0x01;        //1000
        msdelay(1);

}
void motor2()
{
            P0=0x01;        // 0001
        msdelay(1);
        P0=0x02;        //0010
        msdelay(1);
        P0=0x04;        //0100
        msdelay(1);
        P0=0x08;        //1000
        msdelay(1);
}
void motor2Ters()
{
            P0=0x08;        // 0001
        msdelay(1);
        P0=0x04;        //0010
        msdelay(1);
        P0=0x02;        //0100
        msdelay(1);
        P0=0x01;        //1000
        msdelay(1);
}
void motor3()
{
            P0=0x10;        // 0001
        msdelay(1);
        P0=0x20;        //0010
        msdelay(1);
        P0=0x40;        //0100
        msdelay(1);
        P0=0x80;        //1000
        msdelay(1);
}
void motor3Ters()
{
            P0=0x80;        // 0001
        msdelay(1);
        P0=0x40;        //0010
        msdelay(1);
        P0=0x20;        //0100
        msdelay(1);
        P0=0x10;        //1000
        msdelay(1);
}
void servoAC()
{
```

```
        PWM0L = 0x31;
  PWM0H  = 0x08;
        PWM1L = 0xEA;
  PWM1H  = 0x51;
  PWMCON = 0x1B ;
        msdelay(250);
}
void servoKapat()
{


        PWM0L = 0x19;
  PWM0H  = 0x04;
        PWM1L = 0xEA;
  PWM1H  = 0x51;
  PWMCON = 0x1B ;
        msdelay(250);
}
void calistir(char c)
{
        unsigned i1 ;
        unsigned c1 ;
        unsigned c2 ;
        unsigned c3 ;

        if (c =='A')
        {
                c1 = _getkey();
                if (c1 ==' ')
                {
                        c1=0;
                }
                else
                {
                        c2 = _getkey();
                        if (c2 ==' ')
                                {
                                        c2=0;
                                        c1 = toint(c1);
                                }
                                else
                                {
                                        c3 = _getkey();
                                        if (c3 ==' ')
                                                {
                                                        c3=0;
                                                        c1 = toint (c1);
                                                        c2 = toint (c2);
                                                        c1 = 10*c1 +c2;
                                                }
                                                else
                                                {
                                                        c1 = toint (c1);
                                                        c2 = toint (c2);
```

```
                                        c3 = toint (c3);

                                        c1 = (100*c1) + (10*c2) +c3 ;
                                }
                        }
                }
        for(i1=0;i1<c1;i1++)
                {
                        motor1();
                }
        }

else if(c =='B')
{
        c1 = _getkey();
        if (c1 ==' ')
        {
                c1=0;
        }
        else
        {
                c2 = _getkey();
                if (c2 ==' ')
                        {
                                c2=0;
                                c1 = toint(c1);
                        }
                        else
                        {
                                c3 = _getkey();
                                if (c3 ==' ')
                                        {
                                                c3=0;
                                                c1 = toint (c1);
                                                c2 = toint (c2);
                                                c1 = 10*c1 +c2;
                                        }
                                        else
                                        {
                                                c1 = toint (c1);
                                                c2 = toint (c2);
                                                c3 = toint (c3);

                                                c1 = (100*c1) + (10*c2) +c3 ;
                                        }
                                }
                        }
        for(i1=0;i1<c1;i1++)
                        {
                                motor2();
                        }
                }

else if(c =='C')
{
```

```
        c1 = _getkey();
        if (c1 ==' ')
        {
                c1=0;
        }
        else
        {
                c2 = _getkey();
                if (c2 ==' ')
                        {
                                c2=0;
                                c1 = toint(c1);
                        }
                        else
                        {
                                c3 = _getkey();
                                if (c3 ==' ')
                                        {
                                                c3=0;
                                                c1 = toint (c1);
                                                c2 = toint (c2);
                                                c1 = 10*c1 +c2;
                                        }
                                        else
                                        {
                                                c1 = toint (c1);
                                                c2 = toint (c2);
                                                c3 = toint (c3);

                                                c1 = (100*c1) + (10*c2) +c3 ;
                                        }
                                }
                        }
        for(i1=0;i1<c1;i1++)
                        {
                                motor3();
                        }
                }
else if (c =='D')
{
        c1 = _getkey();
        if (c1 ==' ')
        {
                c1=0;
        }
        else
        {
                c2 = _getkey();
                if (c2 ==' ')
                        {
                                c2=0;
                                c1 = toint(c1);
                        }
                        else
                        {
```

```
                                        c3 = _getkey();
                                        if (c3 ==' ')
                                                {
                                                        c3=0;
                                                        c1 = toint (c1);
                                                        c2 = toint (c2);
                                                        c1 = 10*c1 +c2;
                                                }
                                                else
                                                {
                                                        c1 = toint (c1);
                                                        c2 = toint (c2);
                                                        c3 = toint (c3);

                                                        c1 = (100*c1) + (10*c2) +c3 ;
                                                }
                                        }
                                }
        for(i1=0;i1<c1;i1++)
                                {
                                motor1Ters();
                                }
                        }
else if (c =='E')
{
        c1 = _getkey();
        if (c1 ==' ')
        {
                c1=0;
        }
        else
        {
                c2 = _getkey();
                if (c2 ==' ')
                        {
                                c2=0;
                                c1 = toint(c1);
                        }
                        else
                        {
                                c3 = _getkey();
                                if (c3 ==' ')
                                        {
                                                c3=0;
                                                c1 = toint (c1);
                                                c2 = toint (c2);
                                                c1 = 10*c1 +c2;
                                        }
                                        else
                                        {
                                                c1 = toint (c1);
                                                c2 = toint (c2);
                                                c3 = toint (c3);

                                                c1 = (100*c1) + (10*c2) +c3 ;
```

```
                                        }
                                }
                        }
        for(i1=0;i1<c1;i1++)
                        {
                        motor2Ters();
                        }
                }
else if (c =='F')
{
        c1 = _getkey();
        if (c1 ==' ')
        {
                c1=0;
        }
        else
        {
                c2 = _getkey();
                if (c2 ==' ')
                        {
                                c2=0;
                                c1 = toint(c1);
                        }
                        else
                        {
                                c3 = _getkey();
                                if (c3 ==' ')
                                        {
                                                c3=0;
                                                c1 = toint (c1);
                                                c2 = toint (c2);
                                                c1 = 10*c1 +c2;
                                        }
                                        else
                                        {
                                                c1 = toint (c1);
                                                c2 = toint (c2);
                                                c3 = toint (c3);

                                                c1 = (100*c1) + (10*c2) +c3 ;
                                        }
                                }
                        }
        for(i1=0;i1<c1;i1++)
                        {
                        motor3Ters();
                        }
                }
else if (c =='G')
{
        servoAC();
}
else if (c =='H')
{
        servoKapat();
```

```
        }
        else
        {
                c = _getkey();
                calistir(c);
        }

}

void main()
{

        char c;
  P0=0x00;
          P1=0x00;
          P2=0x00;

        UART_Init();
        calistir(c);

}
```

3.3.1.1.Ctype

```
/*---------------------------------------------------------------------------
CTYPE.H

Prototypes for character functions.
Copyright (c) 1988-2002 Keil Elektronik GmbH and Keil Software, Inc.
All rights reserved.
---------------------------------------------------------------------------*/

#ifndef __CTYPE_H__
#define __CTYPE_H__

#pragma SAVE
#pragma REGPARMS
extern bit isalpha (unsigned char);
extern bit isalnum (unsigned char);
extern bit iscntrl (unsigned char);
extern bit isdigit (unsigned char);
extern bit isgraph (unsigned char);
extern bit isprint (unsigned char);
extern bit ispunct (unsigned char);
extern bit islower (unsigned char);
extern bit isupper (unsigned char);
extern bit isspace (unsigned char);
extern bit isxdigit (unsigned char);
extern unsigned char tolower (unsigned char);
extern unsigned char toupper (unsigned char);
extern unsigned char toint (unsigned char);

#define _tolower(c) ( (c)-'A'+'a' )
#define _toupper(c) ( (c)-'a'+'A' )
#define toascii(c)  ( (c) & 0x7F )
#pragma RESTORE

#endif
```

3.3.1.2.Intrins

```
/*---------------------------------------------------------------------------

INTRINS.H


Intrinsic functions for C51.

Copyright (c) 1988-2010 Keil Elektronik GmbH and ARM Germany GmbH

All rights reserved.

---------------------------------------------------------------------------*/
```

```
#ifndef __INTRINS_H__
#define __INTRINS_H__

#pragma SAVE

#if defined (__CX2__)
#pragma FUNCTIONS(STATIC)
/* intrinsic functions are reentrant, but need static attribute */
#endif

extern void          _nop_     (void);
extern bit           _testbit_ (bit);
extern unsigned char _cror_    (unsigned char, unsigned char);
extern unsigned int  _iror_    (unsigned int,  unsigned char);
extern unsigned long _lror_    (unsigned long, unsigned char);
extern unsigned char _crol_    (unsigned char, unsigned char);
extern unsigned int  _irol_    (unsigned int,  unsigned char);
extern unsigned long _lrol_    (unsigned long, unsigned char);
extern unsigned char _chkfloat_(float);
#if defined (__CX2__)
extern int           abs       (int);
extern void          _illop_   (void);
#endif
#if !defined (__CX2__)
extern void          _push_    (unsigned char _sfr);
extern void          _pop_     (unsigned char _sfr);
#endif

#pragma RESTORE

#endif
```

3.3.1.3.Lib842

//Function declaration for ADuC842.lib file for Keil tools.

//File: lib842.h

```
extern char AdcCfg(char);

extern char AdcGo(char, char);

extern int AdcRd(void);

extern char AdcBsy(void);

extern int AdcGet(char);

extern char AdcCal(char);

extern char DacCfg(char);

extern char DacOut(char, int);

extern char PllDly(int);

extern char PllRcd(void);

extern char PllWcd(int);

extern char PwmCfg(char);

extern char PwmW16(char,int);
```

```
extern char PwmW8(char,char,char);

extern char TicCfg(char);

extern char TicHr(void);

extern char TicMin(void);

extern char TicSec(void);

extern char TicHth(void);

extern char TicVal(char,char);

extern char TicGo(char,char,char,char);

extern char UrtCfg(char, unsigned int);

extern char UrtBsy(void);

extern char WdtCfg(char);

extern char WdtKk(void);


//Variable declaration for ADuC842.lib file.

char data       cUrtVar;

char data       cSpiFlag;

//Declaration
End==================================================Declaration End
```

3.3.1.4.Reg51

```
/*--------------------------------------------------------------------------
REG51.H

Header file for generic 80C51 and 80C31 microcontroller.
Copyright (c) 1988-2002 Keil Elektronik GmbH and Keil Software, Inc.
All rights reserved.
--------------------------------------------------------------------------*/

#ifndef __REG51_H__
#define __REG51_H__

/*  BYTE Register  */
sfr P0   = 0x80;
sfr P1   = 0x90;
sfr P2   = 0xA0;
sfr P3   = 0xB0;
sfr PSW  = 0xD0;
sfr ACC  = 0xE0;
sfr B    = 0xF0;
sfr SP   = 0x81;
sfr DPL  = 0x82;
sfr DPH  = 0x83;
sfr PCON = 0x87;
sfr TCON = 0x88;
sfr TMOD = 0x89;
sfr TL0  = 0x8A;
sfr TL1  = 0x8B;
sfr TH0  = 0x8C;
sfr TH1  = 0x8D;
sfr IE   = 0xA8;
sfr IP   = 0xB8;
```

```
sfr SCON = 0x98;
sfr SBUF = 0x99;


/*  BIT Register  */
/*  PSW   */
sbit CY   = 0xD7;
sbit AC   = 0xD6;
sbit F0   = 0xD5;
sbit RS1  = 0xD4;
sbit RS0  = 0xD3;
sbit OV   = 0xD2;
sbit P    = 0xD0;

/*  TCON  */
sbit TF1  = 0x8F;
sbit TR1  = 0x8E;
sbit TF0  = 0x8D;
sbit TR0  = 0x8C;
sbit IE1  = 0x8B;
sbit IT1  = 0x8A;
sbit IE0  = 0x89;
sbit IT0  = 0x88;

/*  IE   */
sbit EA   = 0xAF;
sbit ES   = 0xAC;
sbit ET1  = 0xAB;
sbit EX1  = 0xAA;
sbit ET0  = 0xA9;
sbit EX0  = 0xA8;
```

```
/*  IP   */
sbit PS   = 0xBC;
sbit PT1  = 0xBB;
sbit PX1  = 0xBA;
sbit PT0  = 0xB9;
sbit PX0  = 0xB8;

/*  P3  */
sbit RD   = 0xB7;
sbit WR   = 0xB6;
sbit T1   = 0xB5;
sbit T0   = 0xB4;
sbit INT1 = 0xB3;
sbit INT0 = 0xB2;
sbit TXD  = 0xB1;
sbit RXD  = 0xB0;

/*  SCON  */
sbit SM0  = 0x9F;
sbit SM1  = 0x9E;
sbit SM2  = 0x9D;
sbit REN  = 0x9C;
sbit TB8  = 0x9B;
sbit RB8  = 0x9A;
sbit TI   = 0x99;
sbit RI   = 0x98;

#endif
```

3.3.1.5.Stdio

```
/*---------------------------------------------------------------------------
STDIO.H

Prototypes for standard I/O functions.
Copyright (c) 1988-2002 Keil Elektronik GmbH and Keil Software, Inc.
All rights reserved.
---------------------------------------------------------------------------*/

#ifndef __STDIO_H__
#define __STDIO_H__

#ifndef EOF
 #define EOF -1
#endif

#ifndef NULL
 #define NULL ((void *) 0)
#endif

#ifndef _SIZE_T
 #define _SIZE_T
 typedef unsigned int size_t;
#endif

#pragma SAVE
#pragma REGPARMS
extern char _getkey (void);
extern char getchar (void);
extern char ungetchar (char);
extern char putchar (char);
extern int printf   (const char *, ...);
```

```
extern int sprintf  (char *, const char *, ...);
extern int vprintf  (const char *, char *);
extern int vsprintf (char *, const char *, char *);
extern char *gets (char *, int n);
extern int scanf (const char *, ...);
extern int sscanf (char *, const char *, ...);
extern int puts (const char *);

#pragma RESTORE

#endif
```

SONUÇLAR

Robot uzuv ve parçaları SolidWorks programıyla çizilmiştir. Çizilen parçalar Zortax M200 3D Yazıcı ve yazıcının kendi software programıyla basılmıştır. Robotun ileri ve geri kinematik hesapları yapılmış ve MatLab programı tarafından simule edilmiştir. Robotun programlaması için Aduc842 Mikrodenetleyici kart kullanılmış ve assembly dilinde programlaması yapılmıştır. Devresi ve bağlantıları ilgili şemalarda gösterildiği üzere yapılmıştır. Robotun kontrolü bilgisayar yardımıyla elle yapılmaktadır. İsteniler ekseni, istenilen derecede (kablo düzeneği dikkate alınarak) döndürme hareketi sağlanabilmiştir. İkinci eksendeki step motorların gücü zayıf kaldığı tespit edilmiştir.

KAYNAKLAR

[1]. Mendi, F., Külekçi, M.K. (2001). PLC denetimli üç eksenli pnömatik bir işlem ünitesi tasarımı ve imalatı Gazi Üniversitesi Fen Bilimleri Dergisi, 14 (22), 35-42

[2]. Gilles, M. (1992). Programme Logic Controllers (1. Baskı). Massachusetts: Baffins Lane

[3]. Koiva, A.J. (1989) Fundamentals for Control of Robotic Manipulators (1. Baskı). New York: John Wiley & Sons INC

[4]. Bolt, W. (2006) Mechatronics: Electronic Control Systems in Mechanical and Electrical Engineering (4. Baskı). New Jersey: Prentice Hall.

[5]. www.dicle.edu.tr/Contents/cda8a682-629a-4cb4-bf74-e25dcdfd8208.ppt

ÖZGEÇMİŞ

Fatih Can Kaya 1995'te İstanbul'da doğdu; ilk ve orta öğrenimini aynı şehirde Tantavi İlköğretim Okulu'nda tamamladı; İstanbul Ticaret Odası Anadolu Teknik Lisesi, Elektrik-Elektronik alanında Endüstriyel Bakım Onarım bölümünden mezun olduktan sonra 2013 yılında Karabük Üniversitesi, Mühendislik Fakültesi, Mekatronik Bölümü'ne girdi. Halen bu bölümde eğitimini sürdürmektedir.

İletişim Bilgileri
E-posta: fatihcankaya95@gmail.com