

**ÜÇ BOYUTLU UZAYDA HAREKET EDEBİLEN ROBOT KOLUN,
BİLGİSAYAR DESTEKLİ KONTROLÜ, TASARIMI VE UYGULAMASI**

MEHMET ALBAYRAK

**YÜKSEK LİSANS TEZİ
ELEKTRİK EĞİTİMİ ANABİLİM DALI**

**GAZİ ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ**

**EYLÜL - 1997
ANKARA**

Mehmet ALBAYRAK tarafından hazırlanan ÜÇ BOYUTLU UZAYDA HAREKET EDEBİLEN ROBOT KOLUN, BİLGİSAYAR DESTEKLİ KONTROLÜ, TASARIMI VE UYGULAMASI adlı bu tezin Yüksek Lisans tezi olarak uygun olduğunu onaylarım.



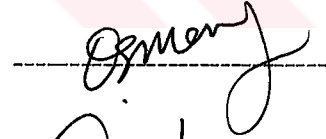
Y. Doç. Dr. Osman GÜRDAL
Tez Yöneticisi

Bu çalışma, jürimiz tarafından Elektrik Eğitimi Anabilim Dalında yüksek lisans tezi olarak kabul edilmiştir.

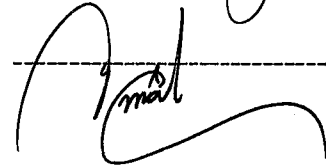
Başkan : Doç.Dr. Çetin ELMAS



Üye : Y.Doç.Dr. Osman GÜRDAL



Üye : Y.Doç.Dr. Erkan İMAL



Bu tez, Gazi Üniversitesi Fen Bilimleri Enstitüsü tez yazım kurallarına uygundur.



İÇİNDEKİLER

	Sayfa
ÖZET	i
ABSTRACT	ii
TEŞEKKÜR	iii
ŞEKİLLERİN LİSTESİ.....	iv
ÇİZELGELERİN LİSTESİ.....	v
SİMGELER VE KISALTMALAR DİZİNİ	vi
1. GİRİŞ	1
2. ROBOTLAR	5
2.1. Robotların Kullanım Alanları	5
2.2. Endüstriyel Robotların Sınıflandırılması	6
2.2.1. Tiplerine göre	6
2.2.2. Yeteneklerine göre	7
2.2.3. Koordinat sistemine göre	7
2.2.4. Kuşaklarına (nesillerine) göre	7
2.3. Robotların Yapısı	7
2.3.1. Manipülatör ve el aparatları	7
2.3.2. Aktivatörler	7
2.3.3. Kontrol sistemi	8
2.4. Robotların Hareket Uzayı	8
2.5. Robotların Alt Sistemleri ve Fonksiyonel Yapısı	8
2.5.1. Kavrayıcı kol kinematiği	8
2.5.2. Kontrol sistemi	9
2.5.3. Sürme	9
2.5.4. Ölçme sistemi	9
2.5.5. Sensörler	9
3. BİLGİSAYAR DESTEKLİ KONTROL	10
3.1. I/O Birimi, Giriş ve Çıkış Portları	10

3.2.	Yollar	10
3.2.1.	Paralel I/O portları	12
3.3.	Kart Bağlantı Uçları	14
4.	GERÇEKLEŞTİRİLEN SİSTEM	16
4.1.	Adres Çözücü (Kod Çözücü, Yol Çoklayıcı) Katı	17
4.2.	Programlanabilir I/O Elemanı 8255'in Özellikleri	20
4.2.1.	8255'in çalışma modları	25
4.2.1.1.	Mod 0'da çalışma	26
4.2.1.2.	Mod 1'de çalışma	27
4.2.1.3.	Mod 2'de çalışma	27
4.3.	Step Motorlar	28
4.3.1.	Step motor türleri ve çalışması	30
4.3.1.1.	Sabit mıknatıslı step motor	30
4.3.1.2.	Değişken relüktanslı step motorlar	33
4.3.1.3.	Karma step motor	34
4.3.2.	Step motor parametreleri	35
4.3.2.1.	Adım cevabı (Step response)	35
4.3.2.2.	Adımlama oranı (Stepping rate)	35
4.3.2.3.	Adım açısı (Step angle)	36
4.3.3.	Step motor tahrik modları	37
4.4.	Step Motor Sürücü Katı	38
4.5.	Step Motor Frenleme Katı	40
4.6.	Robot Kol Kontrol ve Simülasyon Programı.....	41
4.7.	Robot Kol Mekanik Tasarımı	44
5.	SONUÇ VE ÖNERİLER	46
6.	KAYNAKLAR	47
7.	EKLER	48
8.	ÖZGEÇMİŞ	67

ÜÇ BOYUTLU UZAYDA HAREKET EDEBİLEN ROBOT KOLUN, BİLGİSAYAR DESTEKLİ KONTROLÜ, TASARIMI VE UYGULAMASI

(Yüksek Lisans Tezi)

Mehmet ALBAYRAK

**GAZİ ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ
Eylül 1997**

ÖZET

Bu çalışmada, bilgisayar destekli kontrol yöntemi ile üç boyutlu uzayda hareket edebilen, üç serbestlik dereceli robot kol tasarımı, kontrolü ve simülasyonu yapılmıştır. Kontrol için, bilgisayarın 16 bitlik ISA slotlarından birine yerleştirilebilecek yapıda; adres çözücü katı, programlanabilir giriş / çıkış katı, dört adet motor sürücü katı ve step motor frenleme katının bulunduğu devreler tek bir kontrol kartı üzerinde toplanmıştır. Hazırlanan kart üzerinde 48 bit giriş / çıkış hattı mevcuttur. Bu hatlardan 17 tanesi çıkış biti olarak kullanılmış, geri kalan bitler geri besleme için ayrılmıştır. Hazırlanan sistemin, kontrolü ve simülasyonu bilgisayarda gerçekleştirilmiştir. Eğitim amaçlı kullanılabilecek robot kolun, prototip olarak hazırlanan mekanik tasarımı bu kart ile kontrol edilmiştir.

Bilim Kodu : 626.01.00

Anahtar Kelimeler : Robot Kol, Bilgisayar Destekli Kontrol, Simülasyon

Sayfa Sayısı : 67

Danışman : Y.Doç.Dr. Osman GÜRDAL

**COMPUTER AIDED CONTROL, DESIGN AND APPLICATION OF
ROBOT ARM THAT CAN BE MOVED IN THREE DIMENSIONAL SPACE**

(M. Sc. Thesis)

Mehmet ALBAYRAK

**GAZİ UNIVERSITY
INSTITUTE OF SCIENCE AND TECNOLOGY
September 1997**

ABSTRACT

In this study, by computer aided control system, the design, control and the simulation of the degress of freedom robot arm that can be moved in three dimensional space has been effected. For the control, address decoder device, programmable input / output device, four stepping motor driver device and the devices which has stepping motor breaking device were put on only one control card, that can be placed in one of the 16 bits ISA slots of computer. On prepared card, 48 bits input / output lines are available. 17 of these lines are used as output lines. And the rest of lines are reserved for feedback. The control and simulation of prepared system were effected on computer. Mechanical design which is prepared as prototype of the robot arm that can be used with the aim of education was controlled by this card.

Science Code : 626.01.00
Key Words : Robot arm, Computer Aided Control, Simulation
Page Number : 67
Adviser : Assoc.Prof.Dr. Osman GÜRDAL

TEŞEKKÜR

Bu çalışmanın gerçekleşmesinde hoşgörü, sonsuz sabır ve katkılarından dolayı tez danışmanım Y.Doç.Dr. Osman GÜRDAL 'a, tecrübe ve ilgileri ile elektronik donanımın hazırlanmasında bana yol gösteren Erciyes Üniversitesi Kayseri Meslek Yüksek Okulu Öğr.Gör. Mustafa YAZ ve Öğr.Gör. H.Murat SERT 'e , mekanik kısmın hazırlanmasındaki katkılarından dolayı S.Demirel Üniversitesi Teknik Eğitim Fakültesi Öğr.Gör. Ş.Abdurrahman ÇELİK ve Isparta Teknik Lisesi Makine Bölümü Öğretmenlerine, bilgisayar programının hazırlanması ve karşılaşılan problemlerdeki yardımları için mesai arkadaşlarım Arş.Gör. Tuncay AYDOĞAN ve Arş.Gör. Abdülkadir ÇAKIR 'a, tez çalışmamda her türlü yardım ve kolaylığı esirgemeyen Gazi Üniversitesi Teknik Eğitim Fakültesi öğretim üyesi Y.Doç.Dr.Adnan SÖZEN 'e, S.Demirel Üniversitesi Teknik Eğitim Fakültesi akademik ve idari personeline, Eterno Bilgisayar Ltd. Şti. Genel Müdürü Fizik Müh. Birol KESER 'e, tez yazılmasında gösterdiği fedakarlık ve hiç eksiltmediği manevi desteği için Eşime ve beni yalnız bırakmayan Aileme teşekkürü bir borç bilir, şükranlarımı sunarım.

ŞEKİLLERİN LİSTESİ

Şekil	Sayfa
Şekil 3.1.	Verinin bir giriş portundan nasıl okunduğunu ve bir çıkış portuna nasıl yazıldığını gösteren diagram.....13
Şekil 4.1.	Kurulan sistemin blok şeması.....16
Şekil 4.2.	74 LS 138 Bağlantı uçları.....17
Şekil 4.3.	Adres çözücü katı ve 8255 ile bağlantısı.....20
Şekil 4.4.	8255 Entegresinin ayak bağlantıları21
Şekil 4.5.	8255 'in iç bağlantı blok diagramı.....22
Şekil 4.6.	8255 'e ait temel mod tanımlamaları ve yol arabirim.....24
Şekil 4.7.	Sabit mıknatıslı step motor.....31
Şekil 4.8.	Disk tipi PM step motor32
Şekil 4.9.	Üç fazlı değişken relüktanslı step motorun yapısı.....33
Şekil 4.10.	Karma tip step motorun şafta paralel kesiti34
Şekil 4.11.	İki fazlı step motorda yapılan tahrik.....38
Şekil 4.12.	Step motor kontrol sistemi.....38
Şekil 4.13.	Unipolar ve bipolar sürücü devreler39
Şekil 4.14.	Hazırlanan step motor sürücü devre.....39
Şekil 4.15.	Motor sürücü katına kadar olan katların blok şeması.....40
Şekil 4.16.	Step motor frenleme katı.....41
Şekil 4.17.	Yazılım Algoritması.....43
Şekil 4.18.	Ana gövde montaj şeması.....43
Şekil 4.19.	Robot kol mekanik yapısı.....43

ÇİZELGELERİN LİSTESİ

Çizelge	Sayfa
Çizelge 3.1. ISA Slottaki pin bağlantıları.....	14
Çizelge 4.1. 74LS138 Doğruluk tablosu.....	18
Çizelge 4.2. IBM / PC için giriş / çıkış adres haritası.....	19
Çizelge 4.3. Adres çözücü devrede kullanılan adresleme yapısı.....	19
Çizelge 4.4. 8255 Entegresinin ayak numaraları ve görevleri.....	21
Çizelge 4.5. 8255 'in temel işlem fonksiyon tablosu.....	23
Çizelge 4.6. Kontrol kelimesinin yapısı.....	26



SİMGELER VE KISALTMALAR

Bu çalışmada kullanılmış, fakat tez metni içinde açıklanmamış bazı simgeler ve kısaltmalar, açıklamaları ile birlikte aşağıda sunulmuştur.

Simgeler

f	frekans
Hz	Hertz
t	zaman
sn	saniye
V	Volt

Kısaltmalar

AC	Alternate current (alternatif akım)
CW	Clock wise (saat ibresi yönü)
CCW	Counter clock wise (saat ibresi tersi yönü)
CMOS	Complementary metal oxide semiconductor
DC	Direct current (doğru akım)
Fak.	Fakültesi
Fen Bil. Ens.	Fen Bilimleri Enstitüsü
I / O	Input / output (giriş / çıkış)
IBM	International Business Machines
İTÜ	İstanbul Teknik Üniversitesi
PPI	Programmable peripheral interface
PM	Permanent magnet (sabit mıknatıs)
SM	Step motor
TTL	Transistor transistor logic
VR	Variable Reluctance (Değişken relüktans)
Ün.	Üniversitesi

1. GİRİŞ

Robot kelimesi uluslararası literatüre, ilk defa Çekoslovak yazar Karl Capek tarafından sahnelenen “ Rossum ’un Evrensel Robotları “ adlı tiyatro eseri ile girmiştir. Bu eserde geçen ‘ Robota’ sözcüğü, “ köle, işçi “ anlamlarına gelmektedir. Çağımızın ünlü bilim-kurgu yazarlarından Dr. Isaac Asimov, robot davranışlarını gerçekçi bir biçimde incelemiş, 1950 ’li yıllarda yayınlanan eserlerinde, bu davranışları inceleyen bilim dalını ‘Robotik’ olarak adlandırmıştır. Dr. Asimov robot - insan ilişkileriyle, robotları belli bir oranda tanımlayan üç kanun ortaya koymuştur. Bunlar;

1. Robot hiçbir zaman insana zarar verecek hareketler yapmamalı ve insanın zarar görebileceği hallerde hareketsiz kalmalıdır.
2. Birinci kanunu çiğnememek şartıyla robot, insana her zaman itaat etmelidir.
3. Birinci ve ikinci kanunu çiğnememek şartıyla robot, kendini de korumalıdır.

Konu ile ilgili kurum ve kişilerce yapılan, uluslararası literatürde geçmiş robot tanımlamaları ;

1. Amerikan Robot Enstitüsü (RIA);

Robot; değişik türdeki işlerin yerine getirilmesi, programlanmış hareketler ile malzemeleri, parçaları, takım ve özel araçları taşıyabilecek, yeniden programlanabilme özelliğine sahip çok amaçlı bir manipülatördür.

2. İngiliz Robot Birliği;

Robot; belirli imalat işlemlerinin yapılması için parça, takım ve alet ya da özel imalat aygıtlarının programlanmış değişik hareketlerle taşınması, yerinin değiştirilmesi veya kullanılması için tasarlanmış, programlanabilir bir aygıttır.

3. Japonya Endüstriyel Robotlar Birliği (JIRA);

Sanayide kullanılan robotları, bilgi girdileri, eğitim açısından ele alarak sınıflandırmaktadır. Bu sınıflandırmaya göre robot tanımına, el ile idare edilen manipülatör ve sabit işlem sıralı robotlar da dahil edilmektedir.

Endüstride, robotlar hızla insanın yerini almaktadır. Robot teknolojisinin bir çok avantajı vardır. Bunlar ;

1. Robotlar sınırlı çalışma süreleri olmaksızın, verilen görevi 24 saat yerine getirmektedir.
2. İnsan sağlığını olumsuz yönde etkileyebilecek iş ortamlarında çalışabilirler.
3. İnsan gücünün yetmediği yerlerde çalışabilirler.
4. Hassas çalışma ölçüsü istenen yerlerde çalışabilirler.
5. Arıza halinde tamir edilebilirler. Normal bakım şartları yerine getirildiğinde arıza riski düşüktür.
6. El aparatı değişmesi suretiyle, aynı robot farklı işlerde kullanılabilir.
7. Uygun şartlarda çalışması halinde , mali bakımdan insana göre ekonomiktir.
8. İnsana göre daha hızlı hareket etme yeteneğine sahiptir [4].

Dezavantajları ise;

1. İlk tesis maliyeti yüksektir.
2. Kendi kendine karar verme yetenekleri yoktur.
3. Yaptıkları işe bağlı olarak kabiliyetleri sınırlıdır.
4. Çalışılan bölgede servisi yoksa, yedek parça ve onarım açısından, zaman ve maddi kayıplara neden olabilir.

Robotlar, mekaniği bakımından farklı serbestlik derecelerine sahiptirler. İnsan kolu beş serbestlik derecesine sahiptir. Robot kolların tasarımı insan kolu örnek alınarak yapılmaya başlanmıştır. Robotların mekanik yapısı gibi,

programlama teknikleri ve elektronik aksamaları da farklıdır. Tasarım yapılırken öncelikle, farklı çalışma şartları dikkate alınmalıdır.

Dünya 'da endüstriyel olarak kalkınmış birçok ülke, robot teknolojisine 1970 li yıllardan itibaren geçmiş olmasına rağmen, ülkemize robot teknolojisi ve robotik ile ilgili çalışmalar 1990 'lı yıllarda girmeye başlamıştır. Robotik ile ilgili çalışmalar üniversiteler, yüksek teknoloji enstitülerlerinde yapılmaktadır. Ülkemiz şartlarında üretim maliyetleri ve dolayısıyla satış fiyatları, arz / talep dengesi henüz tam olarak oluşmadığı için birçok alanda henüz kullanımı yaygınlaşmamıştır.

Ülkemizde robotik alanında çalışan çok az sayıda kuruluş bulunmaktadır. Bunların başında üniversiteler yer almaktadır. Bu konuda yapılan yüksek lisans ve doktora çalışmaları, araştırma projeleri robotik alanında önemli roller oynamaktadır. Üniversitelerimizin pek azında robotik anabilim dalı ve robotik laboratuvarları bulunmakta, pek çoğunda ise yeni kurulma aşamasındadır. Robotik konusunda taranan makale, tez ve araştırma projelere göre, ODTÜ, İTÜ, Gaziantep Ün., Erciyes Ün., Gebze Yüksek Tek. Ens. ön sıralarda yer almaktadır.

Robotik konusu elektrik, elektronik, bilgisayar, kontrol, haberleşme, mekanik vs. gibi pek çok bilim dalı ile bağlantılı olduğundan hazırlanacak projelerde ekip çalışmasına sıklıkla ihtiyaç duyulmaktadır.

Ülkemizde robotik konusunda yapılan çalışmalar incelendiğinde, hareket ve uzaktan algılama sistemleri, kontrol sistemleri, değişik amaçlara yönelik tasarımlar, robot dinamiği, robotların programlanması ve simülasyonu gibi konularda çalışıldığı görülmektedir.

Sanayileşme bakımından gelişmiş ülkelere bakıldığında, çalışılan konular açısından paralellik görülmekle beraber, özel amaçlı hazırlanmış uygulamaların daha geniş oranda yer aldığı dikkati çekmiştir. Yine diğer

lkeler incelendiğinde bu konu ile ilgili alıřan zel kuruluřların ve yan sanayinin fazlalığı grlmekte ayrıca niversiteler ve zel sektrn ortak projelere imza attığı dikkati ekmektedir.

Bu alıřmada  boyutlu uzayda hareket edebilen,  serbestlik dereceli, takım tezgahları ile uyumlu alıřabilecek, eėitim amalı kullanılabilecek bir robot kol tasarım ve imalatı, bilgisayar destekli kontrol ve simlasyonu gerekleřtirilmiřtir.



2. ROBOTLAR

Yeni teknolojinin kullanıldığı birçok gelişen ülkede üretim teknikleri değişmektedir. Günümüzde endüstriyel amaçlı olarak tasarlanmış robotlar, üretimde çeşitli aşamalarda insanın yerini almaktadır [2].

Endüstriyel robotlar, denetim birimindeki bilgisayar programı yardımıyla hareketlerini seri ve çevresindeki değişimlere bağlı olarak tamamlarlar.

2.1. Robotların Kullanım Alanları

Günümüzde robotlar birçok alanda kullanılmaktadır. Bunlar:

A. Yalnız yer değiştirmenin gerektiği uygulamalar

1. Makinaların yükleme ve boşaltılması
 - a. Takım tezgahları
 - b. Plastik parça imalatı
 - c. Pres döküm
 - d. Hassas döküm
 - e. Dövme
 - f. Fırınlara doldurulup, boşaltılması
 - g. Isıl işlem
 - h. Dökümhane işleri
 - i. Pres işleri
2. Malzeme manipülasyonu
3. İstifleme

B. Yer değiştirmenin ve işlem yapmanın gerektiği uygulamalar

1. Punta ve ark kaynağı
2. Mekanik, elektrik, elektronik ilgili parça montajı
3. Boya püskürtme
4. Kesme
5. El aletleri ile yapılan diğer işlemler

Robotlar bunların yanı sıra aşağıdaki uygulamalar için de kullanılmaktadır.

1. Deri ile ilgili işlemler
2. Ayakkabı imalatı
3. Kauçuk ile ilgili işlemler
4. Asbest ile ilgili işlemler
5. Gıda maddeleri ile ilgili işlemler
6. Kil ve çimento ile ilgili işlemler
7. Cam endüstrisi
8. Giyim endüstrisi
9. Ağaç endüstrisi

Robotlar, insanlar için tehlikeli ve zararlı olabilecek uygulama alanlarında da kullanılmaktadır. Bunlar ;

1. Kömür madenleri
2. Uzay çalışmaları
3. Sualtı çalışmaları
4. Radyoaktif malzeme manipülasyonu
5. Savunma sanayi
6. Boya sanayi

2.2. Endüstriyel Robotların Sınıflandırılması

Endüstriyel robotlar aşağıdaki gibi sınıflandırılabilir.

2.2.1. Tiplerine göre

1. Servo kontrollü robotlar
2. Servo kontrollü olmayan robotlar
3. Noktadan noktaya robotlar
4. Sürekli yörüngeli robotlar

2.2.2. Yeteneklerine göre

1. Sıra kontrollü robotlar
2. Öğretilebilir (play back) robotlar
3. Kontrollü yörünge robotlar
4. Adaptif robotlar
5. Akıllı robotlar

2.2.3. Koordinat sistemine göre

1. Kartezyen koordinatlı
2. Silindirik koordinatlı
3. Kutupsal (küresel) koordinatlı
4. Döner koordinatlı

2.2.4. Kuşaklarına (nesillerine) göre

1. Birinci kuşak
2. İkinci kuşak
3. Üçüncü kuşak [4].

2.3. Robotların Yapısı

Bir robotu üç bölümde sınıflandırmak mümkündür.

2.3.1. Manipülatör ve el aparatı

Robot dizayn edilirken, yapacağı iş göz önüne alınarak mekanik yapısı belirlenir. Bu yapıya manipülatör adı verilir. Manipülatöre gövde, omuz, dirsek, bilek, el, mafsallar, el aparatı ve denge ağırlıkları dahil edilebilir.

2.3.2. Aktivatörler

Aktivatörler, eklem hareketleri için kullanılan motorlarla hidrolik ve pnömatik silindirleri içerir. Aktivatörler, tahrik sistemi olarak da adlandırılabilir.

2.3.3. Kontrol sistemi

Kontrol ünitesi, mikroişlemci veya bilgisayar yardımıyla işlevini yerine getirir. Kontrol sistemi yapılacak işin tipine, hassasiyetine ve yükün ağırlığına bağlı olarak geri beslemeli veya geri beslemesiz olabilir [4].

2.4. Robotların Hareket Uzayı

Robotun hareket uzayı ne kadar büyük ise robot o kadar çok alana (noktaya) erişebilir.

Hareket uzayını belirleyen etkenler şunlardır:

1. Manipölatör ve el aparatının büyüklüğü
2. Serbestlik derecesi
3. Hareket açısı

Bu üç madde belirlenirken belli sınırlar içerisinde kalma zorunluluğu ortaya çıkabilir. Bunun sebebi robotun ağırlığının artması, ağırlık artışı sonucu aktivatör kuvvetlerinin artması, sistemin hızı ve ivmesinin değişmesidir.

Serbestlik derecesi artırıldığında sistem karmaşık hale gelir. Her bir serbestlik derecesi aynı bir tahrik ve kontrol ünitesine ihtiyaç doğurur. Bu işlem ise maliyeti artırır. Öteleme miktarını değiştirmekse sistemin ağırlık merkezinin büyük bir uzayı taramasına ve bu sebepten yataklara gelen kuvvetlerin artmasına neden olur [2].

2.5. Robotların Alt Sistemleri ve Fonksiyonel Yapısı

2.5.1. Kavrayıcı kol kinematiği

1. Üç boyutlu çalışma sahası ve iş parçası, alet, üretim gereci arasındaki ilişkinin oluşturulması
2. İş parçasının tutulması
3. İş parçasının durum emniyeti

2.5.2. Kontrol sistemi

1. İşlem sırasının saptanması, kontrolü ve izlenmesi
2. Yöneltilme gereci ve üretimle, sayısal bağlantı yapılması

2.5.3. Sürme

Hareketli tüm eksenler, taşımanın korunmasında istenen güç

2.5.4. Ölçme sistemi

Hareketli tüm eksenlerin hız ve durumun ölçülmesi

2.5.5. Sensörler

1. Fiziksel değerlendirilmenin ölçülmesi
2. Model, parça ve pozisyon sezilmesi için kullanılır [4].

3. BİLGİSAYAR DESTEKLİ KONTROL

Elektronikte teknolojinin gelişmesine paralel olarak en sık uygulanan kontrol yöntemlerinden birisi de bilgisayar destekli kontroldür. Sık karşılaşılan başka bir kontrol yöntemi de mikroişlemci ile kontroldür. Mikroişlemci ile kontrolde aynı program tekrar edilerek uygulandığı halde, bilgisayar destekli kontrolde kullanılan devre değiştirilmeksizin farklı kontroller yapılabilir.

Bilgisayar destekli kontrol, seri ve paralel porttan ya da sonradan takılan, istenilen amaca uygun olarak tasarlanmış kontrol kartları ile yapılabilir. Seri ve paralel portlar ile kontrolde, önceden standartlarla belirlenmiş sınırlar içerisinde kalarak kontrol yapılabilir.

3.1. I / O Birimi, Giriş ve Çıkış Portları

Giriş portu (Input Port) dış aygıtlardan gelen sinyalleri mikroişlemciye aktaran bir entegre devre (IC, Integrated Circuit)'dir. Çıkış portu (output port) ise mikroişlemcinin çıkış sinyallerini dış aygıtlara aktaran bir IC 'dir.

Giriş ve çıkış portlarının işlevleri genellikle bir birimde toplanır ki bu birime I/O birimi denir. I / O birimi bir mikroişlemci entegre devresi büyüklüğünde bir entegre devredir. I / O birimi veri kaynağından veri girmede ve çalıştırılan bir programın belirlediği veriyi dışarıya aktarmada mikroişlemci tarafından kontrol edilir.

Genel olarak I / O birimleri mikroişlemciler ile giriş ve çıkış aygıtları arası bağdaştırmayı (interface) sağlarlar.

3.2. Yollar

Mikroişlemci, bellek ve I / O entegre devreleri üç yol yardımı ile birbirlerine bağlanmıştır. Yol (bus) deyimi sayısal sinyaller için bir yol vazifesi gören ve aynı işleve sahip bir grup iletken teli veya iletim yolunu belirtmek için

kullanılmaktadır. Mikroişlemcilerde kullanılan üç ana yol bulunmaktadır. Bunlar özetle ;

a. Kontrol Yolu (Control Bus) :

Mikroişlemcinin zamanlama ve kontrol devrelerinde üretilen kontrol sinyallerini belleğe ve I / O birimlerine göndermek için kullanılan yoldur. Örneğin oku/yaz (read / write) sinyallerini belleğe, giriş ve çıkış portlarına taşır.

b. Veri Yolu (Data Bus) :

Veri yolu makine komutlarını ve verileri bellekten mikroişlemciye taşır. Aynı zamanda giriş / çıkış transferleri ile ilgili veriler de taşımak için kullanılır. Kontrol ve adres yollarından farklı olarak, veri yolu iki yönlüdür (bidirectional). Veri iki yönde hareket edebilir. Yani hem dışarıdan mikroişlemciye doğru hem de mikroişlemciden dışarıya doğrudur. Birçok mikroişlemci sisteminin veri yolu sekiz yoldan oluşur. Bu yüzden aynı anda 8 bit veya 1 bayt taşınabilir.

c. Adres Yolu (Address Bus) :

Adres yolu bellekteki bir yerin veya veri transferinde görev alan giriş / çıkış portunun adresini iletmekte kullanılır. ROM ve RAM bellekte saklanan her komut ve her parça bilginin bir adresi vardır. Bu adres daima 16 bit 'den oluşan bir ikilik sayıyı içerir. Programın çalışması sırasında verilen bir yerin içeriği gerekli olduğunda, mikroişlemci o yerin adresini adres yoluna koyar. Adres yolu verini saklandığı yere ulaşmakta kullanılan adresi iletmekte kullanılır. Ulaşılan yerin içeriği daha sonra veri yoluna konur ve bu yolun içeriği mikroişlemciye okunur. eğer bir veri RAM belleğe depolanacaksa belirtilen yeri seçen adres kodu mikroişlemci tarafından adres yoluna konur. Daha sonra gönderilen yaz komutu sinyali veri yolundaki bilginin bir kopyasının belirlenen yere yazılmasını sağlar.

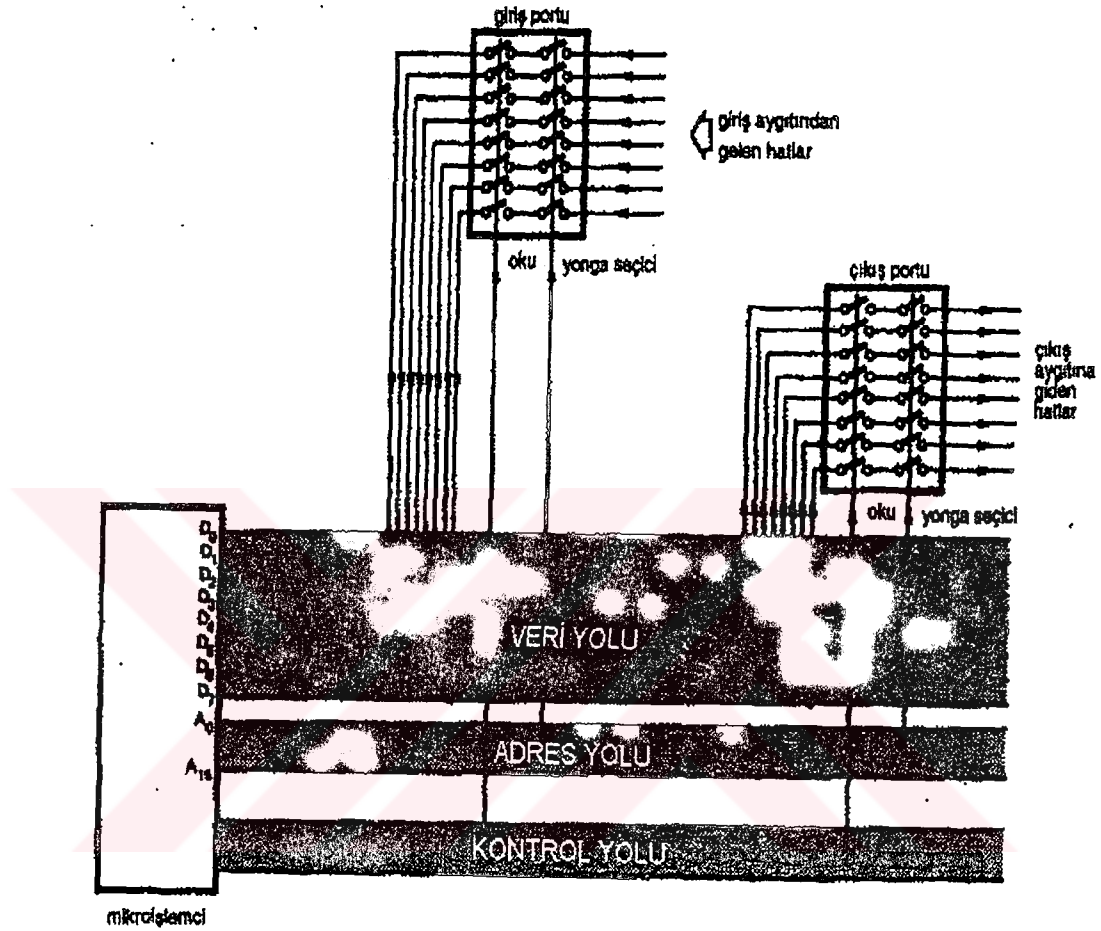
Adres yollarının çoğu 16 hattan meydana gelir. Her hat 0 veya 1'den oluşan bir adres biti taşır. Bundan dolayı söz konusu hattın $2^{16} = 65\ 536$ değişik durumu mevcuttur. Bunun anlamı 16 adres hattı kullanılarak 65 536 tane farklı saklama yerine ve giriş/çıkış aygıtına ulaşılabilmesidir.

3.2.1. Paralel I / O portları

Adresleme şekli, I / O portlarına yazma ve I / O portlarından okuma, bellek için yapılanların çok benzeridir. 8 bitlik sözcükler veya 16 bitlik adresler, veri ve adres yolları boyunca paralel olarak dağılmaktadır. Bazı giriş çıkış aygıtları seri şekilde olan ikilik veriler üretmektedir. Böyle durumlarda, seri girişler mikroişlemci sistemine verilmeden önce paralel forma dönüştürülmek zorundadır. Üniversal asenkron alıcı verici (UART, Üniversal Aysnchronous Receiver Transmitter) olarak bilinen seriden-paralele ve paralelden-seriye dönüşüm yapan IC 'ler bulunmaktadır.

Şekil 3.1 'de bir giriş kapısından verini nasıl okunduğunu ve bir çıkış portuna verinin nasıl yazıldığını göstermektedir. Kolay olması bakımından bir I / O portunun içerisindeki devreler sekiz anahtardan meydana gelen iki adet anahtar grubu şeklinde gösterilmiştir. Giriş portundan bir veri verileceği zaman, söz konusu giriş portu MPU tarafından adres yolu kullanılarak adreslenir. Örneğin en yüksek düzeyli adres biti çip seçme (CS, Çip Select) sinyalini taşır. Bu sinyal anahtar gruplarından birinin kapanmasını sağlar. MPU tarafından bir kontrol yolu ile gönderilen oku (RD, Read) sinyali ikinci anahtar grubunu kapatır. Böylece giriş aygıtının paralel veri hatları mikroişlemcinin veri hatlarına bağlanmış olur. Mikroişlemci gerekli olan yerin adresini adres yolunu 16 hattına koyar. Adres hatları hangi bellek çipinin seçilmesi gerektiğini ve seçilen çipte hangi yerin adreslenmesi gerektiğini belirtir. MPU tarafından veri yoluna konan verinin bir kopyası çıkış portuna verileceği zaman, önce çıkış portuna adres yoluna konan adres (ve çip seçici sinyal) ile erişilir. Sonra yaz kontrol sinyali gönderilir. Bu iki sinyal (çip

seçici ve yaz kontrol) iki tane anahtar grubunu kapatır ve böylece veri yolunda 8 tane hat paralel çıkış aygıtının veri hatlarına bağlanmış olur.



Şekil 3.1. Verinin bir giriş portundan nasıl okunduğunu ve bir çıkış portuna nasıl yazıldığını gösteren diyagram

Oku komutu için, giriş portuna adres yolundaki adres ile erişilir. Sonra, giriş aygıtına kontrol yolu yardımı ile oku sinyali MPU tarafından gönderilir. Daha sonra, giriş aygıtındaki veri, veri yoluna yerleştirilir. Yaz komutu için çıkış portuna adres yolu ile erişilir. Kontrol yolu ile gönderilen yaz sinyali 8 veri yolu hattını çıkış kapısına bağlar [6].

3.3. Kart Bağlantı Uçları

Hazırlanan kart, IBM uyumlu 80XXX serisi mikroişlemci kullanan sistemlerde boş bir ISA slota, prototip kart olarak takılmak suretiyle çalıştırılmaktadır.

Çizelge 3.1. ISA slottaki pin bağlantıları.

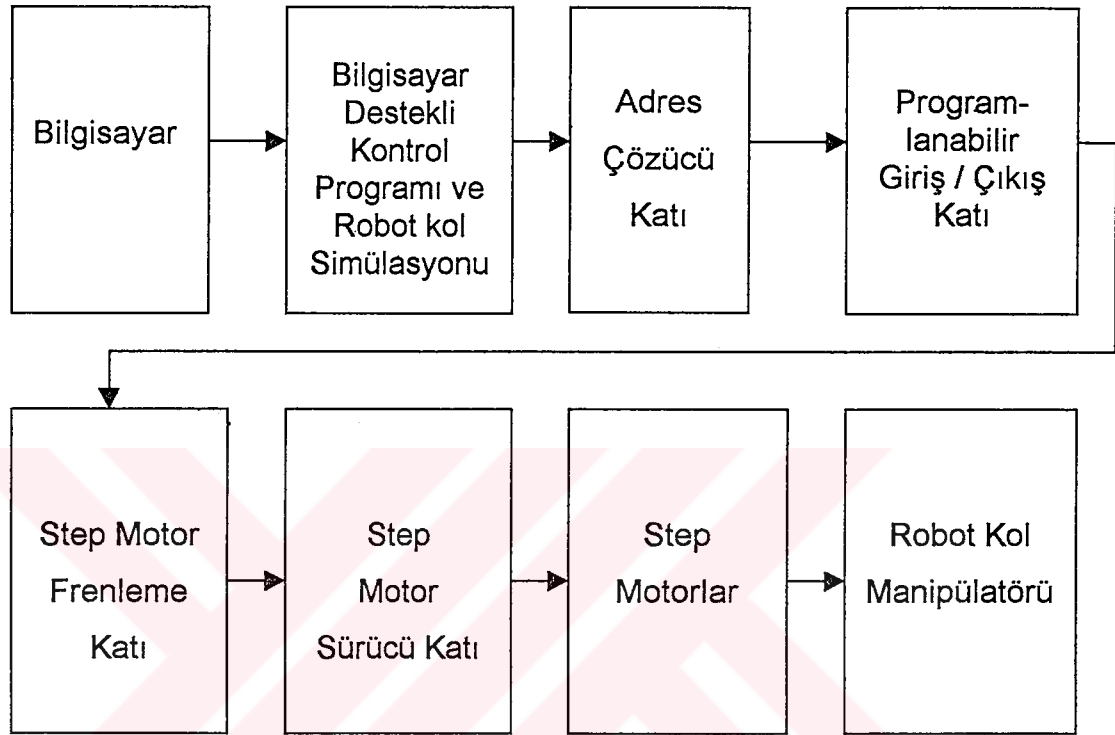
SLOT PIN	SIGNAL	SLOT PIN	SIGNAL
A1	---	B1	---
A2	D7	B2	RESET
A3	D6	B3	+5 V
A4	D5	B4 - B9	---
A5	D4	B10	GND
A6	D3	B11	---
A7	D2	B12	---
A8	D1	B13	I/O $\overline{WR}$
A9	D0	B14	I/O $\overline{RD}$
A10	---	B15 - B28	---
A11	$\overline{AEN}$	B29	+5 V
A12 -A21	---	B30	---
A22	A9	B31	GND
A23	A8		
A24	A7		
A25	A6		
A26	A5		
A27	A4		
A28	A3		
A29	A2		
A30	A1		
A31	A0		

Bilgisayar ana kartındaki 16 bitlik ISA slotun 62 uçlu soketinde bulunan bağlantı uçları tüm kartlarda standart olarak kullanılmaktadır. Bu kart ile ana karttan veri, adres, kontrol iletim kartları ve besleme gerilimi (RD, WR, AEN, Reset, +5 V, GND) sağlanmaktadır. Standartlarda belirtilen bağlantı uçları Çizelge 3.1. 'de gösterilmektedir. Kartın malzeme yüzeyi A, lehim yüzeyi B olarak kabul edilerek, bilgisayar kasasının yan tarafına gelen kısmını her iki yüzünde de 1 numara olarak tasarlanmıştır.



4. GERÇEKLEŞTİRİLEN SİSTEM

Gerçekleştirilen sistemin genel yapısı, aşağıda blok şema ile gösterilmiştir.



Şekil 4.1. Kurulan sistemin blok şeması

Tasarlanan katlar tek bir kart üzerinde toplanmıştır. Bu kart bilgisayarın 16 bitlik ISA slotuna yerleşebilecek yapıdadır. Bilgisayar slotlarında kullanılan adres ve dataların farklı yüzlerde olması sebebiyle yapılan kartın çift yüzlü olması gerekmiştir. Bu kart sayesinde 48 bitlik giriş/çıkış kontrolü sağlanabilmektedir. Kart üzerinde dört adet, iki kutuplu step motor sürücü devresi bulunmaktadır. Hazırlanan mekaniğin prototip olması nedeniyle ağırlık problemi söz konusu olmadığından geri beslemeye ihtiyaç duyulmamıştır.

Tasarlanan kart, IBM ve IBM uyumlu 80XXX serisi CPU (Central Process Unit) kullanan sistemlerin 16 bitlik ISA slotlarına takılabilmekte ve yüksek

seviyeli programlama dillerinden (Basic, Pascal, C++, vs.) herhangi biriyle programlanabilir.

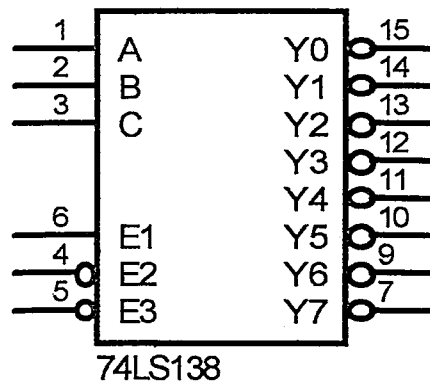
Kart üzerinde dört adet motor sürücü devre hazırlanmış fakat üç tanesi kullanılmıştır. Bir step motorun kontrolü için dört bit kullanılmıştır. Kart üzerindeki bir bit de frenleme için kullanılmıştır. Geri kalan 35 bit daha sonradan eklenebilecek uygulamalar için ayrılmıştır.

Tasarlanan sistemi oluşturan bölümler aşağıda açıklanmıştır.

4.1. Adres Çözücü (Kod Çözücü, Yol Çoklayıcı) Katı

Okunacak ya da içine yazılacak hücreleri seçen, sistemin önemli bir parçası kod çözücülerdir. Bu özel devre, çoktan bire kod çözücü (many - to - one - decoder), kod çözücü matris veya basitçe kod çözücü diye adlandırılır. n giriş hücresi tarafından alınabilen 2^n tane mümkün ikilik giriş için, matrisin 2^n çıkış hattından seçilen tek çıkışı olacaktır.

Sayısal makinalarda kullanılan üç tip kod çözücü matris çalıştıktan sonra, kod çözücü devresini n girişli ve 2^n çıkışlı bir kutu gibi çizilebilir. Bundan kutunun içindeki üç tip devreden bir tanesinin kullanıldığı anlaşılır. Genellikle tamamlayıcı alınmamış girişler kod çözücüye bağlanır ve ters çeviriciler de kod çözücü paketinin içinde bulunur.



Şekil 4.2. 74LS138 Bağlantı uçları [3].

Bu üç girişli kod çözücü $2^n = 2^3 = 8$ çıkış hattına sahiptir. Kullanılan 74LS138 in bağlantı uçları Şekil 4.2. 'de, doğruluk tablosu da Çizelge 4.1. de görülmektedir

Çizelge 4.1. 74LS138 Doğruluk tablosu

E1	E2	E3	C	B	A	Y0	Y1	Y2	Y3	Y4	Y5	Y6	Y7
1	0	0	0	0	0	L	H	H	H	H	H	H	H
1	0	0	0	0	1	H	L	H	H	H	H	H	H
1	0	0	0	1	0	H	H	L	H	H	H	H	H
1	0	0	0	1	1	H	H	H	L	H	H	H	H
1	0	0	1	0	0	H	H	H	H	L	H	H	H
1	0	0	1	0	1	H	H	H	H	H	L	H	H
1	0	0	1	1	0	H	H	H	H	H	H	L	H
1	0	0	1	1	1	H	H	H	H	H	H	H	L
0	X	X	X	X	X	H	H	H	H	H	H	H	H
X	1	X	X	X	X	H	H	H	H	H	H	H	H
X	X	1	X	X	X	H	H	H	H	H	H	H	H

Adres çözücü devrede 74LS138 entegresinden iki adet kullanılmıştır. CPU dan 8255 entegresine gelen adresin ayarlanması adres çözücü devre ile sağlanmıştır [3].

Bilgisayara takılan ek donanımlar rasgele adreslere bağlanamazlar. Çizelge 4.2. 'de IBM uyumlu sistemlere bağlanacak donanımlara ait için adres haritası verilmiştir. İlave edilen ek donanımların bu adreslere bağlanma standartlar gereğidir. Bu standartlara uyulduğu takdirde takılan kartlar sonucu adres çakışmaları ve sistemin hata vermesi ortadan kaldırılmış olur.

Çizelge 4.2. 'de görüldüğü üzere hazırlanan kart 300 - 31F (Hexadecimal) sınırları arasında çalışmak zorundadır. Bu sınırlar içinde çalışma yapabilmek

için seçilen adresin hem donanımdan hem de yazılımdan ayarlanması gerekir.

İkinci 74LS138 çıkışında jumper 'larla adres seçme imkanı vardır. Buna göre adres seçimi Çizelge 4.3. 'de verildiği gibidir.

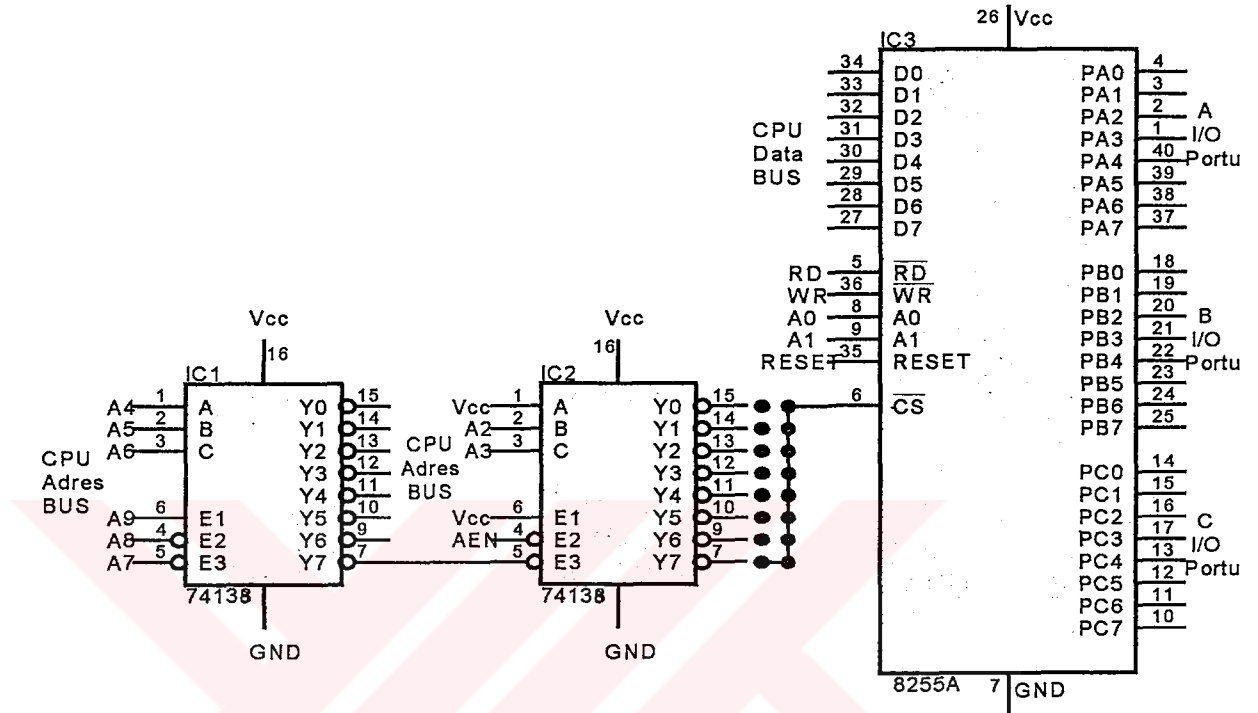
Çizelge 4.2. IBM / PC için giriş / çıkış adres haritası [7].

DESİMAL DEĞERİ	KULLANIM ALANI
0-15	8237A DMA Yongası
32-33	8259A Kesilme
64-67	8253 Zamanlayıcı
96-99	PPI 8255
128-131	DMA Sayfa yazmacı
160	NMI Maske yazmacı
512-527	Oyun Kontrolü
528-535	Genişletme ünitesi
760-767	Asenkron iletişimler (ikincil)
768-792	MIC 926 adresler sınırı
800-815	Sabit disk
888-895	Printer (LPT1)
896-908	SDLC İletişimler
896-905	Binary senkron iletişimler (ikincil COM2)
912-915	Demet
928-937	Binary senkron iletişimler (birincil)
944-959	IBM Monokrom ekran / Printer
967-991	Renkli / Grafik ekran
1008-1015	Disket sürücü
1016-1034	Asenkron iletişimler (Birincil COM1)
1936-1939	Demet (Adaptör 1)
2960-2963	Demet (Adaptör 2)
5008-5011	Demet (Adaptör 3)
9140-9143	Demet (Adaptör 4)

Çizelge 4.3. Adres çözücü devrede kullanılan adresleme yapısı [7].

A ₉	A ₈	A ₇	A ₆	A ₅	A ₄	A ₃	A ₂	A ₁	A ₀
1	0	0	1	1	1	x	x	X	x

Adreslerden $A_1 A_0 = XX$ 8255 kontrol girişine bağlı olup 8255 mod seçimi için kullanılırlar. Bu mod seçiminde A_1 ve A_0 den faydalanılır.

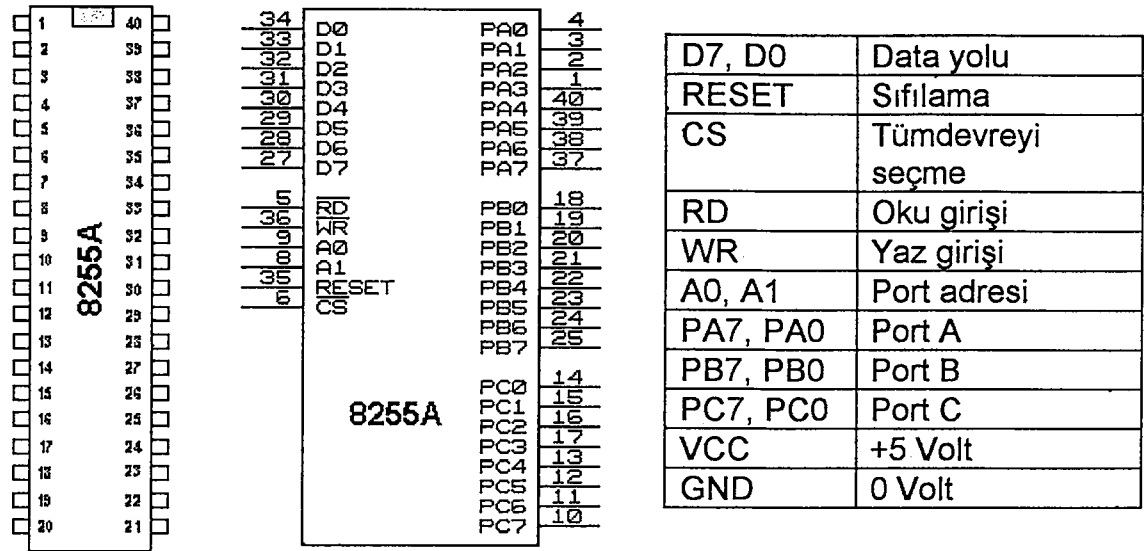


Şekil 4.3. Adres çözücü katı ve 8255 ile bağlantısı [7].

4.2. Programlanabilir I/O Elemanı 8255 'in Özellikleri

8255 entegresi, 24 adet giriş / çıkış ucuna sahiptir. TTL, CMOS ve INTEL in tüm mikroişlemci ailesiyle tam uyumlu olarak çalışır. Reset ayağı sayesinde, mikroişlemci sisteminin resetlenmesi ile doğrudan resetlenerek korunma yapısına sahiptir. Üç değişik modda çalışabilir. Genel amaçlı bir giriş çıkış entegresidir.

Şekil 4.4 ve Çizelge 4.3. 'de de görüldüğü üzere ayak bağlantıları ve görevleri şöyledir:



Şekil 4.4. 8255 Entegresinin ayak bağlantıları [7].

Çizelge 4.4. 8255 Entegresinin ayak numaraları ve görevleri

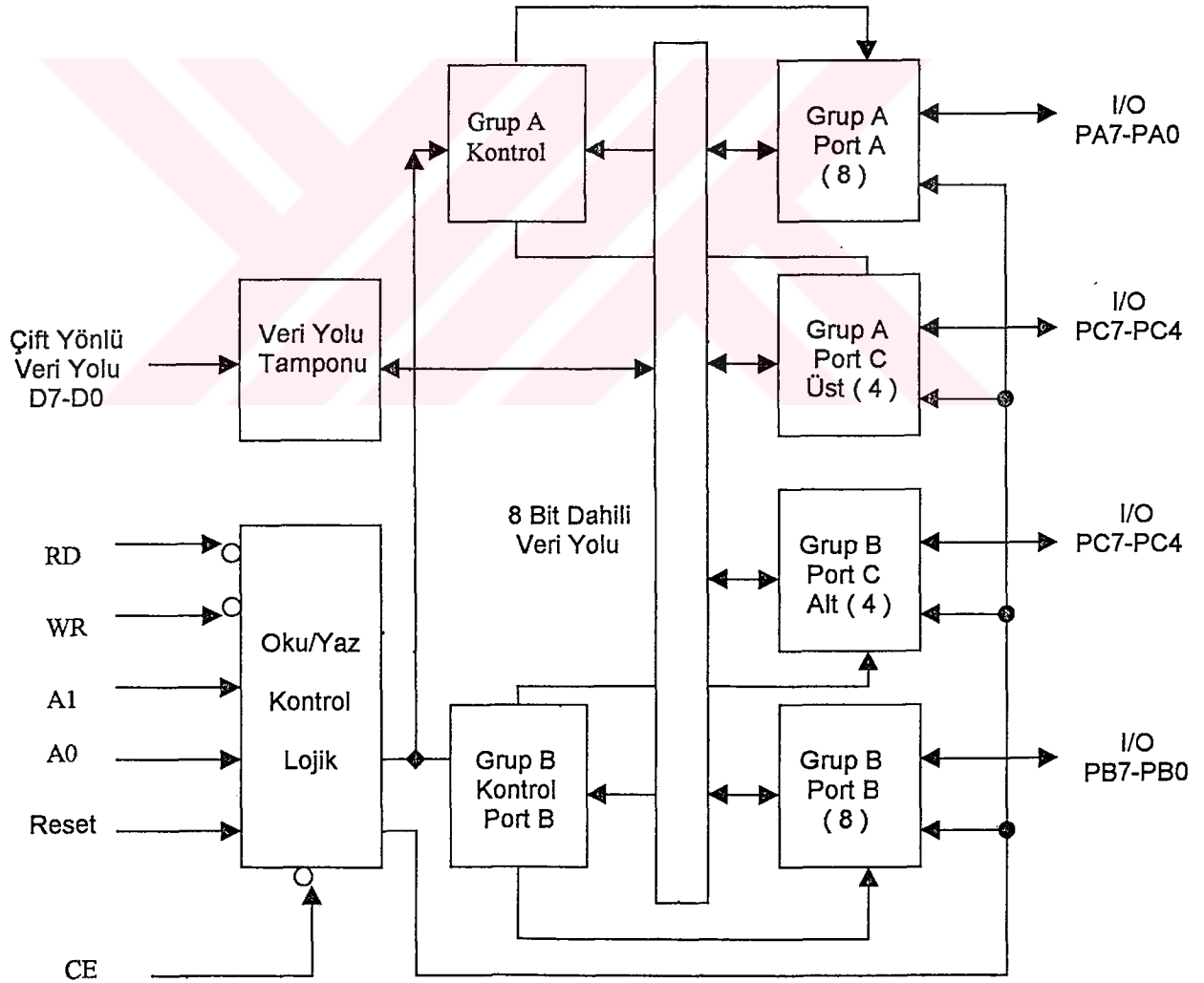
Ayak No :	Sinyal :	Görevi :
27,28,29,30,31,32,33,34	D7,D6,...,D1,D0	Veri Yolu (Datalar)
35	Reset	Sıfırlama
36	$\overline{WR}$	Yaz Girişi
5	$\overline{RD}$	Oku Girişi
8,9	A1,A0	Port Adresi
6	$\overline{CS}$	Çip Seçici
18,19,...,24,25	PB0,PB1,...,PB6,PB7	Port B
10,11,...,16,17	PC0,PC1,...,PC6,PC7	Port C
4,3,2,1,40,39,38,37	PA0,PA1,...,PA6,PA7	Port A
26	V _{CC}	+5 Volt
7	GND	0 Volt

8255 'in 6 nolu çip seçici ayağı kullanıcı için boş bırakılmış adrese, adres çözücü devre yardımı ile bağlanır. Adres çözücü devre tasarımı yapılırken, CPU 'dan gelen A0 ve A1 adresleri kullanılamaz. Çünkü bu adresler

doğrudan, 8255 'in programlanabilmesi için, 8255 'in A0 ve A1 (8 ve 9 nolu) ayaklarına bağlanır.

8255 'in 27,28,...,31,32 no 'lu ayakları (D7,D6,...,D1,D0) , 8 bitlik CPU veri yoluna bağlanır. Veri, CPU tarafından giriş ya da çıkış komutunun gönderilmesi ile, 8255 'e iletilir ya da 8255 'den alınır. 8255 'in programlanması da bu yola olur.

A0 ve A1 uçları 8255 'in kontrol girişlerine bağlı olup, mod seçimi için kullanılır. 8255'e ait temel işlem fonksiyon tablosu Çizelge 4.4. 'de verildiği gibidir.

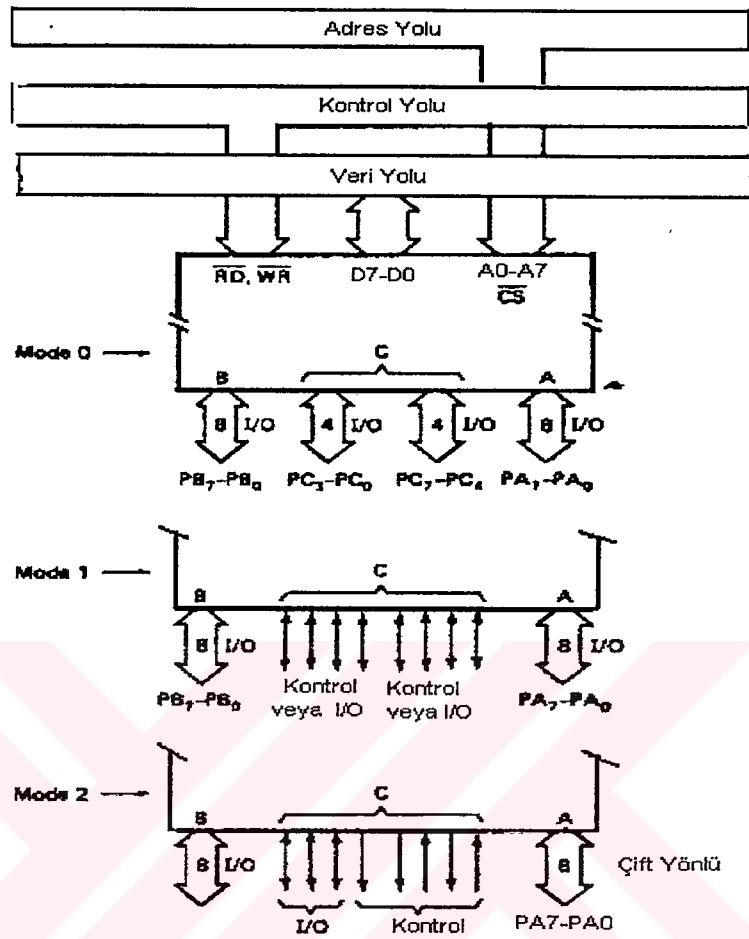


Şekil 4.5. 8255 'in iç bağlantı blok diyagramı [3].

8255 'in giriş/çıkış portları A ve B olmak üzere iki bölüme ayrılmıştır. A bölümü, A portunun tamamı ile C portunun C_{üst} olarak adlandırılan son dört (PC4, PC5, PC6, PC7) bitini içine alır. B bölümü ise, B portunu tamamı ile C portunun C_{alt} olarak adlandırılan ilk dört (PC0, PC1, PC2, PC3) bitini içine alır. A, B ve C portlarının kontrolü, üç farklı yöntemle (modda) yapılır. Bunlar mod0, mod1 ve mod2 olarak adlandırılır.

Çizelge 4.5. 8255 'in temel işlem fonksiyon tablosu [7].

A1	A0	RD	WR	CS	Giriş İşlemi (Okuma)
0	0	0	1	0	Port A → Veri Yolu
0	1	0	1	0	Port B → Veri Yolu
1	0	0	1	0	Port C → Veri Yolu
					Çıkış İşlemi (Yazma)
0	0	1	0	0	Veri Yolu → Port A
0	1	1	0	0	Veri Yolu → Port B
1	0	1	0	0	Veri Yolu → Port C
1	1	1	0	0	Veri Yolu → Kontrol
					Geçersiz Fonksiyonlar
X	X	X	X	1	Veri Yolu → Tristate
1	1	0	1	0	İllegal Durum
X	X	1	1	0	Veri Yolu → Tristate



Şekil 4.6. 8255 'e ait temel mod tanımlamaları ve yol arabirimi [3].

8255 'in çalışmasının daha iyi anlaşılması için kullanılan kontrol terimlerini bilinmesi gereklidir. Bunlar ;

$\overline{RD}$: Okuma işlemini belirtir. Bu sinyal geldiğinde mikroişlemci seçilen giriş / çıkış biriminden bilgi okur. Aktif durumu lojik "0" dır.

$\overline{WR}$: Yazma işlemini belirtir. Bu sinyal ile mikroişlemci seçilen giriş / çıkış birimine istenilen bilgiyi yazar. Aktif durumu lojik "0" dır.

Bu iki sinyal, bilgi ve kontrol kelimelerinin iç ve dış transferini yönetir. Bu blok CPU nun kontrol ve adres iletim hatlarından girişleri alır ve kontrol gruplarının her ikisini de yönlendirir.

RESET : Reset giriři, lojik “1” de aktif olur. Reset giriřinin “1” yapılması sonucunda A, B ve C portları ve denetim kütüğünün içerikleri sıfır ile yüklenmiř olur. Tüm kapılar çıkıř moduna kurulur.

$\overline{CS}$: Bu giriř ucundaki lojik “0” sinyali, CPU ile iletiřim saęlar. Lojik “0” sinyali, veri iletim hattı üzerinden 8255’in bir veriyi ya da kontrol kelimesini CPU ’ya göndermesini saęlar. Yani CPU ’nun 8255’den bilgi okumasını saęlar. Bu sayede 8255 ’in istenilen adreste çalışması saęlanır.

A0, A1 : A0 ve A1 giriřleri 8255 içindeki alt birimlerin sečilmesinde, okuma ve yazma giriřleri ile birlikte kullanılır. Bu sinyallere baęlı olarak A,B,C portlarından ya da kontrol kaydedicisinden birinin sečilmesini saęlar [7].

4.2.1. 8255 ’in çalışma modları

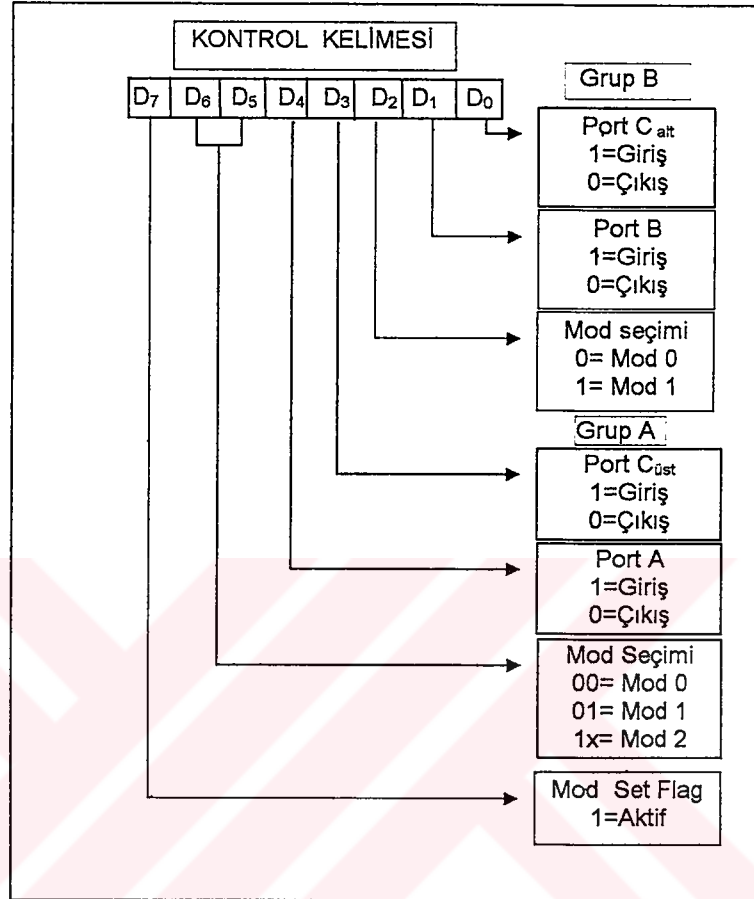
Sistemin program tarafından seçilen üç temel çalışma modu vardır. Bunlar;

1. Mod 0 : Temel giriř / çıkıř işlemleri
2. Mod 1 : Denetimli giriř / çıkıř işlemleri
3. Mod 2 : Çift yönlü yol işlemleri için kullanılır.

Reset ucu lojik “1” ’e geldięi zaman, bütün uçlar giriř moduna kurulur. Reset kaldırıldıktan sonra, 8255 hala giriř modunda kalır. Sistem, programın icrası sırasında tek bir çıkıř komutu kullanarak dięer modlardan herhangi birini seçebilir.

A portu ve B portu için modlar ayrı ayrı tanımlanabilir. Bu durumda C portu ikiye bölünür. C portunun iki ayrı bölümü A ve B portları ile beraber çalışır. Mod deęiřiklięi olduęu zaman bütün çıkıř kaydedicileri resetlenir.

Çizelge 4.6. Kontrol kelimesinin yapısı [7].



Kontrol kelimesinde D7 bitinin lojik “0” yapılması ile tek bit set / reset etme özelliği elde edilir. C portunun 8 bitinden herhangi bir tanesi set veya reset edilebilir. Bu işlem tek bir OUT komutu ile gerçekleştirilir. Bu özellik kontrol amaçlı uygulamalarda yazılım açısından kolaylık sağlayan bir özelliktir [8].

4.2.1.1. Mod 0 'da çalışma

Bu modun seçilmesi A ve B portlarının tüm kapıları ya giriş ya da çıkış olarak seçilir. C portunun kapıları ise, denetim kütüğünün D0, D1, D3, ve D4 bitlerine uygun olarak şartlanırlar. Bu modun uygulanmalarında sadece basit okuma / yazma işlemleri yapılır. Mod 0 'ın temel fonksiyonları şunlardır:

1. İki adet 8 bitlik port ve iki adet 4 bitlik port seçilebilir.

2. Herhangi bir port giriş yada çıkış seçilebilir.
3. Çıkışa aktarılan bilgiler çıkışta sürekli olarak tutulur.
4. Çıkışlar kilit (latch) devrelerine bağlıdır.
5. Girişler kilit devrelerine bağlı değildir.
6. Bu modda 16 farklı giriş/çıkış düzenlemesi mevcuttur [8].

4.2.1.2. Mod 1 'de çalışma

Bu modun seçilmesi durumunda A ve B portlarının tüm kapıları ya alıcı ya da verici olarak şartlanır. C portunun üst kısmı A, alt kısmı B portuna el sıkışma (handshaking) işlemleri için destek verir. Veri gönderme veya alma işlemleri, el sıkışma veya strobe hatları yardımı ile gerçekleştirilir. C portunun bazı bitleri strobe hatları olarak ayrılır. Mod 1 'in temel fonksiyonları şunlardır:

1. A grubu ve B grubu birbirlerinden bağımsız olarak düzenlenebilir.
2. Her grup 8 bitlik bir port ve 4 bitlik kontrol veri portu içerir.
3. 8 bitlik portlar giriş veya çıkış portlu olarak kullanılabilir.
4. Hem giriş, hem de çıkış kilit devrelere bağlıdır.
5. 4 bitlik portlar, 8-bitlik portları kontrol etmek için kullanılabilir.

Bu mod, çeşitli dijital sistemlerin birlikte kontrollü olarak çalışması sırasında bilgi alış-verişinde arabirim olarak kullanılır. Örneğin bilgisayarın printer veya klavye ile haberleşmesi bu modda sağlanır.

4.2.1.3. Mod 2 'de çalışma

Bu mod, çalışması sırasında dijital sistemler arası 8 bitlik bir bilgi yolunun çift yönlü (bidirectional) olarak kullanılmasını sağlar. Bilgi giriş / çıkış işlemlerinde, giriş ve çıkış bilgileri daima tutulur. Mod 2 'de, sadece 8 bitlik iki yönlü hat oluşturulur. C portunun 5 biti, A kapısının durum kontrolü için kullanılarak, Mod 1 'e benzer şekilde el sıkışma özelliğini ortaya koyar [8].

4.3. Step Motorlar

Step motor sürücü devresi anlatmadan önce step motorların özelliklerinin bilinmesi gereklidir. Herhangi bir mekanik sistemin hareketinin sağlanabilmesi için motorlara ihtiyaç vardır. Robot mekaniğinin uzayda herhangi bir noktaya ulaşabilmesi için, pozisyon kontrollü motorlar veya pnömatik, hidrolik sistemler kullanılır. Mekanik sistemin motor gücünden fazla güce veya daha hızlı harekete ihtiyaç varsa pnömatik veya hidrolik kontrollü sistemler kullanılır. Fakat pnömatik - hidrolik sistemlerde pozisyon hassasiyeti daha düşüktür. Pozisyon hassasiyetini arttırmak için yükün durumuna göre step motorlar, servo motorlar veya fırçasız DC motorlar kullanılabilir. Bu çalışmanın prototip olması nedeniyle robot kol kontrolü için step motorlar tercih edilmiştir.

Servo veya fırçasız DC motorlarda çalışma gerilimi veya yük değişikliğinde motor devri değişebilmektedir. Bu, motorlarda geri beslemeli bir kontrol sistemine ihtiyaç duyulur. Geri beslemesiz sistemlerde bu motorlar sabit pozisyonda kalmazlar. Motor gücü, yük torkundan fazla olan step motor uygulamalarında geri beslemeli sistemlere ihtiyaç duyulmayabilir. Böylelikle geri besleme problemi ortadan kaldırılabilir. Fakat uygulamada, robot kol son noktasının (elin) istenilen noktaya gittikten sonra kaymasını engellemek ve daha az torkla hareket için denge ağırlıkları ve fren sistemine ihtiyaç duyulur.

Step motorlar da diğer motor türleri gibi elektrik enerjisini, mekanik enerjiye çevirirler, Fakat step motorların diğer motor türlerinden farklı yönü, dijital kontrollü olmasıdır. Bu özelliğinden dolayı, bilgisayar destekli uygulamalarda sık sık karşımıza çıkar.

Step motorları, diğer motorlardan ayıran başka bir özellik ise uygulanan her pals için belli açıda dönmesidir. Her pals için aynı miktar dönen motor mili, tekrar harekete başlamak için, ikinci palsin uygulanmasını bekler. Her pals için aynı olay tekrarlanır.

Step motorlar; robot teknolojisi, bilgisayar donanımı (printer, floppy disk, hard disk, teyp back - up unit, ... , vs.), kamera, konum kontrollü sistemler ve tıp elektroniğinde kullanılır.

Step motorların kullanılmasının sağladığı avantajlar şunlardır:

1. Rotorun açısal hızı yeterince küçük olduğundan hareket sırasında adım kaybı olmaz. Açısal mesafe kesin olarak tanımlanırsa, motorun dönmesi uygun sayıda pals ile kontrol edilebilir.
2. Step motorların dijital kontrol sistemlerine uygun olması, bilgisayar kontrollü uygulamaları kolaylaştırır.
3. Step motorların ısınma problemleri çok azdır.
4. Step motorlar, yüksek torklarda düşük açısal hıza sahip olması, yüke yeterli torku sağlar.
5. Oluşan hatalar sadece step (basamak) hatasıdır. Aşırı yüklenme problemi olmadığı durumlarda, pozisyon hassasiyeti çok iyidir.
6. Step motorlar DC uyarmada geniş bir tutma torkuna (holding torque) sahiptir. Rotor hareketi sabitken, otomatik kilitleme (self locking) özelliği vardır. Rotor sadece uç gerilimi zamanla değiştiği sürece hareket eder.
7. Motorda açık çevrim kontrolünün olması sebebiyle takometre veya encoder 'ın kullanılmasına gerek kalmaz. Buna bağlı olarak da sistemin maliyeti aşağıya çekilebilmektedir. Servo motorlara göre de daha az maliyetlidir.
8. Servo motorlarda frenleme olmadığı takdirde hareket, mekaniki bir zorlama veya başka bir engelle karşılaşınca kadar devam eder. Sonuçta tehlikeli bir durum ortaya çıkabilir. Buna karşılık step motorlarda uygulanan gerilim ile step hareketi veya tutma (durdurma) işlemi gerçekleştirilebilir. Yani step motor yük durumuna bağlı olarak frenleme olmaksızın durdurulabilir.

Bu avantajlara karşılık step motorların da dezavantajları vardır. Bu tasarımda en sık karşılaşılanı, bu uygulamada da olduğu gibi, hız kontrolü CPU 'dan gelen clock darbeleri ile tetiklenmesi sonucu CPU 'nun ürettiği clock

darbelerinin hızından daha hızlı hareket sağlanamamaktadır. Belli bir hızdan daha hızlı hareket isteniyorsa harici bir pals jeneratörüne ihtiyaç duyulur. Motor hızı belli bir seviyenin üzerine çıktığında pozisyon hataları artar. Başka bir dezavantaj ise motorların ağırlık problemidir. Bu problem denge ağırlıkları ile çözülmeye çalışılır.

Step motorun alan sargıları (stator sargıları) birbirleri ile belli miktarda açılarla yerleştirilmiştir. Alan sargılarının belli bir düzen içinde enerjilendirilmesi zorunludur. Step motora pals uygulanırken, sayısal bilgi (dijital sinyal) bu düzene göre gönderilir [8].

4.3.1. Step motor türleri ve çalışması

Step motorların yapı bakımından üç ana gruba ayrılır. Bunlar :

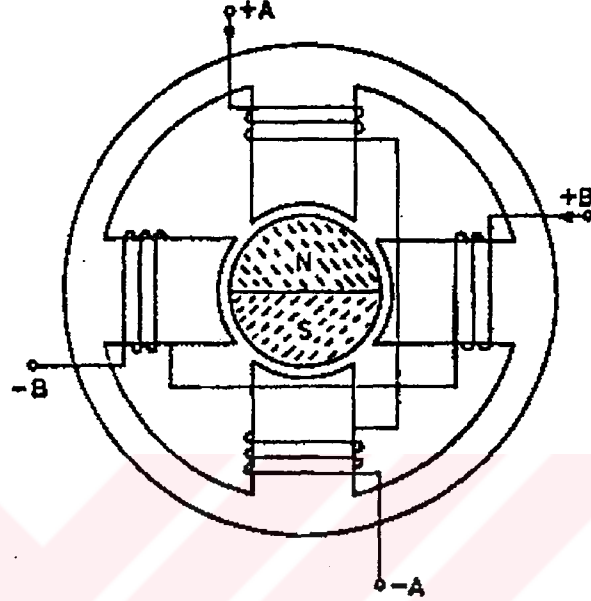
1. Sabit mıknatıslı step motorlar (PMSM, Permanet Magnet Step Motor)
2. Değişken relüktanslı step motorlar (VRSM, Variable Reluctance Step Motor)
3. Karma step motor (HSM, Hybrid Step Motor)

4.3.1.1. Sabit mıknatıslı step motor

Bu motorun rotoru, adından da anlaşılacağı gibi sabit mıknatıstan oluşur. Mıknatısın elektromagnetik alan içinde hareket etmesi prensibine göre çalışır. Rotoru uygulanan sinyalin durumuna göre her iki yönde de hareket edebilir, Stator sargılarının yönü, diğer motorlarda olduğu gibi önemlidir. Birbirine zıt yönlü sargıların oluşturduğu alanlar birbirlerini yok eder. Böyle durumlarda kutuplaşma olmadığından rotor hareket edemez.

Şekil 4.7. 'de görülen step motor iki kutupludur. Bu motor iki fazlı olarak da adlandırılabilir. 1. faza DC sinyali uygulandığında, sağ el kuralına göre, bobinin üst kısmında N, alt kısmında S kutbu oluşur. Bobinin S kutbu, rotorun N kutbunu çeker. 2. Faz enerjilendirildiğinde aynı olay bu faz için

tekrarlanır. Her iki fazda düzenli bir sıra ile enerjilendirilmeye devam edilirse, rotorun 90 'ar derecelik açılarla hareket ettiği görülür. Fazlar düzenli bir sıra ile enerjilendirildiğinde, rotor sürekli olarak dönmeye devam eder.



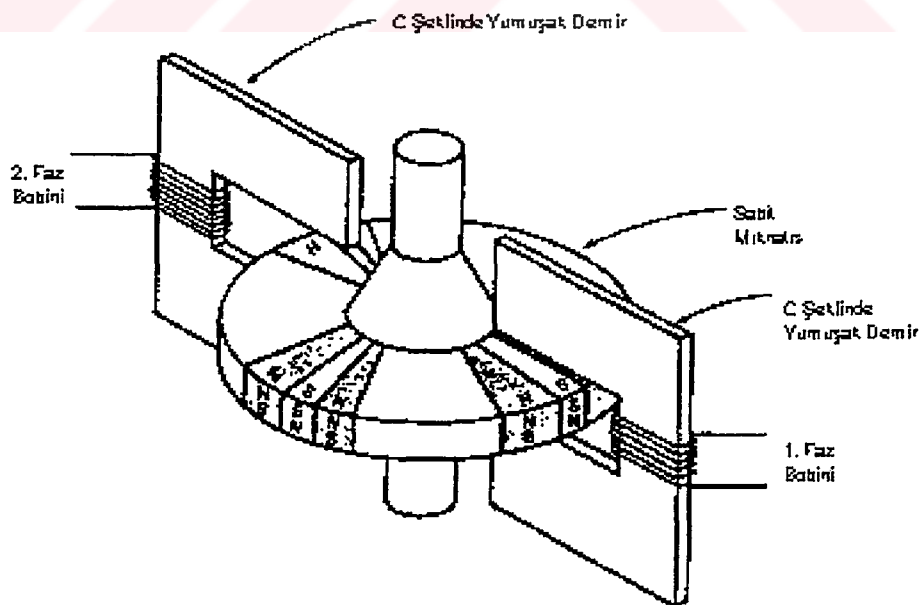
Şekil 4.7. Sabit mıknatıslı step motor [1].

Sabit mıknatıslı step motorlar genellikle, kare dalga pulslerle enerjilendirilir. Rotorda oluşan tork, stator bobinlerine uygulanan akımla orantılıdır. Uygulanan akım ne kadar fazla ise, oluşacak magnetik alan dolayısıyla tork da o kadar fazladır. Bu olay stator kutupları doyuma ulaşınca kadar devam eder. Kutuplar doyuma ulaştığında akım veya gerilim ne kadar arttırılırsa arttırılsın, oluşacak tork arttırılamaz.

Step motor ile DC motorlar arasında önemli farklar vardır. Bunlardan biri elde edilen tork türüdür. Stator bobinlerine DC sinyal tatbik edildiğinde, rotor sabit bir pozisyonda kalır. Buna tutma torku (holding torque) denir. Bu tork aynı zamanda rotoru sabit bir pozisyonundan kurtarmak için de gerekli olan torktur. Tutma torku motorun akım ve gerilimi ile ilgilidir. DC motorlar, step motorlarla aynı parametrelerle sahip değildir. DC motorun rotoru sabit pozisyonda tutulmak istenirse, bu işlem mekanik yöntemlerle yapılır. Step

motorun stator sargılarından akım geçmezken, rotorun harekete karşı gösterdiği dirence artık tork (residual torque) veya alıkoyma torku (dedent torque) denir. Bu tork, bobinlerin içinden geçen sürekli akı sebebiyledir. Düşük hızlarda ise bu tork, artık mıknatısıyetten dolayıdır. PMSM 'nin rotorunu harekete geçirebilmek için, belirli bir büyüklükte sinyal uygulanmalıdır. Rotor hacminin büyük olması, motora büyük atalet verir. Büyük atalette, rotorun ataletini yavaşlatır ve zorlaştırır. Motorun başka bir parametresi de maksimum adımlama oranı (bir saniyede atabileceği maksimum adım sayısı) 'dır.

Rotorun yukarıda anlatılan dezavantajlarından kurtulması için değişik bir tip ortaya atılmıştır. Bu tipe disk tipi PMSM denir. Bu tasarımla rotor ağırlığı ve hacim problemi çözülmüştür. Disk tipi rotorlu motorda, rotor doğal mıknatıstan yapılmış ve yaklaşık 1 mm kalınlığındadır. Diskin üzerinde 100 'den fazla ince mıknatıs bulunur. Bu tip motorun statoru C şeklinde yumuşak demirden yapılmıştır. C şeklinde her bir parça motorun bir fazını oluşturur. Fazlar sıra ile enerjilendirilerek rotorun dönmesi sağlanır.



Şekil 4.8. Disk tipi PM step motor [1].

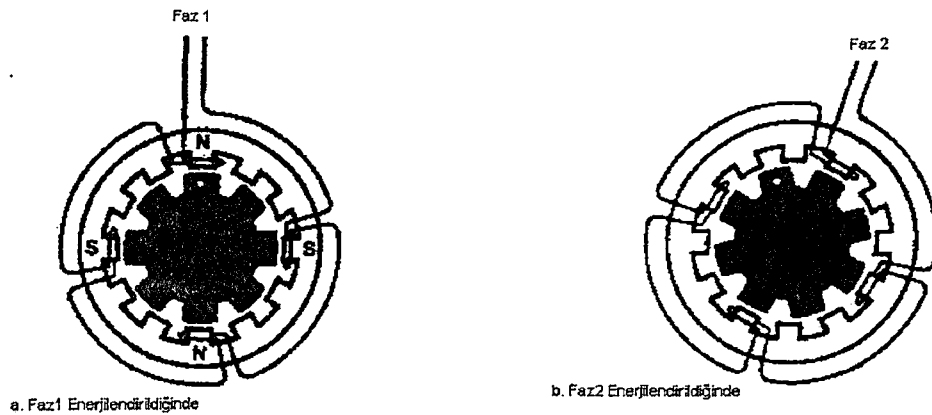
4.3.1.2. Değişken relüktanslı step motorlar

Değişken relüktanslı step motorlar, sabit mıknatıslı step motorlardan farklı, rotorun sabit mıknatıslı olmamasıdır. Rotor mıknatıslanmayan yumuşak çelikten yapılır.

Bu tip motorun rotoru üzerinde, dişler ve yarıklar bulunur. VRSM 'ler ikiye ayrılır :

1. Tek parçalı (single stack)
2. Çok parçalı (multi stack)

Şekil 4.9. 'de üç fazlı bir VRSM görülmektedir. Şekil incelendiğinde rotor dişleri stator kutuplarından daha azdır. Şekilde 12 stator kutbuna karşılık 8 rotor dişi bulunmaktadır. Stator kutupları $360 / 12 = 30$ 'ar derecelik aralıklarla ayrılmıştır. Rotor dişleri ise ; $360 / 8 = 45$ 'er derece aralıktır. Birinci faz enerjilendirildiğinde, rotor dişleri ile stator kutupları karşılıklıdır. Bu durumda ikinci faz ile dişler karşılıklı değildir. Birinci faz kesilip ikinci faz enerjilendirildiğinde, rotor dişleri sağ ve solundaki en yakın kutbun altına gelir. Yeni oluşturulan kutup saat ibresi (CW, Clock Wise) yönünde ise motor ileri, saat ibresi tersi (CCW, Counter Clock Wise) yönünde ise motor geri yönde dönüyor denir.

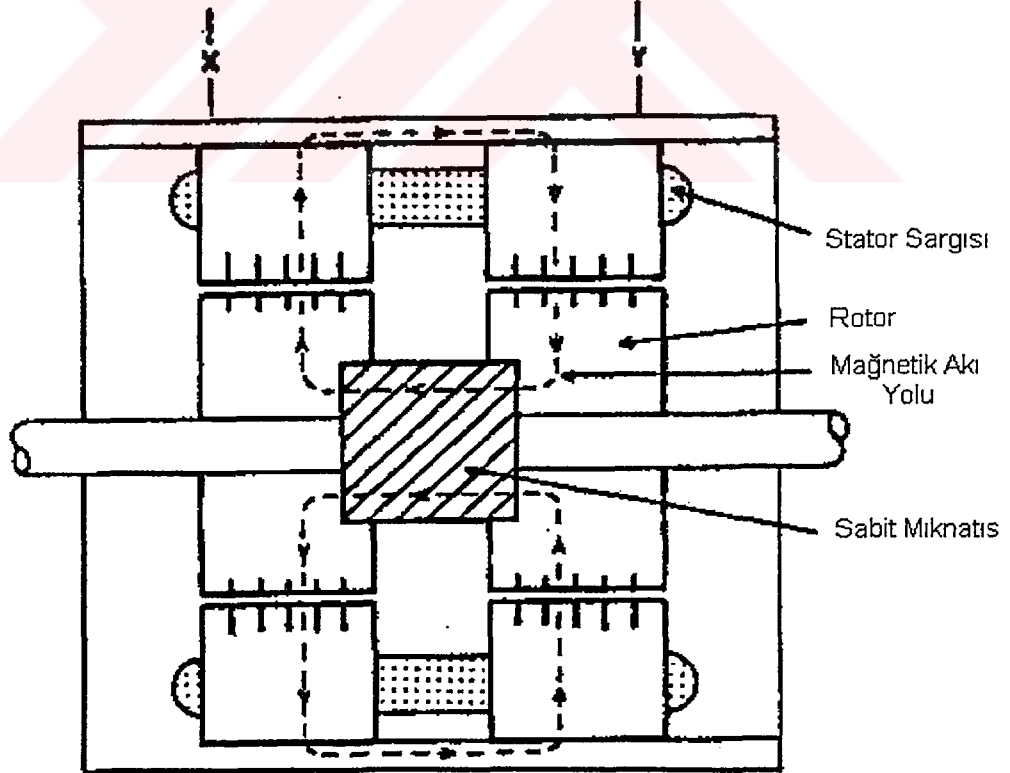


Şekil 4.9. Üç fazlı değişken relüktanslı step motorun yapısı.

VRSM 'nin rotoru mıknatısiyete sahip olmadığından, artık tork da yoktur. Fakat PMSM 'de olduğu gibi tutma torkuna sahiptir. VRSM 'de tork akımın karesi ile orantılı olarak değişir. Motorun rotorunda mıknatıs olmadığından dolayı, hafiflik ve küçük hacim sağlanır. Motor hacminin küçüklüğü, ataleti düşürür ve uygulanan sinyale daha hızlı cevap verir. VRSM 'ler, PMSM lerden daha hızlı kalkınır adımlar ve durur.

4.3.1.3. Karma step motor

Karma tip step motorlar, daha önce anlatılan iki tipin karışımı olarak düşünülebilir. Karma tip motorun rotorunda da sabit mıknatıs bulunur. Şekil 4.9 'da karma tip step motorun yapısı gösterilmiştir. Karma tip step motorlar da, stator kutup sayısı ile rotor diş sayısı birbirinden farklı olabilir. Bu özellik sayesinde daha küçük açılarda hareket yapabilme imkanı verir. Karma tip step motorda da çok parçalı rotor özelliği bulunabilir



Şekil 4.10. Karma tip step motorun şafta paralel kesiti [1].

Uygulamada kullanılan hepsi aynı tip ve karma step motor olup, hareket açısı 1.8° dir. Motor iki fazlı ve 20 stator diş sayısına sahiptir. Rotoru iki parçalı yapıdadır.

Karma tip motorun rotorunda üretilen tork, değişken relüktanslı motora göre daha fazladır. Hareket açısının küçüklüğü ve yüksek çıkış torku, uygulamalarda karma tip step motoru cazip hale getirmiştir.

Değişken relüktanslı motorlarda hareket açısı minimum 15° 'dir. Açının büyük olması hassa işlemler için yeterli değildir. Bu durumda hareket açısı, dişli sistemi yardımı ile düşürülür. VRSM 'lerin HSM 'lere göre tek avantajı, rotorunda sabit mıknatıs olmadığından, rotor ataleti daha düşüktür.

4.3.2. Step motor parametreleri

Step motorlar, genel olarak 1KW 'a kadar ve maksimum 3000 d / dk 'lık uygulamalarda tercih edilir. Step motorların, DC motorlardan farklı olan parametreleri aşağıda verilmiştir [8].

4.3.2.1. Adım tepkisi (Step response)

Step motorun en önemli parametrelerinden birisi tepki hızıdır. Motor sinyal uygulandıktan belli bir zaman sonra tepki verir. Bu zaman miktarına adım cevabı denir. Yüksüz motorlarda bu süre milisaniyeler civarındadır [8].

4.3.2.2. Adımlama oranı (Stepping rate)

Motorun 1 saniyede yapabileceği maksimum adım sayısına, adımlama oranı denir. Bu oran genellikle 300 ile 800 adım arasında değişir. Bu parametre motorun devrini bulmada kullanılır [8].

4.3.2.3. Adım açısı (Step angle)

Motora bir pals uygulandığında, rotorun hareket edebileceği açı miktarına adım açısı denir. İki kutuplu PMSM 'larda kutupların enerjilendirme özelliğine göre bu açı; 45 veya 90 derece olur. Kutup sayısı veya rotor diş sayısı arttırıldığında adım açısı küçülür.

Motor devri gerçekte iki parametreye bağlıdır. Bunlar; adım açısı ve adımlama oranıdır. Rotorun ortalama devri n (d / dk), adım açısı Y mekaniki derece olsun. Bu iki parametre arasındaki ilişki $n = Y / 6$ 'dır. Bu ifade, Y saniyedeki adım sayısıdır (step / second). Örneğin adım oranı 12 adım / sn. , adım açısı 45° olan bir motorun dakikadaki devri;

$$n = 45 \times \frac{12}{6} = 90 \text{ d / dk. 'dır.}$$

Step motorlarda uygulanan palsler ne kadar hızlı olursa, adımlama hızı da o kadar fazla olur. Adım hızındaki artış ve atalet, rotorun istenilen pozisyondan farklı bir pozisyonda durmasına neden olabilir. Bu duruma sınır aşımı veya sarkma (over shoot) denir. Bu olay gerçekte bir miktar devam eder. Fakat iki kutup arasındaki magnetik çekim rotoru olması gereken yere getirir. Bir sonraki harekette de bu olay tekrarlanır. Bir pals uygulandıktan sonra rotorun durgun hale gelmesi için geçen zamana durulma zamanı (setting time) denir. Bazı uygulamalarda minimum sarkma ve çok hızlı durma gerekebilir. Bu durumda sönümlendirme (frenleme) işlemine gerek duyulur. En çok kullanılan sönümlendirme metotları şunlardır :

1. Mekaniki sönümlendirme
 - a . Slip clutch (balatalı) sönümlendirme
 - b . Viscous (sıvılı) sönümlendirme
2. Elektriki sönümlendirme
 - a. Dirençle sönümlendirme
 - b. Kapasitif sönümlendirme

- c. İki fazlı sönümlendirme
- d. Ters torkla sönümlendirme [8].

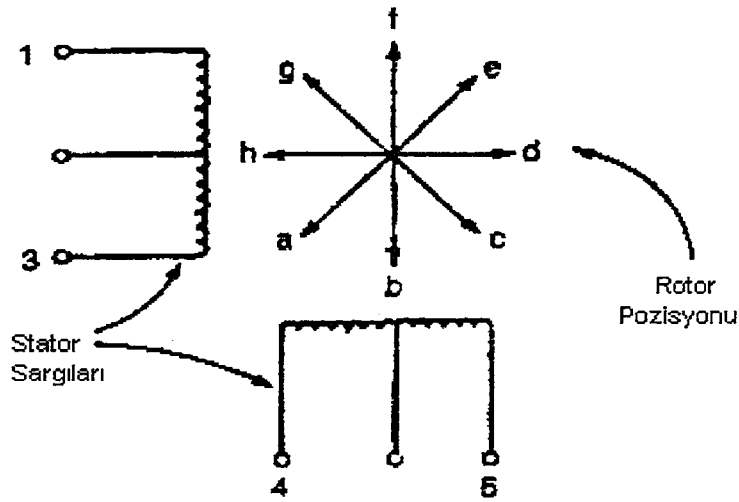
4.3.3. Step motor tahrik modları

Step motorlar genel olarak üç yöntemle tahrik edilir :

1. İki fazlı tahrik,
2. Üç fazlı tahrik,
3. Dört fazlı tahrik.

Bazı step motorlar, orta uçlu stator sargısına sahiptir. Bu durumda orta uçlu stator sargısına sahip iki fazlı step motorlar, orta ucu da kullanılarak dört fazlı tahrik edilebilir.

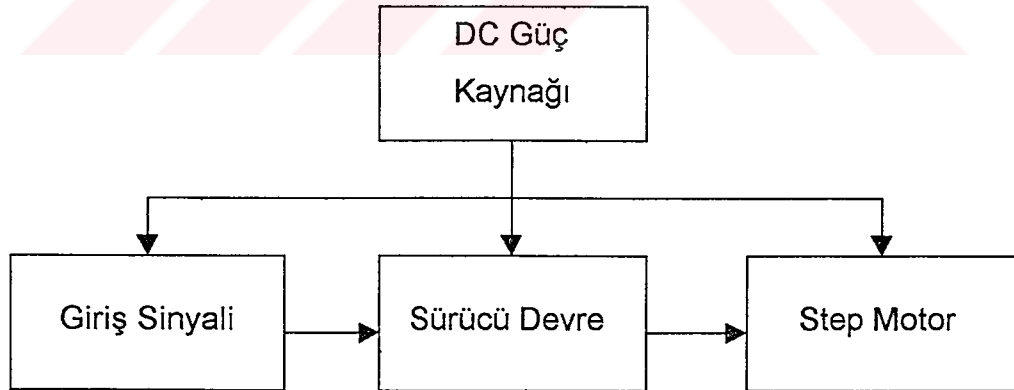
Uygulamada kullanılan motor iki fazlıdır. İki fazlı motorların tahrik edilmesi, dört fazlı motorlara göre daha zordur. Sebebi, stator direncinin iki kat daha fazla olmasıdır. Giriş akımı sabitse, güç yarıya düşer. İki fazlı motorların sürücü devre tahrikinde en yaygın olan metot, aynı anda iki fazın sürülmesi metodudur. Her fazın tek tek sürülmesi metoduna dalga tahriki de denir. İki fazın aynı anda sürülmesi metodunda, her iki sargı aynı anda enerjilendirilir. Şekil 4.10 'da görüldüğü gibi, ilk adımda 3 - 1 ve 6 - 4 stator bobinlerinde, 3 - 1 sargısında akım bir yönde akar. İkinci adımda stator bobinindeki akım ters yönde akar. Bu durum 1 - 3 şeklinde gösterilir [8].



Şekil 4.11. İki fazlı step motorda yapılan tahrik [1].

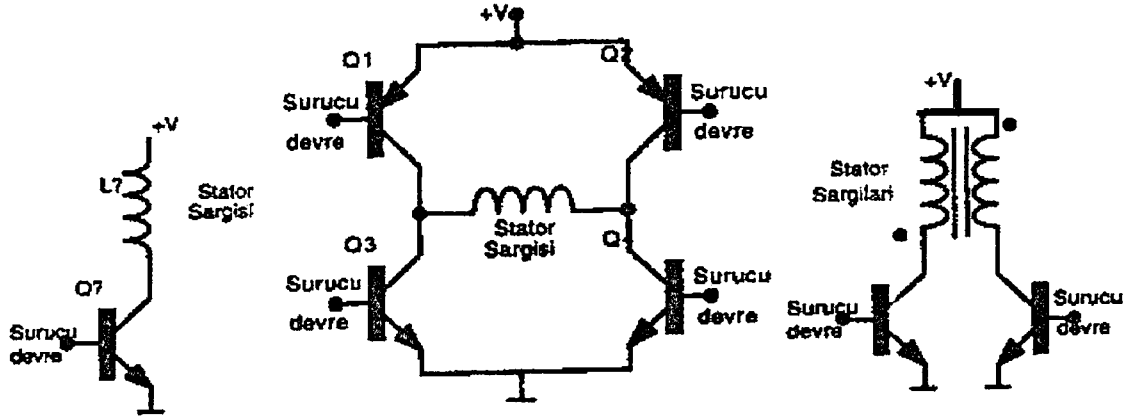
4.4. Step Motor Sürücü Katı

Şekil 4.12 'de temel bir step motor sistemi görülmektedir. Sistemde DC kaynak, darbe jeneratörü destekli DC giriş sinyali, sürücü devre ve step motor bulunmaktadır.



Şekil 4.12. Step motor kontrol sistemi [8].

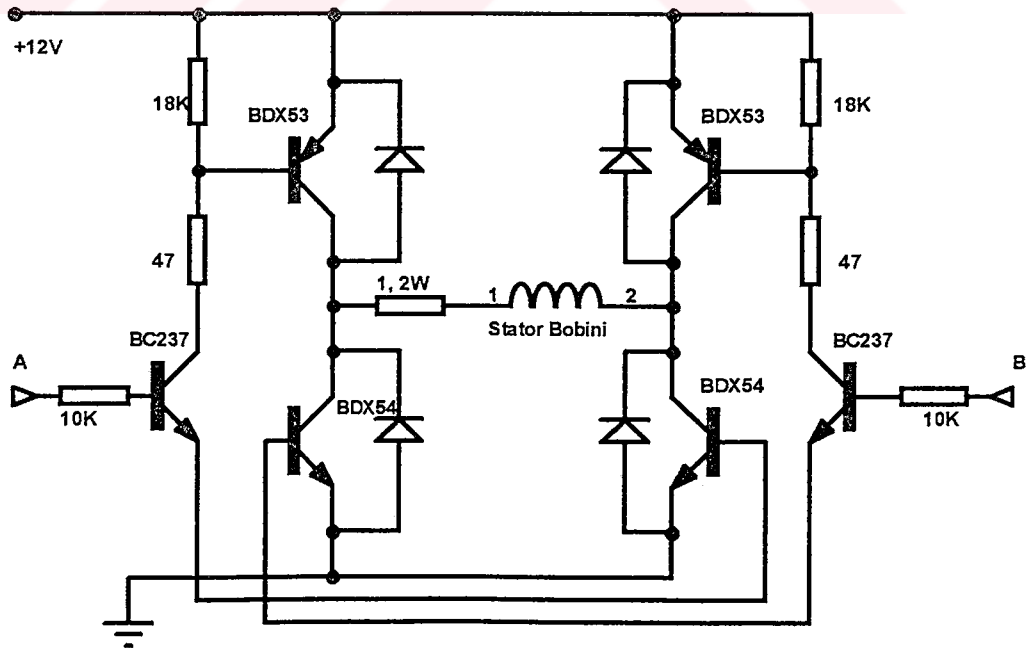
Step motor sürücü devreleri unipolar ve bipolar sınıflandırılır. Unipolar sürücü devrelerde, stator bobinlerinden tek yönde akım geçerken, bipolar sürücü devrelerde akım geçişi iki yönde gerçekleşir.



Şekil 4.13. Unipolar (tek kutuplu) ve bipolar (iki kutuplu) sürücü devreler [8].

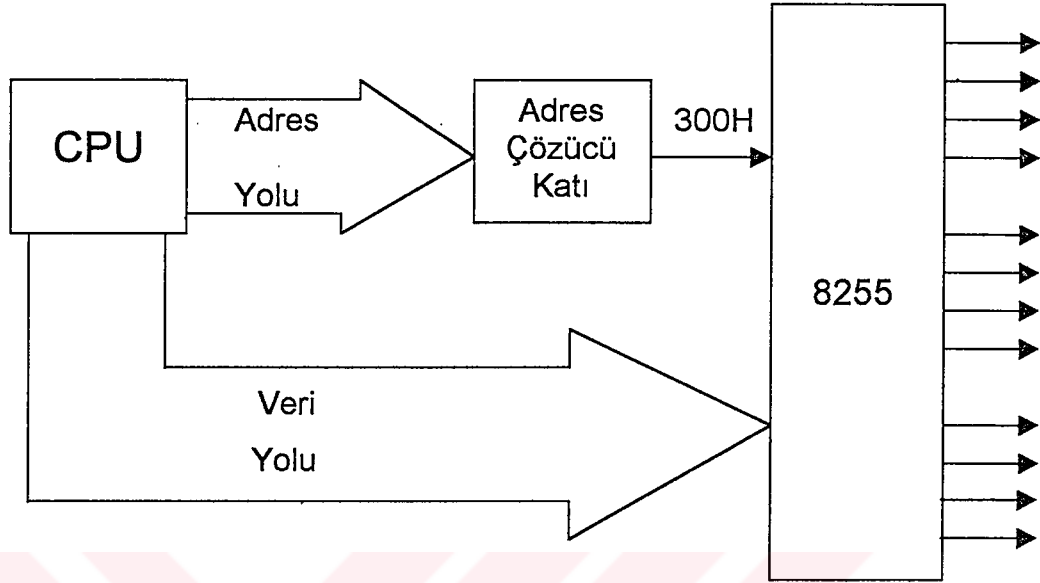
Orta uçlu step motorlarda unipolar sürücü devreler, orta uçlu olmayan step motorlarda ise bipolar sürücü devreler tercih edilir. Uygulamada kullanılan sürücü devre bipolar bir sürücü devredir.

Şekil 4.14 'de gösterilen motor sürücü devreden, her motor için iki adet kullanılmıştır.



Şekil 4.14. Hazırlanan step motor sürücü devre.

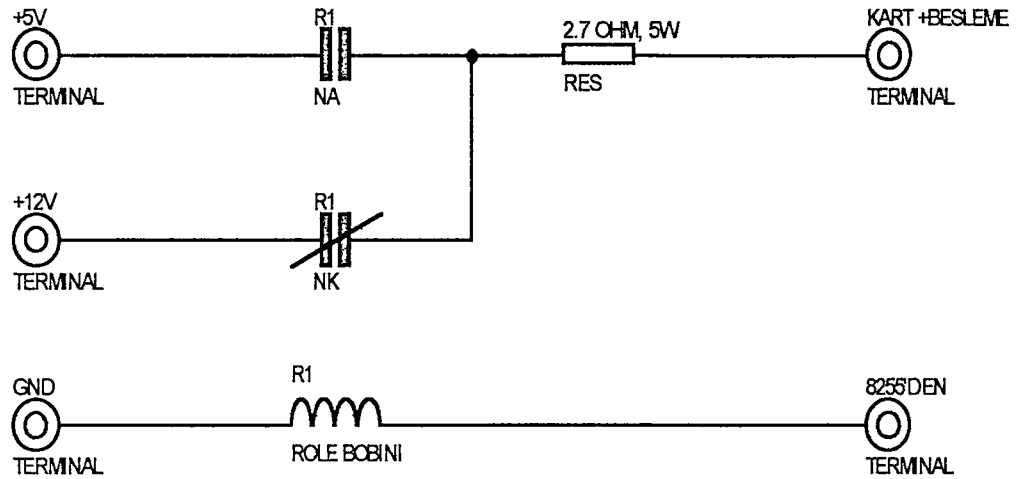
Motor sürücü kartına kadar olan kısmın blok şeması Şekil 4.15 'de görülmektedir.



Şekil 4.15. Motor sürücü katına kadar olan katların blok şeması

4.5. Step Motor Frenleme Katı

Robot kolun son noktasının, istenilen noktadan başka bir noktaya mekaniki ağırlıklar sonucu gitmesini engellemek amacıyla motor frenleme katı hazırlanmıştır. Bu katta sinyal bilgisayar kontrollü olarak hazırlanan kart üzerinden 8255 'in bir biti kullanılarak sinyal gönderilmiştir. Frenleme sinyali programdan gönderilmeden önce devrenin besleme gerilimi bilgisayarın içindeki güç kaynağının DC 12V ucu olduğu halde, frenleme sinyali gönderildikten sonra bir röle yardımıyla bilgisayarın içindeki DC 5V 'luk besleme sinyali kullanılmaktadır. Frenlemenin 5V 'la yapılma sebebi kol frenlenirken motorların ısınmasını engellemektir. Frenleme için 5V 'luk besleme sinyali gönderildiğinde motorların rotorları kilitlenmektedir. Frenleme sinyali iptal edilene kadar rotorların kilit durumu devam eder. Uygulanan gerilim direnç kullanılarak da düşürüldüğünden, frenlemede motorların ısınması ve kolun ulaştığı noktadan kayması söz konusu olmamaktadır.



Şekil 4.16. Step Motor Frenleme Katı

4.6. Robot Kol Kontrol ve Simülasyon Programı

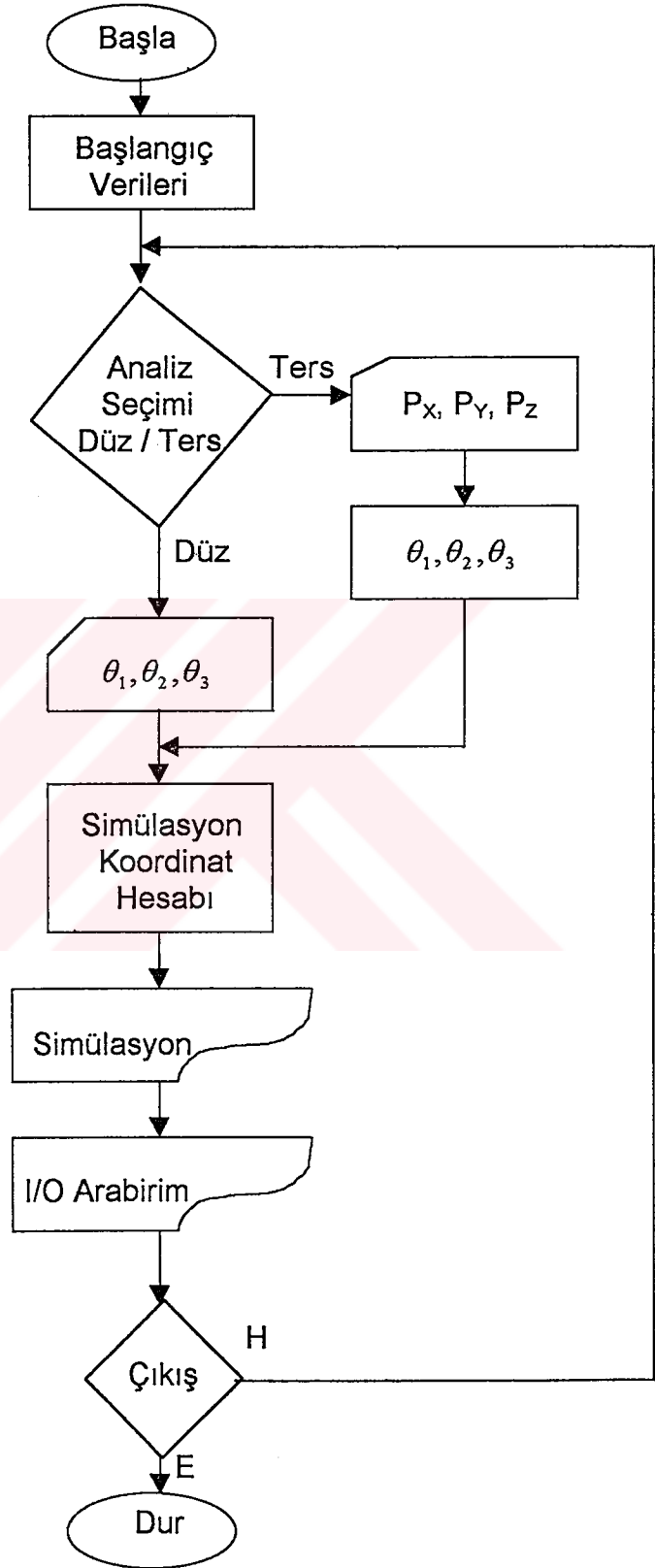
Robot kol kontrolü için gerçekleştirilen yazılım iki kısımdan oluşmaktadır. Birinci kısım, manipülatörün yaptığı / yapacağı hareketleri birebir olarak görebilmemizi sağlayan simülasyon programıdır. İkinci kısım ise manipülatör hareketinin sağlanabilmesi için, bilgisayardan gerekli sinyalleri göndererek motorların hareketini sağlayan kontrol programıdır.

Simülasyon programı, başlangıçta referans konumunda bulunan robot kolun düz ve ters kinematik analizlerini simüle etmektedir. Düz kinematik analiz, referans durumundaki kolu istenilen açı miktarlarında $(\theta_1, \theta_2, \theta_3)$, istenilen yönde (düz veya ters) ve istenilen adımda hareketi sağlar. Ters kinematik analizde ise, kol referans durumunda iken eli mevcut durumundan, üç boyutlu uzay koordinat sisteminde istenilen başka bir konuma hareket ettirilmesi işlemi sağlanır. Kolun üç boyutlu gerçek koordinat sistemi üzerindeki konumu ve konum değişiklikleri girilen veriler doğrultusunda hesaplanıp, son değerler dikkate alınarak simüle edilmiştir. Simülasyon üç boyutlu grafik kural ve algoritmaları kullanılarak ekran üzerinde x-y-z koordinat sisteminde gösterilmiştir. Program editörü iki seçimli basit

kullanımlı bir menüden oluşmaktadır. Düz kinematik seçimi ile robotun eklemlerine istenilen $\theta_1, \theta_2, \theta_3$ dönme açı değerleri veri olarak girilir. Ters kinematik seçimi ile, robotun son noktasının gitmesi istenilen hedef noktanın koordinatları (P_x, P_y, P_z) veri olarak girilir. Girilen veriler doğrultusunda amaçlanan işlerin simülasyonları gerçekleştirilir.

Kontrol programı, simülasyon programında anlatılan şekilde girilip, işlenen verilerin simülasyon sonuçlarını kontrol ünitelerinde işleterek manipülatöre ileten programdır. Kontrol programında önce simülasyon kısmının sonuç verileri olan $\theta_1, \theta_2, \theta_3$ dönme açıları robot donanımında kullanılan step motorların adım açıları göz önüne alınarak, dizayn edilen I / O arabirim kartına gönderilecek sinyallerin sıra ve sayıları hesaplanır. Hesaplanan sinyaller, motorları sürmek ve kolları hareket ettirmek için kullanılan I / O kartına gönderilmek üzere, bilgisayar donanımından hexadecimal olarak 300, 301, 302, 303 seçili adreslerine gönderilir. 303H adresi, kartı tanıtan kontrol sinyalinin adresidir. 300H adresi birinci kolu (taban), 301H adresi ikinci kolu (omuz), 302H adresi ise üçüncü kolu (dirsek) hareket ettirecek motorların bağlı olduğu katları sürer.

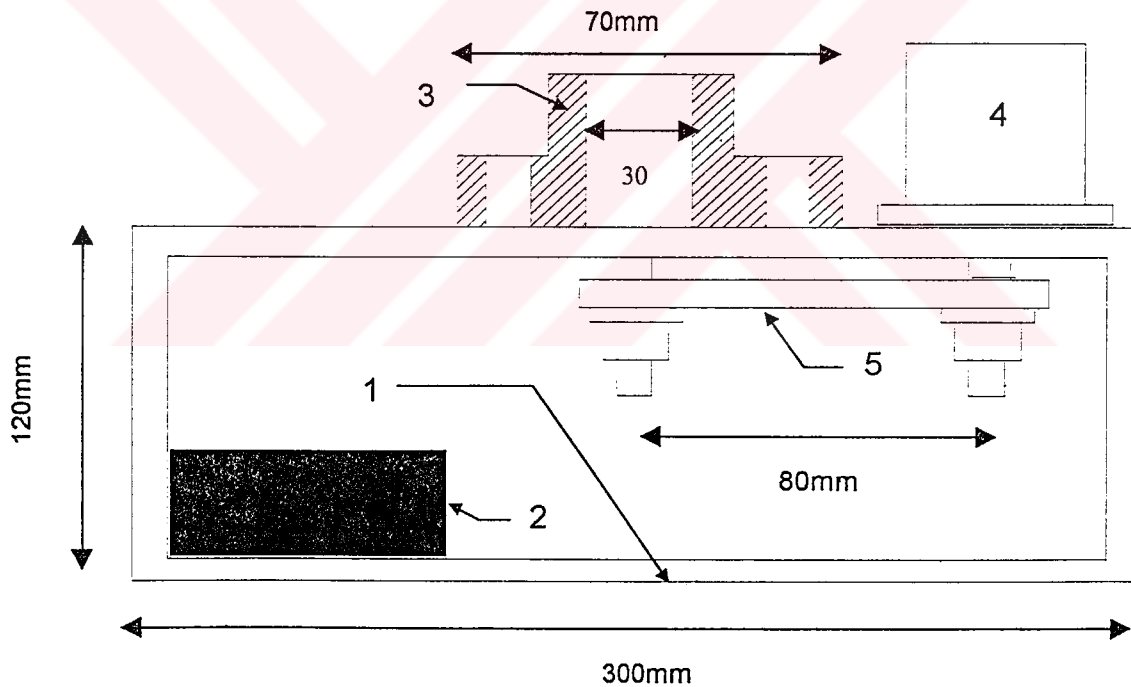
Sistem yazılımını oluşturan simülasyon programı ve onun alt programı olarak çalışan kontrol programı tek bir paket altında birleştirilmiştir. Programın tamamı Turbo Pascal 7.0 programlama dili ile yazılmıştır. Hazırlanan programa ait algoritma Şekil 4.17 'de verilmiştir. Hazırlanan yazılım çalıştırıldığında elde edilen ekran görüntüleri eklerde yer almaktadır.



Şekil 4.17. Yazılım Algoritması

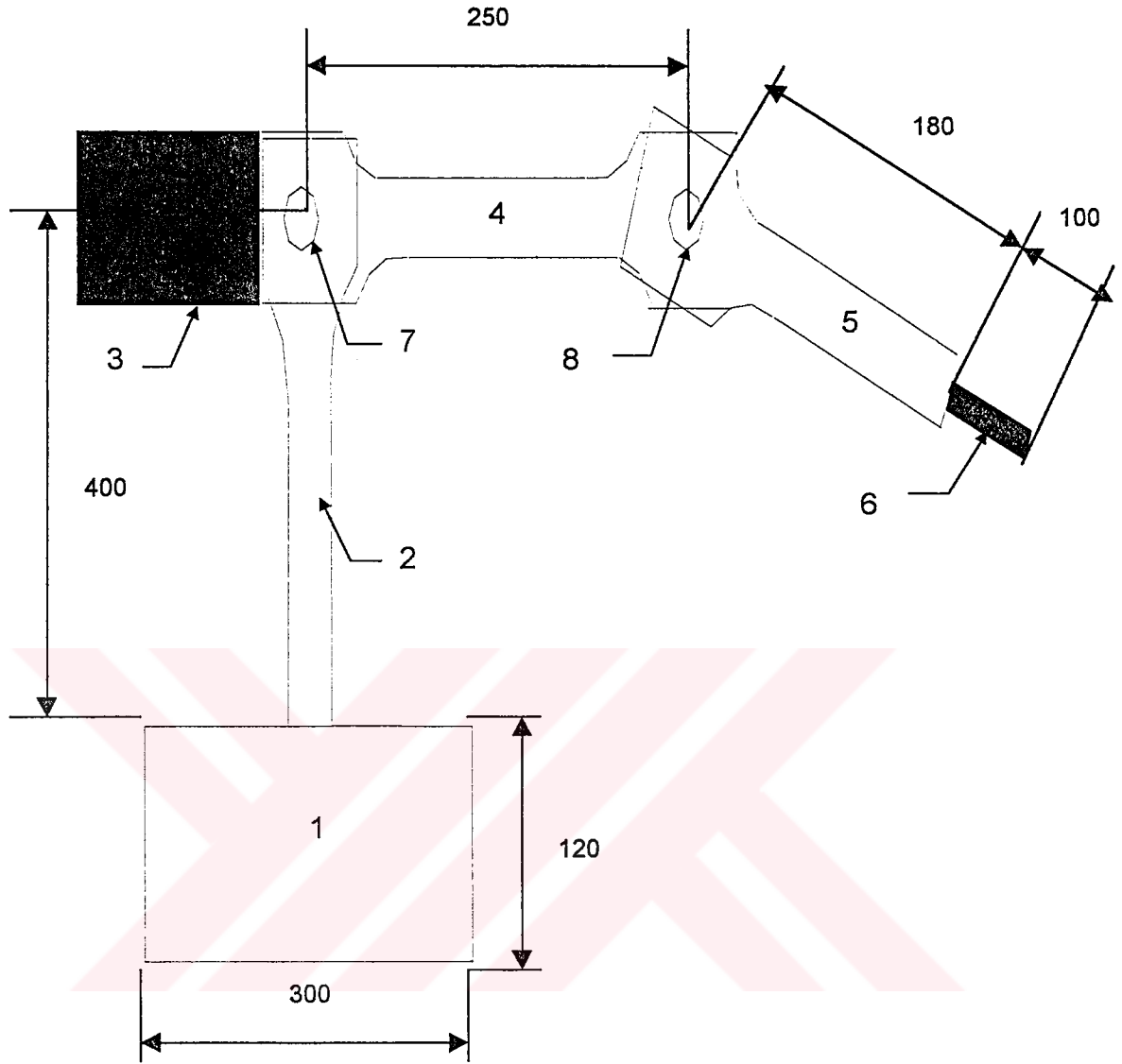
4.7. Robot Kol Mekanik Tasarımı

Hazırlanan yazılım ve kontrol kartı Şekil 4.19. 'da verilen kol mekaniği ile birlikte çalışmaktadır. Kolun mekanik tasarımı ve imalatı gerçekleştirilmiştir. Mekanik tasarım yapılırken öncelikle kollar arasında bağlantıyı sağlayan step motorların taşıyacağı min – max yük ve moment hesapları, 2. kol ve 3. kolun ağırlıklarından dolayı oluşturdukları momentler hesaplanarak, 2. kolun üzerinde bulunan 2 nolu step motorun bulunduğu eksen ağırlık merkezi kabul edilerek denge ağırlıkları hesaplanmıştır. 1. Kolun ana gövdeye montaj şeması Şekil 4.18. 'de verilmiştir. 1. Kol ana gövdeye $\Phi 20$ mm 'lik bir rulmanlı yatak ile montajlanarak, 1 nolu step motordan kramayer kayış sistemi ile ilk hareketi almaktadır.



1. Ana gövde
2. Sistemin ana ağırlığı
3. Bilyeli yatak
4. 1 nolu step motor
5. Kramayer kayış

Şekil 4.18. Ana gövde montaj şeması



- | | |
|----------------------|----------------------|
| 1. Ana gövde | 2. 1.kol |
| 3. Ağırlık | 4. 2.kol |
| 5. 3.kol | 6. El aparatı |
| 7. 2 nolu step motor | 8. 3 nolu step motor |

Şekil 4.19. Robot kol mekanik yapısı

5. SONUÇ VE ÖNERİLER

Bu çalışmada, bilgisayar destekli kontrol yöntemi ile üç serbestlik dereceli, prototip olarak mekanik tasarımı yapılan robot kolun, hazırlanan kart yardımı ile kontrolü yapılmıştır.

Hazırlanan mekanik tasarımda, üç adet 1.8° 'lik step motor kullanılmıştır. Mekanik tasarım hazırlanırken, üç boyutlu uzayda hareket edebilecek yapıda tasarlanmıştır. Kullanılan step motorlar düşük güçlü (14.5 W ve 15.5 W) olması sebebiyle büyük ağırlıklar kaldırılmaya çalışıldığında ağırlık problemi ile karşılaşılabilir. Fakat bu çalışmada el aparatı olarak kalem kullanıldığı için böyle bir problemle karşılaşılmamıştır.

Bu tasarımda el aparatına kalem takılarak daire çizdirilebilmektedir. El aparatına kalem yerine taşıma kancası takılarak, noktadan noktaya taşıma işlemleri gerçekleştirilebilir. Mekanik tasarım, yörünge planlaması ve program bakımından hazırlanan prototip geliştirilerek farklı uygulamalar gerçekleştirilebilir.

Kontrol kolaylığı ve hareket hassasiyetinden dolayı step motorlar tercih edilmiştir. Büyük ağırlıkların taşınması söz konusu olduğunda mekanik tasarımı uygun hale getirip, pnömatik kontrollü sistemlere dönüştürülebilir. Ayrıca motor güçlerinin düşük olma problemine karşı daha büyük güçlü step veya servo motorlar seçilebilir.

Tasarımı yapılan kart üzerinde, robot kolun kontrolü için gerekli olan tüm katlar hazırlanmıştır. Hazırlanan kontrol kartı ile farklı ünitelerin de kontrolü yapılabilir. Kartın beslemesi için bilgisayarın güç kaynağı kullanılmıştır. Büyük güçlere ihtiyaç duyulduğunda bilgisayar kasası içine ikinci bir güç kaynağı yerleştirilebilir.

Robot kolun kontrol ve simülasyonu için Turbo Pascal 7.0 programlama dilinde bir program hazırlanmıştır.

KAYNAKLAR

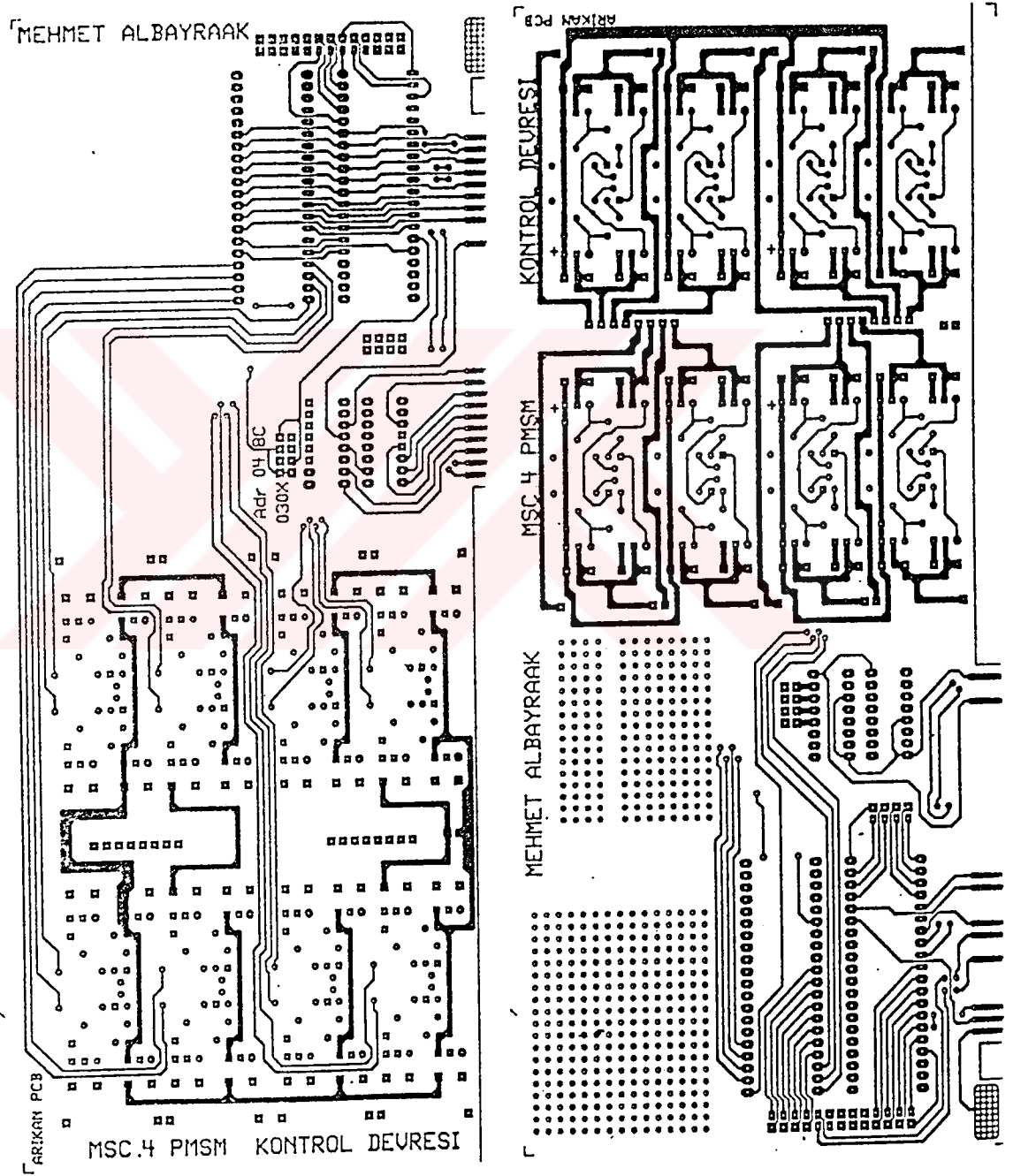
1. Acarnley, P.P., 1984, Stepping motors: a guide to modern theory and pratice, Peter Peregrinus Ltd., Revised 2nd edition, London
2. Bal, G., 1997, Özel elektrik makinaları, Gazi Üniversitesi, Ankara
3. Ata, F., 1996, Türkçe 'den seçilmiş bir grup kelimenin robot kolu kontrolünde kullanılması, Fırat Ün. Fen Bil. Ens., Doktora Tezi, Elazığ
4. Dailey, D.J., 1988, Small computer theory and applications, McGraw Hill Book Co., International edition, Singapore
5. Durak, E., 1993, Türkiye 'de robotlar üzerine bir araştırma, Akdeniz Ün. Fen Bil. Ens. ,Yüksek lisans tezi, Isparta
6. Kenzo, T., 1985, Stepping motors and their microprocessor controls, Oxford Science Publications, Oxford
7. Klafter, R.D., Chmielewski, A.T., Negin, M., 1989, Robotic engineering an integrated approach
8. Gürdal, O., 1995, Endüstriyel kontrol ders notları, Ankara
9. Meadows, R., Parsons, A., 1994, Mikroişlemciler esaslar, elemanlar ve sistemler, Milli Eğitim Bakanlığı yayınları, Eskişehir
10. Sert, H.M., 1996, Elektrik sayaçlarının bilgisayarla test edilmesi, Erciyes Ün. Fen Bil. Ens., Yüksek lisans tezi, Kayseri
11. Slemon, G.R., 1992, Electrical machines, drivers, Addison – Wesley Publishing Company Inc.
12. Yaz, M., 1996, Dijital elektronik ders notları, Kayseri



EKLER

EK - 1

KONTROL KARTI BASKI DEVRE ŞEMALARI



EK – 2

HAZIRLANAN YAZILIMA AİT EKRAN GÖRÜNTÜLERİ

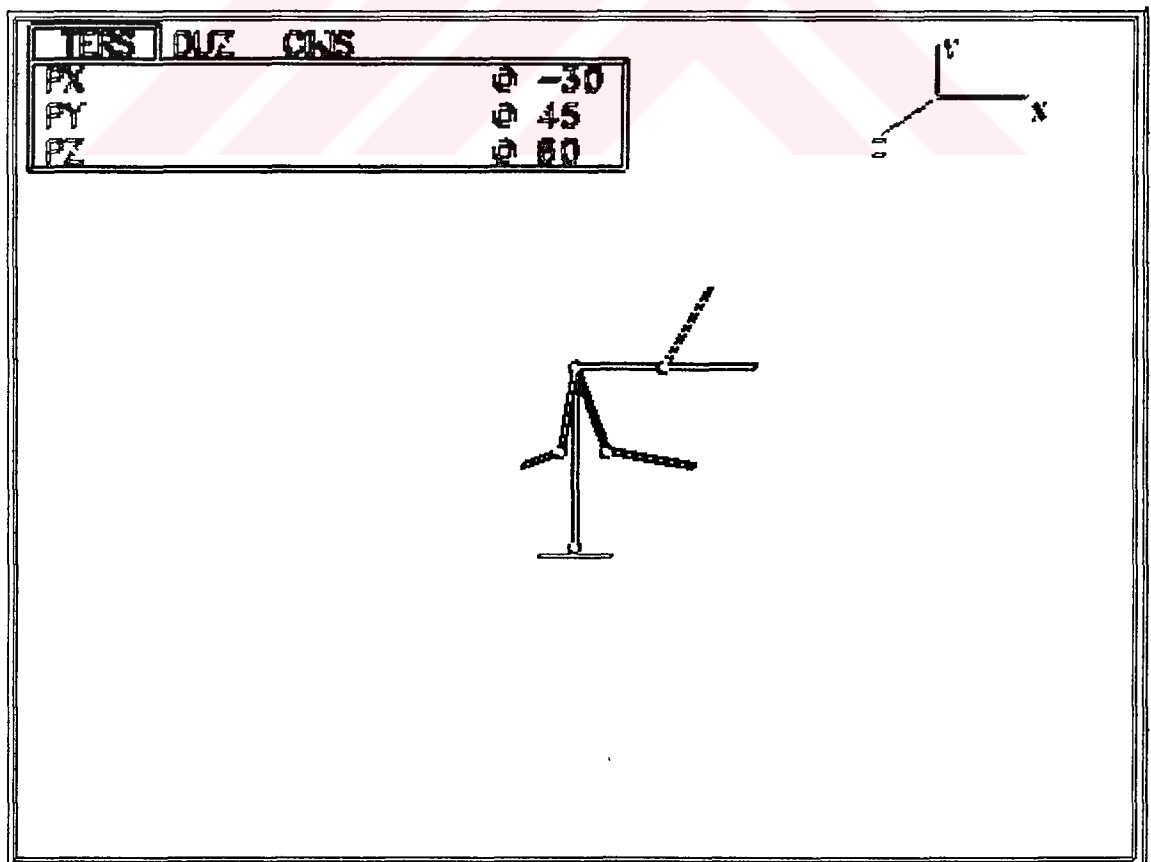
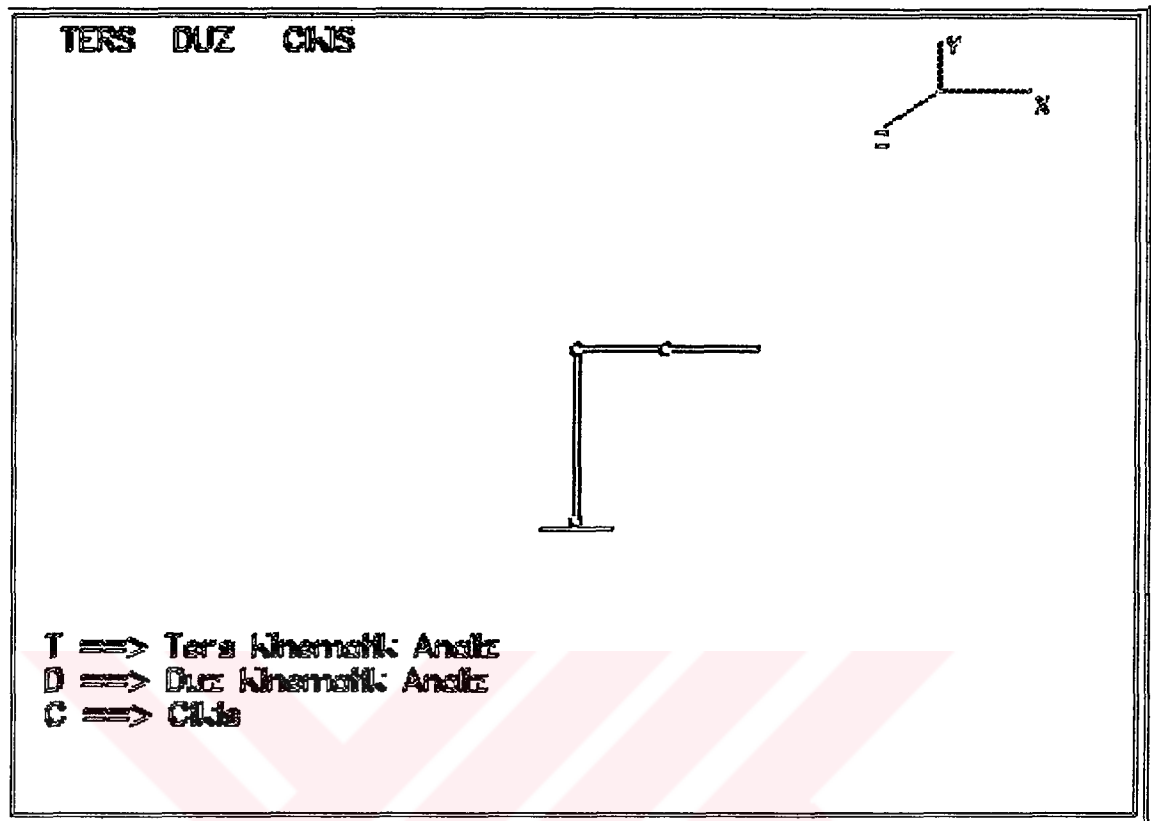
**GAZİ ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ
ELEKTRİK EĞİTİMİ ANABİLİM DALI**

YÜKSEK LİSANS TEZİ

**ÜÇ BOYUTLU UZAYDA HAREKET EDERİLEN
ÜÇ SERBESTLİK DERECELİ ROBOT KOL TASARIMI,
BİLGİSAYAR DESTEKLİ KONTROLÜ VE SİMULASYONU**

**MEHMET ALBAYRAK
9411001**

ESC**ENTER**



EK – 3

HAZIRLANAN YAZILIM

{ BU PROGRAM ÜÇ SERBESTLİK DERECELİ BİR MANİPÜLATÖRÜN DÜZ VE TERS KİNEMATİK ANALİZ SONUÇLARININ SİMÜLASYONUNU VE BU SONUÇLARA GÖRE ROBOT KOL KONTROLÜNÜ YAPAR.}

```

program robot;
uses crt, Graph;
var
  k,r,delta,t1,d1,d2,d3,px,py,pz,aa,bb,c,s,st,stt:real;
  Q,Q1,Q2,Q3,a,b,a1,a2,a3,a4,b1,b2,b3,b4:real;
  kod,say,x1,x2,x3,x4,y1,y2,y3,y4,z1,z2,z3,z4:integer;
  l1,l2,l3,gd,gm,x,y,z,fs:integer;
  ik,iks:string;
  kar,but:char;
  h1,h2:integer;
  kontrol1,kontrol2:real;
  Q1A,Q1D,Q1M,Q2A,Q2D,Q2M,Q3A,Q3M,Q3D,adim:integer;
const
  IO_8255:word=$303;{kontrol adresi}
  A_portu:word=$300;{A portunun adresi}
  B_portu:word=$301;{B portunun adresi}
  C_portu:word=$302;{C portunun adresi}
  bekleme:word=2;

procedure ciz;
begin
  SetLineStyle(0,4,3);
  setcolor(15);circle(320+y1,300-z1,2); line(320+y1,300-z1,320+y2,300-z2);
  setcolor(12); circle(320+y2,300-z2,2); line(320+y2,300-z2,320+y3,300-z3);
  setcolor(14); circle(320+y3,300-z3,2); line(320+y3,300-z3,320+y4,300-z4);
end;

procedure ze;
begin
  a1:=x1*cos(Q1)+y1*(-sin(Q1));
  b1:=x1*sin(Q1)+y1*cos(Q1);
  a2:=x2*cos(Q1)+y2*(-sin(Q1));
  b2:=x2*sin(Q1)+y2*cos(Q1);
  a3:=x3*cos(Q1)+y3*(-sin(Q1));
  b3:=x3*sin(Q1)+y3*cos(Q1);
  a4:=x4*cos(Q1)+y4*(-sin(Q1));
  b4:=x4*sin(Q1)+y4*cos(Q1);
  x1:=round(a1);y1:=round(b1);
  x2:=round(a2);y2:=round(b2);
  x3:=round(a3);y3:=round(b3);
  x4:=round(a4);y4:=round(b4);

```

```

    setcolor(12);
    delay(500); ciz;
end;
procedure xe;
begin
    a3:=(y3*cos(Q2))+((z3-z2)*(-sin(Q2)));
    b3:=(y3*sin(Q2))+((z3-z2)*cos(Q2));
    a4:=(y4*cos(Q2))+((z4-z2)*(-sin(Q2)));
    b4:=(y4*sin(Q2))+((z4-z2)*cos(Q2));
    y3:=round(a3);z3:=round(b3)+z2;
    y4:=round(a4);z4:=round(b4)+z2;
    setcolor(4);
    delay(500); ciz;
end;
procedure xee;
begin
    a4:=((y4-y3)*cos(Q3))+((z4-z3)*(-sin(Q3)));
    b4:=((y4-y3)*sin(Q3))+((z4-z3)*cos(Q3));
    y4:=round(a4)+y3;z4:=round(b4)+z3;
    setcolor(10);
    delay(500);ciz;
end;

procedure kisim1_i(portX:word);
begin
    port[portX]:=5 ;delay(bekleme);
    port[portX]:=6 ;delay(bekleme);
    port[portX]:=10;delay(bekleme);
    port[portX]:=9 ;delay(bekleme);
end;
procedure kisim1_g(portX:word);
begin
    port[portX]:=9 ;delay(bekleme);
    port[portX]:=10;delay(bekleme);
    port[portX]:=6 ;delay(bekleme);
    port[portX]:=5 ;delay(bekleme);
end;

procedure Q3aktif;
begin
    port[$300]:=0;port[$301]:=0;port[$302]:=0;port[$303]:=0;
    port[IO_8255]:=ABC_out;
    if Q3>0 then begin
        if q3d>0 then begin
            for adim:=1 to q3d do begin {outtextxy(100,100,'adim');}
                {delay(25);}
            end
        end
    end
end

```

```

        kisim1_i(c_portu);end;
        end;
        if q3m=1 then begin {outtextxy(100,120,'q3m1');}port[c_portu]:=5
;delay(bekleme);end;
        if q3m=2 then begin {outtextxy(100,140,'q3m2');}port[c_portu]:=5
;delay(bekleme);
                port[c_portu]:=6 ;delay(bekleme);end;
        if q3m=3 then begin {outtextxy(100,160,'q3m3');}port[c_portu]:=5
;delay(bekleme);
                port[c_portu]:=6 ;delay(bekleme);
                port[c_portu]:=10;delay(bekleme);end;
        end;
        if Q3<0 then begin
                if q3d>0 then begin
                        for adim:=1 to q3d do begin {outtextxy(100,100,'adim');}
                        {delay(25);}
                        kisim1_g(c_portu);end;
                        end;
                        if q3m=1 then begin {outtextxy(100,120,'q3m1');}port[c_portu]:=9
;delay(bekleme);end;
                        if q3m=2 then begin {outtextxy(100,140,'q3m2');}port[c_portu]:=9
;delay(bekleme);
                                port[c_portu]:=10 ;delay(bekleme);end;
                        if q3m=3 then begin {outtextxy(100,160,'q3m3');}port[c_portu]:=9
;delay(bekleme);
                                port[c_portu]:=10 ;delay(bekleme);
                                port[c_portu]:=6;delay(bekleme);end;
                        end;
                port[$300]:=0;port[$301]:=0;port[$302]:=0;port[$303]:=0;
                port[IO_8255]:=0;
        end;

```

```

procedure Q3reset;
begin
port[$300]:=0;port[$301]:=0;port[$302]:=0;port[$303]:=0;
port[IO_8255]:=ABC_out;
        if Q3>0 then begin
                if q3m=1 then begin {outtextxy(100,180,'sq3m1');}
                        port[c_portu]:=6 ;delay(bekleme);
                        port[c_portu]:=10 ;delay(bekleme);
                        port[c_portu]:=9 ;delay(bekleme);end;
                if q3m=2 then begin {outtextxy(100,200,'sq3m2');}
                        port[c_portu]:=10 ;delay(bekleme);
                        port[c_portu]:=9 ;delay(bekleme);end;
                if q3m=3 then begin {outtextxy(100,220,'sq3m3');}

```

```

        port[c_portu]:=9 ;delay(bekleme);end;
    end;
    if Q3<0 then begin
        if q3m=1 then begin {outtextxy(100,180,'sq3m1');}
            port[c_portu]:=10 ;delay(bekleme);
            port[c_portu]:=6 ;delay(bekleme);
            port[c_portu]:=5 ;delay(bekleme);end;
        if q3m=2 then begin {outtextxy(100,200,'sq3m2');}
            port[c_portu]:=6 ;delay(bekleme);
            port[c_portu]:=5 ;delay(bekleme);end;
        if q3m=3 then begin {outtextxy(100,220,'sq3m3');}
            port[c_portu]:=5 ;delay(bekleme);end;
        end;
        port[$300]:=0;port[$301]:=0;port[$302]:=0;port[$303]:=0;
        port[IO_8255]:=0;
    end;

    procedure Q2aktif;
    begin
        port[$300]:=0;port[$301]:=0;port[$302]:=0;port[$303]:=0;
        port[IO_8255]:=ABC_out; if Q2>0 then begin
            if q2d>0 then begin
                for adim:=1 to q2d do begin {outtextxy(150,100,'adim');}
                    { delay(25);}
                    kisim1_i(b_portu);end;
                end;
            if q2m=1 then begin {outtextxy(150,120,'q2m1');}port[b_portu]:=5
;delay(bekleme);end;
            if q2m=2 then begin {outtextxy(150,140,'q2m2');}port[b_portu]:=5
;delay(bekleme);
                port[b_portu]:=6 ;delay(bekleme);end;
            if q2m=3 then begin {outtextxy(150,160,'q2m3');}port[b_portu]:=5
;delay(bekleme);
                port[b_portu]:=6 ;delay(bekleme);
                port[b_portu]:=10;delay(bekleme);end;
            end;
        if Q2<0 then begin
            if q2d>0 then begin
                for adim:=1 to q2d do begin {outtextxy(150,100,'adim');}
                    {delay(25);}
                    kisim1_g(b_portu);end;
                end;
            if q2m=1 then begin {outtextxy(150,120,'q2m1');}port[b_portu]:=9
;delay(bekleme);end;
            if q2m=2 then begin {outtextxy(150,140,'q2m2');}port[b_portu]:=9
;delay(bekleme);

```

```

        port[b_portu]:=10 ;delay(bekleme);end;
    if q2m=3 then begin {outtextxy(150,160,'q2m3');}port[b_portu]:=9
;delay(bekleme);
        port[b_portu]:=10 ;delay(bekleme);
        port[b_portu]:=6;delay(bekleme);end;
    end;
    port[$300]:=0;port[$301]:=0;port[$302]:=0;port[$303]:=0;
port[IO_8255]:=0;
end;

```

```

procedure Q2reset;
begin
port[$300]:=0;port[$301]:=0;port[$302]:=0;port[$303]:=0;
port[IO_8255]:=ABC_out;
    if Q2>0 then begin
        if q2m=1 then begin {outtextxy(150,180,'sq2m1');}
            port[b_portu]:=6 ;delay(bekleme);
            port[b_portu]:=10 ;delay(bekleme);
            port[b_portu]:=9 ;delay(bekleme);end;
        if q2m=2 then begin {outtextxy(150,200,'sq2m2');}
            port[b_portu]:=10 ;delay(bekleme);
            port[b_portu]:=9 ;delay(bekleme);end;
        if q2m=3 then begin {outtextxy(150,220,'sq2m3');}
            port[b_portu]:=9 ;delay(bekleme);end;
        end;
    if Q2<0 then begin
        if q2m=1 then begin {outtextxy(150,180,'sq2m1');}
            port[b_portu]:=10 ;delay(bekleme);
            port[b_portu]:=6 ;delay(bekleme);
            port[b_portu]:=5 ;delay(bekleme);end;
        if q2m=2 then begin {outtextxy(150,200,'sq2m2');}
            port[b_portu]:=6 ;delay(bekleme);
            port[b_portu]:=5 ;delay(bekleme);end;
        if q2m=3 then begin {outtextxy(150,220,'sq2m3');}
            port[b_portu]:=5 ;delay(bekleme);end;
        end;
    port[$300]:=0;port[$301]:=0;port[$302]:=0;port[$303]:=0;
port[IO_8255]:=0;
end;

```

```

procedure Q1aktif;
begin
port[$300]:=0;port[$301]:=0;port[$302]:=0;port[$303]:=0;
port[IO_8255]:=ABC_out;
    if Q1>0 then begin
        if q1d>0 then begin

```

```

        for adim:=1 to q1d do begin {outtextxy(200,100,'adim');}
        { delay(25);}
        kisim1_i(a_portu);end;
        end;
        if q1m=1 then begin {outtextxy(200,120,'q1m1');}port[a_portu]:=5
;delay(bekleme);end;
        if q1m=2 then begin {outtextxy(200,140,'q1m2');}port[a_portu]:=5
;delay(bekleme);
                port[a_portu]:=6 ;delay(bekleme);end;
        if q1m=3 then begin {outtextxy(200,160,'q1m3');}port[a_portu]:=5
;delay(bekleme);
                port[a_portu]:=6 ;delay(bekleme);
                port[a_portu]:=10;delay(bekleme);end;
        end;
    if Q1<0 then begin
        if q1d>0 then begin
            for adim:=1 to q1d do begin {outtextxy(200,100,'adim');}
            {delay(25);}
            kisim1_g(a_portu);end;
            end;
            if q1m=1 then begin {outtextxy(200,120,'q1m1');}port[a_portu]:=9
;delay(bekleme);end;
            if q1m=2 then begin {outtextxy(200,140,'q1m2');}port[a_portu]:=9
;delay(bekleme);
                    port[a_portu]:=10 ;delay(bekleme);end;
            if q1m=3 then begin {outtextxy(200,160,'q1m3');}port[a_portu]:=9
;delay(bekleme);
                    port[a_portu]:=10 ;delay(bekleme);
                    port[a_portu]:=6;delay(bekleme);end;
            end;
            port[$300]:=0;port[$301]:=0;port[$302]:=0;port[$303]:=0;
port[IO_8255]:=0;
end;

procedure Q1reset;
begin
port[$300]:=0;port[$301]:=0;port[$302]:=0;port[$303]:=0;
port[IO_8255]:=ABC_out;
    if Q1>0 then begin
        if q1m=1 then begin {outtextxy(200,180,'sq1m1');}
                port[a_portu]:=6 ;delay(bekleme);
                port[a_portu]:=10 ;delay(bekleme);
                port[a_portu]:=9 ;delay(bekleme);end;
        if q1m=2 then begin {outtextxy(200,200,'sq1m2');}
                port[a_portu]:=10 ;delay(bekleme);
                port[a_portu]:=9 ;delay(bekleme);end;
    end;
end;

```

```

        if q1m=3 then begin {outtextxy(200,220,'sq1m3');}
            port[a_portu]:=9 ;delay(bekleme);end;
        end;
    if Q1<0 then begin
        if q1m=1 then begin {outtextxy(200,180,'sq1m1');}
            port[a_portu]:=10 ;delay(bekleme);
            port[a_portu]:=6 ;delay(bekleme);
            port[a_portu]:=5 ;delay(bekleme);end;
        if q1m=2 then begin {outtextxy(200,200,'sq1m2');}
            port[a_portu]:=6 ;delay(bekleme);
            port[a_portu]:=5 ;delay(bekleme);end;
        if q1m=3 then begin {outtextxy(200,220,'sq1m3');}
            port[a_portu]:=5 ;delay(bekleme);end;
        end;
        port[$300]:=0;port[$301]:=0;port[$302]:=0;port[$303]:=0;
        port[IO_8255]:=0;
    end;

```

```

procedure duz;
begin
    ciz;
    setcolor(11); settextstyle(3,0,1);
    rectangle(90,30,440,50);
    setcolor(15);outtextxy(95,27,'(Q1) TABAN DÖNÜŞ AÇISI ');
    circle(380,40,5);
    moveto(390,27); ik:="";say:=0;
    repeat
        setcolor(15);
        iks:=readkey;outtext(iks); ik:=ik+iks; say:=say+1;
        if iks=#27 then
            begin closegraph;halt; end;
        until iks=#13;
        delete(ik,say,3); val(ik,Q1,kod);
        setcolor(0);rectangle(90,30,440,50);
        setcolor(11);rectangle(90,30,440,70);
        setcolor(12);outtextxy(95,47,'(Q2) OMUZ DÖNÜŞ AÇISI');
        circle(380,60,5);
        moveto(390,47); ik:="";say:=0;
        repeat
            setcolor(12);
            iks:=readkey;outtext(iks); ik:=ik+iks; say:=say+1;
            if iks=#27 then
                begin closegraph;halt; end;
            until iks=#13;
        delete(ik,say,3); val(ik,Q2,kod);
    end;

```

```

setcolor(0);rectangle(90,30,440,70);
setcolor(11);rectangle(90,30,440,90);
setcolor(14);outtextxy(95,67,'(Q3) DİRSEK DÖNÜŞ AÇISI');
setcolor(14); circle(380,80,5);
moveto(390,67); ik:="";say:=0;
repeat
setcolor(14);
iks:=readkey;outtext(iks); ik:=ik+iks; say:=say+1;
if iks=#27 then
begin closegraph;halt; end;
until iks=#13;
delete(ik,say,3); val(ik,Q3,kod);

```

```

if Q3<>0 then begin
    Q3:=(Q3/57.3); Q3A:=round(abs(Q3)*57.3/1.8);
    Q3D:=Q3A div 4;
    Q3M:=Q3A mod 4;

    xee;
    Q3aktif;
end;

```

```

if Q2<>0 then begin
    Q2:=(Q2/57.3); Q2A:=round(abs(Q2)*57.3/1.8);
    Q2D:=Q2A div 4;
    Q2M:=Q2A mod 4;

    xe;
    Q2aktif;
end;

```

```

if Q1<>0 then begin
    Q1:=(Q1/57.3); Q1A:=round(abs(Q1)*57.3/1.8);
    Q1D:=Q1A div 4;
    Q1M:=Q1A mod 4;

    ze;
    Q1aktif;
end;

```

```

readln;
Q3reset;
Q2reset;
Q1reset;
readln;
end;

```

```

procedure ters;
begin
  settextstyle(3,0,1);
  setcolor(11);rectangle(15,30,348,50);
  setcolor(15);outtextxy(25,27,'PX');circle(280,40,5);{PX}
  moveto(300,27); ik:="";say:=0;
  repeat
    repeat
      setcolor(15);iks:=readkey;outtext(iks); ik:=ik+iks; say:=say+1;
      setcolor(0);outtextxy(48,50,' BU DEĞER SIFIRA EŞİT OLAMAZ !!');
      if iks=#27 then begin closegraph;halt; end;
    until iks=#13;
    delete(ik,say,3); val(ik,PX,kod);
    if px=0 then begin setcolor(15);
      outtextxy(48,50,' BU DEĞER SIFIRA EŞİT OLAMAZ !!');
      setcolor(0);outtextxy(300,27,ik);
      moveto(300,27);ik:="";say:=0;end;
    until px<>0;

    setcolor(0);rectangle(15,30,348,50);
    setcolor(11);rectangle(15,30,348,70);
    setcolor(12);outtextxy(25,47,'PY');circle(280,60,5);{PZ}
    moveto(300,47); ik:="";say:=0;

    repeat
      iks:=readkey;outtext(iks); ik:=ik+iks; say:=say+1;
      if iks=#27 then
        begin closegraph;halt; end;
      until iks=#13;
      delete(ik,say,3); val(ik,PZ,kod);

      setcolor(0);rectangle(15,30,348,70);
      setcolor(11);rectangle(15,30,348,90);
      setcolor(14);outtextxy(25,67,'PZ');circle(280,80,5);{PY}
      moveto(300,67); ik:="";say:=0;
      repeat
        repeat
          setcolor(14);iks:=readkey;outtext(iks); ik:=ik+iks; say:=say+1;
          setcolor(0);outtextxy(48,90,' BU DEĞER SIFIRA EŞİT OLAMAZ !!');
          if iks=#27 then
            begin closegraph;halt; end;
          until iks=#13;
          delete(ik,say,3); val(ik,PY,kod);
          if py=0 then begin setcolor(14);
            outtextxy(48,90,' BU DEĞER SIFIRA EŞİT OLAMAZ !!');

```

```

        setcolor(0);outtextxy(300,67,ik);
        moveto(300,67);ik:="";say:=0;end;
until py<>0;
begin
py:=abs(py);
d1:=arctan(py/px);

c:=cos(d1);
s:=sin(d1);
aa:=(c*px+s*py);
bb:=pz-l1;
st:=((aa*aa)+(bb*bb)-(50*50)-(50*50))/(2*50*50);
stt:=st*st;
kontrol1:=1-stt;
if kontrol1<0 then begin
setcolor(12);
outtextxy(10,97,' ÖLÜ BÖLGE , LÜTFEN YENİ VERİ GİRİNİZ !!!');
readln;exit;end;
d3:=arctan(sqrt(kontrol1)/st);

k:=l2+l3*cos(d3);
r:=l3*sin(d3);
kontrol2:=(4*k*k)-4*(l1-r-pz)*(r+l1-pz);
if kontrol2<0 then begin
setcolor(12);
outtextxy(10,97, ' ÖLÜ BÖLGE , LÜTFEN YENİ VERİ GİRİNİZ !!!');
readln;exit;end;
delta:=sqrt(kontrol2);
t1:=(-2*k+delta)/(2*(l1-r-pz));
d2:=2*arctan(t1);

if (px<0) and (py>0) then d1:=((90+(90-(d1*57.3)))/57.3);
if (px<0) and (py<0) then d1:=180+d1;
Q1:=d1;
Q2:=d2;
Q3:=d3;

if Q3<>0 then begin Q3:=(Q3);
Q3A:=round(abs(Q3)*57.3/1.8);
Q3D:=Q3A div 4;
Q3M:=Q3A mod 4;

xee;
Q3aktif;
end;

if Q2<>0 then begin Q2:=(Q2);

```

```

        Q2A:=round(abs(Q2)*57.3/1.8);
        Q2D:=Q2A div 4;
        Q2M:=Q2A mod 4;

        xe;
        Q2aktif;
        end;

if Q1<>0 then begin Q1:=(Q1);
        Q1A:=round((Q1*57.3)/1.8);
        Q1D:=Q1A div 4;
        Q1M:=Q1A mod 4;

        ze;
        Q1aktif;
        end;

end;

readln;
Q3reset;
Q2reset;
Q1reset;
readln;
end;
procedure kapak;
begin
    SETBKCOLOR(1);
    setcolor(15);rectangle(1,1,638,478);rectangle(6,6,632,472);
    settextstyle(1,0,3); {103}
    outtextxy(220,20,'GAZI UNIVERSITESI');
    outtextxy(175,45,'FEN BILIMLERI ENSTITUSU');
    outtextxy(135,70,'ELEKTRIK EGITIMI ANABILIMDALI');
    outtextxy(200,130,'YUKSEK LISANS TEZI');
    outtextxy(95,190,'UC BOYUTLU UZAYDA HAREKET EDEBILEN');
    outtextxy(60,220,'UC SERBESTLIK DERECELI ROBOT KOL TASARIMI,');
    outtextxy(40,250,'BILGISAYAR DESTEKLI KONTROLU VE
SIMULASYONU');
    outtextxy(200,345,'MEHMET ALBAYRAK');
    outtextxy(260,370,'9411001');
    outtextxy(25,445,'ESC');
    outtextxy(545,445,'ENTER');
    but:=readkey;
    if but=#27 then begin closegraph;halt;end;
    for fs:=0 to 240 do
        begin
            setcolor(1);line(0,240-FS,640,240-FS);line(0,240+FS,640,240+FS);
            setcolor(4);line(1,240-FS,1,240+FS);line(6,240-FS,6,240+FS);
                line(638,240-FS,638,240+FS);line(632,240-FS,632,240+FS);

```

```

    delay(5);
end;
end;

```

```

procedure menu;

```

```

begin

```

```

    l1:=100;

```

```

    l2:=50;

```

```

    l3:=50;

```

```

    x1:=0;y1:=0;z1:=0;

```

```

    x2:=0;y2:=0;z2:=100;

```

```

    x3:=0;y3:=50;z3:=100;

```

```

    x4:=0;y4:=100;z4:=100;

```

```

    repeat

```

```

        cleardevice;

```

```

    { Q1A:=0; Q1D:=0; Q1M:=0; Q2A:=0; Q2D:=0; Q2M:=0;
      Q3A:=0; Q3M:=0; Q3D:=0; adim:=0;}

```

```

        port[$300]:=0;port[$301]:=0;port[$302]:=0;port[$303]:=0;

```

```

    port[IO_8255]:=0;

```

```

    setcolor(15);SetLineStyle(0,0,1);rectangle(1,1,638,478);

```

```

    rectangle(6,6,632,472);line(300,305,340,305);

```

```

    SetLineStyle(1,0,1);line(520,50,570,50);line(520,50,520,20);

```

```

    line(520,50,490,70);

```

```

    settextstyle(2,0,5);

```

```

    outtextxy(575,50,'X');outtextxy(525,15,'Y');outtextxy(485,70,'Z');

```

```

    settextstyle(3,0,1);

```

```

    setcolor(12);outtextxy(33,7,'T');setcolor(11);outtextxy(16,7,' ERS');

```

```

    setcolor(12);outtextxy(95,7,'D');setcolor(11);outtextxy(106,7,'UZ');

```

```

    setcolor(12);outtextxy(155,7,'C');setcolor(11);outtextxy(167,7,'IKIS');

```

```

    setcolor(14);outtextxy(25,360,'T ==> Ters Kinematik Analiz');

```

```

        outtextxy(25,380,'D ==> Duz Kinematik Analiz');

```

```

        outtextxy(25,400,'C ==> Cikis');

```

```

ciz;

```

```

ik:=readkey;if ik=#27 then begin closegraph;halt; end;

```

```

if((ik<>#27)and((ik<>'T') and (ik<>'t'))and((ik<>'D') and

```

```

(ik<>'d'))and((ik<>'C') and (ik<>'c'))))then write(#07);

```

```

setcolor(1);outtextxy(25,360,'T ==> Ters Kinematik Analiz');

```

```

        outtextxy(25,380,'D ==> Duz Kinematik Analiz');

```

```

        outtextxy(25,400,'C ==> Cikis');

```

```

if ((ik='T') or (ik='t')) then begin setcolor(11);rectangle(15,30,85,10);

```

```

        ters; end;

```

```

if ((ik='D') or (ik='d')) then begin setcolor(11);rectangle(90,30,150,10);

```

```
        duz; end;  
    if ((ik='C') or (ik='c')) then exit;  
until false;  
end;  
begin  
    Gd := Detect; InitGraph(Gd, Gm, 'c:\Tp\bgi');  
    if GraphResult <> grOk then Halt(1);  
    kapak;  
    menu;  
    closegraph;  
end.
```



EK – 4**HAZIRLANAN ROBOT KOL İLE YAPILAN DENEY ÖRNEĞİ 1**

Bu örnek deney ile robot kolun ucuna kalem takılarak, zemine çember çizdirilmiştir. Bu işlemin yapılması için gerekli işlem basamakları aşağıda verilmiştir.

1. Hazırlanan yazılım çalıştırılır
2. Programda düz kinematik analiz 'D' harfine basılarak seçilir
3. $Q1=-90$, $Q2=0$, $Q3=360$ değerleri girilerek robot programlanmış olur. Bu değerlerden $Q1$ uç kolu 90° aşağı yönlendirir, $Q3$ kolu tabandan 360° döndürür. Bu işlem gerçekleştirildiğinde tam bir çember çizdirilmiş olur.



EK – 5**HAZIRLANAN ROBOT KOL İLE YAPILAN DENEY ÖRNEĞİ 2**

Bu örnek deney ile robot kolun ucuna kalem takılarak, zemine 90° 'lik yay çizdirilmiştir. Bu işlemin yapılması için gerekli işlem basamakları aşağıda verilmiştir.

1. Hazırlanan yazılım çalıştırılır
2. Programda düz kinematik analiz 'D' harfine basılarak seçilir
3. $Q1=-90$, $Q2=0$, $Q3=90$ değerleri girilerek robot programlanmış olur. Bu değerlerden $Q1$ uç kolu 90° aşağı yönlendirir, $Q3$ kolu tabandan 90° döndürülerek çeyrek çemberlik yay çizdirilmiş olur.



ÖZGEÇMİŞ

1974 Yılında İzmir 'de doğdu. Öğrenimini Isparta, Alaybeyoğlu İlkokulu, Gülkent Ortaokulu ve Teknik Lise 'de tamamladı. 1990 - 1991 öğretim yılında Gazi Üniversitesi, Teknik Eğitim Fakültesi, Elektrik Eğitimi Bölümünü kazandı. 1993 - 1994 öğretim yılında mezun oldu. Öğrenimi sırasında; Elektrik - Elektronik ve Bilgisayar firmalarının teknik servislerinde çalıştı. 1994-1995 öğretim yılında Gazi Üniversitesi, Fen Bilimleri Enstitüsü, Elektrik Eğitimi Bölümünde Yüksek Lisansa girdi. 1994 - 1995 öğretim yılında Kayseri Seyyid Burhaneddin Anadolu Teknik, Teknik ve Endüstri Meslek Lisesi Elektrik Bölümü öğretmeni olarak çalıştı. 1995 – 1996 öğretim yılında Süleyman Demirel Üniversitesi, Teknik Eğitim Fakültesi, Elektrik Eğitimi Bölümü, Elektrik Makinaları Anabilimdalı 'na Araştırma Görevlisi olarak girdi.

Halen bu kurumda Araştırma Görevlisi olarak çalışmaktadır. Evli ve yabancı dili İngilizce 'dir.

Mehmet ALBAYRAK

15.08.1997