T. C.
YÜZÜNCÜ YIL ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ
ELEKTRİK – ELEKTRONİK MÜHENDİSLİĞİ ANABİLİM DALI

BİLGİSAYAR DESTEKLİ VEKTÖREL ÇİZİM YAPAN KARTEZYEN ROBOT TASARIMI VE PROGRAMLANMASI

YÜKSEK LİSANS TEZİ

HAZIRLAYAN: M. Kemalettin TORAMAN
DANIŞMAN: Prof. Dr. Nihat İNANÇ

VAN – 2008

T. C.
YÜZÜNCÜ YIL ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ
ELEKTRİK – ELEKTRONİK MÜHENDİSLİĞİ ANABİLİM DALI

BİLGİSAYAR DESTEKLİ VEKTÖREL ÇİZİM YAPAN KARTEZYEN ROBOT TASARIMI VE PROGRAMLANMASI

YÜKSEK LİSANS TEZİ

HAZIRLAYAN: M. Kemalettin TORAMAN

VAN – 2008

ONAY SAYFASI

Yüzüncü Yıl Üniversitesi Fen Bilimleri Enstitüsü Elektrik-Elektronik Mühendisliği Anabilim Dalı'nda Prof. Dr. Nihat İNANÇ danışmanlığında, M. Kemalettin TORAMAN tarafından sunulan ***"Bilgisayar Destekli Vektörel Çizim Yapan Kartezyen Robot Tasarımı ve Programlanması"*** isimli bu çalışma "Lisansüstü Eğitim ve Öğretim Yönetmeliği" ve "Fen Bilimleri Enstitüsü Yönergesi"nin ilgili hükümleri gereğince 13/10/2008 tarihinde aşağıdaki jüri tarafından oy birliği / oy çokluğu ile başarılı bulunmuş ve Yüksek Lisans tezi olarak kabul edilmiştir.

Başkan: Prof. Dr. Sabir Rüstemli imza:

Üye: Prof. Dr. Nihat İNANÇ imza:

Üye: Yrd. Doç. Dr. Remzi TUNTAŞ imza:

Fen Bilimleri Enstitüsü Yönetim Kurulu'nun …./…./2008 tarih ve …….. sayılı kararı ile onaylanmıştır.

imza

Enstitü Müdürü

ÖZET

BİLGİSAYAR DESTEKLİ VEKTÖREL ÇİZİM YAPAN KARTEZYEN ROBOT TASARIMI VE PROGRAMLANMASI

TORAMAN, M. Kemalettin
Yüksek Lisans Tezi, Elektrik-Elektronik Mühendisliği Anabilim Dalı
Tez Danışmanı: Prof. Dr. Nihat İNANÇ
Ekim 2008, 99 Sayfa

Bu çalışmada, adım motorlarının kombinasyonlu bir şekilde, kalemi X,Y,Z eksenlerinde hareket ettirerek, bilgisayar ile çizilen vektörel grafiklerin, kağıt üzerine çizilmesini sağlayan, kartezyen robotun mekanik konstrüksiyonunun tasarımı, incelenmesi ve programlanması amaçlanmıştır.

Çalışma, üç ana konuda incelenmiş ve dizayn edilmiştir. Birinci konu, X,Y,Z eksenlerinde doğrusal hareket yapan çizim robotunun konstrüksiyonunun tasarımı ve tasarımda kullanılan adım motorlarının incelenmesidir. İkinci konu, bilgisayardan gelen veriler ile robotta kullanılan motorları çalıştıracak olan elektronik kontrol kartın tasarımıdır. Üçüncü konu, vektörel çizim yapan ve adım motorlarının kontrolünü sağlayan programın tasarlanmasıdır.

Mekanik tasarımda, hybrid unipolar adım motorları, kayışlar, kasnaklar, çelik yataklama milleri ve sınır anahtarları kullanılmıştır. Çizim ve kontrol programı Delphi 7 programlama dili ile yazılmıştır. Program ve robot ile A4 kâğıdına çizgi, dikdörtgen, kare, daire, elips, yay, çokgen çizilebilmektedir. Bilgisayar ile elektronik kontrol kartın adaptasyonunda, paralel port kullanılmıştır.

Anahtar Kelimeler: Adım motoru, Bilgisayarla kontrol, Delphi, Kartezyen robot, Ploter, Paralel Port, Vektörel çizim.

ABSTRACT

CARTESIAN ROBOT DESIGN THROUGH VECTORAL DRAWING AND ITS PROGRAMMING

TORAMAN, M. Kemalettin

MSc, Electrical-Electronics Engineering Department

Supervisor: Prof. Dr. Nihat İNANÇ

October 2008, 99 Pages

In this study the design of a mechanic construction analysis and programming of a cartesian robot facilitating the vectoral drawing on the paper in the computer environment through moving the pen on X, Y, Z axis with a combination of step motors has been aimed.

The study was conducted under three main categories, and the due designs have been made. In the first stage, step motors were analyzed in order to be designed through a draw robotic construction having the capability of vertical movement on X, Y ,Z axis; in the second stage, the design of on electronic control card to be able to run the motors to be used in the robot and the data coming from the computer was made. The third stage, the design of a program capable of making vectoral drawing and the control of step motors will be made.

In the mechanic construction, hybrid unipolar step motors, belts, pulleys, steal embedded axles and border switches were used. The drawing and control programmers were written in Delphi 7 software. Through the programmed and robot, lines, rectangles, squares, circles, ellipses, bows, polygons can be drawn on A4 paper. In the adaptation of the electronic control card with the computer parallel port was used.

Keywords: Step motor, Control of computer, Delphi, Cartesian robot, Plotter, Parallel port, Vectoral drawing.

ÖNSÖZ

Bu çalışmada, bilgisayar ekranında vektörel grafiklerin oluşturulması, saklanması ve robota gönderilmesi için bir arayüz program tasarlanmıştır. Grafiklerin kağıt üzerine çizebilmesi için adım motorları, kayış, kasnak, çelik miller ve sınır anahtarları ile üç boyutlu bir kartezyen robot sistemi tasarlanmıştır.

Bu çalışmanın yapılmasında her türlü desteğini esirgemeyen, başta danışman hocam sayın Prof. Dr. Nihat İNANÇ'a, Prof. Dr. Sabir Rüstemli'ye, Yrd. Doç. Dr. Nuri ALMALI'ya Bilg. Yük. Müh. Musa ATAŞ'a, Elektronik Yük. Müh. Zeydin PALA'ya, Yrd. Doç. Dr. Haluk KULAZ'a, Öğr. Gör. Haluk YERGİN'e, robotun yapımında atölyelerini kullanmama izin veren ve yardımcı olan Teknisyen Süleyman DEMİR ve Teknisyen Munci MÜLAYİM'e ayrı ayrı teşekkürlerimi sunarım.

VAN-2008 M. Kemalettin TORAMAN

İÇİNDEKİLER

	sayfa
ÖZET	i
ABSTRACT	iii
ÖNSÖZ	v
İÇİNDEKİLER	vii
ŞEKİLLER DİZİNİ	ix
ÇİZELGELER DİZİNİ	xiii
EKLER	xv
SİMGELER ve KISALTMALAR DİZİNİ	xvii
1. GİRİŞ	1
2. KAYNAK BİLDİRİŞLERİ	5
3. MATERYAL ve YÖNTEM	8
3.1. Materyal	8
3.1.1. Adım motorları	8
3.1.1.1. Adım motoru çeşitleri	9
3.1.1.1.1. Değişken relüktanslı (VR) adım motorları	10
3.1.1.1.2. Sürekli mıknatıslı (PM) adım motorları	11
3.1.1.1.3. Karışık yapılı (Hybrid) adım motorları	12
3.1.1.2. Adım motoru sargıları	13
3.1.1.2.1. Bipolar sargılar	14
3.1.1.2.2. Unipolar sargılar	14
3.1.2. Paralel port (DB-25 Connector)	16
3.2. Yöntem	18
3.2.1. Vektörel çizim yapan program	24
3.2.2. Robotun grafikleri çizme yöntemi ve interpolasyon	31
3.2.2.1. Lineer çizgi çizme	32
3.2.2.2. Eğri çizgi çizme ve interpolasyon	33
3.2.2.2.1. Eğri çiziminde interpolasyon	36
3.2.2.3. Dikdörtgen ve kare çizme	39
3.2.2.4. Daire çizme ve interpolasyon	39
3.2.2.5. Elips çizme ve interpolasyon	41
3.2.2.6. Yay çizme ve interpolasyon	42
3.2.2.7. Çokgen çizme ve interpolasyon	43
4. BULGULAR	44
5. TARTIŞMA ve SONUÇ	51
6. KAYNAKLAR	54
7. EKLER	56
8. ÖZGEÇMİŞ	99

ŞEKİLLER DİZİNİ

sayfa

Şekil 1.1. Endüstriyel robotların mekanik yapılarına göre sınıflandırılması (Eğilmez, 2005). 3
Şekil 1.2. Kartezyen robot (Bayrak, 2007). 3
Şekil 3.1. Çok katmanlı değişken relüktanslı adım motoru (Baytuğan, 2000). 10
Şekil 3.2. Değişken relüktanslı adım motoru (Baytuğan, 2000). 10
Şekil 3.3. Sürekli mıknatıslı adım motoru (Altunsaçlı, 2003). 11
Şekil 3.4. Karışık yapılı (Hybrid) adım motoru. 12
Şekil 3.5. İki fazlı adım motoru (Benson, 2001). 13
Şekil 3.6. Dört fazlı adım motoru (Benson, 2001). 13
Şekil 3.7. Bipolar sargı (Apaydın, 2006). 14
Şekil 3.8. Bipolar sargı uyartılması (Eğilmez, 2005). 14
Şekil 3.9. Unipolar sargı (Apaydın, 2006). 15
Şekil 3.10. Unipolar sargının tam adım dizisi ile uyartılması (Eğilmez, 2005). 15
Şekil 3.11. Unipolar adım motorunun yüksek moment ve yarım adım dizisi ile uyartılması (Eğilmez, 2005). 15
Şekil 3.12. DB-25 Connector (paralel port). 16
Şekil 3.13. Paralel port pinleri. 17
Şekil 3.14. Robot sisteminin blok diyagramı. 18
Şekil 3.15. Robot sistemi. 19
Şekil 3.16. Robotun yapım aşaması. 19
Şekil 3.17. Robotun yapım aşaması. 19
Şekil 3.18. Robotun son hali. 19
Şekil 3.19. Switch anahtarlar, a.Mikro b.SMD 20
Şekil 3.20. Güç besleme devresi. 21
Şekil 3.21. ISIS programı ile devrenin tasarımı. 21
Şekil 3.22. Elektronik kartın 3 boyutlu devresi. 22
Şekil 3.23. ARES programı ile baskı devre tasarımı. 22
Şekil 3.24. ARES programı ile baskı devre tasarımının 3D görüntüsü. 22
Şekil 3.25 Elektronik devrenin baskı devresi. 23
Şekil 3.26. Elektronik devrenin montajdan sonraki hali. 23
Şekil 3.27. Paralel porta motorların ve switch anahtarların bağlanması. 23
Şekil 3.28. Programın yapım aşaması. 25
Şekil 3.29. Vektörel çizim yapan program. 25
Şekil 3.30. Program menüleri. 26
Şekil 3.31. Status menüsü. 26
Şekil 3.32. Göstergeler ve manüel hareket butonları. 26
Şekil 3.33. Program hakkında formu. 26
Şekil 3.34. Program ile çizilen örnek bir çizim. 30
Şekil 3.35. Adım sayısının X ve Y yönünde eşit olduğu eğri çizimi. 35
Şekil 3.36. Adım sayısının X yönünde fazla olduğu eğri çizimi. 35

Şekil 3.37. Adım sayısının Y yönünde fazla olduğu eğri çizimi. 35
Şekil 3.38. (25-11) olan eğri çizimi. 38
Şekil 3.39. Örnek fazla kalan adımların dağılış tablosu. 38
Şekil 3.40. Koordinat bulma. 41
Şekil 4.1. Düz çizgi çizme, a.Çizim aşaması, b.Ekrandaki çizim, c.Robot çizimi.45
Şekil 4.2. Eğri çizgi çizme, a.Çizim aşaması, b.Ekrandaki çizim, c.Robot çizimi.45
Şekil 4.3. Dikdörtgen çizme, a.Çizim aşaması, b. Ekrandaki çizim, c. Robot çizimi. 45
Şekil 4.4. Kare çizme, a.Çizim aşaması, b. Ekrandaki çizim, c. Robot çizimi. 45
Şekil 4.5.Bir daire çizimindeki veriler. 46
Şekil 4.6. Daire çizme, a.Çizim aşaması, b. Ekrandaki çizim, c. Robot çizimi. 47
Şekil 4.7. Elips çizme, a.Çizim aşaması, b. Ekrandaki çizim, c. Robot çizimi. 47
Şekil 4.8. Yay çizme, a.Çizim aşaması, b. Ekrandaki çizim, c. Robot çizimi. 48
Şekil 4.9. Çokgen çizme, a.Çizim aşaması, b. Ekrandaki çizim, c. Robot çizimi. 48
Şekil 4.10. Kaydedilen dosya çeşitleri ve büyüklükleri ile ilgili üç örnek. 49
Şekil 4.11. Başlangıç Koordinatı. 49

ÇİZELGELER DİZİNİ

sayfa

Çizelge 3.1. Adım motorların karşılaştırması (Atabey, 2007) 9
Çizelge 3.2. Paralel port adresleri 16
Çizelge 3.3. 888 nolu porta bilgi gönderme tablosu 29
Çizelge 3.4. 890 nolu porta bilgi gönderme tablosu 29
Çizelge 3.5. 889 nolu porttan bilgi okuma tablosu 29

EKLER DİZİNİ

Sayfa

Ek 1. Elektronik kontrol kartın ön ve arka yüzü. 56
Ek 2. Robot konstrüksiyonunda ve elektronik devrede kullanılan malzemeler. 56
Ek 3. Yapılan kartezyen robot. 56
Ek 4. Yapılan kartezyen robot ayrıntıları. 57
Ek 5. Örnek çizim ve Robot çizimi1. 57
Ek 6. Yukarıdaki Örnek Çizim1'in veri kaydı. 57
Ek 7. Örnek çizim ve Robot çizimi2. 58
Ek 8. Örnek çizim ve Robot çizimi3. 58
Ek 9. Kullanılan adım motorlarının karakteristik özellikleri. 59
Ek 10. Step motorlarda kablo bağlama diyagramları. 59
Ek 11. Bipolar ve Unipolar adım motorlarının sürülmesi. 59
Ek 12. Kartezyen robot kontrol program komutlarının tamamı. 60

SİMGELER ve KISALTMALAR DİZİNİ

Simgeler

α	Adım Açısı.
r	Adım motoru milindeki dişli yarıçapı.
l	Adım motorunun bir turdaki aldığı yol.
l_n	Adım motorunun bir adımdaki aldığı yol.
N	Adım motorunun bir turdaki adım sayısı.
R	Adım motoru milindeki dişli çapı.
3D	3 Dimension (3 Boyutlu).

Kısaltmalar

ASIC	Application Specific Integrated Circuit (Uygulamaya Özel Tümleşik Devre)
CNC	Computer Numerical Control (Bilgisayarlı Nümerik Kontrol).
DC	Direct Current (Doğru Akım).
Hybrid	Melez.
PIC	Peripheral Interface Controller (Çevresel Arayüz Kontrolörü).
PLC	Programmable Logic Controller (Programlanabilir Lojik Kontrolör).
PM	Permanent Magnet (Kalıcı Mıknatıs).
SA	Sınır anahtarı.
SMD	Surface Mount Device (Yüzeye Takılan Malzeme).
TTL	Transistor-Transistor Logic.
VR	Variable Reluctance (Değişken Relüktans).

1. GİRİŞ

İnsanlar, fiziksel yapılarından dolayı bütün işleri yapma imkânına sahip değillerdir ve hızın, üretim kalitesinin, hassaslığın gerektiği ve insan gücünün yetmediği yerlerde kullanmak üzere değişik makineler geliştirmişlerdir. İlk zamanlarda insan yardımı ile çalışan ve fonksiyonel olmayan bu makineler, sonraları insan yardımı olmadan, çok fonksiyonlu ve otomatik olarak çalışan makineler üretilmiştir. Otomasyon teknolojisinin çok hızla ilerlediği bu zaman diliminde, elektrik makinelerinin otomasyonu ile akıllı cihazlar üretilmektedir (Çengelci ve Çimen, 2005).

Sanayide, bir işlevin yerine getirilmesi için, üç boyutlu otomatik konumlama sistemi çok kullanılmaktadır. Örnek olarak; baskı devre delme makineleri, CNC freze tezgâhları (Metal ve ağaç işleme sanayisinde), CNC torna tezgâhları, yazıcılar, çiziciler, cam kesme makineleri, CNC oksijen-propan, plazma veya lazer kesim tezgâhları, ağaç işleme sanayisinde kullanılan rutırlar, CNC (Computer Numerical Control) planya tezgâhları gibi. Türkiye'de, klasik robotlara göre kartezyen robotların kullanımı daha fazladır (Oysu ve ark., 2004).

Robot; Üç veya daha fazla programlanabilir ekseni olan, otomatik kontrollü, tekrar programlanabilir, çok amaçlı manipülatördür (ISO 8373:1994). Manipülatörler, cisimleri önceden sabitlenmiş ilk konumlarından son konumlarına taşıyan sistemlerdir (Özyalçın, 2006).

Robot; programlandıkları şekilde cisimleri hareket ettirmek için tasarlanmış, programlanabilir, kendi kendini kontrol edebilen, fonksiyonel manipülatörlerdir. Robotlar hareketlerini, kontrolleri algılamaları ile otomatik olarak gerçekleştirir. Bu algılamalar, konum, ısı, ışık, hareket ve kimyasal gibi algılamalardır. Robot, bu algılamalar sonucu elde ettiği verileri kullanarak, karar vermeli ve uygulamaya sokmalıdır (Ersöz, 2007).

Robot 4 kısımdan meydana gelmektedir. Bunlar; mekanik yapı, hareketlendiriciler (elektrik motorları, hidrolik, pnömatik), algılayıcılar (sensörler) ve kontrol sistemidir (Sciavicco ve Siciliano, 1996). Robotlar, genellikle maliyeti

azaltmak, kaliteyi arttırmak için, insan sağlığının riskli olduğu, insan elinin ulaşamayacağı işlerde, insan gücünün yetmediği yerlerde kullanılmaktadır. Endüstride robotları ilk kullanan ülke Japonya'dır (Çengelci ve Çimen, 2005).

Robotlar, kesme, şekil verme, yüzey kaplama, silindirik ve düzlem yüzey taşlama gibi imalat işlemlerini gerçekleştirmekte ve montaj, kontrol uygulamalarında kullanılmaktadır. Günümüzde robotları inceleyen, robotik ve mekatronik bilim alanları ortaya çıkmıştır. İki bilim dalı da makine, elektronik ve bilgisayarın mühendisliklerinin ortak çalışma alanıdır. Robotik, canlılarla otomatik çalışan makineler arasındaki çalışma benzerliklerini araştırır (Ersöz, 2007; Bayrak, 2007). Robot sistemleri yapıları ve fonksiyonlarına göre 3 grupta toplanabilir. Bunlar;

1- Endüstriyel robotlar.
2- Operasyonel robotlar.
3- Tıp ve sağlık robotlarıdır.

Ayrıca, bunların dışında sibernetik robotlar, hobi robotları, sinema ve oyuncak endüstrisinde kullanılan robotlar gibi sektörel robotlarda kullanılmaktadır (Özyalçın, 2006).

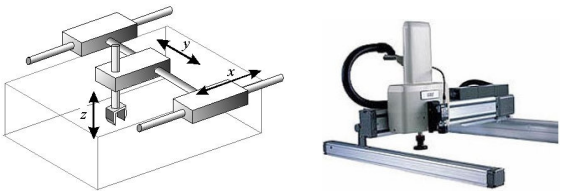
Robotların önemli amaçlarından biri, iş verimliliğini arttırmaktır. Robot kullanımında en verimli sonuca ulaşmak için, kullanım kolaylığını ve zaman kaybını en aza indirmek başta gelmektedir (Arslan, 1998; Ünlü, 1997).

Robotun hareketleri eklemler ile sağlanır. Robotlar; kartezyen (cartesian), silindirik (cylindrical), küresel (spherical), SCARA ve mafsallı (anthropomorphic) olarak sınıflandırılırlar (Koca, 2007).

Robotları farklı gruplara ayırabiliriz. Aşağıda robotların mekanik yapılarına göre sınıflandırılması görülmektedir (Şekil 1.1) (Eğilmez, 2005).

Robot	Eksenler		Örnekler
Prensip	Kinematik Yapı	Çalışma Alanı	Resim
Kartezyen Robot			
Silindirik Robot			
Küresel Robot			
SCARA Robot			
Mafsallı Robot			
Paralel Robot			

Şekil 1.1. Endüstriyel robotların mekanik yapılarına göre sınıflandırılması (Eğilmez, 2005).



Şekil 1.2. Kartezyen robot (Bayrak, 2007).

Kartezyen robot, birbirine dik olan üç prizmatik eklem kullanılarak elde edilir ve çok iyi derecede mekanik sertliğe sahiptir (Şekil 1.2) (Koca, 2007). Robot eklemleri birbirine diktir ve her bir eklem X,Y,Z koordinatlarından birine dik hareket eder. Sadece tutma ve taşıma yeteneği olan bu tip robot, doğrusal hareket etme yeteneğine sahiptir. Eğilme ve bükülme işlemlerini gerçekleştiremez. Çalışma alanları, kare veya dikdörtgen prizma şeklindedir. Yük taşıma kapasitesi diğer robot türlerine göre daha büyüktür. Tahrik sistemleri, elektrik motorları, hidrolik veya pnömatik tahrik sistemleri ile sağlanmaktadır. Islak, nemli, rutubetli çalışma ortamlarında kullanılabilir (Sciavicco ve Siciliano, 1996).

Elektrik motorları, elektrik enerjisini mekanik enerjiye dönüştüren makinelerdir. Manyetik alan içersinde kalan bir iletkene gerilim uygulandığında, biot-savart kanununa göre iletkene bir kuvvet tesir eder ve iletken endüklenen moment etkisi ile döner. Böylece elektrik enerjisi mekanik enerjiye dönüşür (Sarıoğlu, 1990).

Adım motorları, elektrik, elektronik, otomasyon ve mekatronik mühendisliği alanlarında yapılan çalışmalarda, konum hassalığı gereken uygulamalarda, tahrik ve pozisyon denetimi konularında, otomatik işlemleri gerçekleştiren ve kartezyen yapıda olan, yazıcılar, çiziciler, delme ve kesme makineleri gibi makinelerde en çok tercih edilen elektrik motorlarıdır. Doğrudan dijital sinyallerle çalışabilmesi, kontrol edilebilme kolaylığı, kapalı çevrim kontrol düzeneğine ihtiyaç duymaması nedeniyle tercih edilirler (Görgün, 1998).

2. KAYNAK BİLDİRİŞLERİ

Elektrik makinelerinin, bir yanında elektrik enerjisi diğer bir yanında mekanik enerji mevcuttur. Elektrik makinelerinin çözümlerinde, hem elektrik hem de mekanik bağıntılar kullanılmalıdır (Sarıoğlu, 1990).

Adım motorlarının performansı, kendi karakteristik değerlerinden sonra tamamen sürücü devrelerine ve kontrol yöntemlerine bağlıdır (Usta, 1991).

Günümüzde kullanılan robotlar çeşitli sınıflara ayrılabilirler. Bunlar; kullanılan eksen takımlarına göre, tiplerine göre, kullanılan tahrik elemanının çeşidine göre olabilir (Sciavicco ve Siciliano, 1996).

Adım motorunun davranışı, motoru süren güç kaynağına bağlıdır. Güç kaynağından elde edilecek palsler mikroişlemci veya bilgisayar tarafından kontrol edilir (Bal, 1997).

Adım motorları, bilgisayar kontrol ünitesinin girişine uygulanan her pals için, adım açısı kadar döner. Adım motorlarında problemlerden biri, motorun minimum zamanda istenilen referans hıza erişmesidir (Taştan, 1997).

Endüstriyel makinenin kullanımında, kullanıcı ile makine arasındaki iletişim kolay olmalıdır. Akıllı sistem ve makinelerin tasarlanması günümüz endüstriyel şartlarında artık kaçınılmazdır (Ünlü, 1997).

2 kutuplu adım motorunun sürülmesi için 8 bit, 4 kutuplu adım motorunun sürülmesi için 4 bit kontrol sinyali gereklidir. Maliyet ve kontrol açısından, 4 kutuplu adım motoru daha avantajlıdır (Arslan, 1998).

Adım motorları, sargılarının belli kombinasyonlarla enerjilendirilmesi ile hareket ederler. Adım motorlarının kullanıldığı konstrüksiyonlar basit ve ucuzdur. Sürücü devreleri için az sayıda komponent gereklidir. Genellikle 1W 3HP aralığında imal edilirler. Adım açıları 0.78^{o} ile 90^{o} arasındadır (Görgün, 1998).

Robotlar, kısa mesafelerde kullanıldığında hareket mesafeleri kısadır. Hareket etme ve durma ani olmalıdır. Sürücü devreler basit olmalıdır. Bu sebeplerden dolayı adım motorları çok kullanılmaktadır (Çetişli, 1998).

Sistemlerde kullanılan çok çeşitli elektrik makineleri vardır. Bu makinelerin ortak yanları, elektrik enerjisini mekanik enerjiye dönüştürmeleridir. Konum ve hız kontrolünde uygun motorun seçilmesi gereklidir (Özkan,1999).

Paralel portlar 9 yâda 12 bit çıkış verebilir ve basit bir devre ile harici üniteler kontrol edilebilir. 4 kontrol, 5 durum ve 8 veri hattına sahiptir. Yeni paralel portlar IEEE 1284 standardını desteklerler (Baytuğan, 2000).

Adım motorları, girişlerine uygulanan palslere karşılık, analog dönme hareketi yapabilen elektromagnetik elemanlardır. Bu özellikleri nedeniyle dijital sistemlerde çok kullanılırlar (Benson, 2001).

Adım motorları robot teknolojisinde sıkça kullanım alanı bulmuştur. Ayıca, maliyetinin düşük olması DC servo motorlara karşı bir üstünlüğüdür. Adım motorlarının kullanılmasının ikinci bir nedeni de tutma karakteristiğinin robotlarla bağdaşmasıdır (Altunsaçlı, 2003).

Delphi 7 çok gelişmiş bir programlama dilidir. Gerek veri tabanı uygulamaları, gerekse diğer uygulamalarda son derece gelişmiş projeler oluşturmak, Delphi ile çok kolaylaşmıştır (Demirli ve İnan, 2003).

Windows'ta paralel port adresleri bios aracılığıyla değiştirilebilir. Bilgisayardaki port adreslerini öğrenebilmek için Denetim Masası – Sistem bölümündeki Aygıt Yöneticisi kullanılabilir (Pala, 2003).

Günümüzde, otomasyona dayalı imalat sistemi genişlemiş, optimizasyon devrine geçilmiş, prodüktiviteyi arttırarak yatırım masraflarını azaltan, daha fazla üretim yapmanın yanı sıra kaliteyi de artıran ve bu arada çok daha insancıl çalışma koşulları sunan "robot" devri başlamıştır (Zeylan, 2003).

Sanayide motorların, XYZ boyutlarında hareketi ve istenen pozisyona hızlı ve doğru olarak gelmeleri çok önemlidir. Kartezyen robotların Türkiye de klasik seri robotlara göre kullanımı daha fazladır (Oysu ve ark., 2004).

Elektrik makinelerinde elektromanyetik sistem önemli bir rol oynar. Bu makinelerde hareket, enerji dönüşümü ve enerji transferi işlemleri ile gerçekleştirilir (Okumuş ve Gümüşoluk, 2004).

Otomasyon teknolojilerini, ekonomik olarak gelişmiş olan ülkeler etkin bir şekilde kullanmaktadır. Buna bağlı olarak endüstriyel robotların önemi her geçen gün artmaktadır (Çengelci ve Çimen, 2005).

Kartezyen koordinatlı kaynak robotu ile yapılan çalışmaların test sonuçlarına bakıldığında robotun, elle yapılan gazaltı ark kaynağından daha başarılı sonuçlar alabildiği gözlenmiştir (Eğilmez, 2005).

Delphide portlara erişmek için herhangi bir komut bulunmaz. Windows 95/98 ve ME serisinde portlara direk erişilebilirken, XP ve VISTA serisinde portlara direk erişilememektedir (Karagülle, 2006).

Gelişmiş ülkeler, imalat sistemlerinde otomasyon teknolojilerini ve robot manipülatörlerini yaygın olarak kullanmaktadırlar. Bu sistemler içerisinde en yaygın kullanılanı, açık yapılı uyarlanabilir olanlarıdır (Özyalçın, 2006).

Adım motoru kullanımında, motorların sürülmesi ve kontrolü konusu büyük önem arzetmektedir. Teknolojik gelişmeler ile birlikte, günümüzde bilgisayarların, mikroişlemcilerin, microdenetleyicilerin gelişmesi ile bu sorunların üstesinden rahatlıkla gelinebilmektedir. (Apaydın, 2006).

Robot kolunun ters kinematik problemi, Paden-Kahan alt problemlerine indirgenerek çözülebilir. Daha sonra ters kinematik probleminde, robot kolunun uç noktasına gitmesi istenen noktanın x,y,z koordinatları ve başlangıç pozisyonuna göre açısı verilerek eklem açıları hesaplanabilir (Bayrak, 2007).

Günümüzde endüstriyel robotlar, daha ekonomik ve daha kaliteli ürün üretebilmektedirler. Bunun neticesinde, ürünlerin girdi maliyetleri azalır ve firmalar rekabet ortamında öne geçmeye başlarlar (Ersöz, 2007).

Robot manipülatörün hareket yeteneğini arttırmak için, eksen sayısının arttırılması gerekir. Fakat eksen sayısının arttırılmasıyla, robot manipülatörün denetimi de zorlaşmakta ve denetim algoritması daha karmaşık hale gelmektedir (Koca, 2007).

Konum kontrolünde, akla gelen ilk elektrik motor tipi sürekli mıknatıslı adım motorlardır. Adım motorları, konum kontrolü için optimize edilmiş olup, kullanımları çok kolaydır (Atabey, 2007).

3. MATERYAL ve YÖNTEM

3.1. Materyal

3.1.1. Adım motorları

Uygulamalarda kullanılan elektrik motorları birbirlerine göre farklılıklar gösterirler. Sistemlerde, konum ve hız kontrolü büyük önem taşımaktadır. Bu nedenle, sisteme uygun motorun seçilmesi gereklidir. Konum ve hız kontrolünün gerekli olduğu yerlerde, genellikle adım veya servo motorlar kullanılır. Adım motorları daha çok küçük güçlü sistemlerde ve düşük moment gerektiren kontrollerde tercih edilir. Bunun yanında, büyük güç, yüksek moment ve hızlı tepki gerektiren sistemlerde ise daha çok servo motorlar kullanılır (Özkan, 1999).

Adım motorları, uygulanan elektriksel sinyalleri mekanik harekete çeviren, moment-atalet oranları yüksek olan elektromekanik cihazlardır. Genellikle, yüksek çözünürlük ve minimum hata gerektiren, hassas konum kontrolü gerektiği yerlerde kullanılır. Ürettikleri momente göre boyutları küçüktür (Atabey, 2007).

Diğer klasik elektrik motorlarından farklı olarak, sürekli bir dönme hareketi yerine, eşit büyüklükte adım açıları ile hareket yapabilen elektrik motorlarıdır. Motor girişine bir tek pals verildiğinde, rotor bir adım hareket eder ve durur. Daha fazla pals uygulanınca, pals adedi kadar hareket eder ve durur. Palslerin arka arkaya uygulanmasıyla, rotor devamlı olarak döner. Güvenilir olması, iyi pozisyon alması, dijital sinyallerle çalışabilmesi, devir sayısının geniş sınırlar içersinde değiştirilebilmesi, çoğunlukla geri besleme ünitelerine ihtiyaç duymaması, ucuz olması, rotor dururken tutma momentinin yüksek olması nedenleriyle çok kullanılmaktadır. (Apaydın, 2006).

Adım motoru, bir güç kaynağı tarafından beslenir. Güç kaynağından adım motoruna uygulanan gerilim, dijital bir kontrolör tarafından kontrol edilir. Bilgisayarla kontrol edilen adım motorlarında, darbeler bilgisayar tarafından sayılır ve kaydedilir. İleri yöndeki darbeler (+), geri yöndeki darbeler (-) olarak

işlem görür. Böylece, adım motorunun devir sayısı ve pozisyonu her zaman hassas olarak bilinir. Adım motorlarında, stator ve rotor kutup sayıları birbirlerine asla eşit olamaz (Çetişli, 1998).

Çok hassas pozisyon kontrolü gerektiren cihazlarda, x,y koordinat işaretleyicilerde, disket okuyucularda, şerit sürücü ve okuyucularda, optik okuyucularda, yazıcı ve çizicilerde, nümerik kontrollü CNC tezgâhlarında, robot uygulamalarında, uydu antenlerinin yön kontrolünde, faks ve telefaks makinelerinde, araba saatlerinde, ölçü aletlerinde, kağıt sürme düzenlerinde, imalat makinelerinin programlanmasında, çeşitli uzay çalışmalarında, adım motorları etkin olarak kullanılmaktadır (Görgün, 1998).

Adım motoru, bazı Türkçe literatürlerde step motoru, yabancı literatürlerde ise step motor, stepper motor veya stepping motor olarak da yer almaktadır.

3.1.1.1. Adım motoru çeşitleri

Üretilen adım motorları çok çeşitli olup; selenoid, değişken relüktanslı, hybrid, rotoru sabit mıknatıslı, elektrohidrolik ve elektromekanik gibi türleri olmasına karşın en yaygın kullanılan tipleri;

• Değişken Relüktanslı Adım Motorları (VR)

• Sabit Mıknatıslı Adım Motorları (PM)

• Karma Tip Adım Motorlarıdır (HYBRID) (Apaydın, 2006).

Adım motorları moment, hız aralıkları ve adım çözünürlükleri açısından Çizelge 3.1'de karşılaştırılmaktadır (Atabey, 2007).

Çizelge 3.1 Adım motorların karşılaştırması (Atabey, 2007)

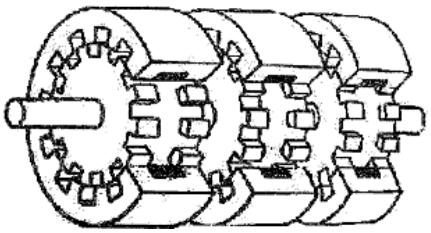
	Moment	**Hız**	**Çözünürlük**
VR	En düşük	En yüksek	Düşük
PM	En yüksek	En düşük	İyi
Hybrid	İyi	İyi	En yüksek

3.1.1.1.1. Değişken relüktanslı (VR) adım motorları

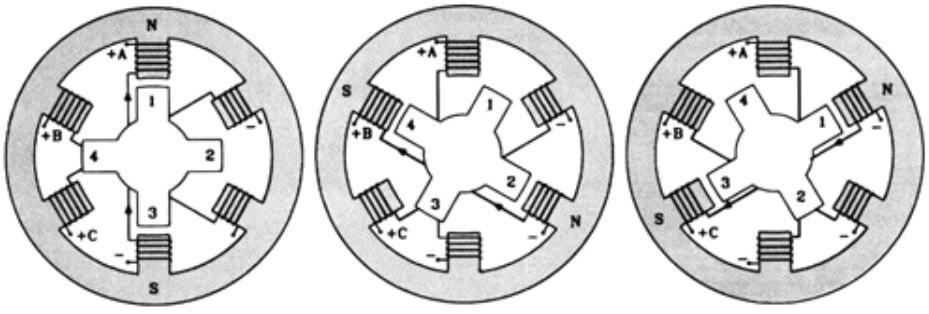
Rotor ve stator çekirdeği, ince tabakalar halinde üst üste sıkıştırılmış silikon çelikten yapılır ve yüksek magnetik geçirgenliği vardır. Hava aralığı mümkün olduğunca küçük olmalıdır. Stator ve rotor dişleri aynı doğrultuda değilse, manyetik relüktans çok büyüktür. Çok ve tek katmanlı yapılabilirler. Çok katmanlı yapılmasındaki amaç, adım açısını küçültmektir (Şekil 3.1) (Baytuğan, 2000).

Bu motorlar, stator ve rotor hava aralığını minimum tutarak, minimum relüktans prensibiyle çalışırlar (Görgün, 1998).

Bu tür motorlarda, stator ve rotor diş sayısı aynı değildir, en az 4 kutuplu stator bulunur. Sargılar üzerinden çift yönlü akım geçer, rotorun pozisyonu bir sonraki hareket için önemlidir (Şekil 3.2). Rotor, artık mıknatıslık özelliği göstermeyen ve dişleri açılmış yumuşak demirden yapılır (Altunsaçlı, 2003).



Şekil 3.1. Çok katmanlı değişken relüktanslı adım motoru (Baytuğan, 2000).



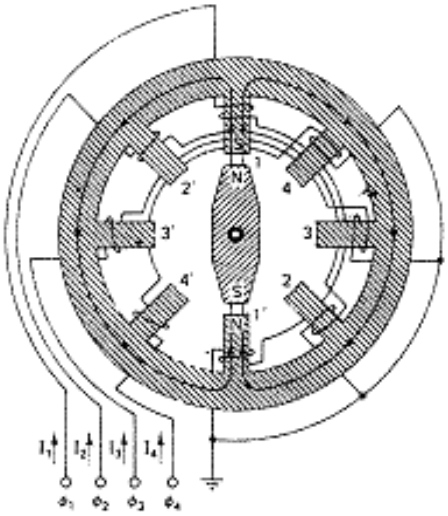
Şekil 3.2. Değişken relüktanslı adım motoru (Baytuğan, 2000).

3.1.1.1.2. Sürekli mıknatıslı (PM) adım motorları

Bu motorların rotoru daimi mıknatıslı, statoru ise çok faz sargılıdır (Şekil 3.3). PM ve VR adım motorlarında daha küçük adım açısı, rotor diş sayısının veya stator kutup sayısının arttırılmasıyla elde edilir. Kullanım alanları azdır. Günümüzde kullanılan bütün PM motorları küçük adımlara sahiptirler. Dört fazlı adım motorlarının tipik adım açıları 30°, 15°, ve 7.5°dir. Motorun meydana getirdiği moment, stator sargılarındaki akımla orantılıdır (Okumuş ve Gümüşoluk, 2004).

PM motorları ile VR motorları karşılaştırılırsa, PM motorları aynı ölçüdeki VR motorlarına göre, küçük motor boyutları için büyük moment üretirler (Baytuğan, 2000).

Bu tip motorların rotorları sabit mıknatıstan yapıldığı için, en son uyartılan bobin altında kalan kutup, tutma momenti nedeniyle, enerji kesildiğinde dahi olduğu yerde sabit kalır. Bu tip motorların rotorlarında diş bulunmaz. Stator sargıları karşılıklı olarak seri bağlıdırlar (Bal, 1997).



Şekil 3.3. Sürekli mıknatıslı adım motoru (Altunsaçlı, 2003).

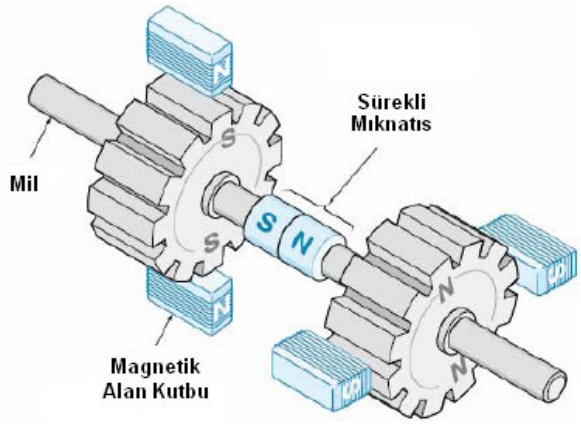
3.1.1.1.3. Karışık yapılı (Hybrid) adım motorları

Hybrid adım motorlarının kutupları VR adım motorları gibidir. Rotoru ise sabit mıknatıstan yapılmıştır (Okumuş ve Gümüşoluk, 2004).

Bu tip motorların rotorunda, eksenel doğrultuda monte edilmiş sabit bir mıknatıs bulunur. Hem stator hem de rotor uç parçaları dişli yapıdadır. Genel olarak altı stator kutbu bulunur ve her kutupta yaklaşık iki ile altı arasında diş vardır (Şekil 3.4). Küçük adım ve yüksek moment gerektiren uygulamalarda tercih edilir (Apaydın, 2006).

Hybrid adım motorları, PM ve VR adım motorlarının özelliklerini birleştirir ve günümüzde en yaygın kullanılan adım motorlarıdır. Bu motorlarda, birbirinin aynı olan, yumuşak demirden yapılmış rotor kutupları aynı mil üzerine monte edilmiştir (Bal, 1997).

Hybrid adım motorları, aynı boyutlardaki VR ve PM adım motorları ile karşılaştırıldığında, hybrid adım motorları daha büyük senkronlama, tutma, dinamik momente ve hızlı adımlama oranına sahiptir. Tüm adım motorları içinde hybrid adım motorları, her türlü doğrusal ve açısal konumlama sistemlerinde en yaygın kullanılan tiptir (Görgün, 1998).



Şekil 3.4. Karışık yapılı (Hybrid) adım motoru.

Bir adım motorunda adım açısı, stator ve rotor dişlerinin derecelerinin farkından bulunabilir. Örneğin; 50 stator kutbu, 40 rotor kutbu olan adım motorunun bir adım açısı;

Stator diş derecesi = 360^o / stator diş sayısı = 360^o / 50 = 7.2^o

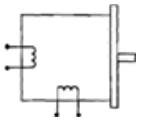
Rotor diş derecesi = 360^o / rotor diş sayısı = 360^o / 40 = 9^o

Adım açısı = Diş derecelerindeki fark = $9^o - 7.2^o = 1.8^o$ olduğundan, bu adım motorunun adım açısı 1.8^odir. Bu adım motoru bir turu; tam adımda 360^o / 1.8^o = 200 adım da, yarım adımda ise 2 x 200 = 400 adımda alır.

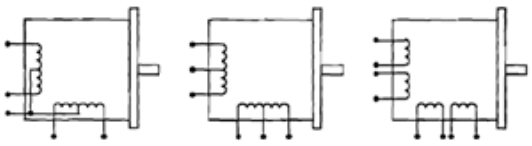
Diğer adım motorlarının yapısı ve çalışma prensipleri, bu sayılan üç motor göz önüne alınarak geliştirilmiştir. Ayrıca, bu motorlardan farklı olarak, hidrolik, pnömatik ve lineer adım motorları da vardır (Usta, 1991).

3.1.1.2. Adım motoru sargıları

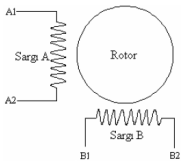
Adım motorları genelde 2 veya 4 fazlıdır. 2 fazlı adım motorları, 2 bobinli 4 tellidirler (Şekil 3.5), 4 fazlı adım motorları 4 bireysel sargıya veya 2 merkezli sargıya sahiptir ve sargı ekleri, motor içinde veya dışında yapılabilir (Şekil 3.6). 2 fazlı motorların kontrolleri, bipolar sürücü devrelerdir ve sargı polariteleri her adım sırasında çaprazlama bir şekilde değişir. 4 fazlı motorların kontrolleri ise unipolar sürücü devrelerdir ve sargı yönleri ile polariteleri değişmez (Benson, 2001).



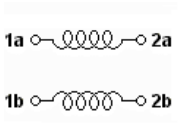
Şekil 3.5. İki fazlı adım motoru (Benson, 2001).



Şekil 3.6. Dört fazlı adım motoru (Benson, 2001).



Şekil 3.7. Bipolar sargı (Apaydın, 2006).



	1a	1b	2a	2b
1	+	–	–	–
2	–	+	–	–
3	–	–	+	–
4	–	–	–	+

Şekil 3.8. Bipolar sargı uyartılması (Eğilmez, 2005).

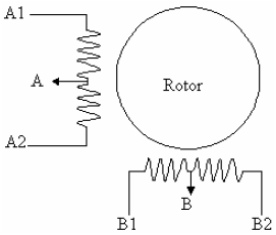
3.1.1.2.1. Bipolar sargılar

Bipolar adım motorlarında 2 sargı bulunmaktadır ve bu sargıların uçları dört iletken halinde dışarıya alınmıştır (Şekil 3.7). Daha karmaşık sürücü devresi gerektirir. Unipolar motorlardan daha fazla moment üretirler. Sürücü olarak H-köprü devreleri veya özel olarak tasarlanmış entegreler (L297, L298, LMD18T245) kullanılır. Bu entegrelerin içinde genelde tek bir H-köprüsü vardır. Fakat bu motoru sürmek için iki H-köprüye ihtiyaç olduğundan, bu entegrelerden iki tane kullanılır (Eğilmez, 2005).

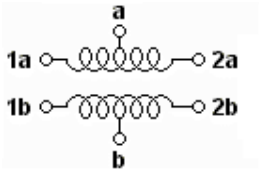
Bipolar sargı uyartılması, herhangi bir uca (+) verildiğinde diğer tüm uçların (-) olması gerekiyor (Şekil 3.8). Bu ise, sürücü devre yapmayı zorlaştırıcı bir etkendir (Apaydın, 2006).

3.1.1.2.2. Unipolar sargılar

Unipolar adım motorlarında, her biri ota uçlu olmak üzere iki faz sargısı bulunmaktadır (Şekil 3.9). İki faz sargısının orta uçları, ayrı ayrı ilkten halinde dışarıya alınabildiği gibi, içten birbirlerine bağlanarak tek bir iletken halinde de dışarıya alınabilmektedir (Şekil 3.6). Unipolar adım motorunun kontrolü nispeten kolaydır. Sargılardan tek yönde akım geçer (Benson, 2001).



Şekil 3.9. Unipolar sargı (Apaydın, 2006).



	1a	1b	2a	2b
1	1	0	0	0
2	0	1	0	0
3	0	0	1	0
4	0	0	0	1

Şekil 3.10. Unipolar sargının tam adım dizisi ile uyartılması (Eğilmez, 2005).

	1a	1b	2a	2b
1	1	0	0	1
2	1	1	0	0
3	0	1	1	0
4	0	0	1	1

Değişimli tam adım dizisi
(Yüksek moment)

	1a	1b	2a	2b
1	1	0	0	0
2	1	1	0	0
3	0	1	0	0
4	0	1	1	0
5	0	0	1	0
6	0	0	1	1
7	0	0	0	1
8	1	0	0	1

Yarım adım dizisi

Şekil 3.11. Unipolar adım motorunun yüksek moment ve yarım adım dizisi ile uyartılması (Eğilmez, 2005).

Ortak uca göre diğer bobin uçlarına uygun sıralamada palsler uygulanarak çalıştırılırlar. Standart sürüm sırasına (Şekil 3.10) ek olarak yüksek moment ve yarım adım sürüm sıraları da (Şekil 3.11) kullanılabilmektedir. Bipolar adım motorlarına göre avantajı sürücü devrelerinin daha basit olmasıdır. (Apaydın, 2006).

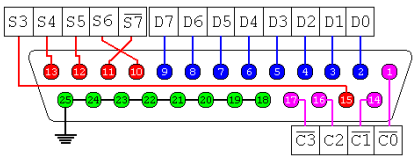
3.1.2. Paralel port (DB-25 Connector)

Paralel port, 25 pinden oluşan 8 bit veri aktarabilen porttur (Şekil 3.12). Pinlerin değeri müdahale olmadığı zaman “1” dir. Bu pinlerden istenen pinin değeri değiştirilebilir. Bu pinlerden her hangi bir tanesinin; "1" olması o pinden okunacak gerilimin +5V, "0" olması ise 0V olması anlamına gelir. Ancak bu pinlerden çekilebilecek akım değerleri çok düşüktür (Zeylan, 2003).

Paralel port, bilgisayarın arkasında DB tipi 25 pin dişi konnektördür. Çıkışları TTL (Transistor-Transistor Logic) mantığı ile yapılmıştır. Portun akımı, ASIC’e (Application Specific Integrated Circuit) göre yaklaşık 12mA dir (Baytuğan, 2000).

Windows’ta sayısal bir sıralamayla, birinci port LPT1, ikinci LPT2, üçüncü LPT3 olarak adlandırılır. LPT1 genellikle 378h adresindedir. Ancak üç adresten herhangi birinde de olabilir. LPT2 ise, eğer mevcutsa, 378h veya 278h adresinde; LPT3 ise yalnızca 278h adresinde olabilir. Bu adresler çeşitli yapılandırma teknikleriyle değiştirilebilir (Çizelge 3.2).

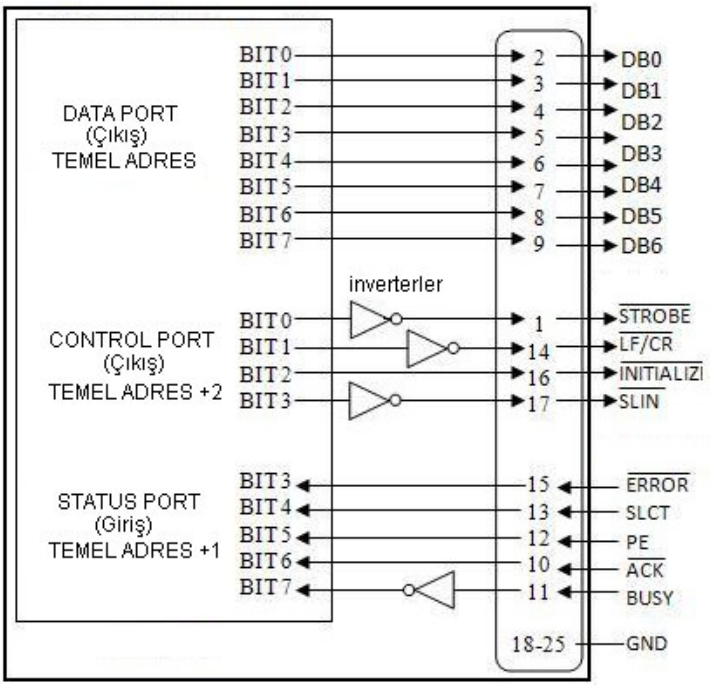
Her paralel port data, status ve control olmak üzere 3 ayrı port adresinden oluşur. Bunlara paralel port yazmaçları da denir. Bilgisayarın en kolay programlanabilir portudur (Şekil 3.13). En fazla 5m kablo kullanılabilir.



Şekil 3.12. DB-25 Connector (paralel port).

Çizelge 3.2. Paralel port adresleri

İsim	Taban Adres	Düşük Bayt	Yüksek Bayt	Hex	Data	Status	Control	Dec	IRQ
LPT1	0000:0408	78	03	0378	378	379	37Ah	888	07
LPT2	0000:0408	78	02	0278	278	279	27Sh	632	07
LPT3	0000:0408	BC	03	03BC	3BC	3BD	3BE	956	07



Şekil 3.13. Paralel port pinleri.

1.DATA PORT : 8 bit (D0-D7) sayısal çıkış yapılabilir.

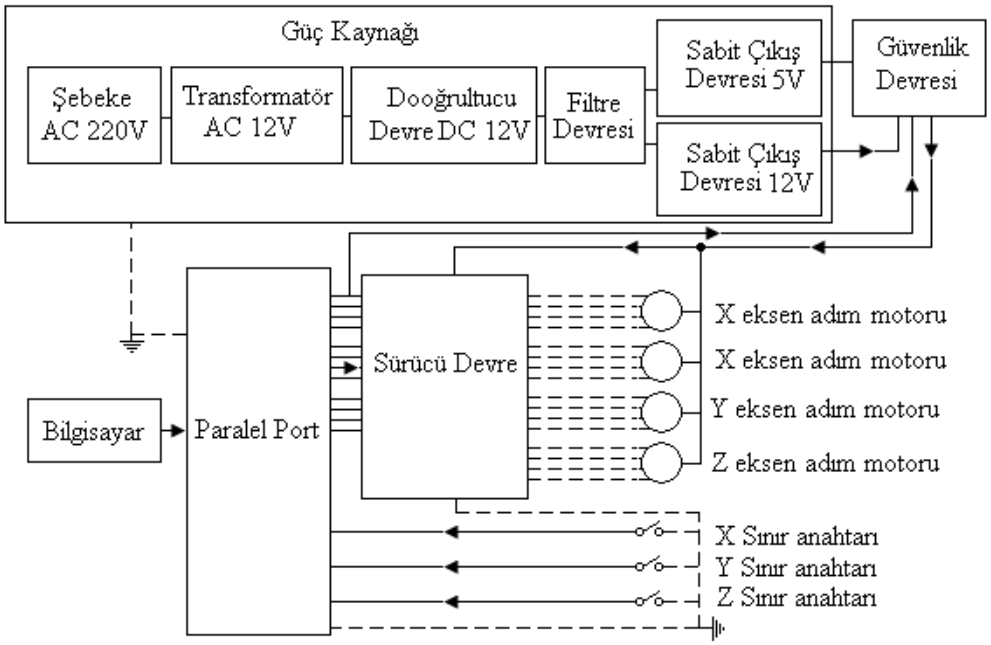
2.STATUS PORT : 5 bit sayısal giriş yapılabilir. 11 nolu pin donanım tarafından tersindirilmiştir.

3. CONTROL PORT: 4 adet kontrol sinyalini, hem giriş hem de çıkış için kullanmak mümkündür Data ve status pinlerinin yetmediği zaman, control portları ile giriş veya çıkış yapmak mümkündür (Şekil 3.13).

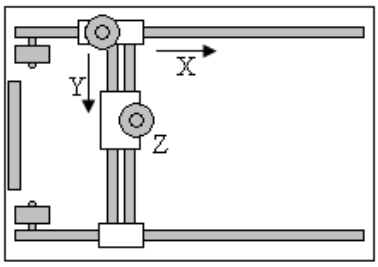
3.2. Yöntem

Bu çalışmada yapılan robot, endüstriyel hizmet verebilecek şekilde tasarlanmayıp, ihtiyacımıza cevap verecek ölçüde model bir robottur. Sistem oluşturulmadan önce blok diyagramı oluşturulmuştur (Şekil 3.14). Çizimde kullanılan kalem, esas kontrol edilmek istenen hareketli kısımdır. Yazma sistemi, köprü üzerinde Z ekseninde aşağı-yukarı hareket ederken, köprüde gövde üzerinde X ve Y ekseninde hareket etmektedir. Dolayısıyla, yazma sistemi tabla üzerinde bir dikdörtgensel bölge içersinde istenilen grafikleri kağıt üzerine çizmektedir (Şekil 3.15).

Atölyede, adım motorları, kayış, kasnak ve çelik yataklama milleri kasa üzerine hassas bir şekilde yerleştirilerek kartezyen koordinatlı robotun mekanik konstrüksiyonu oluşturulmuştur (Şekil 3.15; Şekil 3.16; Şekil 3.17; Şekil 3.18).



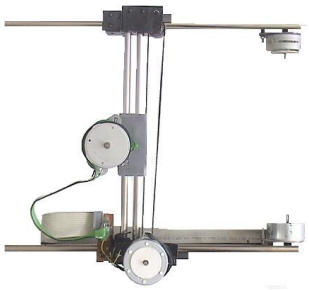
Şekil 3.14. Robot sisteminin blok diyagramı.



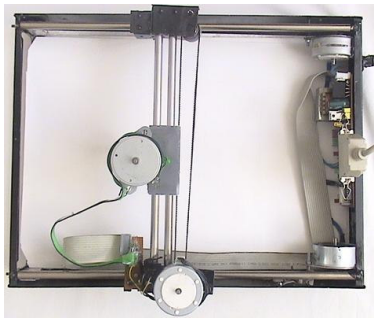
Şekil 3.15. Robot sistemi.



Şekil 3.16. Robotun yapım aşaması.



Şekil 3.17. Robotun yapım aşaması.



Şekil 3.18. Robotun son hali.



(a) (b)

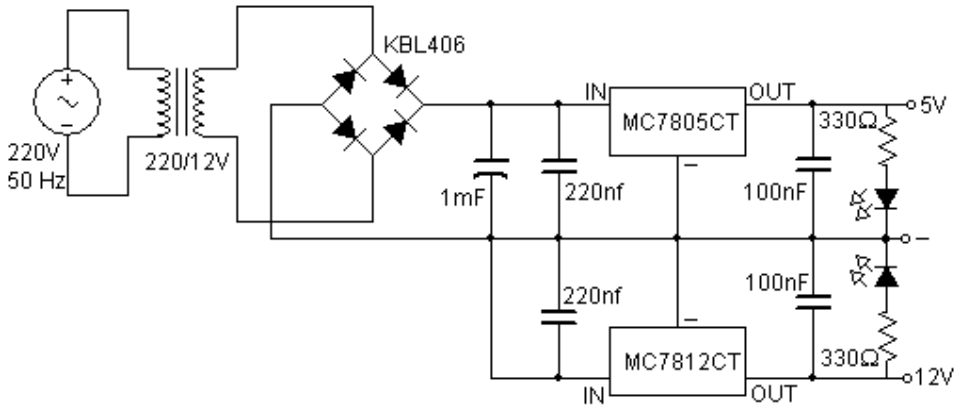
Şekil 3.19. Switch anahtarlar, a.Mikro b.SMD

Sistemde oluşabilecek mekanik hataları gidermek için, motorlar monte edilirken, motor bağlantı saplama vidalarında yay kullanılmıştır. Aynı şekilde, robot mekaniğinde kullanılan kayışların gerginliklerinin ayarlanabilmesi için, kasnakların sabitlenmesinde kullanılan saplama vidalarında yay kullanılmıştır. Robot üzerine, adım motorlarının başlangıç pozisyonlarında olup olmadıklarının kontrolü için, 2 adet mikro switch anahtar (limit sensör), 1 adet SMD (Surface Mount Device) switch anahtar kullanılmıştır (Şekil 3.19). SMD switch anahtar, alan dar olduğundan kalemi kontrol eden motor kontrolü için kullanılmıştır.

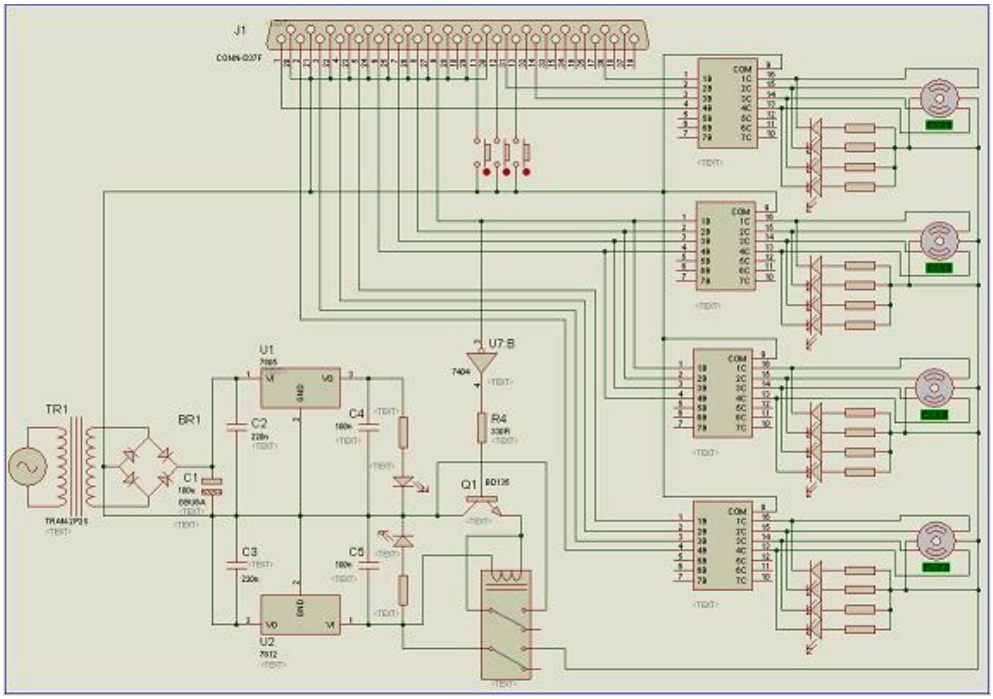
Kullanılan kalemin tüm yüzeye aynı basınçla basabilmesi için, kalemin arkasına yay takılmıştır. Adım motorlarına giden sinyalleri gözlemleyebilmek için, adım motorlarını süren ULN2003AN entegresinin her çıkışına birer led diyot takılmıştır.

Robot yapımında, 4 adet karışık yapılı unipolar hybrid adım motoru kullanılmıştır. Çizim kalemini X ekseninde hareket ettirmek için 2 adım motoru, Y ve Z ekseninde hareket ettirmek için ise birer adım motoru kullanılmıştır. Atölye şartlarında, çizim kalemini X ekseninde hareket ettirmek için bir motorun gücü yetmediğinden iki adım motoru kullanılmıştır. X ve Y ekseninde kullanılan unipolar adım motorları yarım adımlı, Z eksenindeki motor ise tam adımlı çalıştırılmıştır.

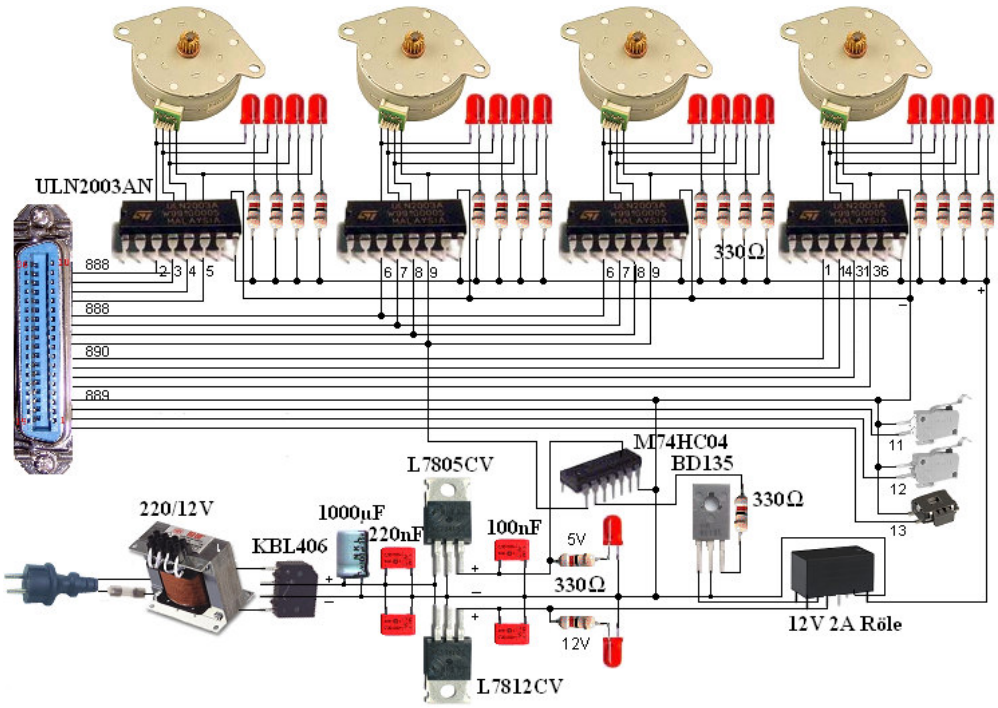
Kartezyen robot atölyede yapıldıktan sonra, ISIS programı ile elektronik devrenin güç besleme katı (Şekil 3.20) ve kontrol katı devresi tasarlanarak iki devre birleştirilmiştir (Şekil 3.21). Ardından, kartın 3 boyutlu devresi çizilmiştir (Şekil 3.22). Devre tasarımı yapıldıktan sonra, ARES programı ile baskı devre dizayn edilerek (Şekil 3.23), 3 boyutlu görüntüsü oluşturulmuştur (Şekil 3.24). Atölyede, Perhidrol ve tuzruhu kullanılarak baskı devre çıkartılmıştır (Şekil 3.25).



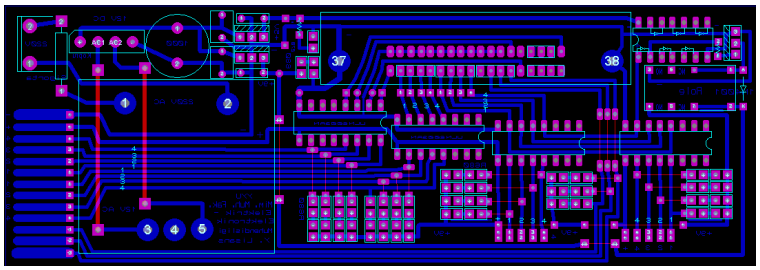
Şekil 3.20. Güç besleme devresi.



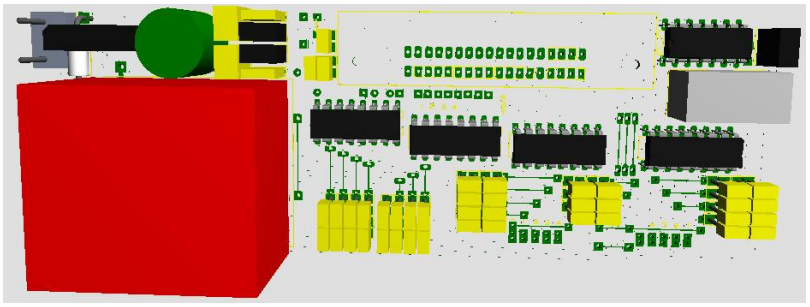
Şekil 3.21. ISIS programı ile devrenin tasarımı.



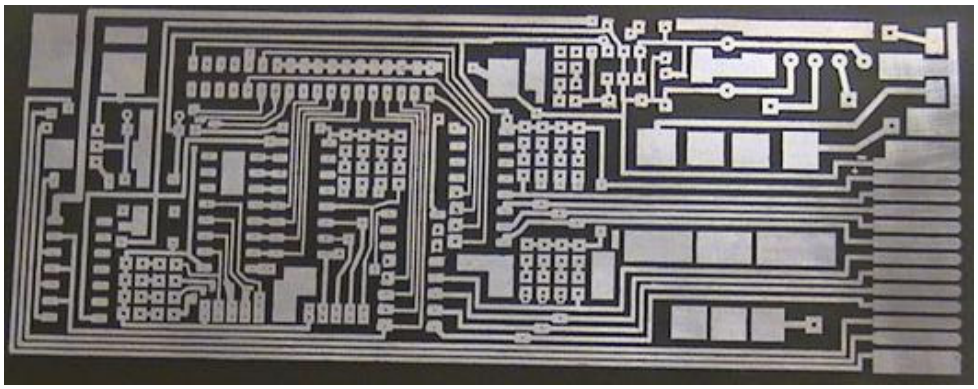
Şekil 3.22. Elektronik kartın 3 boyutlu devresi.



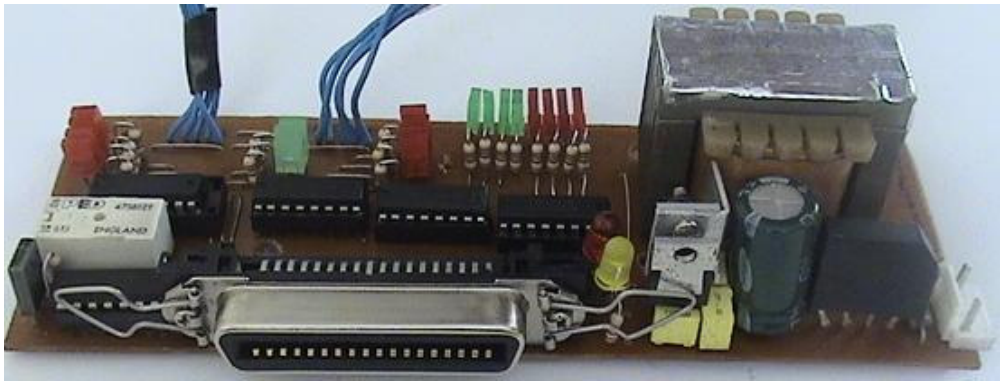
Şekil 3.23. ARES programı ile baskı devre tasarımı.



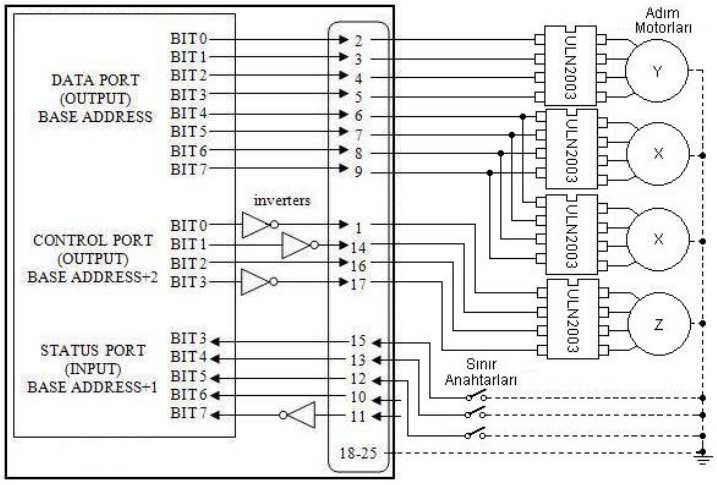
Şekil 3.24. ARES programı ile baskı devre tasarımının 3D görüntüsü.



Şekil 3.25 Elektronik devrenin baskı devresi.



Şekil 3.26. Elektronik devrenin montajdan sonraki hali.



Şekil 3.27. Paralel porta motorların ve switch anahtarların bağlanması.

Elektronik devre yapımının son aşamasında malzemeler baskı devreye yerleştirilerek (Şekil 3.26) robot üzerine monte edilmiştir. Yapılan elektronik devre, akıllı bir devre olmayıp, yalnızca bilgisayarla motorlar arasında köprü vazifesi görmektedir.

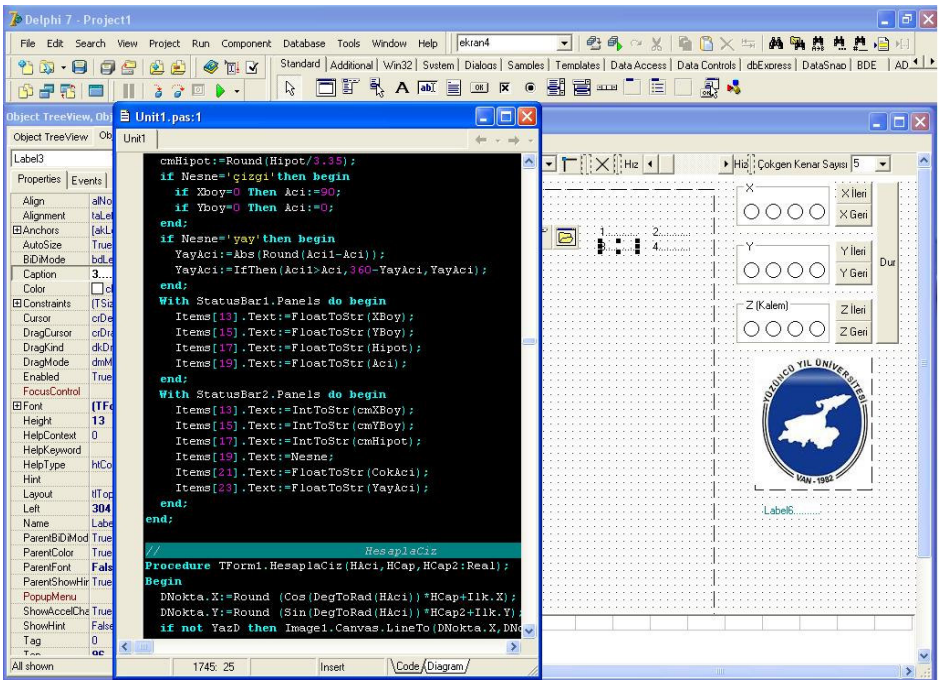
Elektronik devrede, 220V alternatif gerilim, 220/12V trafo ile 12V'ta düşürülmüştür. Ardından, köprü diyot (KBL406) ile doğru gerilime dönüştürüldükten sonra, kondansatör (1000 µF) ile L7805CV ve L7812CV regülatör entegreleri ile 5V ve 12V'luk iki ayrı sabit gerilim elde edilmiştir. 12V kontrol siteminde, 5V ise güvenlik devresinde kullanılmıştır.

Paralel portun, decimal 888 adresindeki data yazmacının ilk 4 biti Y motorunu, son 4 biti ise iki X motorunu çalıştırmak için kullanılmıştır. 890 (888+2) adresindeki control yazmacının 4 biti ise Z motorunu çalıştırmak için kullanılmıştır. 889 (888+1) status yazmacının 3 biti ise SA (sınır anahtarlarından) gelecek bilgileri kontrol için kullanılmıştır (Şekil 3.27).

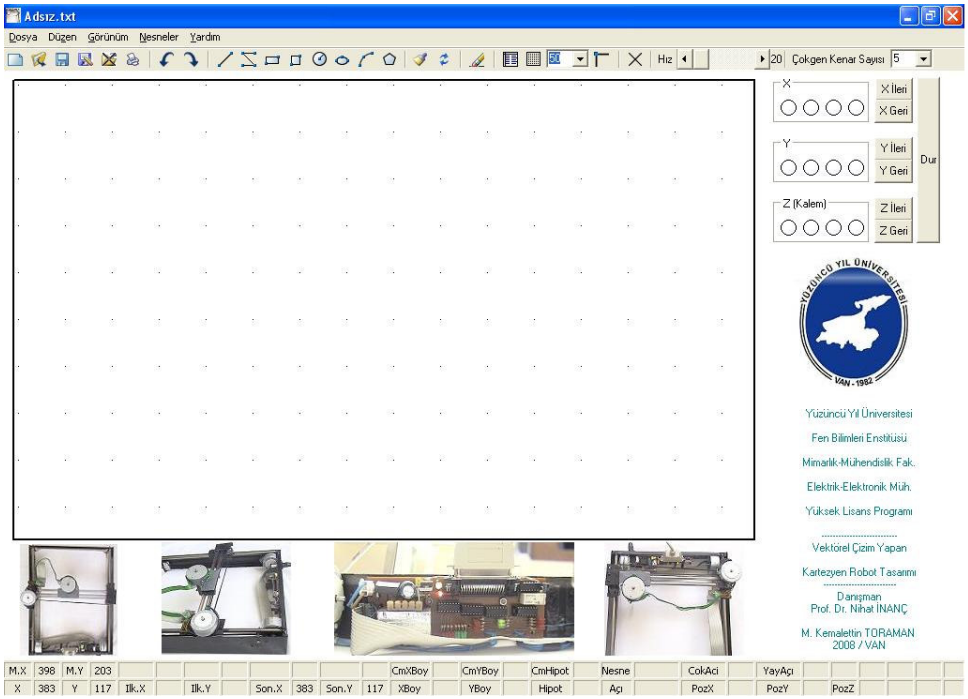
3.2.1. Vektörel çizim yapan program

Robot yapılıp hazır hale getirildikten sonra, bilgisayar ile kontrolü için, arayüz programı yapılmıştır (Şekil 3.29). Programın yapımı için Delphi 7.0 programlama dili kullanılmıştır (Şekil 3.28). Tasarlanan donanımın tüm fonksiyonları bilgisayar tarafından kontrol edilmektedir.

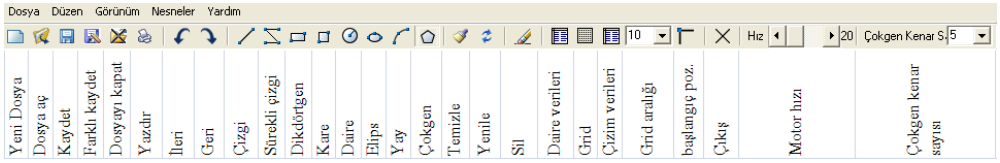
Programda tüm kontroller menüler ile yapılır (Şekil 3.30) ve durumlar status çubuğu ile gözlemlenebilir (Şekil 3.31). Programda, motorlara giden sinyalleri gözlemleyebilmek için göstergeler kullanılmıştır ve motorları manüel kullanabilmek için butonlar eklenmiştir (Şekil 3.32). Ayrıca, program hakkında kısaca bilgi vermek için "Program hakkında" ikinci formu eklenmiştir (Şekil 3.33).



Şekil 3.28. Programın yapım aşaması.



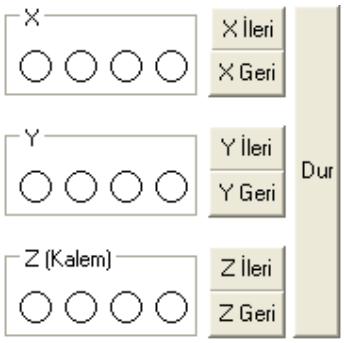
Şekil 3.29. Vektörel çizim yapan program.



Şekil 3.30. Program menüleri.

M.X	314	M.Y	278									CmXBoy	24	CmYBoy	14	CmHipot	27	Nesne	çokgen	CokAci	72
X	299	Y	192	Ilk.X	220	Ilk.Y	144	Son.X	299	Son.Y	192	XBoy	79	YBoy	48	Hipot	92	Açı	31	PozX	

Şekil 3.31. Status menüsü.



Şekil 3.32. Göstergeler ve manüel hareket butonları.



Şekil 3.33. Program hakkında formu.

Vektörel çizim yapan programın bazı özellikleri:

1- Program, çizimleri vektörel olarak yapar.
2- Program, çizilen grafikleri, hem vektörel (txt formunda) hem de bitmap (BMP) olarak kaydedebilir.
3- Programda grafik olarak; çizgi, sürekli çizgi, dikdörtgen, kare, daire, elips, yay ve çokgen çizilebilir.
4- Yapılan çizimler, ileri geri alınabilir.
5- Izgara (grid) gizlenebilir ve ölçüsü değiştirilebilir.
6- Verileri tutan tablo, aktif veya pasif edilebilir.
7- Robot, manüel olarak sıfır pozisyona alınabilir.
8- Robot çizim yaparken, hız ayarı yapılabilir.
9- Çokken için kenar sayısı seçilebilir.
10- Robot, manüel olarak (X-Y-Z) yönlerinde hareket ettirilebilir.
11- Çizim yapılırken, tüm veriler status barından okunabilir.
12- Robot çizimi yaparken, hangi motora hangi bilginin gittiği izlenebilir.

Windows XP işletim sistemi, güvenlik nedenlerinden dolayı, Windows 98 işletim sistemi gibi portlara direkt erişmeye izin vermemektedir. Bu nedenle, portlara erişmek için “io.dll” kütüphanesi kullanılmıştır. Programa eklenen fonksiyonlarla, robotun manüel olarak çalıştırılması programa esneklik kazandırmıştır.

Grafikler, bitmap yerine vektörel olarak tasarlanırlarsa daha yüksek kalite elde edilir ve sabit disk üzerinde daha az yer kaplarlar. Vektör grafikler; çözünürlükten bağımsız, her bir nesne matematiksel ifadelerle oluşturulan grafik türleridir. Vektörel grafikler gerektiğinde bitmap haline kolaylıkla dönüştürülebilirler.

Program ile çizim yapıldığında, her nesnenin ilk ve son koordinatları, gerekli ise açı ve kenar bilgileri bir stringgrid tabloda toplanır. Yazdır ikonu seçildiğinde, stringgridin her satırını teker teker taranarak gerekli bilgiler robota gönderilir.

X,Y motorunun tüm uçlarındaki gerilimi kesmek için 888 nolu yazmaca "0" (çizelge 3.3), Z motorunun tüm uçlarındaki gerilimi kesmek için ise 890 nolu yazmaca "11" bilgisi gönderilir (çizelge 3.4).

Adım motorları, çizelge 3.3 ve çizelge 3.4 deki verilerin paralel porta gönderilmesi ile hareket ettirilirler. Veriler programda bir tablo halinde saklanarak sırasıyla motorlara gönderilir. Motorların sağa veya sola dönmesi, tablodaki verilerin motorlara düz veya ters sıra ile gönderilmesiyle sağlanır.

```
PortX:array[0..7] of byte=(16,48,32,96,64,192,128,144);  // X Motor
PortY:array[0..7] of byte=(1,  3, 2, 6, 4, 12,  8,  9);          // Y Motor
PortZ:array[0..3] of byte=(10,9,15,3);                           // Z Motor
```

Motorların başlangıç konumlarında olup olmadıklarının kontrolü için kullanılan sınır anahtarlarından gelen bilgiler, paralel portun 889 adresindeki status yazmacı ile değerlendirilir. Bu yazmaçtaki bilgi;

a) 127 olursa; hiçbir motor başlangıç konumunda değildir.
b) 111 olursa; X motoru başlangıç konumundadır.
c) 95 olursa; Y motoru başlangıç konumundadır.
d) 63 olursa; Z motoru başlangıç konumundadır.
e) 79 olursa; XY motoru başlangıç konumundadır.
f) 47 olursa; XZ motoru başlangıç konumundadır.
g) 31 olursa; YZ motoru başlangıç konumundadır.
h) 15 olursa; XYZ motoru başlangıç konumundadır (Çizelge 3.5).

Yukarıdaki veriler programda bir tablo halinde saklanarak, SA'larının durumları kontrol edilir. Motorlar, başlangıç konumlarına geldiklerinde, SA'nın kontakları kapanır. PortOku tablolarındaki veriler ile status yazmacının içeriği karşılaştırılarak hangi motorların başlangıç konumunda oldukları tespit edilir. Motorlar başlangıç konumlarına geldiklerinde, motorların konumlarını kontrol eden değişkenler (PozX, PozY, PozZ) "1" değerini alır.

Çizelge 3.3. 888 nolu porta bilgi gönderme tablosu

	X Motoru				Y Motoru					
Sıra	3	2	1	0	3	2	1	0	Y	X
Adım	128	64	32	16	8	4	2	1		
	0	0	0	0	0	0	0	0	Boş	Boş
1	0	0	0	0	0	0	0	1	1	
2	0	0	0	0	0	0	1	1	3	
3	0	0	0	0	0	0	1	0	2	
4	0	0	0	0	0	1	1	0	6	
5	0	0	0	0	0	1	0	0	4	
6	0	0	0	0	1	1	0	0	12	
7	0	0	0	0	1	0	0	0	8	
8	0	0	0	0	1	0	0	1	9	
1	0	0	0	1	0	0	0	0		16
2	0	0	1	1	0	0	0	0		48
3	0	0	1	0	0	0	0	0		32
4	0	1	1	0	0	0	0	0		96
5	0	1	0	0	0	0	0	0		64
6	1	1	0	0	0	0	0	0		192
7	1	0	0	0	0	0	0	0		128
8	1	0	0	1	0	0	0	0		144

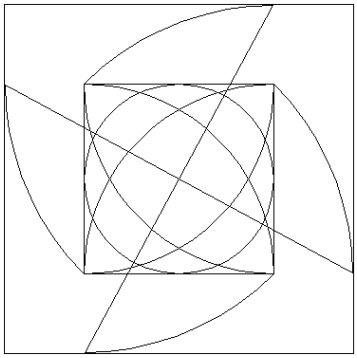
Çizelge 3.4. 890 nolu porta bilgi gönderme tablosu

	Z Motoru				
Sıra	3	2	1	0	Z Motoru
Adım	8	4	2	1	
	1	0	1	1	11- Boş
1	1	0	1	0	10
2	1	0	0	1	9
3	1	1	1	1	15
4	0	0	1	1	3

Çizelge 3.5. 889 nolu porttan bilgi okuma tablosu

Sıra	**7**	6	**5**	**4**	3	2	1	0	
Motor	128	64	32	16	8	4	2	1	
Boş	0	1	1	1	1	1	1	1	127
X	0	1	1	0	1	1	1	1	111
Y	0	1	0	1	1	1	1	1	95
Z	0	0	1	1	1	1	1	1	63
XY	0	1	0	0	1	1	1	1	79
XZ	0	0	1	0	1	1	1	1	47
YZ	0	0	0	1	1	1	1	1	31
XYZ	0	0	0	0	1	1	1	1	15

14	No	Nesne	IlkX	IlkY	SonX	SonY	Hipot	Hipot1	Aci1	Aci2	CokAci	Kenar	SayPline	AcıBolge
17	1	daire	350	200	450	200	100							
	2	kare	250	100	450	300								
	3	yay	250	100	450	100		200	90	0				
	4	yay	450	100	450	300		200	180	90				
	5	yay	450	300	250	300		200	270	180				
	6	yay	250	300	251	100		200	360	270				
	7	yay	250	100	450	300		283	90	45				
	8	yay	450	100	250	300		283	180	135				
	9	yay	450	300	250	100		283	270	225				
	10	yay	250	300	450	100		283	360	315				
	11	çizgi	533	299	167	101			208					0
	12	çizgi	449	17	251	383			118					0
	13	çizgi	166	384	166	16			270					0
	14	çizgi	166	16	534	16			0					0
	15	çizgi	534	16	534	384			90					0
	16	çizgi	534	384	166	384			180					0



Şekil 3.34. Program ile çizilen örnek bir çizim.

Değişkenler; motorlar, ileri adım attıklarında (+1), geri adım attıklarında ise (-1) değerini alırlar. Böylece kalemin nerede olduğu her zaman tespit edilebilir.

```
PortOkuX:TPortOku=[111,79,49,15];      // X – XY – XZ – XYZ
PortOkuY:TPortOku=[95,79,31,15];       // Y – YX – YZ – XYZ
PortOkuZ:TPortOku=[63,47,31,15];       // Z – ZX – ZY – XYZ
```

Program ile grafikler çizildikten sonra, yazdır ikonuna () seçildiğinde, öncelikle, sırası ile Z, X ve Y motorları başlangıç konumlarına gelirler. Motorların gideceği noktalar, bu sıfır konumları referans alınarak belirlenmektedir. Motorlar sıfır konumlarından ayrılır ayrılmaz, adım sayıları bilgisayar tarafından sayılır.

Program ile çizime ilk başlandığında, tıklanan noktanın koordinatı okunarak *IlkNokta(X,Y)* değişkenine aktarılır. Mouse hareket ettikçe, imlecin bulunduğu son nokta koordinatı ise *Son:=Point(X,Y)* komutuyla *Son(X,Y)* değişkeninde aktarılır. Daha sonra aşağıdaki komutlarla çizginin; Xboy, YBoy, Açı ve Boyu (Hipot) hesaplanarak bulunur.

```
XBoy:=Son.X-Ilk.x;
YBoy:=Son.Y-Ilk.y;
Hipot:=Hypot(XBoy,YBoy);
if Hipot<0.1 then Aci:=0.0
else Aci:=RadToDeg(ArcCos(XBoy / Hipot));
if YBoy<0 then Aci:=360-Aci;
```

3.2.2. Robotun grafikleri çizme yöntemi ve interpolasyon

Robotun doğru, yay veya daire çizebilmesi için, iki motorunda aynı anda farklı hız ve konum değerlerinde senkronize çalışması gereklidir (Taştan, 1997).

Robot çizim yaparken üç temel hareket yapmaktadır. Bunlar; X ve Y ekseninde ileri-geri, Z ekseninde ise yukarı-aşağı hareketlerdir. Bunlardan; X ve Y eksenindeki hareket, kalemi istenilen koordinatlara taşıyan harekettir, Z eksenindeki hareket ise kalemi, istenilen koordinatlara geldikten sonra aşağı-yukarı hareket ederek kalem ucunun kağıda temas etmesini sağlayan harekettir (Arslan, 1998; Ünlü, 1997).

Adım motorunun bir turdaki adım sayısı: $N = \frac{360}{\alpha}$

Adım motorunun bir adımdaki aldığı yol $l_n = \frac{2\pi r}{N}$

Adım motorunun bir turdaki aldığı yol $l = 2\ \pi\ r$

Formülleri ile hesaplanır. Bu formüller kullanılan motorlara uygulanırsa:

R=5.4mm r=2,7mm $\alpha=7{,}5^{o}$

$$N = \frac{360}{7{,}5^{0}} = 48 \text{ adım}$$

$$l_n = \frac{2.3{,}14.2{,}7}{48} = 0{,}35mm$$

$$l = 2.3{,}14.2{,}7 = 17\,mm = 1{,}7\,cm$$

Yapılan çalışmada kullanılan adım motorlarının bir adımda aldığı yol l_n 0,35mm dir. Adım motorunun mili, bir tur (360^{o}) attığında kalem 17mm (1.7cm) hareket eder. Ekranda çizilen grafiklerin piksel bazındaki uzunlukları 0,35 ile çarpılarak mm'ye dönüştürülür. Böylece ekranda label etiketinde görülen uzunluk, robotun çizdiği çizim ile birebir ölçeklendirilir.

3.2.2.1 Lineer çizgi çizme

Bilgisayar, çizgiyi aşağıdaki komutlarla ekrana çizer. Çizim, çizginin "*LineTo*" komutu ile ilk noktadan son noktaya çekilmesiyle gerçekleşir. Çizim yapıldığında, önceki çizginin bıraktığı iz "*Image1.Canvas.Pen.Mode:=pmNot;"* komutuyla silinir. Her çizilen çizim, birer nesne olarak bir stringgride eklenir.

```
Image1.Canvas.Pen.Mode:=pmNot;
Image1.Canvas.MoveTo(Ilk.X, Ilk.Y);
Image1.Canvas.LineTo(Gecici.X, Gecici.Y);
Gecici:=Point(Ilk.X,Ilk.Y);
```

Çizim ekranda çizildikten sonra yazdır ikonu ile robota yazdırılır. Robot çiz komutunu aldığında; öncelikle motorları, konumlarını kontrol edebilmek için başlangıç konumlarına getirir. Nesne ve veriler stringgridden sırası ile teker teker okunur. Motorlar, kalemi ilk nesnenin ilk koordinatına götürerek çizime başlar. Motorlar, çizginin uzunluğu (ilk koordinattan son koordinat çıkarılarak) hesaplandıktan sonra o kadar adım hareket ederek, ilk çizimi tamamlar. İkinci ve diğer çizimler için motorlar, aşağıdaki komutlarla "*PozisyonAl"* proceduru ile sonraki nesnenin başlangıç koordinatına götürülür ve diğer çizimler yapılır.

```
for i:=1 to Adim do begin
  if MotorD then begin
    iX:=IfThen(iX=7,0,iX+1);
    PortCikis(888,PortX[iX]);
    if Dur then begin PortOut(888,0);exit;end;
    SinirKontrol('XI');
    Inc(PozX);
    StatusBar1.Panels.Items[21].Text:=IntToStr(PozX);
  end;  end;
 MotorD:=True;
 PortOut(888,0);
```

3.2.2.2 Eğri çizgi çizme ve interpolasyon

Programda en zor olan kısım eğri çizme kısmıdır. Düz çizgi haricindeki tüm çizimler, eğri çizimi ile gerçekleştirilir. Eğri çizme işleminde, öncelikle ilk koordinat ile son koordinat farkları (XFark, YFark) hesaplanır, ardından bu farklardan büyük olanı ve küçük olanı bulunur. Sonra XFark ve YFark değerleri küçük olan sayıya bölünerek X ve Y yönündeki adım oranları bulunur. Eğer bu oranlar birbirlerinin katları şeklinde ise, başka bir işleme gerek kalmadan çizim yapılır.

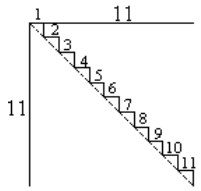
```
XFark:=YSon.X - YIlk.X;
YFark:=YSon.Y - YIlk.Y;
AXFark:=Abs(XFark);
AYFark:=Abs(YFark);
Buyuk:=Max(AXFark,AYFark);
Kucuk:=Min(AXFark,AYFark);
AdimX :=Sign(XFark)*(Buyuk div AYFark);
AdimY :=Sign(YFark)*(Buyuk div AXFark);
```

Komutları ile değerler bulunur.

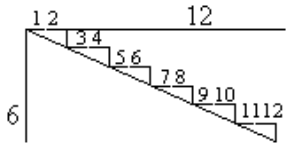
```
for i:=1 to Kucuk do begin
  AdimAyarla(AdimX+Sign(AdimX)*F,AdimY,FAdimX,FAdimY);
end;
Procedure TForm1.AdimAyarla(AdimX,AdimY,FAdimX,FAdimY:Integer);
begin
  YonMotorKontrol('X',AdimX+Sign(AdimX)*FAdimX);
  YonMotorKontrol('Y',AdimY+Sign(AdimY)*FAdimY);
end;
procedure
TForm1.YonMotorKontrol(Motor:ShortString;Adim:Smallint);
begin
```

```
        if Motor='X' then begin
          if Adim >0 then XIleri(Adim);
          if Adim <0 then Xgeri(Adim);
        end;
        if Motor='Y' then begin
          if Adim >0 then YIleri(Adim);
          if Adim <0 then Ygeri(Adim);
        end;
        if Motor='Z' then begin
          if Adim >0 then ZIleri(Adim);
          if Adim <0 then Zgeri(Adim);
        end;
        end;
Procedure TForm1.XIleri(Adim:Smallint);
var i:Smallint;
begin
  for i:=1 to Adim do begin
     PortOut(890,11);
     iX:=IfThen(iX=7,0,iX+1);
     PortCikis(888,PortX[iX]);
     if Dur then begin PortOut(888,0);exit;end;
     SinirKontrol('XI');
     Inc(PozX);
     StatusBar1.Panels.Items[21].Text:=IntToStr(PozX);
   end; end; PortOut(888,0);end;
```

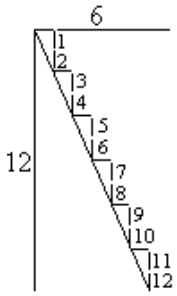
Burada, "*AdimAyarla"* Proceduru ile motorların, hangi yöne, kaç adım atacakları belirlenir. Motorlar, "*kucuk*" değişkeni kadar *for* döngüsüne alınarak, önce X yönünde *AdimX* kadar, sonra Y yönünde *AdimY* kadar adım atarak çizimi tamamlarlar.



Şekil 3.35. Adım sayısının X ve Y yönünde eşit olduğu eğri çizimi.



Şekil 3.36. Adım sayısının X yönünde fazla olduğu eğri çizimi.



Şekil 3.37. Adım sayısının Y yönünde fazla olduğu eğri çizimi.

Örneğin:

a) *XFark*=11, *YFark*=11 olan çizim; 11 defa, önce X yönüne 1 adım, sonra Y yönüne 1 adım atarak yapılır (Şekil 3.35).

b) *XFark*=12, *YFark*=6 olan çizim; 6 defa, önce X yönüne 2 adım, sonra Y yönüne 1 adım atarak yapılır (Şekil 3.36).

c) *XFark*=6, *YFark*=12 olan çizim, 6 defa, önce X yönüne 1 adım, sonra Y yönüne 2 adım atarak yapılır (Şekil 3.37).

3.2.2.2.1 Eğri çiziminde interpolasyon

Eğri çiziminde önemli olan diğer bir husus, *XFark* ve *YFark* değişkenlerinin birbirlerinin katları olmamasıdır. Bu durumda; oranlarla beraber fazla kalan adımlar (*AdimXK*, *AdimYK*) bulunur. Bulunan adımlar (*AdimXK*, *AdimYK*), çizime dengeli bir şekilde dağıtılarak, motorların aralarda 1'er adım fazla atarak çizimi yapması sağlanır.

Örneğin: Ilk=(100,132) Son=(125,143) ise;

XFark:=125 – 100 = 25

YFark:= 143 – 132 = 11

Buyuk:= 25

Kucuk:= 11

AdimX :=25 div 11=2

AdimXK:=25 mod 11 = 3

AdimY :=25 div 25 = 1

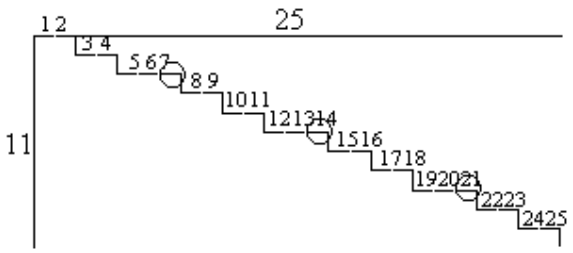
AdimYK:= 25 mod 25 = 0

Orta:= Round(11 / 2) = 6

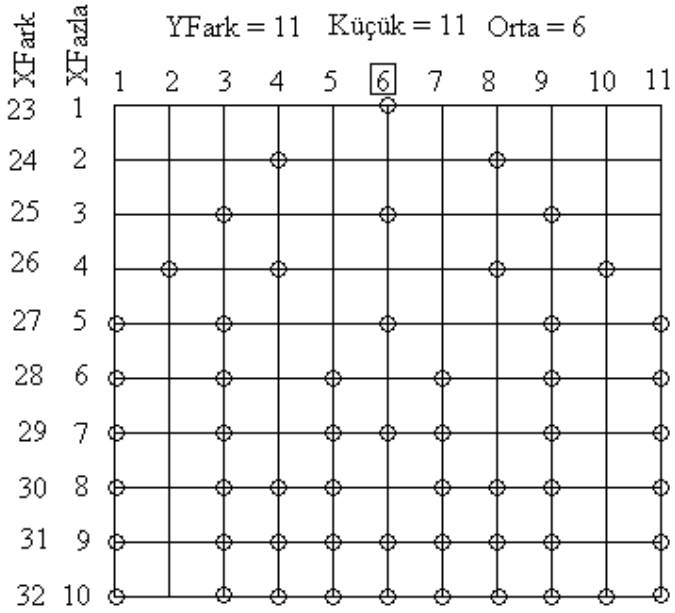
Hesaplamalarda; *AdimX*=2, *AdimY*=1, *Kucuk*=11 dir. Bu durumda; 11 defa, X motoru 2, Y motoru 1 adım atarak çizim yapılacaktır. Fakat bu şekilde yapılan bir çizimde; hem kalem tam son koordinatta olmayacaktır, hem de çizginin uzunluğu tam çıkmayacaktır. Çünkü X motoru 11*2=22 adım atacak, Y motoru ise 11*1=11 adım atacaktır. Çizginin fark koordinatı, (22,11) ve son koordinatı (122,143) olacaktır. Hâlbuki çizginin gerçek fark koordinatı (25,11) ve son koordinatı ise (125,143) dür. X motoru 3 adım eksik atmıştır.

Bunun çözümü için, fazla kalan 3 adımın dengeli bir şekilde aralara dağıtılması gerekmektedir. Böylece, hem çizgi düz ve eksiksiz olur, hem de kalem tam olarak son koordinata gelir. Bu işlemi yapmak için *FazlaKont* Proceduru kullanılır.

```
Procedure   FazlaKont(Buyuk,Kucuk,Adim,AdimK,Orta:Integer;var
TFazla:TFAdim;var F:Integer);
var
  i,c,YKalan:Integer;
  ilk :Real;
  x,y :integer;
begin
  SetLength(TFazla,AdimK);
  ilk:=Kucuk/(AdimK+1);
  if AdimK>Orta then begin
    F:=1;
  if Adim=1 then form1.Caption:='';
    x:=Abs(Adim+Sign(Adim)*1);
    y:=x*kucuk;
    Ykalan:=y-Buyuk;
    SetLength(TFazla,YKalan);
    ilk:=Round(Kucuk/(Ykalan+1));
    for i:=1 to YKalan do begin
      c:=Round(ilk*i);
      if c>kucuk then c:=kucuk;
      TFazla[i-1]:=c;
    end;
  end else begin
    for i:=1 to AdimK do begin
      F:=0;
      c:=Round(ilk*i);
      if c>kucuk then c:=kucuk;
      TFazla[i-1]:=c;
    end;  end; end;
```



Şekil 3.38. (25-11) olan eğri çizimi.



Şekil 3.39. Örnek fazla kalan adımların dağılış tablosu.

Şekil 3.38'de görüldüğü gibi, daire içinde olan adımlarda motor bir adım fazla atarak çizim yapılmıştır.

Şekil 3.39'da, XFark=11, YFark=23-32 olan bir çizginin çiziminde, fazla kalan adımların dengeli bir şekilde nasıl dağıtıldığı görülmektedir.

Dağıtım yapabilmek için, önce küçük sayının ortası (*Orta*) değişkeni bulunur. Sonra, küçük olan sayı, fazla kalan adım sayısından bir fazlasına bölünerek hangi aralıkta dağılacağı bulunur. Bulunan aralıklarda motor, 1 fazla adım atarak çizimi yapar.

Eğer kalan adım sayısı küçük sayının iki katını geçiyorsa; adımların tümü 1 arttırılır ve eksik kalan adım bulunur. Ardından, yukarıda yapılan işlemler tekrarlanarak, motor 1 eksik adım atarak çizimi yapar.

3.2.2.3 Dikdörtgen ve kare çizme

Ekranda aşağıdaki komutlar ile çizim yapılır. Eğer kare çizilecekse; X yönündeki uzunluk kadar Y yönünde çizgi çizilir.

```
Image1.Canvas.MoveTo(Ilk.X, Ilk.Y);
Image1.Canvas.LineTo(Gecici.X,Ilk.Y);
Image1.Canvas.LineTo(Gecici.X,Gecici.Y);
Image1.Canvas.LineTo(Ilk.X,Gecici.Y);
Image1.Canvas.LineTo(Ilk.X, Ilk.Y);
if nesne ='kare' then
  if Son.Y-Ilk.Y>0 then
     Son.Y:=Ilk.Y+Abs(Son.X-Ilk.X)
  else Son.Y:=Ilk.Y-Abs(Son.X-Ilk.X);
```

Dikdörtgen veya kare çiziminde motor, *AdimAyarla(XFark,YFark,0,0);* komutu ile X ve Y yönündeki çizgileri çizer. Sonra, motor *AdimAyarla(-XFark,-YFark,0,0);* komutu ile ters yönde X ve Y çizgilerini çizerek çizimi tamamlar.

```
PozisyonAl(YIlk.X,YIlk.Y);
AdimAyarla(XFark,YFark,0,0);
AdimAyarla(-XFark,-YFark,0,0);
```

3.2.2.4 Daire çizme ve interpolasyon

Ekranda, aşağıdaki komutlar ile dairenin yarıçapı bulunur.

```
XBoy:=Son.X-Ilk.x;
YBoy:=Son.Y-Ilk.y;
Hipot1:=Hypot(XBoy,YBoy);
```

Sonra, aşağıdaki komutlar ile dairenin 36 adet nokta koordinatları bulunur.

```
for i:=0 to 36 do begin
  HesaplaCiz(i*10,Hipot1,Hipot1);
end;
```

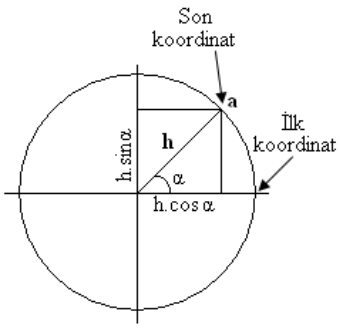
Sonra, aşağıdaki komutlar ile 36 adet nokta arasına çizgiler çizilerek daire çizimi tamamlanır.

```
DNokta.X:=Round (Cos(DegToRad(HAci))*HCap+Ilk.X);
DNokta.Y:=Round (Sin(DegToRad(HAci))*HCap2+Ilk.Y);
if not YazD then Image1.Canvas.LineTo(DNokta.X,DNokta.Y);
```

Görüldüğü gibi, daire çizimi aslında bir çokgen çizimidir. Fakat çizimdeki kenar sayısı fazla olduğundan çizim, çokgen gibi değil de daire gibi gözükür. Nitekim birçok grafik programı (AutoCAD gibi) daire çizimlerini bu şekilde yapmaktadır.

Kağıt üzerine daire çizimi için, önce kalem ilk noktaya gelir. Sonra, motorlar 36 defa ilk noktadan sonraki noktaya çizgi çizerek daire çizimini tamamlarlar.

```
  PozisyonAl((YIlk.X+Round(YHipot)),YIlk.Y);
  for i:=1 to 36 do begin
    HesaplaYaz(i*10,YHipot,YHipot);    end;
Procedure TForm1.HesaplaYaz(HYAci,HYHipot1,HYHipot2:Real);
var
HXFark,HYFark:Integer;
begin
  HesaplaCiz(HYAci,HYHipot1,HYHipot2);
  HXFark:=DNokta.X-Gecici2.X;
  HYFark:=DNokta.Y-Gecici2.Y;
  CizgiYaz(HXFark,HYFark);
  Gecici2:=Point(DNokta.X,DNokta.Y); end;
Procedure TForm1.HesaplaCiz(HAci,HCap,HCap2:Real);
```



Şekil 3.40. Koordinat bulma.

a noktasının koordinatı: a.X=*ilkX*+(h.cos α), a.Y=*ilkY*+(h.sin α) dır (Şekil 3.40).

```
Begin
 DNokta.X:=Round (Cos(DegToRad(HAci))*HCap+Ilk.X);
 DNokta.Y:=Round (Sin(DegToRad(HAci))*HCap2+Ilk.Y);
 if not YazD then Image1.Canvas.LineTo(DNokta.X,DNokta.Y); end;
```

3.2.2.5 Elips çizme ve interpolasyon

Elips çizimi, daire çizimi ile aynıdır. Sadece yarıçap çarpanları farklıdır. Dairede, X yönündeki uzunluk ile Y yönündeki uzunluk aynı olduğundan, aynı yarıçap ile çarpılmıştır. Elipste ise; X yönündeki uzunluk ile Y yönündeki uzunluk farklı olduğundan, çarpanlar farklı olacaktır. Ekrana çizim yapılırken;

```
for i:=0 to 36 do begin
 HesaplaCiz(i*10, Gecici.X-Ilk.X, Gecici.Y-Ilk.Y);
end;
```

Komutları kullanılır. Kağıda çizim yapılırken;

```
PozisyonAl(Son.X,Ilk.Y);
Gecici2:=Point(Son.X,Ilk.Y);
for i:=1 to 36 do begin
 HesaplaYaz(i*10,Son.X-Ilk.X,Son.Y-Ilk.Y);    end;
```

Komutları kullanılır.

3.2.2.6 Yay çizme ve interpolasyon

Aşağıdaki komutlar ile yay ekran çizilir. Yay çiziminde, ilk noktadan sonra her 5'er derecede yeni koordinatlar bulunur. İlk koordinat ile sonraki koordinatlar arasına çizgi çizilerek yay çizimi tamamlanır. Yay çiziminde, çizimin daha hassas olması için aralıklar 5'er derece alınmıştır.

```
if Aci1>Aci then Aci:=360+Aci;
if Aci1<Aci then begin
 Tur:=Round(GeciciAci-Aci1) div 5;
 DegisenAci:=Aci1;
 Image1.Canvas.MoveTo(Son.X,Son.Y);
 for i:=0 to Tur do begin
  HesaplaCiz(DegisenAci,Hipot1,Hipot1);
  if GeciciAci-DegisenAci<10 then DegisenAci:=GeciciAci
  else DegisenAci:=DegisenAci+5;
 end;
```

Kağıt üzerine yay çizimi için, önce kalem ilk noktaya gelir. Sonra, motorlar her 5'er derece sonraki koordinatlara çizgi çekerek yay çizimi tamamlanır.

```
PozisyonAl(Son.X,Son.Y);
Gecici2:=Point(Son.X,Son.Y);
Tur:=Round(YAci-YAci1) div 5;
YDegisenAci:=YAci1;
for i:=0 to Tur do begin
 HesaplaYaz(YDegisenAci,YHipot1,YHipot1);
 if YAci-YDegisenAci<10 then YDegisenAci:=YAci
 else YDegisenAci:=YDegisenAci+5;
end;
```

3.2.2.7 Çokgen çizme ve interpolasyon

Aşağıdaki komutlar ile ekrana çokgen çizilir. Kenar sayısı bir ComboBox ile seçilir. Kenar sayısına bağlı olarak *CokAci* hesaplanır. Her *CokAci* değeri kadar sonraki koordinatlar bulunur. İlk koordinat ile sonraki koordinatlar arasına çizgi çizilerek çizim tamamlanır.

```
Kenar:=StrToInt(ComboBox2.Text);
CokAci:=RoundTo(360/Kenar,-2);
Kenar:=Kenar+1;
Hesap;  Etiket;
Image1.Canvas.MoveTo(Gecici.X,Gecici.Y);
for i:=1 to Kenar do begin
  HesaplaCiz(Aci1,Hipot1,Hipot1);
  Aci1:=Aci1+CokAci;
end;
Cokgen2;
Gecici:=Point(Son.X,Son.Y);
Aci1:=Aci;
Hipot1:=Hipot;
```

Kağıt üzerine çokgen çizimi için, önce kalem ilk noktaya gelir. Sonra, motorlar *CokAci* değişkeni kadar sonraki koordinatlara çizgi çekerek çizim tamamlanır.

4. BULGULAR

Çalışmada, kullanım kolaylığı, sürücü devresinin basit ve boyutunun küçük olmasından dolayı adım motorları kullanılmıştır.

Motorların 1 adımı 7.5^{o} dir. Motor, 360^{o} için (360/7.5=48) 48 adım atmaktadır. Kalem, motorun 48 adımında (1 turda) 17 mm, 1 adımında ise (17/48=0.35) 0.35 mm çizim yapmaktadır.

Programda kullanılan image1'in boyutları 780x480 pikseldir. Çizim için A4 kağıdı kullanılmıştır. Kağıdın çizim alanı; 780.0,35 = 273 mm, 480.0,35 = 168 mm (273x168 mm) dir.

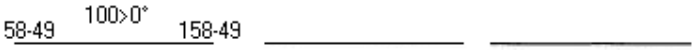
Düz çizgi çizmek için; ilk koordinat ile son koordinat farkı bulunmuştur (*XFark, YFark*). Bulunan fark kadar, motor adım atarak robot çizimi tamamlamıştır (Şekil 4.1).

Eğri çizgi çizmek için; ilk koordinat ile son koordinat farkı bulunmuştur (*XFark, YFark*). Bu farklar birbirleri ile oranlanarak, motorların X ve Y ekseninde atacakları adımlar bulunmuştur. Bu oranlar tam sayı ise; robot, küçük olan sayı kadar önce X ekseninde, sonra Y ekseninde, bulunan fark kadar adım atarak çizimi yapmıştır. Eğer oranlar tam sayı değilse; farklara ek olarak fazla kalan adımlar bulunmuştur. Robot, dengeli olarak fazla kalan adımlar kadar, aralarda bir fazla veya bir eksik adım atarak çizimi tamamlamıştır (Şekil 4.2).

Dikdörtgen çizmek için; robot, önce ilk koordinattan başlayarak X eksenindeki, ardından Y eksenindeki çizgileri çizmiştir. Sonra X ekseninin tersine, ardından Y ekseninin tersine çizgiler çizerek çizimi yapmıştır (Şekil 4.3).

Kare çizmek için; robot, önce ilk koordinattan başlayarak X eksenindeki çizgiyi çizmiştir. Sonra X eksenindeki uzunluk kadar Y eksenindeki çizgiyi çizmiştir. Sonra iki çizgiyi ters yönde çizerek çizimi tamamlamıştır (Şekil 4.4).

Daire çizmek için; önce dairenin yarıçapı, sonra yarıçap ile dairenin merkez koordinatı toplanarak çizimin ilk noktası bulunmuştur. İlk noktadan sonra 10 ar derecelik açıların koordinatları bulunmuştur (Şekil 4.5). Robot, bu koordinatlar arasına çizgi çizerek çizimi tamamlamıştır (Şekil4.6).



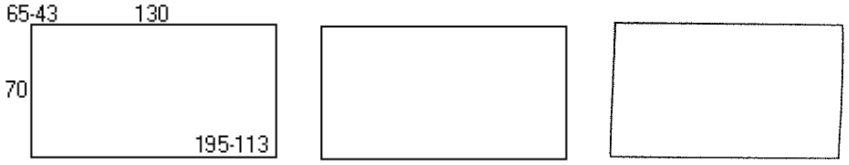
a b c

Şekil 4.1. Düz çizgi çizme, a.Çizim aşaması, b. Ekrandaki çizim, c. Robot çizimi.



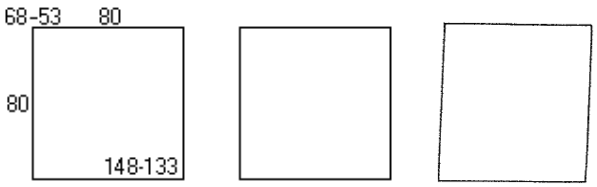
a b c

Şekil 4.2. Eğri çizgi çizme, a.Çizim aşaması, b. Ekrandaki çizim, c. Robot çizimi.



a b c

Şekil 4.3. Dikdörtgen çizme, a.Çizim aşaması, b. Ekrandaki çizim, c. Robot çizimi.



a b c

Şekil 4.4. Kare çizme, a.Çizim aşaması, b. Ekrandaki çizim, c. Robot çizimi.

Açı	İlk.X	İlk.Y	Son.X	Son.Y		Xfark	Yfark		AdimX	AdimXK	AdimY	AdimYK	Orta
	189	175	242	149									
10	248	175	247	185		-1	10		-1	0	10	0	0
20	247	185	244	195		-3	10		-1	0	3	1	2
30	244	195	240	204		-4	9		-1	0	2	1	2
40	240	204	234	213		-6	9		-1	0	1	3	3
50	234	213	227	220		-7	7		-1	0	1	0	4
60	227	220	218	226		-9	6		-1	3	1	0	3
70	218	226	209	230		-9	4		-2	1	1	0	2
80	209	230	199	233		-10	3		-3	1	1	0	2
90	199	233	189	234		-10	1		-10	0	1	0	0
100	189	234	179	233		-10	-1		-10	0	-1	0	0
110	179	233	169	230		-10	-3		-3	1	-1	0	2
120	169	230	160	226		-9	-4		-2	1	-1	0	2
130	160	226	151	220		-9	-6		-1	3	-1	0	3
140	151	220	144	213		-7	-7		-1	0	-1	0	4
150	144	213	138	204		-6	-9		-1	0	-1	3	3
160	138	204	134	195		-4	-9		-1	0	-2	1	2
170	134	195	131	185		-3	-10		-1	0	-3	1	2
180	131	185	130	175		-1	-10		-1	0	-10	0	0
190	130	175	131	165		1	-10		1	0	-10	0	0
200	131	165	134	155		3	-10		1	0	-3	1	2
210	134	155	138	146		4	-9		1	0	-2	1	2
220	138	146	144	137		6	-9		1	0	-1	3	3
230	144	137	151	130		7	-7		1	0	-1	0	4
240	151	130	160	124		9	-6		1	3	-1	0	3
250	160	124	169	120		9	-4		2	1	-1	0	2
260	169	120	179	117		10	-3		3	1	-1	0	2
270	179	117	189	116		10	-1		10	0	-1	0	0
280	189	116	199	117		10	1		10	0	1	0	0
290	199	117	209	120		10	3		3	1	1	0	2
300	209	120	218	124		9	4		2	1	1	0	2
310	218	124	227	130		9	6		1	3	1	0	3
320	227	130	234	137		7	7		1	0	1	0	4
330	234	137	240	146		6	9		1	0	1	3	3
340	240	146	244	155		4	9		1	0	2	1	2
350	244	155	247	165		3	10		1	0	3	1	2
360	247	165	248	175		1	10		1	0	10	0	0

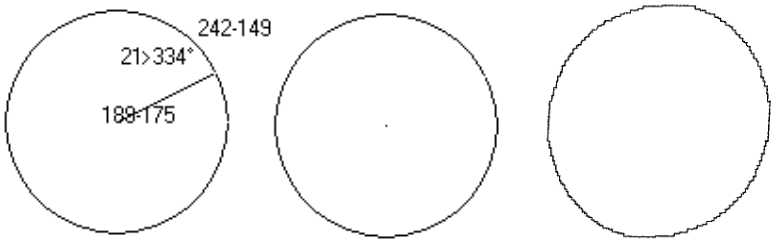
Şekil 4.5.Bir daire çizimindeki veriler.

Elips çizmek için; önce elipsin X eksenindeki yarıçapı, sonra yarıçap ile elipsin merkez koordinatı toplanarak çizimin ilk noktası bulunmuştur. İlk

noktadan sonra 10 ar derecelik açıların koordinatları bulunmuştur. Robot, bu koordinatlar arasına çizgi çizerek çizimi tamamlamıştır (Şekil4.7).

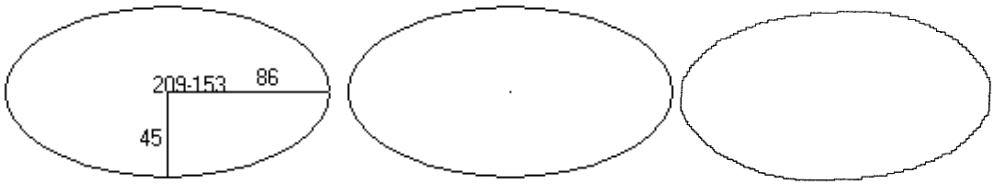
Yay çizmek için; yayın başlangıç noktasından sonra 5 er derecelik açıların koordinatları bulunmuştur. Robot, bu koordinatlar arasına çizgi çizerek çizimi tamamlamıştır (Şekil 4.8).

Çokgen çizmek için; ComboBox ile çokgenin kenar sayısı seçilmiştir. Çokgenin açısı *"CokAci:=RoundTo(360/Kenar,-2);"* komutuyla bulunmuştur. İlk noktadan sonra, kenar sayısı kadar açıların koordinatları bulunmuştur. Robot, bu koordinatlar arasına çizgi çizerek çizimi yapmıştır (Şekil 4.9).



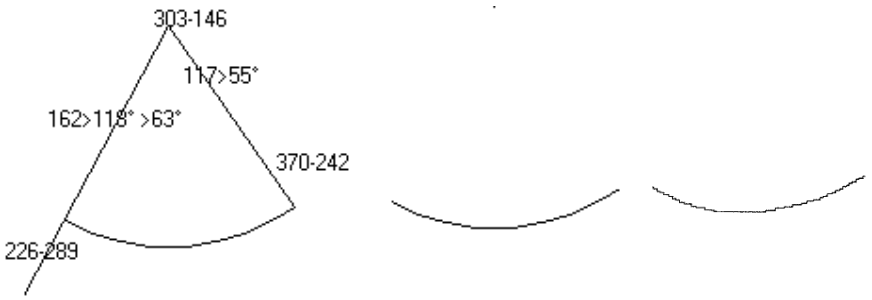
a b c

Şekil 4.6. Daire çizme, a.Çizim aşaması, b. Ekrandaki çizim, c. Robot çizimi.



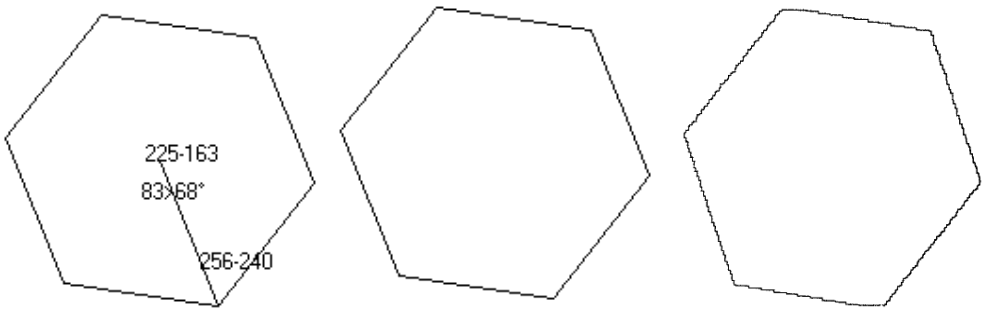
a b c

Şekil 4.7. Elips çizme, a.Çizim aşaması, b. Ekrandaki çizim, c. Robot çizimi.



a b c

Şekil 4.8. Yay çizme, a.Çizim aşaması, b. Ekrandaki çizim, c. Robot çizimi.



a b c

Şekil 4.9. Çokgen çizme, a.Çizim aşaması, b. Ekrandaki çizim, c. Robot çizimi.

Robotun çizdiği çizimler üzerinde çeşitli denemeler yapılarak, 200 adımda 70 mm (7cm) çizdiği tespit edilmiştir. Buradan, robot bir adımda 70/200=0.35mm çizeceğinden 0.35 oranı bulunmuştur. Bu nedenle, ekranda çizilen grafiklerin uzunluklarının, robotun çizeceği gerçek uzunlukları göstermesi için, piksel bazındaki uzunluk 0.35 oranı ile çarpılmıştır. Örneğin, ekranda 200 piksel olan uzunluk, motorun 200 adım atmasını sağlar. Fakat motor 200 adım attığında robot 70 mm uzunluğundaki çizgi çizecektir. Ekranda, 200 piksellik uzunluk 0.35 ile çarpıldıktan sonra 70 olarak gösterilir.

Çalışmada elde edilen grafikler, vektörel olarak kaydedilmiştir. Vektörel olarak kaydedilen dosyalar, içerdikleri verilerin okunabilmesi için “txt”

formatında kaydedilmiştir. Ayrıca, grafiklerin çizimine program olmadan bakılabilmesi için bitmap formatında "bmp" olarak da kaydedilmiştir.

İki dosya çeşidi karşılaştırıldığında, "bmp" dosyaların, "txt"(vektörel) dosyalardan çok daha fazla yer kapladığı görülmüştür. Bununla ilgili 3 örnek yapılmıştır (Şekil 4.10). "bmp" dosyası vektörel dosyanın, 1. Örnekte 1468 katı (Şekil 4.10.a), 2. örnekte yaklaşık 293 katı (Şekil 4.10.b), 3. örnekte (içinde 500 grafik var) yaklaşık 52 katı (Şekil 4.10.c) olduğu görülmüştür.

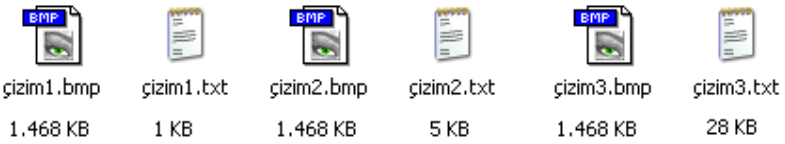
Programda, vektörleri çizmek için kullanılan sayfanın başlangıç koordinatı, ekranın sol üst köşesidir (Şekil4.11).

Matematikte, Y ekseninin yönü yukarıya doğru ve açıların yönleri saat ibresinin tersi yönündedir. Programda ise, Y ekseninin yönü aşağı doğru ve dolayısıyla açıların yönleri saat ibresi yönündedir. Açı hesaplamalarında, 180-360 arası açılarda, açılar 180-0 arası olmaktadır. 180^o den sonraki açılar *if YBoy<0 then Aci:=360-Aci;* komutu ile düzeltilmiştir.

$Cos(10^o) = 0{,}984$ $\quad$ $Acos(0{,}984) \cong 10^o$

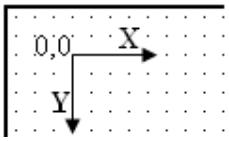
$Cos(180^o) = -1$ $\quad$ $Acos(-1) = 180^o$

$Cos(190^o) = -0{,}984$ $\quad$ $Acos(-0{,}984) \cong 170^o$ olmamalıdır. Bu açı gerçekte 190^o dir.



(a) (b) (c)

Şekil 4.10. Kaydedilen dosya çeşitleri ve büyüklükleri ile ilgili üç örnek.



Şekil 4.11. Başlangıç Koordinatı.

Windows XP işletim sistemi paralel porta direk erişmeye izin vermemektedir. Bundan dolayı, paralel porta erişmek için "io.dll" kütüphanesi kullanılmıştır. Paralel porta, bilgi göndermek için *PortOut*, bilgi okumak için *PortIn* komutu kullanılmıştır. Programa, "io.dll" kütüphanesi aşağıdaki şekilde eklenmiştir;

```
implementation {$R *.dfm}
procedure PortOut(Port : Word; Data : Byte);
stdcall;
external 'io.dll';
function PortIn(Port:Word):Byte;
stdcall;
external 'io.dll';
```

5. TARTIŞMA ve SONUÇ

Endüstriyel otomasyon sistemleri hızla gelişmekte olup, birçok iş sahasında insan gücünün yerini almaktadır. Sağladıkları faydalardan dolayı, dünyada hızlı bir gelişme sağlayan otomasyon sistemlerinin, endüstrinin birçok alanında kullanımı yaygınlaşmıştır.

Robot kullanılan sistemlerde, insan kaynaklı hatalar ortadan kalkmakta, üretim kalitesi artmakla kalmayıp bunun yanında verimlilikte artmaktadır.

Kartezyen robot, robotlar içinde dinamik donanım yapılandırması kolay, performansları yüksek, kullanım kolaylıkları olan robot çeşididir. Kartezyen Robot manipülatörün, 3 boyutlu eksen denetlenmesinde, geliştirilen ve uygulanan denetim algoritmasının ve pozisyon denetiminin basit olduğu görülmektedir. Kartezyen robot kullanımındaki amaç, sistemi mümkün olduğunca basit tutmak ve güvenilir bir denetleyici tasarlamaktır. Robot konstrüksiyonunun yapımında, hazır donanımsal yapıların incelenmesinin büyük etkisi vardır.

Günümüzde birçok uygulamada kartezyen denetim uygulamaları kullanılmaktadır. Kartezyen denetimde, robotun daha kesin ve doğru denetiminin sağlanabilmesi için geri besleme elemanı olarak sınır anahtarları yanında optik veya manyetik sensörler kullanılabilir. Böylece, robotun verilen işi daha az hatayla ve daha güvenli yapması sağlanabilir.

Kartezyen robot sistemlerinde, adım motorundan farklı olarak, servo motor, hidrolik veya pnömatik tahrik sistemleri kullanılabilir. Kontrol ünitesi olarak, bilgisayarın yanında, özel üretilmiş elektronik kartlar, PLC veya mikrodenetleyici (PIC) kullanılabilir.

Adım motorları, uygulamaları hızlandırmak için kullanılabilecek genel amaçlı motorlardır. Hassas mekaniki hareketin istenildiği, konum ve pozisyon kontrol sistemlerinde, birçok üstün özellikleri nedeniyle adım motorları tercih edilmektedir. Adım motorlarının adım açıları ne kadar küçük olursa o kadar hassas kontrol yapılabilir. ULN2003AN entegresi, adım motorlarını sürmek için

kullanılan, basit, ekonomik, programlama algoritması ve kontrolü kolay olan entegredir.

Kullanılan adım motorların adım açıları ne kadar küçük olursa çizilen çizimler o kadar hassas olur. Daire, elips, yay çizimlerinde fark açıları ne kadar küçük olursa daha kaliteli çiziler elde edilebilir. Ayrıca, dairenin çapı büyüdüğünde kenar sayısının büyütülmesi daha yararlı olacaktır.

Motorlar yarım adım çalıştırılarak hassasiyet değerleri arttırılmıştır. Çizimi yaparken yüksek hızlarda motorların titreşim yaptığı görülmüş, bu nedenle en uygun hızın 20 ms olduğu tespit edilmiştir. Motorların çalışmasında kayma ve adım hatası tespit edilmemiştir.

Bitmap dosyalar, vektörel dosyalardan daha fazla yer kaplamaktadırlar. Bundan dolayı günümüz profesyonel grafik programları, grafikleri bitmap dosya olarak değil de vektörel olarak kaydederler. AutoCAD, 3D Studio Max, Macromedia Flash örnek olarak gösterilebilir.

Paralel portu, elektronik devreden ve doğabilecek arızalardan yalıtmak için optokuplörler (4N25) kullanılabilir.

Bu çalışmada, endüstriyel amaçlı bir robotun bilgisayar tarafından nasıl kontrol edileceğini incelemeye ve tasarlamaya çalıştık. Geliştirilen kartezyen robot sistemi, kullanılan kalemin yerine değişik malzeme ve aparatların kullanılması ile değişik amaçlar için kullanılabilecek sistemler haline dönüştürülebilir. Örnek olarak, optik okuyucu, oksijen ile kaynak yapma ve kesme, cam veya bez kesme, baskı devresi delme, 3 boyutlu işleme, yük taşıma işlemlerini yapabilecek sistemler gibi.

Bu çalışma, bir robot kol uygulamasının temelini teşkil etmekte ve birçok uygulamaya temel teşkil edecek şekilde geliştirmeye çok elverişlidir. Sistemin çalışır modeli dikkate alınarak, imkânlar dâhilinde gerçek bir prototipi yapılabilir. Sistem, daha ileri versiyonlarında, bluetooth, wireless veya RF teknikleri ile kablosuz olarak çalıştırılabilir.

Eğri çiziminde AutoCAD programının çizim tekniği izlenmiştir. İstenirse program, yaygın olarak kullanılan AutoCAD gibi CAD programlarıyla veri

alışverişine uygun hale getirilerek daha komplike çizimlerin yapılması sağlanabilir.

Robotun rijitliğini arttırabilmek ve oluşabilecek titreşimleri elimine etmek için, kullanılan kasa gövdesinin sert malzemeden ve yerleştirilen aparatların kaynak, punto ve vidalamalarının sağlam yapılması gerekmektedir. Çalışmalar model robot üzerinde denenmiştir.

Robot, elektronik devre, gerekli ölçme, kablolama, boyama ve robotun çalıştırılması ile ilgili tüm deneysel çalışmalar YYÜ Hakkari MYO Elektrik bölümü laboratuarında yapılmıştır.

6. KAYNAKLAR

Altunsaçlı, A., 2003. ***"Elektrik Motorları ve Sürücüleri".*** Color Ofset. İskenderun. 309.

Apaydın, H., 2006***. Adım Motorlarının Karakteristikleri ve Bilgisayar İle Konum Kontrolü Uygulaması.*** (Yüksek lisans tezi, basılmamış). Marmara Üniversitesi, Fen Bilimleri Enstitüsü, Göztepe, İstanbul.

Arslan, M., 1998***. Bilgisayar Kontrollü Baskı Devre Delgi Cihazına Ait Ek Donanım Yapımı ve Uygun Bir Yazılım Geliştirimi.*** (Yüksek lisans tezi, basılmamış). Afyon Kocatepe Üniversitesi, Fen Bilimleri Enstitüsü, Afyon.

Atabey, Y. E., 2007***. Geri Beslemeyle Doğrusallaştırma Yönteminin Sürekli Mıknatıslı Adım Motoru Denetiminde Kullanılması.*** (Yüksek lisans tezi, basılmamış). Gazi Üniversitesi, Fen Bilimleri Enstitüsü, Ankara.

Bal, G., 1997. ***Özel Elektrik Makineleri.*** ISBN 975-94799-0-7, Ankara. 164.

Bayrak, A., 2007***. Beş Eksenli Bir Robot Kolunun Simülasyonu ve Kontrolü.*** (Yüksek lisans tezi, basılmamış). Gazi Üniversitesi, Fen Bilimleri Enstitüsü, Ankara.

Baytuğan, F., 2000***. Bilgisayar Kontrollü Baskı Devre Delgi Makinesi Tasarımı ve Kontrolü.*** (Yüksek lisans tezi, basılmamış). Ondokuz Mayıs Üniversitesi, Fen Bilimleri Enstitüsü, Samsun.

Benson, D., 2001. ***An Introduction To Stepper Motors For The Experimenter.*** Square Electronics. Kelseyville,CA 95451 USA. 199.

Çengelci, B., Çimen H., 2005. ***Endüstriyel Robotlar***. Makine Teknolojileri Elektronik Dergisi, Afyon Kocatepe Üniversitesi, Afyonkarahisar. 2005 (2) 69-78

Çetişli, B., 1998***. Adım Motorlarının Bilgisayarla Kontrolü.*** (Yüksek lisans tezi, basılmamış). Pamukkale Üniversitesi, Fen Bilimleri Enstitüsü, Denizli.

Demirli, N., İnan, Y., 2003. ***Delphi7.*** Prestige Yayınları. Ankara. 813.

Eğilmez, M., 2005. ***Kartezyen Koordinatlı Kaynak Robotu.*** (Yüksek lisans tezi, basılmamış).Yıldız Teknik Üniversitesi, Fen Bilimleri Enstitüsü, İstanbul.

Ersöz, H., 2007. ***Endüstriyel Robotlar ve Uygulama Alanları*** (Yüksek lisans tezi, basılmamış). Gazi Üniversitesi, Fen Bilimleri Enstitüsü, Ankara.

Görgün, H., 1998. ***Step Motor Kullanılarak Ploter Uygulamasının Gerçekleştirilmesi.*** (Yüksek lisans tezi, basılmamış). Yıldız Teknik Üniversitesi, Fen Bilimleri Enstitüsü, İstanbul.

Karagülle, İ., 2006. ***Borland Delphi 7.*** Türkmen Kitapevi. İstanbul. 1167.

Koca, H., 2007. ***Robot Manipülatör Denetimi.*** (Yüksek lisans tezi, basılmamış). Gazi Üniversitesi, Fen Bilimleri Enstitüsü, Ankara.

Okumuş, M. T., Gümüşoluk, A., 2004. ***"Elektrik Makineleri".*** Color Ofset. İskenderun. 92.

Oysu, C. ve ark., 2004. ***Termal Kesme İşlemleri İçin Üç Eksenli Kartezyen Robot Tasarımı.*** Kocaeli Üniversitesi, Mekatronik Mühendisliği, Kocaeli. BAPB-Proje No: 2004/019.

Özkan, A., 1999. PLC and SCADA Supported Position Control. ***Graduate Thesis***, Erciyes Üniversitesi, Fen Bilimleri Enstitüsü, 1-2, Kayseri.

Özyalçın, İ., 2006. ***Kartezyen Robot Tasarımı.*** (Yüksek lisans tezi, basılmamış). Mustafa Kemal Üniversitesi, Fen Bilimleri Enstitüsü, Antakya.

Pala, Z., 2003. ***Borland Delphi 7 Uygulama Geliştirme Rehberi 34 Konuda Delphi 7.*** Türkmen Kitapevi. İstanbul. 1272.

Sarıoğlu, M. K., 1990. ***Elektrik Makinelerinin Temelleri.*** İTÜ Ofset baskı atölyesi. İstanbul. 124.

Sciavicco, L., Siciliano, B., 1996. ***Modeling And Control Of Robot Manipulators.*** The McGraw-Hill Companies, Inc., New York, 11-26.

Taştan, L., 1997. ***Bilgisayar Yardımı İle Step Motorun Hareket Kontrolü*** (Yüksek lisans tezi, basılmamış). İTÜ, Fen Bilimleri Enstitüsü, İstanbul.

Usta, A. T., 1991. ***"HYBRID Adım Motorunun Modellenmesi ve Simülasyonu"***. (Yüksek lisans tezi, basılmamış). İstanbul Teknik Üniversitesi, Fen Bilimleri Enstitüsü, İstanbul.

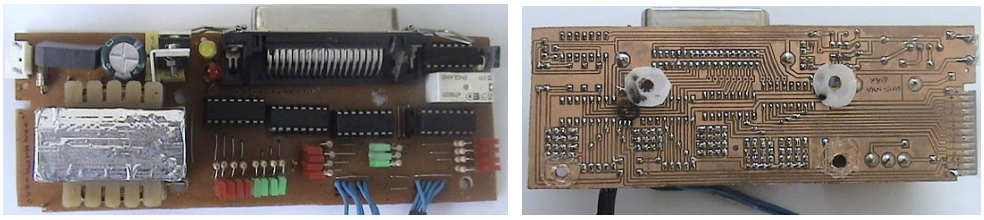
Ünlü, M., 1997. ***Bilgisayar Kontrollü Cam Kesme Tezgahı.*** (Yüksek lisans tezi, basılmamış). Osmangazi Üniversitesi, Fen Bilimleri Enstitüsü, Eskişehir.

Zeylan, F., 2003. ***Kartezyen Koordinatlı Kaynak Robot Kontrolü*** (Yüksek lisans tezi, basılmamış).Yıldız Teknik Üniversitesi, Fen Bilimleri Enstitüsü, Beşiktaş, İstanbul.

ISO Standardı, ISO 8373 (1994).

7. EKLER

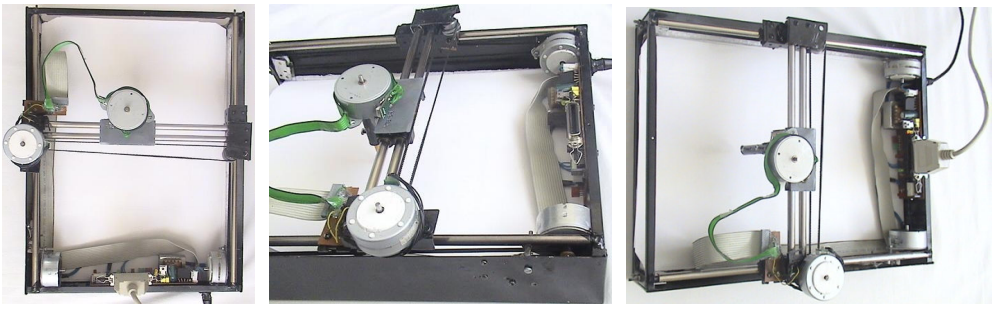
Ek 1. Elektronik kontrol kartın ön ve arka yüzü



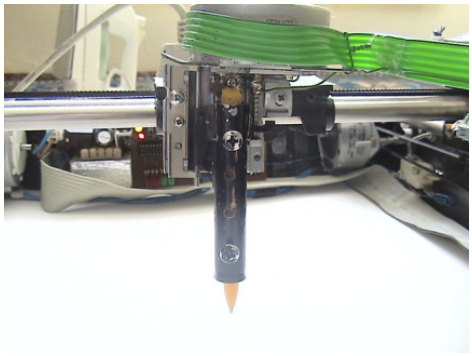
Ek 2. Robot konstrüksiyonunda ve elektronik devrede kullanılan malzemeler

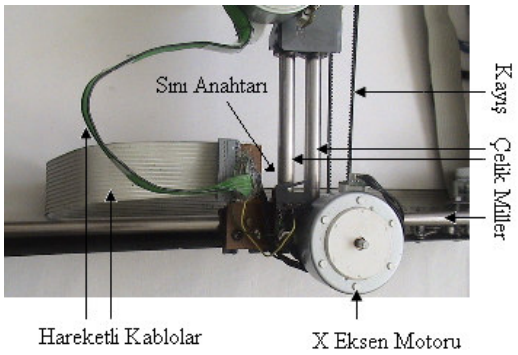
4 adet	PM55L Unipolar Hyprid adım motoru	2 adet	100nF kondansatör
4 adet	Çelik mil	1 adet	L7805CV
3 adet	kayış	1 adet	L7812CV
3 adet	Kasnak	18 adet	LED
1 adet	Kasa	19 adet	330Ω Direnç
2 adet	Mikro switch	1 adet	M74HC04
1 adet	SMD switch	1 adet	BD135
1 adet	2A cam sigorta	1 adet	12V 2A Röle
1 adet	KBL406 köprü diyot	4 adet	ULN2003AN
1 adet	1000 µF elektrolitik kondansatör	1 adet	Centronics connector
			Bakır pertinaks
			Tuzruhu
2 adet	220nF kondansatör		Perhidrol

Ek 3. Yapılan kartezyen robot

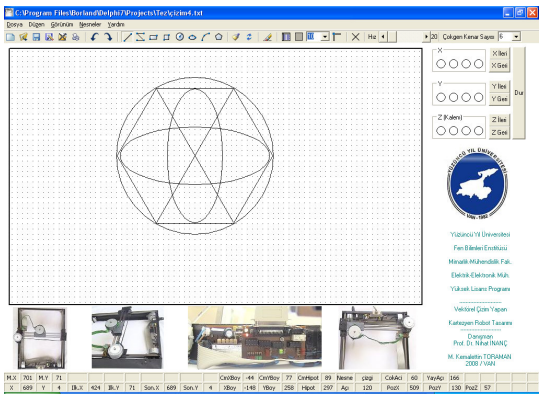


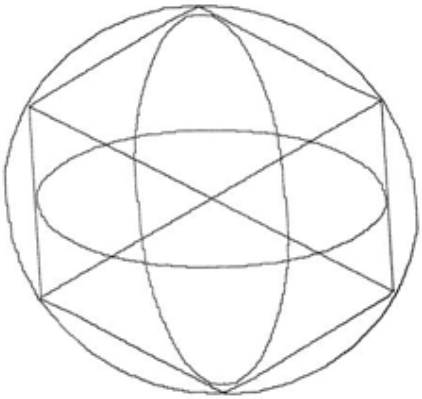
Ek 4. Yapılan kartezyen robot ayrıntıları





Ek 5. Örnek çizim ve Robot çizimi1

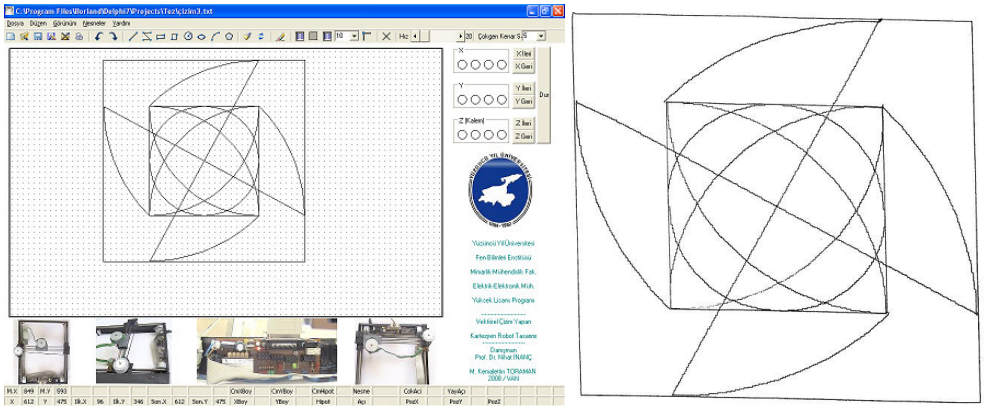




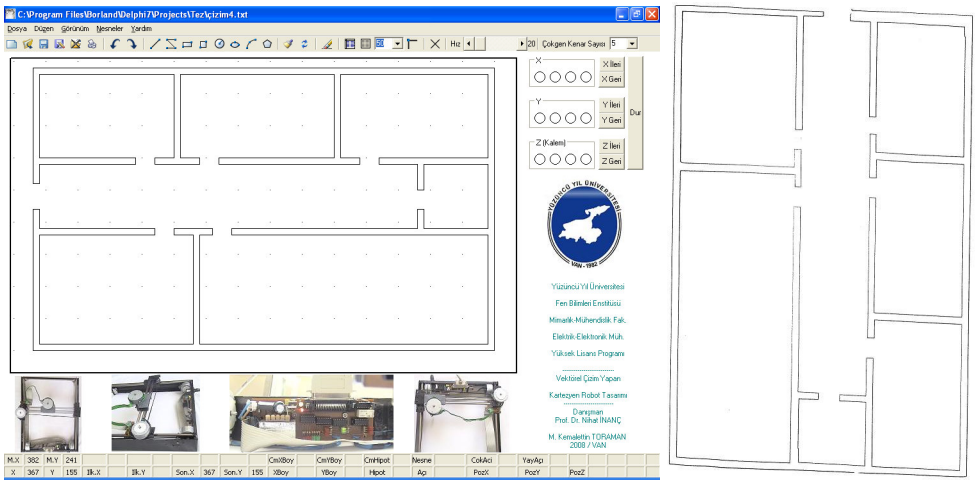
Ek 6. Yukarıdaki Örnek Çizim1'in veri kaydı

No	Nesne	IlkX	IlkY	SonX	SonY	Hipot	Hipot1	Aci1	Aci2	CokAci	Kenar	SayPlin	AcıBolg
1	daire	350	200	500	200	150		180				1	0
2	çokgen	350	200	499	200	149		0		60	7	2	0
3	elips	350	200	493	255	98	200	0	0	60	7	3	0
4	elips	350	200	403	328		168	133	45			4	0
5	çizgi	276	71	424	328		169	60	228			5	0
6	çizgi	424	71	276	329	76	62	120	357	45	9	6	0

Ek 7. Örnek çizim ve Robot çizimi2



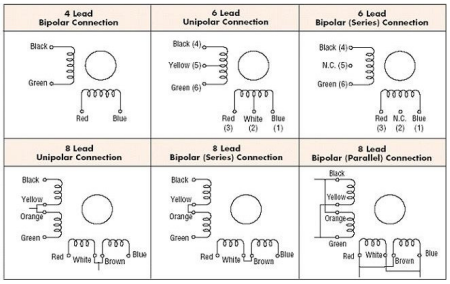
Ek 8. Örnek çizim ve Robot çizimi3



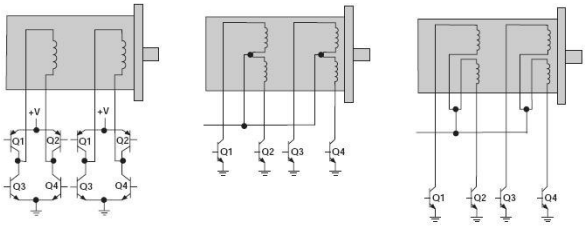
Ek 9. Kullanılan adım motorlarının karakteristik özellikleri

Motor Size	PM55L-048	
Number of Steps per Rotation	48(7.5°/Step)	
Drive Method	2-2 PHASE	
Drive Circuit	UNIPOLAR CONST. VOLT.	BIPOLAR CHOPPER
Drive Voltage	12[V]	12[V]
Current/Phase		800[mA]
Coil Resistance/Phase	30[Ω]	5.5[Ω]
Drive IC	2SC3346	UDN2916B-V
Magnet Material	Ferrite plastic magnet (MSPL) Polar anisotropy ferrite sintered magnet (MS50) Nd-Fe-B bonded magnet (MS70)	
Insulation Resistance	100M[Ω] MIN	
Dielectric Strength	AC 500[V] 1[min]	
Class of Insulation	CLASS E	
Operating Temp.	-10[°C] ~ 50[°C]	
Storage Temp.	-30[°C] ~ 80[°C]	
Operating Hum.	20[%] RH ~ 90[%] RH	

Ek 10. Step motorlarda kablo bağlama diyagramları



Ek 11. Bipolar ve Unipolar adım motorlarının sürülmesi



BIPOLAR

Step	Q_1-Q_4	Q_2-Q_5	Q_3-Q_8	Q_6-Q_7
1	ON	OFF	ON	OFF
2	ON	OFF	OFF	ON
3	OFF	ON	OFF	ON
4	OFF	ON	ON	OFF
1	ON	OFF	ON	OFF

Normal Adım

1	ON	OFF	ON	OFF
2	ON	OFF	OFF	OFF
3	ON	OFF	OFF	ON
4	OFF	OFF	OFF	ON
5	OFF	ON	OFF	ON
6	OFF	ON	OFF	OFF
7	OFF	ON	ON	OFF
8	OFF	OFF	ON	OFF
1	ON	OFF	ON	OFF

Yarım Adım

1	ON	OFF	OFF	OFF
2	OFF	OFF	OFF	ON
3	OFF	ON	OFF	OFF
4	OFF	OFF	ON	OFF
1	ON	OFF	OFF	OFF

Dalgalı sürüş

UNIPOLAR

Step	Q_1	Q_2	Q_3	Q_4
1	ON	OFF	ON	OFF
2	ON	OFF	OFF	ON
3	OFF	ON	OFF	ON
4	OFF	ON	ON	OFF
1	ON	OFF	ON	OFF

1	ON	OFF	ON	OFF
2	ON	OFF	OFF	OFF
3	ON	OFF	OFF	ON
4	OFF	OFF	OFF	ON
5	OFF	ON	OFF	ON
6	OFF	ON	OFF	OFF
7	OFF	ON	ON	OFF
8	OFF	OFF	ON	OFF
1	ON	OFF	ON	OFF

1	ON	OFF	OFF	OFF
2	OFF	OFF	OFF	ON
3	OFF	ON	OFF	OFF
4	OFF	OFF	ON	OFF
1	ON	OFF	OFF	OFF

Ek 12. Kartezyen robot kontrol program komutlarının tamamı

Programın yapında (Demirli ve İnan, 2003; Karagülle, 2006; Pala, 2003) kaynaklarından faydalanılmıştır.

```
{Yüzüncü Yıl Üniversitesi                                                    .
Fen Bilimleri Enstitüsü                                                      .
Mimarlık-Mühendislik Fakültesi                                               .
Elektrik-Elektronik Mühendisliği                                             .
Yüksek Lisans Programı                                                       .
** Vektörel Çizim Yapan Kartezyen Robot Tasarımı Programı **                 .
Danışman                                                                     .
Prof. Dr. Nihat İNANÇ                                                        .
ninanc@yyu.edu.tr                                                            .
M. Kemalettin TORAMAN                                                        .
Kemalettin@hotmail.com                                                       .
2008 / VAN                                                                   .
_____________________________________________________________________________.
İmage1 Özellikleri:                                                          .
780 x 480 piksel                                                             .
kağıt                                                                        .
273 x  168 mm                                                                .
yazıcı                                                                       .
1 adımı 7,5 derece                                                           .
48 adımda 17 mm                                                              .
1 adımda 0,35mm                                                             }

unit Unit1;
interface

uses
  Windows, Messages, SysUtils, Variants, Classes,  Graphics, Controls, Forms,
  Dialogs, ComCtrls, ToolWin,  ImgList,  ExtCtrls, StdCtrls, Math, Grids,Menus,
  jpeg;
type
  TForm1 = class(TForm)
  ToolBar1                  : TToolBar;
  Yeni_ToolButton           : TToolButton;
  Ac_ToolButton             : TToolButton;
  Kayit_ToolButton          : TToolButton;
  FKaydet_ToolButton        : TToolButton;
  DosyaKapat_ToolButton     : TToolButton;
  Yaz_ToolButton            : TToolButton;
```

```
Geri_ToolButton          : TToolButton;
Ileri_ToolButton         : TToolButton;
Cizgi_ToolButton         : TToolButton;
Dortgen_ToolButton       : TToolButton;
Kare_ToolButton          : TToolButton;
Daire_ToolButton         : TToolButton;
Elips_ToolButton         : TToolButton;
Sil_ToolButton           : TToolButton;
Kapat_ToolButton         : TToolButton;
Ayirac_ToolButton1       : TToolButton;
Ayirac_ToolButton2       : TToolButton;
Ayirac_ToolButton3       : TToolButton;
Ayirac_ToolButton4       : TToolButton;
Ayirac_ToolButton6       : TToolButton;
Cokgen_ToolButton        : TToolButton;
Pline_ToolButton         : TToolButton;
Yay_ToolButton           : TToolButton;
Temizle_ToolButton       : TToolButton;
Veri_ToolButton          : TToolButton;
Grid_ToolButton          : TToolButton;
Tazele_ToolButton        : TToolButton;
ToolButton1              : TToolButton;
ToolButton2              : TToolButton;
ToolButton3              : TToolButton;
PSifir_ToolButton        : TToolButton;
MainMenu1                : TMainMenu;
Dosya_MainMenu1          : TMenuItem;
Yeni_MainMenu1           : TMenuItem;
Ac_MainMenu1             : TMenuItem;
Kaydet_MainMenu1         : TMenuItem;
FKaydet_MainMenu1        : TMenuItem;
DKapat_MainMenu1         : TMenuItem;
N1                       : TMenuItem;
Yazdir_MainMenu1         : TMenuItem;
Duzen_MainMenu1          : TMenuItem;
Geri_MainMenu1           : TMenuItem;
Ileri_MainMenu1          : TMenuItem;
N3                       : TMenuItem;
Sil_MainMenu1            : TMenuItem;
N4                       : TMenuItem;
Kes_MainMenu1            : TMenuItem;
Gorunum_MainMenu1        : TMenuItem;
Veri_MainMenu1           : TMenuItem;
Grid_MainMenu1           : TMenuItem;
Kopyala_MainMenu1        : TMenuItem;
```

```
    Yapistir_MainMenu1      : TMenuItem;
    Nesne_MainMenu1         : TMenuItem;
    Cizgi_MainMenu1         : TMenuItem;
    SCizgi_MainMenu1        : TMenuItem;
    Dortgen_MainMenu1       : TMenuItem;
    Kare_MainMenu1          : TMenuItem;
    Daire_MainMenu1         : TMenuItem;
    Elips_MainMenu1         : TMenuItem;
    Yay_MainMenu1           : TMenuItem;
    Cokgen_MainMenu1        : TMenuItem;
    N2                      : TMenuItem;
    Cikis_MainMenu1         : TMenuItem;
    Yardim_MainMenu1        : TMenuItem;
    YKonu_MainMenu1         : TMenuItem;
    N5                      : TMenuItem;
    Hakinda_MainMenu1       : TMenuItem;
    Temizle1                : TMenuItem;
    Yenile1                 : TMenuItem;
    PozisyonSfrla1          : TMenuItem;
    Shape1                  : TShape;
    Shape2                  : TShape;
    Shape3                  : TShape;
    Shape4                  : TShape;
    Shape5                  : TShape;
    Shape6                  : TShape;
    Shape7                  : TShape;
    Shape8                  : TShape;
    Shape9                  : TShape;
    Shape10                 : TShape;
    Shape11                 : TShape;
    Shape12                 : TShape;
    Label1                  : TLabel;
    Label2                  : TLabel;
    Label3                  : TLabel;
    Label4                  : TLabel;
    Label5                  : TLabel;
    Label6                  : TLabel;
    Label7                  : TLabel;
    Label8                  : TLabel;
    Button1                 : TButton;
    Button2                 : TButton;
    Button3                 : TButton;
    Button4                 : TButton;
    Button5                 : TButton;
    Button6                 : TButton;
```

```
  Button9                 : TButton;
  StatusBar1              : TStatusBar;
  StatusBar2              : TStatusBar;
  GroupBox1               : TGroupBox;
  GroupBox2               : TGroupBox;
  GroupBox3               : TGroupBox;
  ComboBox1               : TComboBox;
  ComboBox2               : TComboBox;
  StringGrid1             : TStringGrid;
  OpenDialog1             : TOpenDialog;
  SaveDialog1             : TSaveDialog;
  ImageList1              : TImageList;
  Image1                  : TImage;
  ScrollBar1              : TScrollBar;
  Image2                  : TImage;
  Image3                  : TImage;
  Image4                  : TImage;
  Image5                  : TImage;
  Image6                  : TImage;
  StringGrid2             : TStringGrid;
  DVeri_ToolButton        : TToolButton;

procedure FormCreate (Sender: TObject);
procedure Image1MouseDown (Sender:TObject;Button:TMouseButton;
                           Shift:TShiftState; X,Y:Integer);
  procedure Image1MouseUp  (Sender:TObject;Button:TMouseButton;Shift:
                           TShiftState; X,Y:Integer);
Procedure Yazici (YNo,YKenar:Integer;YYNesne:ShortString;YIlk,YSon:TPoint;
                  YAci,Yaci1,YCokAci,YHipot,YHipot1:Real);
  procedure Image1MouseMove (Sender: TObject;Shift:TShiftState;X,Y:integer);
  procedure StringGrid1KeyPress    (Sender: TObject;var Key: Char);
  procedure FormMouseMove          (Sender: TObject;Shift:TShiftState;X,Y:
                                   Integer);
  procedure FormClose              (Sender: TObject;var Action: TCloseAction);
  procedure Yeni_ToolButtonClick         (Sender: TObject);
  procedure Ac_ToolButtonClick           (Sender: TObject);
  procedure Kayit_ToolButtonClick        (Sender: TObject);
  procedure FKaydet_ToolButtonClick      (Sender: TObject);
  procedure Kapat_ToolButtonClick        (Sender: TObject);
  procedure Geri_ToolButtonClick         (Sender: TObject);
  procedure Ileri_ToolButtonClick        (Sender: TObject);
  procedure Sil_ToolButtonClick          (Sender: TObject);
  procedure Veri_ToolButtonClick         (Sender: TObject);
  procedure DVeri_ToolButtonClick        (Sender: TObject);
  procedure Grid_ToolButtonClick         (Sender: TObject);
```

```
procedureTazele_ToolButtonClick        (Sender: TObject);
procedure Hakinda_MainMenu1Click       (Sender: TObject);
procedure Button1Click                 (Sender: TObject);
procedure Button2Click                 (Sender: TObject);
procedure Button3Click                 (Sender: TObject);
procedure Button4Click                 (Sender: TObject);
procedure Button5Click                 (Sender: TObject);
procedure Button6Click                 (Sender: TObject);
procedure Button9Click                 (Sender: TObject);
procedure ScrollBar1Change             (Sender: TObject);
procedure Yaz_ToolButtonClick          (Sender: TObject);
procedure Temizle_ToolButtonClick      (Sender: TObject);
procedure PSifir_ToolButtonClick       (Sender: TObject);
procedure FormPaint                    (Sender: TObject);
procedure ComboBox1Change              (Sender: TObject);
procedure Cizgi_MainMenu1Click         (Sender: TObject);
procedure SCizgi_MainMenu1Click        (Sender: TObject);
procedure Dortgen_MainMenu1Click       (Sender: TObject);
procedure Kare_MainMenu1Click          (Sender: TObject);
procedure Daire_MainMenu1Click         (Sender: TObject);
procedure Elips_MainMenu1Click         (Sender: TObject);
procedure Yay_MainMenu1Click           (Sender: TObject);
procedure Cokgen_MainMenu1Click        (Sender: TObject);
Procedure Uyari                (MesajNo:Smallint; var MesajCevap:Smallint);
Procedure AdimAyarl            (AdimX,AdimY,FAdimX,FAdimY:Integer);
Procedure YonMotorKontrol              (Motor:ShortString;Adim:Smallint);
Procedure HesaplaYaz                   (HYAci,HYHipot1,HYHipot2:Real);
Procedure PortCikis                    (PortNo,Port:Smallint);
Procedure SinirKontrol                 (Kontrol:ShortString);
Procedure HesaplaCiz                   (HAci,HCap,HCap2:Real);
Procedure CizgiYaz                     (XFark,YFark:Integer);
Procedure PozisyonAl                   (YPozX,YPozY:Integer);
Procedure L3                           (L:ShortString);
procedure IlkNokta                     (X,Y:Smallint);
procedure CizSon                       (X,Y:Smallint);
procedure LabelDurum                   (Durum:Boolean);
Procedure XIleri                       (Adim:Smallint);
Procedure XGeri                        (Adim:Smallint);
Procedure YIleri                       (Adim:Smallint);
Procedure YGeri                        (Adim:Smallint);
Procedure ZIleri                       (Adim:Smallint);
Procedure ZGeri                        (Adim:Smallint);
Procedure Renk                         (port:integer);
Procedure PozisyonSifirla;
Procedure HucreBaslik;
```

```
  procedure AcibolgeBul;
  procedure AcibolgeCiz;
  Procedure YenidenCiz;
  procedure Fkaydet;
  procedure Dortgen;
  Procedure Cokgen;
  Procedure Cokgen2;
  procedure Kaydet;
  Procedure Etiket;
  Procedure Cizgi1;
  Procedure Cizgi2;
  Procedure Cizgi3;
  Procedure Cizgi4;
  Procedure Cizgi4_2;
  Procedure Cizgi5;
  Procedure Cizgi6;
  procedure Daire;
  procedure Daire2;
  procedure Elips;
  procedure Elips2;
  Procedure Cizgi;
  Procedure Hesap;
  Procedure Ekle;
  procedure Grid;
  Procedure Yeni;
  Procedure Yay;
  Procedure Yay2;
  procedure Sil;
  Procedure Ac;
  Procedure L1;
  Procedure L2;
  Procedure L4;

private
public
  BC,Kayit,DosyaAc,Basla,FK,YazD,MotorD,Dur,PozDur   :Boolean;
  Ilk,Son,Gecici,Gecici2,OncekiSon,DNokta            :TPoint;
  Nesne,DosyaAdi                                     :ShortString;
  Hipot,Hipot1,DegisenAci,GeciciAci                  :Real;
  Aci,Aci1,Aci2,CokAci,YayAci                        :Real;
  Apsis,XBoy,YBoy,SayPline,KareKenar,Kenar           :Integer;
  iX,iY,iZ,iXY,PozX,PozY,PozZ,Hiz,AciBolge           :integer;
  SonKayit,MesajCevap,DCizgi,X1,Y1, DSay             :Smallint;
end;
type
```

```
  TPortOku=Set of Byte;
  TFAdim=Array of Integer;
var
  Form1: TForm1;
const
PortOkuX:TPortOku=[111,79,239,207];                    // X XY XZ XYZ
PortOkuY:TPortOku=[95,79,223,207];                     // Y YX YZ XYZ
PortOkuZ:TPortOku=[255,239,223,207];                    // Z ZX ZY XYZ
PortX:array[0..7] of byte=(16,48,32,96,64,192,128,144);    // X Motor
PortY:array[0..7] of byte=(1,  3, 2, 6, 4, 12,  8,  9);   // Y Motor
PortZ:array[0..3] of byte=(10,9,15,3);                   // Z Motor

implementation
Uses Unit2;
{$R *.dfm}

procedure PortOut(Port:Word; Data:Byte);
  stdcall;
  external 'io.dll';
function PortIn(Port:Word):Byte;
  stdcall;
  external 'io.dll';

procedure TForm1.FormCreate(Sender: TObject);
begin
  Label6.Caption:='Yüzüncü Yıl Üniversitesi'+#13#10+#13#10+
  'Fen Bilimleri Enstitüsü'+#13#10+#13#10+'Mimarlık-Mühendislik Fak.'
  +#13#10+#13#10+'Elektrik-Elektronik Müh.'+#13#10+#13#10+
  'Yüksek Lisans Programı'+#13#10+#13#10+'--------------------------'
  +#13#10+'Vektörel Çizim Yapan'+#13#10+#13#10+
  'Kartezyen Robot Tasarımı'+#13#10+'--------------------------'+#13#10
  +'Danışman'+#13#10+'Prof. Dr. Nihat İNANÇ'+
  #13#10+#13#10+'M. Kemalettin TORAMAN '+#13#10+'2008 / VAN';
  Form1.WindowState:=wsMaximized;
  DosyaAdi:='Adsız.txt';
  Caption:=DosyaAdi;
  MotorD:=True;
  Kayit:=True;
  BC:=True;
  FK:=True;
  SayPline:=0;
  LabelDurum(False);
  StringGrid1.Visible:=False;
  StringGrid2.Visible:=False;
  Sil;
```

```
  Grid;
  HucreBaslik;
  Hiz:=scrollbar1.position;
  Label8.Caption:=inttostr(hiz);
  PortOut(888,0);PortOut(890,11);
end;

procedure TForm1.Image1MouseDown(Sender:TObject;Button:TMouseButton;
          Shift:TShiftState;X,Y:Integer);
begin
  if (ssLeft in Shift) and (SayPline=0) and BC then IlkNokta(X,Y);
  if (SayPline<>0) and(Button=mbRight) then begin
    SayPline:=0;
    Case AciBolge of
      0:Cizgi2;
      1:Cizgi4;
      2:Cizgi4_2;
    end;
  end;
end;

procedure
TForm1.Image1MouseMove(Sender:TObject;Shift:TShiftState;X,Y:Integer);
var
  M:TPoint;
begin
  DCizgi:=0;AciBolge:=0;
  GetCursorPos(M);
  Image1.Canvas.Pen.Mode:=pmNot ;
  Son:=Point(X,Y);
  with StatusBar1.Panels do begin
    Items[1].Text:= FloatToStr(X);
    Items[3].Text:= FloatToStr(Y);
    Items[9].Text:=FloatToStr(Son.X);
    Items[11].Text:=FloatToStr(Son.Y);
  end;
  StatusBar2.Panels.Items[1].Text:=IntToStr(M.X);
  StatusBar2.Panels.Items[3].Text:=IntToStr(M.Y);
  if (Shift=[ssLeft]) then begin
    if X>780 then SetCursorPos(795,M.Y);//CurPos(+15) Y(+86).
    if Y>480 then SetCursorPos(M.X,566);
    if X<0   then SetCursorPos(15,M.Y);
    if Y<0   then SetCursorPos(M.X,86);
    if nesne='çizgi'  then begin Cizgi;Hesap;Etiket;end;
    if nesne='pline'  then begin Basla:=True;Cizgi;Hesap;Etiket;end;
```

```
    if nesne='dörtgen'then Dortgen;
    if nesne='kare'   then Dortgen;
    if nesne='daire'  then begin Cizgi;Daire;end;
    if nesne='elips'  then begin Elips;end;
    if (nesne='yay')and BC then begin Cizgi; Hesap; Etiket;end;
    if nesne='çokgen' then begin Cizgi;Cokgen;end;
  end;
  if YazD=False then begin // yazici çalıştığında çizgi çizmesin
    if (nesne='yay') and not BC then begin Cizgi;Hesap;Etiket;Yay;end;
    if (nesne='pline')and (SayPline<>0)then begin
Basla:=True;Cizgi;Hesap;Etiket;end;
    if (ssShift in Shift) and (nesne='pline')and (SayPline<>0)then begin
      AcibolgeBul;
      case AciBolge of
        1:begin Cizgi1;Cizgi5;Hesap;Etiket;end;
        2:begin Cizgi1;Cizgi6;Hesap;Etiket;end;
      end;
  end;
 end;
  if (ssShift in Shift) and (ssLeft in Shift) then begin
    if X>780 then SetCursorPos(796,M.Y);//CurPos(+16) Y(+87).
    if Y>480 then SetCursorPos(M.X,567);
    if X<0   then SetCursorPos(15,M.Y);
    if Y<0   then SetCursorPos(M.X,86);
    AcibolgeBul;
    if Nesne='çizgi'then begin
      if DCizgi=0 then Cizgi1;
      AcibolgeCiz;
      inc(DCizgi);
    end;
    if Nesne='pline'then begin
      Gecici2:=Point(Ilk.X,Ilk.Y);
      if SayPline=0 then begin
        Cizgi1;
        AcibolgeCiz;
      end;
    end;
  end;
end;

procedure TForm1.Image1MouseUp(Sender:TObject;Button:TMouseButton;
          Shift:TShiftState;X,Y:Integer);
begin
  LabelDurum(False);
  if Button=mbLeft then begin
```

```
    if Yay_ToolButton.Down then begin
      if BC and not DosyaAc Then begin
        BC:=False;
        Hipot1:=Hipot;
        Aci1:=Aci;
        GeciciAci:=Aci1;
        Gecici:=Point(Ilk.X,Ilk.Y);
        Gecici2:=Point(Ilk.X,Ilk.Y);
        OncekiSon:=Point(Son.X,Son.Y)
      end else begin
        BC:=True;
        Ilk:=Point(Gecici2.X,Gecici2.Y);
        Son:=Point(OncekiSon.X,OncekiSon.Y);
      end;
    end;
    if Pline_ToolButton.Down then begin
      Gecici2:=Point(Ilk.X,Ilk.Y);
       case AciBolge of
         0:ilk:=Point(X,Y);
         1:ilk:=Point(X,Ilk.Y);
         2:ilk:=Point(Ilk.X,Y);
       end;
      Inc(SayPline);
    end;
    CizSon(X,Y);
    if Basla and not DosyaAc and(Nesne<>'')then Ekle;
  end;
end;

procedure TForm1.Yeni_ToolButtonClick(Sender: TObject);
begin
  if Basla and not Kayit then begin
    Uyari(3,MesajCevap);
    Case MesajCevap of
      mrYes:begin Kaydet; if Kayit then Yeni; end;
      mrNo:Yeni;
      mrCancel:Exit;
    end;
  end
  else Yeni;
end;

procedure TForm1.Ac_ToolButtonClick(Sender: TObject);
begin
  if basla and not Kayit then begin
```

```
  Uyari(3,MesajCevap);
    Case MesajCevap of
      mrYes:begin Kaydet;if Kayit then Ac;end;
      mrNo:Ac;
      mrCancel:Exit;
    end;
  end
  else Ac;
end;

procedure TForm1.Kayit_ToolButtonClick(Sender: TObject);
begin
  if StringGrid1.RowCount=1 then Uyari(1,MesajCevap) else Kaydet;
end;

procedure TForm1.FKaydet_ToolButtonClick(Sender: TObject);
begin
  if StringGrid1.RowCount=1 then Uyari(1,MesajCevap) else FKaydet;
end;

procedure TForm1.Geri_ToolButtonClick(Sender: TObject);
begin
  StringGrid1.RowCount:=StringGrid1.RowCount-1;
  Image1.Picture:=nil;
  Grid;
  YenidenCiz;
end;

procedure TForm1.Ileri_ToolButtonClick(Sender: TObject);
begin
  StringGrid1.RowCount:= IfThen(StringGrid1.RowCount=SonKayit+1,
    StringGrid1.RowCount,StringGrid1.RowCount+1);
  Image1.Picture:=nil;
  Grid;
  YenidenCiz;
end;

procedure TForm1.Tazele_ToolButtonClick(Sender: TObject);
begin
  Image1.Picture:=Nil;
  Grid;
  YenidenCiz;
end;

procedure TForm1.Sil_ToolButtonClick(Sender: TObject);
```

```
begin
  Uyari(2,MesajCevap);
  Case MesajCevap of
    mrYes:begin Sil; Grid; end;
    else Exit;
  end;
end;

procedure TForm1.Veri_ToolButtonClick(Sender: TObject);
begin
  with StringGrid1 do begin
    if Visible then Visible:=False else Visible:= True;
  end;
end;

procedure TForm1.DVeri_ToolButtonClick(Sender: TObject);
begin
  with StringGrid2 do begin
    if Visible then Visible:=False else Visible:= True;
  end;
end;

procedure TForm1.Grid_ToolButtonClick(Sender: TObject);
begin
   Grid;
   YenidenCiz;
end;

procedure TForm1.Kapat_ToolButtonClick(Sender: TObject);
begin
  Dur:=True;close;
end;

procedure TForm1.FormClose(Sender: TObject; var Action: TCloseAction);
begin
  Dur:=True;
end;

procedure TForm1.PSifir_ToolButtonClick(Sender: TObject);
begin
  Dur:=False;
  PozisyonSifirla;
end;

procedure TForm1.Temizle_ToolButtonClick(Sender: TObject);
```

```
begin
  Image1.Picture:=Nil;Grid;
end;

procedure TForm1.ScrollBar1Change(Sender: TObject);
begin
  Label8.Caption:=IntToStr(ScrollBar1.Position);
end;

procedure TForm1.Sil;
begin
  Image1.Picture:=nil;
  StringGrid1.ColCount:=14;
  StringGrid1.RowCount:=1;
end;

procedure TForm1.Grid;
var I,K,A:Smallint;
begin
  a:=StrToInt(ComboBox1.Text);
  Image1.Canvas.Pen.Mode:=pmNot;
  K:=0;
  repeat
    I:=0;
    repeat
      Image1.Canvas.Pixels [I,K]:=clGray;
      I:=I+a;
    until I>Image1.Width;
    K:=K+a;
  until K>Image1.Height;
end;

Procedure TForm1.HucreBaslik;
begin
  with StringGrid1 do begin
    Cells[0,0] :='No';
    Cells[1,0] :='Nesne';
    Cells[2,0] :='IlkX';
    Cells[3,0] :='IlkY';
    Cells[4,0] :='SonX';
    Cells[5,0] :='SonY';
    Cells[6,0] :='Hipot';
    Cells[7,0] :='Hipot1';
    Cells[8,0] :='Aci1';
    Cells[9,0] :='Aci2';
```

```
    Cells[10,0]:='CokAci';
    Cells[11,0]:='Kenar';
    Cells[12,0]:='SayPline';
    Cells[13,0]:='AcıBolge';
  end;
end;

procedure TForm1.AcibolgeBul;
begin
  Case Round(Aci) of
        0..44: AciBolge:=1;
45,135,225,315: AciBolge:=0;
       46..134: AciBolge:=2;
      136..224: AciBolge:=1;
      226..314: AciBolge:=2;
      316..360: AciBolge:=1;
  end;
end;

procedure TForm1.AcibolgeCiz;
begin
  case AciBolge of
    0:begin Cizgi;Hesap;Etiket;end;
    1:begin Aci:=0;Cizgi5;Hesap;Etiket;end;
    2:begin Aci:=90;Cizgi6;Hesap;Etiket;end;
  end;
end;

procedure TForm1.IlkNokta(X,Y:Smallint);
begin
  Kayit:=False;
  DosyaAc:=False;
  Basla:=True;
  Ilk:=Point(X,Y);
  Gecici:=Point(X,Y);
  if Cizgi_ToolButton.Down   then begin Nesne:='çizgi'  ;end;
  if Dortgen_ToolButton.Down then begin Nesne:='dörtgen';end;
  if Kare_ToolButton.Down    then begin Nesne:='kare'   ;end;
  if Daire_ToolButton.Down   then begin Nesne:='daire'  ;end;
  if Elips_ToolButton.Down   then begin Nesne:='elips'  ;end;
  if Cokgen_ToolButton.Down  then begin Nesne:='çokgen' ;end;
  if Temizle_ToolButton.Down then begin Nesne:='nokta'  ;end;
  if Yay_ToolButton.Down     then begin Nesne:='yay'    ;end;
  if Pline_ToolButton.Down   then begin Nesne:='pline'  ;end;
  if (nesne='daire')or(Nesne='çokgen')then begin Aci1:=0; Hipot1:=0; end;
```

```
  L1;
  StatusBar1.Panels.Items[5].Text:= FloatToStr(Ilk.X);
  StatusBar1.Panels.Items[7].Text:= FloatToStr(Ilk.Y);
end;

procedure TForm1.Hesap;
var cmXBoy,cmYBoy,CMHipot:Integer;
begin
  XBoy:=Son.X-Ilk.x;
  YBoy:=Son.Y-Ilk.y;
  Hipot:=Hypot(XBoy,YBoy);
  if Hipot<0.1 then Aci:=0.0
  else Aci:=RadToDeg(ArcCos(XBoy / Hipot));
  if YBoy<0 then Aci:=360-Aci;
  Apsis:=Ilk.X+Round(Hipot);
  Aci:=Round(Aci);
  Hipot:=Round(Hipot);
  cmXBoy:=Round(XBoy/3.35);
  cmYBoy:=Round(YBoy/3.35);
  cmHipot:=Round(Hipot/3.35);
  if Nesne='çizgi'then begin
    if Xboy=0 Then Aci:=90;
    if Yboy=0 Then Aci:=0;
  end;
  if Nesne='yay'then begin
    YayAci:=Abs(Round(Aci1-Aci));
    YayAci:=IfThen(Aci1>Aci,360-YayAci,YayAci);
  end;
  With StatusBar1.Panels do begin
    Items[13].Text:=FloatToStr(XBoy);
    Items[15].Text:=FloatToStr(YBoy);
    Items[17].Text:=FloatToStr(Hipot);
    Items[19].Text:=FloatToStr(Aci);
  end;
  With StatusBar2.Panels do begin
    Items[13].Text:=IntToStr(cmXBoy);
    Items[15].Text:=IntToStr(cmYBoy);
    Items[17].Text:=IntToStr(cmHipot);
    Items[19].Text:=Nesne;
    Items[21].Text:=FloatToStr(CokAci);
    Items[23].Text:=FloatToStr(YayAci);
  end;
end;

Procedure TForm1.HesaplaCiz(HAci,HCap,HCap2:Real);
```

```
Begin
  DNokta.X:=Round (Cos(DegToRad(HAci))*HCap+Ilk.X);
  DNokta.Y:=Round (Sin(DegToRad(HAci))*HCap2+Ilk.Y);
  if not YazD then Image1.Canvas.LineTo(DNokta.X,DNokta.Y);
end;

procedure TForm1.LabelDurum(Durum:Boolean);
Begin
  Label1.Visible:=Durum;
  Label2.Visible:=Durum;
  Label3.Visible:=Durum;
  Label4.Visible:=Durum;
end;

procedure TForm1.Etiket;
begin
  if nesne<>'elips' then L2;
  if (nesne ='dörtgen') or (nesne='kare')or (nesne='elips') then begin
    L3('dke');
    L4;
  end else begin
    L3('cdcy');
    if not BC then
      Label3.Caption:=Label3.Caption+' >'+FloatToStr(YayAci)+'°';
  end;
end;

Procedure TForm1.L1;
begin
  Label1.Visible:=True;
  Label1.Left:=Ilk.X+7;
  Label1.Top:=Ilk.Y+30;
  Label1.Caption:=IntToStr(Ilk.X)+'-'+FloatToStr(Ilk.Y);
end;

Procedure TForm1.L2;
begin
  Label2.Visible:=True;
  case AciBolge of
    0:begin Label2.Left:=Son.X + 5;
          Label2.Top:=Son.Y + 10; end;
    1:Begin Label2.Left:=Son.X + 5;
          Label2.Top:=Ilk.Y + 10; end;
    2:begin Label2.Left:=Ilk.X + 5;
          Label2.Top:=Son.Y + 10; end;
```

```
  end;
  Label2.Caption:= IntToStr(Son.X)+'-'+ IntToStr(Son.Y);
end;

Procedure TForm1.L3(L:ShortString);
begin
  Label3.Visible:=True;
  if L='cdcy' then begin
    case AciBolge of
      0:begin Label3.Left:=((Ilk.X + Son.X ) Div 2)-10;
            Label3.Top :=((Ilk.Y + Son.Y ) Div 2)+12; end;
      1:begin Label3.Left:=((Ilk.X + Son.X ) Div 2)-10;
            Label3.Top :=Ilk.Y+12; end;
      2:begin Label3.Left:=Ilk.X-10;
            Label3.Top :=((Ilk.Y + Son.Y ) Div 2)+12; end;
    end;
    Label3.Caption:= FloatToStr(Round(Hipot*0.35))+'>'+FloatToStr(Aci)+'°';
  end;
  if L='dke' then begin
    Label3.Left:=((Ilk.X + Son.X ) Div 2)-10;
    Label3.Top :=Ilk.Y+12;
    Label3.Caption:= IntToStr(Abs(Round((Son.X - Ilk.X)*0.35)));
  end;
end;

Procedure TForm1.L4;
begin
  Label4.Visible:=True;
  Label4.Left:=Ilk.X-15 ;
  Label4.Top :=((Ilk.Y+Son.Y) Div 2)+20;
  Label4.Caption:=IntToStr(Abs(Round((Son.Y - Ilk.Y)*0.35)));
  if nesne='kare' then begin
    Label4.Top :=((Ilk.Y+KareKenar) Div 2)+20;
    Label4.Caption:=IntToStr(Abs(Round((Son.X - Ilk.X)*0.35)));
  end;
end;

procedure TForm1.Cizgi;
begin
  if nesne<>'elips' then begin
    Image1.Canvas.MoveTo(Ilk.X, Ilk.Y);
    Image1.Canvas.LineTo(Gecici.X, Gecici.Y);
    Cizgi2;
    if nesne<>'çokgen' then Gecici := Point(Son.X,Son.Y);
  end;
```

```
end;

procedure TForm1.Cizgi1;
begin
  Image1.Canvas.MoveTo(Ilk.X, Ilk.Y);
  Image1.Canvas.LineTo(Gecici.X, Gecici.Y);
  Gecici:=Point(Ilk.X,Ilk.Y);
end;

procedure TForm1.Cizgi2;
begin
  Image1.Canvas.MoveTo(Ilk.X, Ilk.Y);
  Image1.Canvas.LineTo(Son.X,Son.Y);
end;

procedure TForm1.Cizgi3;
begin
  Image1.Canvas.MoveTo(Ilk.X, Ilk.Y);
  Image1.Canvas.LineTo(Son.X,Ilk.Y);
  Image1.Canvas.LineTo(Son.X,Son.Y);
  Image1.Canvas.LineTo(Ilk.X,Son.Y);
  Image1.Canvas.LineTo(Ilk.X, Ilk.Y);
end;

Procedure Tform1.Cizgi4;
begin
  Image1.Canvas.MoveTo(Ilk.X, Ilk.Y);
  Image1.Canvas.LineTo(Son.X, Ilk.Y);
end;

Procedure Tform1.Cizgi4_2;
begin
  Image1.Canvas.MoveTo(Ilk.X, Ilk.Y);
  Image1.Canvas.LineTo(Ilk.X,Son.Y);
end;

Procedure Tform1.Cizgi5;
begin
  Image1.Canvas.MoveTo(Ilk.X,Ilk.Y);
  Image1.Canvas.LineTo(Gecici.X,Ilk.Y);
  Cizgi4;
  Gecici:=Point(Son.X,Ilk.Y);
end;

Procedure Tform1.Cizgi6;
```

```
begin
  Image1.Canvas.MoveTo(Ilk.X,Ilk.Y);
  Image1.Canvas.LineTo(Ilk.X,Gecici.Y);
  Cizgi4_2;
  Gecici:=Point(Ilk.X,Son.Y);
end;

procedure TForm1.Dortgen;
begin
  Image1.Canvas.MoveTo(Ilk.X, Ilk.Y);
  Image1.Canvas.LineTo(Gecici.X,Ilk.Y);
  Image1.Canvas.LineTo(Gecici.X,Gecici.Y);
  Image1.Canvas.LineTo(Ilk.X,Gecici.Y);
  Image1.Canvas.LineTo(Ilk.X, Ilk.Y);
  if nesne ='kare' then
    if Son.Y-Ilk.Y>0 then
        Son.Y:=Ilk.Y+Abs(Son.X-Ilk.X)
    else Son.Y:=Ilk.Y-Abs(Son.X-Ilk.X);
  Cizgi3;
  Gecici := Point(Son.X,Son.Y);
  KareKenar:=Son.Y;
  Hesap;
  Etiket;
end;

procedure TForm1.Daire;
var i:Smallint;
begin
  Hesap;Etiket;
  Image1.Canvas.MoveTo(Apsis,Ilk.Y);
  for i:=0 to 36 do begin
    HesaplaCiz(i*10,Hipot1,Hipot1);
  end;
  Daire2;
  Hipot1:=Hipot;
end;

procedure TForm1.Daire2;
var i:Smallint;
begin
  for i:=0 to 36 do begin
    HesaplaCiz(i*10,Hipot,Hipot);
  end;
end;
```

```
procedure TForm1.Elips;
var i:Smallint;
begin
  Image1.Canvas.MoveTo(Ilk.X, Ilk.Y);
  Image1.Canvas.LineTo(Gecici.X, Ilk.Y);
  Image1.Canvas.MoveTo(Ilk.X, Ilk.Y);
  Image1.Canvas.LineTo(Ilk.X, Gecici.Y);
  Cizgi4;Cizgi4_2;
  Hesap;Etiket;
  Image1.Canvas.MoveTo(Son.X,Ilk.Y);
  for i:=0 to 36 do begin
    HesaplaCiz(i*10, Gecici.X-Ilk.X, Gecici.Y-Ilk.Y);
  end;
  Elips2;
  Gecici := Point(Son.X, Son.Y);
end;

procedure TForm1.Elips2;
var i:Smallint;
begin
  for i:=0 to 36 do begin
    HesaplaCiz(i*10, Son.X-Ilk.X, Son.Y-Ilk.Y);
  end;
end;

Procedure TForm1.Cokgen;
var i:Smallint;
begin
  Kenar:=StrToInt(ComboBox2.Text);
  CokAci:=RoundTo(360/Kenar,-2);
  Kenar:=Kenar+1;
  Hesap;  Etiket;
  Image1.Canvas.MoveTo(Gecici.X,Gecici.Y);
  for i:=1 to Kenar do begin
    HesaplaCiz(Aci1,Hipot1,Hipot1);
    Aci1:=Aci1+CokAci;
  end;
  Cokgen2;
  Gecici:=Point(Son.X,Son.Y);
  Aci1:=Aci;
  Hipot1:=Hipot;
end;

Procedure TForm1.Cokgen2;
var i:Smallint;
```

```
begin
  Aci2:=Aci;
  Image1.Canvas.MoveTo(Son.X,Son.Y);
  for i:=1 to Kenar do begin
    HesaplaCiz(Aci2,Hipot,Hipot);
    Aci2:=Aci2+CokAci;
  end ;
end;

Procedure TForm1.Yay;
var  i,Tur:Smallint;
begin
  if Aci1>Aci then Aci:=360+Aci;
  if Aci1<Aci then begin
    Tur:=Round(GeciciAci-Aci1) div 5;
    DegisenAci:=Aci1;
    Image1.Canvas.MoveTo(Son.X,Son.Y);
    for i:=0 to Tur do begin
      HesaplaCiz(DegisenAci,Hipot1,Hipot1);
      if GeciciAci-DegisenAci<10 then DegisenAci:=GeciciAci
      else DegisenAci:=DegisenAci+5;
    end;
    Yay2;
    GeciciAci:=Aci;
  end;
end;

Procedure TForm1.Yay2;
var i,Tur:Smallint;
begin
  Tur:=Round(Aci-Aci1) div 5;
  DegisenAci:=Aci1;
  Image1.Canvas.MoveTo(Son.X,Son.Y);
  for i:=0 to Tur do begin
    HesaplaCiz(DegisenAci,Hipot1,Hipot1);
    if Aci-DegisenAci<10 then DegisenAci:=Aci
    else DegisenAci:=DegisenAci+5;
  end;
end;

procedure TForm1.CizSon(X,Y:Smallint);
begin
  Image1.Canvas.Pen.Mode:=pmBlack;
  if Nesne='çizgi' then
    case AciBolge of
```

```
      0:Cizgi2;
      1:Cizgi4;
      2:Cizgi4_2;
    end;
  if Nesne='pline' then begin
    Image1.Canvas.MoveTo(Gecici2.X,Gecici2.Y);
    case AciBolge of
      0:Image1.Canvas.LineTo(Son.X,Son.Y);
      1:Image1.Canvas.LineTo(Son.X,Gecici2.Y);
      2:Image1.Canvas.LineTo(Gecici2.X,Son.Y);
    end;
  end;
  if (Nesne='dörtgen') or (Nesne='kare') then Cizgi3;
  if Nesne='daire'then begin;
    Daire2; Image1.Canvas.Pen.Mode:=pmNot; Cizgi2; end;
  if Nesne='elips'then begin;
    Elips2; Image1.Canvas.Pen.Mode:=pmNot; Cizgi4;Cizgi4_2;end;
  if Nesne='çokgen'then begin;
  Cokgen2; Image1.Canvas.Pen.Mode:=pmNot; Cizgi2; end;
  if (Nesne='yay') and BC then begin;
    Yay2;
    Image1.Canvas.Pen.Mode:=pmNot;
    Image1.Canvas.MoveTo(Son.X,Son.Y);
    Image1.Canvas.LineTo(Ilk.X,Ilk.Y);
    Image1.Canvas.LineTo(X,Y);
  end;
end;

Procedure TForm1.Yeni;
begin
  DosyaAdi:='Adsız.txt';
  Caption:=DosyaAdi;
  Sil;Grid;YenidenCiz;
end;

procedure TForm1.Kaydet;
var
  DosyaBmp:ShortString;
  Text:TextFile;
  I,K:Smallint;
begin
  if not FK then exit;
  if DosyaAdi='Adsız.txt' then Fkaydet ;
  if Basla and not Kayit then begin
    Grid;
```

```
    DosyaBmp:=StringReplace(DosyaAdi,'.txt','.bmp',[rfReplaceAll]);
    Image1.Picture.SaveToFile(DosyaBmp);
    Grid;
    AssignFile (Text,DosyaAdi);
    Rewrite (Text);
    Writeln (Text,Stringgrid1.ColCount);
    Writeln (Text,Stringgrid1.RowCount);
    For I:=0 to StringGrid1.ColCount-1 do
      For K:=0 to StringGrid1.RowCount-1 do
        Writeln (Text, Stringgrid1.Cells[I,K]);
    CloseFile (Text);
    Kayit:=True;
    Basla:=False;
  end;
end;

procedure TForm1.Fkaydet;
begin
  SaveDialog1.InitialDir:=ExtractFilePath(Application.ExeName);
  if SaveDialog1.Execute then begin
    DosyaAdi:=SaveDialog1.FileName;
    Caption:=DosyaAdi;
    FK:=True;
    Kaydet;
  end
  else begin
    Kayit:=False;
    FK:=False;
    Basla:=False;
  end;
end;

procedure TForm1.Ac;
var
  Text:Textfile;
  Sat,Sut,I,K:Smallint;
  Temp:String;
begin
  DosyaAc:=True;   //Stringe son ekleme yapmasın
  Nesne:='';
  OpenDialog1.InitialDir:=ExtractFilePath(Application.ExeName);
  if OpenDialog1.Execute Then begin
    sil; Grid;YenidenCiz;
    DosyaAdi:=OpenDialog1.FileName;
    Caption:=DosyaAdi;
```

```
    AssignFile (Text,DosyaAdi);
    Reset (Text);
    Readln (Text,Sut);StringGrid1.ColCount:=Sut;
    Readln (Text,Sat);StringGrid1.RowCount:=Sat;
    For I:=0 to StringGrid1.ColCount-1 do
      For K:=0 to StringGrid1.RowCount-1 do begin
        Readln (Text, Temp);
        StringGrid1.Cells[I,K]:=Temp;
      end;
    CloseFile (Text);
    YenidenCiz;
  end
  else exit;
  Kayit:=True;
end;

procedure TForm1.Ekle;
var
  No:Smallint;
  OIlk,OSon:TPoint;
begin
  No:=StringGrid1.RowCount;
  SonKayit:=IfThen(SonKayit>No,Sonkayit,No);
  with StringGrid1 do begin
    Cells[0,No]:=IntToStr(No);
    Cells[1,No]:=Nesne;
    if BC then begin
      Cells[2,No]:=IntToStr(Ilk.X);
      Cells[3,No]:=IntToStr(Ilk.Y);
      Cells[4,No]:=IntToStr(Son.X);
      Cells[5,No]:=IntToStr(Son.Y);
      if (Nesne='çizgi') or (Nesne='pline')then begin
        Cells[13,No]:=IntToStr(AciBolge);
        case AciBolge of
          1:Cells[5,No]:=IntToStr(Ilk.Y);
          2:Cells[4,No]:=IntToStr(Ilk.X);
        end;
        OIlk.X:=StrToInt(Cells[2,No]);OIlk.Y:=StrToInt(Cells[3,No]);
        OSon.X:=StrToInt(Cells[4,No]);OSon.Y:=StrToInt(Cells[5,No]);
        if OSon.X-OIlk.X=0 then
          if  OSon.Y>OIlk.Y Then Aci:=90 else Aci:=270;
        if OSon.Y-OIlk.Y=0 then
          if  OSon.X>OIlk.X Then Aci:=0 else Aci:=180;
        Cells[8,No]:=FloatToStr(Aci);
      end;
```

```
      if Nesne='pline'then begin
        Cells[12,No]:=IntToStr(SayPline);
        Cells[2,No]:=IntToStr(Gecici2.X);
        Cells[3,No]:=IntToStr(Gecici2.Y);
      end;
      if (Nesne='daire') or (Nesne='çokgen')then
        Cells[6,No]:=FloatToStr(Hipot);
      if Nesne='çokgen' then begin
        Cells[8,No]:=FloatToStr(Aci);
        Cells[10,No]:=FloatToStr(CokAci);
        Cells[11,No]:=IntToStr(Kenar);
      end;
      if Nesne='yay' then
        if BC then begin
          Cells[4,No]:=FloatToStr(Son.X);
          Cells[5,No]:=FloatToStr(Son.Y);
          Cells[7,No]:=FloatToStr(Hipot1);
          Cells[8,No]:=FloatToStr(Aci);
          Cells[9,No]:=FloatToStr(Aci1);
      end;
      RowCount:=RowCount +1;
    end;
  end;
end;

Procedure TForm1.YenidenCiz;
var
 i,No      :Smallint;
 YNesne :ShortString;
begin
  Image1.Canvas.Pen.Mode:=pmBlack;
  for i:=1 to StringGrid1.RowCount-1 do begin
    with StringGrid1 do begin
     No:=StrToInt(Cells[0,i]);
      YNesne:=Cells[1,i];
      Ilk.X:=StrToInt(Cells[2,i]);
      Ilk.Y:=StrToInt(Cells[3,i]);
      Son.X:=StrToInt(Cells[4,i]);
      Son.Y:=StrToInt(Cells[5,i]);
      if (YNesne='çizgi') or (YNesne='pline')then begin
        Aci:=StrToFloat(Cells[8,i]);
        AciBolge:=StrToInt(Cells[13,i]);
      end;
      if (YNesne='daire') or (YNesne='çokgen') then
        Hipot:=StrToInt(Cells[6,i]);
```

```
      if YNesne='çokgen' then begin
        Aci:=StrToFloat(Cells[8,i]);
        CokAci:=StrToFloat(Cells[10,i]);
        Kenar:=StrToInt(Cells[11,i]);
      end;
      if YNesne='yay'then begin
        Hipot1:=StrToInt(Cells[7,i]);
        Aci:=StrToFloat(Cells[8,i]);
        Aci1:=StrToFloat(Cells[9,i]);
       end;
      if YNesne='pline' then
        SayPline:=StrToInt(Cells[12,i]);
    end;
//              verileri çiz        .
    if YazD then begin
      Yazici(No,Kenar,YNesne,Ilk,Son,Aci,Aci1,CokAci,Hipot,Hipot1);
    end else begin
      if (YNesne='çizgi')or(YNesne='pline')then Cizgi2;
      if (YNesne='dörtgen')or(YNesne='kare')then Cizgi3;
      if  YNesne='yay'then Yay2;
      if  YNesne='çokgen' then Cokgen2;
      if  YNesne='daire' then begin
       Image1.Canvas.MoveTo((Ilk.X+Round(Hipot)),Ilk.Y);
       Daire2;
      end;
      if YNesne='elips' then begin
        Image1.Canvas.MoveTo(Son.X,Ilk.Y);
        Elips2;
      end;
    end;
  end;
  SayPline:=0;
end;

Procedure TForm1.Uyari(MesajNo:Smallint;var MesajCevap:Smallint);
begin
  Case Mesajno of
   1: ShowMessage('Sayfa Boş Olduğundan Kayıt Yapamazsınız');
   2: MEsajCevap:=MessageDlg('Çizim silinecek Eminmisiniz?',
      MtConfirmation,[mbYes,mbNo,mbCancel] ,0);
   3: MesajCevap:=MessageDlg('Değişiklikler '+' ('+DosyaAdi+') '+
       ' Dosyasına Kaydedilsin mi?',MtConfirmation,
       [mbYes,mbNo,mbCancel] ,0);
  end;
end;
```

```
procedure TForm1.Hakinda_MainMenu1Click(Sender: TObject);
begin Form2.Show;end;

procedure TForm1.Button1Click(Sender: TObject);
begin
  SinirKontrol('XI');
  XIleri(780);
end;

procedure TForm1.Button2Click(Sender: TObject);
begin
  SinirKontrol('XG');
  XGeri(-780);
end;

procedure TForm1.Button3Click(Sender: TObject);
begin
  SinirKontrol('YI');
  YIleri(480);
end;

procedure TForm1.Button4Click(Sender: TObject);
begin
  SinirKontrol('YG');
  YGeri(-480);
end;

procedure TForm1.Button5Click(Sender: TObject);
begin
  SinirKontrol('ZI');
  ZIleri(160);
 end;

procedure TForm1.Button6Click(Sender: TObject);
begin
  SinirKontrol('ZG');
  ZGeri(-160);
end;

procedure TForm1.Button9Click(Sender: TObject);
begin Dur:=True; end;

Procedure FazlaKont(Buyuk,Kucuk,Adim,AdimK,Orta:Integer;var
TFazla:TFAdim;var F:Integer);
```

```
var
  i,c,YKalan:Integer;
  ilk :Real;
  x,y :integer;
begin
  SetLength(TFazla,AdimK);
  ilk:=Kucuk/(AdimK+1);
  if AdimK>Orta then begin
    F:=1;
  if Adim=1 then form1.Caption:='';
    x:=Abs(Adim+Sign(Adim)*1);
    y:=x*kucuk;
    Ykalan:=y-Buyuk;
    SetLength(TFazla,YKalan);
    ilk:=Round(Kucuk/(Ykalan+1));
    for i:=1 to YKalan do begin
      c:=Round(ilk*i);
      if c>kucuk then c:=kucuk;
      TFazla[i-1]:=c;
    end;
  end else begin
    for i:=1 to AdimK do begin
      F:=0;
      c:=Round(ilk*i);
      if c>kucuk then c:=kucuk;
      TFazla[i-1]:=c;
    end;
  end;
end;


procedure TForm1.Yaz_ToolButtonClick(Sender: TObject);
begin
  DSay:=1;
  PozisyonSifirla;
  Dur:=False;
  YazD:=True;
  YenidenCiz;
  YazD:=False;
  PozisyonSifirla;
end;

Procedure TForm1.PozisyonSifirla;
begin
  SinirKontrol('ZG');YonMotorKontrol('Z',-160);
```

```
  SinirKontrol('XG');YonMotorKontrol('X',-780);
  SinirKontrol('YG');YonMotorKontrol('Y',-480);
  PozDur:=False;
end;

Procedure
TForm1.Yazici(YNo,YKenar:Integer;YYNesne:ShortString;YIlk,YSon:TPoint;
          YAci,YAci1,YCokAci,YHipot,YHipot1:Real);
var
  XFark,YFark,i,Tur :Integer;
  YAci2,YDegisenAci:Real;
begin
  if Dur then begin
    PortOut(888,0);
    PortOut(890,11);
    exit;
  end;
  XFark:=YSon.X - YIlk.X;
  YFark:=YSon.Y - YIlk.Y;
  if (YYNesne='çizgi')or(YYNesne='pline') then begin
    PozisyonAl(YIlk.X,YIlk.Y);
    Gecici2:=Point(YIlk.X,YIlk.Y);
    if (Xfark=0)or(YFark=0)then AdimAyarla(XFark,YFark,0,0)
    else CizgiYaz(XFark,YFark);
    Gecici2:=Point(YSon.X,YSon.Y);
  end;
  if (YYNesne='dörtgen')or(YYNesne='kare') then begin
    PozisyonAl(YIlk.X,YIlk.Y);
    Gecici2:=Point(YIlk.X,YIlk.Y);
    AdimAyarla(XFark,YFark,0,0);
    AdimAyarla(-XFark,-YFark,0,0);
    Gecici2:=Point(YSon.X,YSon.Y);
  end;
  if (YYNesne='daire')then begin
    PozisyonAl((YIlk.X+Round(YHipot)),YIlk.Y);
    Gecici2:=Point((YIlk.X+Round(YHipot)),YIlk.Y);
    StringGrid2.Cells[1,0]:=IntToStr(YIlk.X);
    StringGrid2.Cells[2,0]:=IntToStr(YIlk.Y);
    StringGrid2.Cells[3,0]:=IntToStr(YSon.X);
    StringGrid2.Cells[4,0]:=IntToStr(YSon.Y);
    StringGrid2.Cells[6,0]:=FloatToStr(YHipot);
    for i:=1 to 36 do begin
      HesaplaYaz(i*10,YHipot,YHipot);
    end;
  end;
```

```
  if (YYNesne='elips')then begin
    PozisyonAl(Son.X,Ilk.Y);
    Gecici2:=Point(Son.X,Ilk.Y);
    for i:=1 to 36 do begin
      HesaplaYaz(i*10,Son.X-Ilk.X,Son.Y-Ilk.Y);
    end;
  end;
  if YYNesne='çokgen'then begin
    PozisyonAl(Son.X,Son.Y);
    Gecici2:=Point(Son.X,Son.Y);
    YAci2:=YAci;
    for i:=1 to YKenar-1 do begin
      YAci2:=YAci2+YCokAci;
      HesaplaYaz(YAci2,YHipot,YHipot);
    end;
  end;
  if YYNesne='yay'then begin
    PozisyonAl(Son.X,Son.Y);
    Gecici2:=Point(Son.X,Son.Y);
    Tur:=Round(YAci-YAci1) div 5;
    YDegisenAci:=YAci1;
    for i:=0 to Tur do begin
      HesaplaYaz(YDegisenAci,YHipot1,YHipot1);
      if YAci-YDegisenAci<10 then YDegisenAci:=YAci
      else YDegisenAci:=YDegisenAci+5;
    end;
  end ;
end;

Procedure TForm1.CizgiYaz(XFark,YFark:Integer);
var
  i,j,Orta:Integer;
  Buyuk,Kucuk:Integer;
  AdimX,AdimXK,AdimY,AdimYK:Integer;
  FAdimX,FadimY,TOku:Integer;
  TFazla:TFAdim;
  AXFark,AYFark,F  :Integer;
begin
  FAdimX:=0; FAdimY:=0;F:=0;
  AXFark:=Abs(XFark);
  AYFark:=Abs(YFark);
  Buyuk:=Max(AXFark,AYFark);
  Kucuk:=Min(AXFark,AYFark);
  if (Xfark=0)or(YFark=0)then AdimAyarla(XFark,YFark,0,0)
  else begin
```

```
   AdimX :=Sign(XFark)*(Buyuk div AYFark);
   AdimXK:=Buyuk mod AYFark;
   AdimY :=Sign(YFark)*(Buyuk div AXFark);
   AdimYK:=Buyuk mod AXFark;
   Orta:=Round(Kucuk/2);

   StringGrid2.Cells[9,DSay]:=IntToStr(AdimX);
   StringGrid2.Cells[10,DSay]:=IntToStr(AdimXK);
   StringGrid2.Cells[11,DSay]:=IntToStr(AdimY);
   StringGrid2.Cells[12,DSay]:=IntToStr(AdimYK);
   StringGrid2.Cells[13,DSay]:=IntToStr(orta);
   if AXFark>=AYFark then begin
     FazlaKont(Buyuk,Kucuk,AdimX,AdimXK,Orta,TFazla,F);
     for i:=1 to Kucuk do begin
       for j:=Low(TFazla) to High(TFazla) do begin
         TOku:= TFazla[j];
         if i=TOku then begin
           if AdimXK>Orta then FAdimX:=-1
           else FAdimX:=1;
           Break;
         end;
       end;
       AdimAyarla(AdimX+ Sign(AdimX)*F,AdimY,FAdimX,FAdimY);
       FAdimX:=0;FadimY:=0;
     end;
    end;
   if AXFark<AYFark then begin
     FazlaKont(Buyuk,Kucuk,AdimY,AdimYK,Orta,TFazla,F);
     for i:=1 to Kucuk do begin
       for j:=Low(TFazla) to High(TFazla) do begin
         TOku:= TFazla[j];
         if i=TOku then begin
           if AdimYK>Orta then FAdimY:=-1
           else FAdimY:=1;
           Break;
         end;
       end;
       AdimAyarla(AdimX,AdimY+Sign(AdimY)*F,FAdimX,FAdimY);
       FAdimX:=0;FadimY:=0;
     end;
    end;
  end;
end;

Procedure TForm1.HesaplaYaz(HYAci,HYHipot1,HYHipot2:Real);
```

```
var
HXFark,HYFark:Integer;
begin
  HesaplaCiz(HYAci,HYHipot1,HYHipot2);
  HXFark:=DNokta.X-Gecici2.X;
  HYFark:=DNokta.Y-Gecici2.Y;
  CizgiYaz(HXFark,HYFark);
  StringGrid2.Cells[0,DSay]:=IntToStr(kenar+1);
  StringGrid2.Cells[1,DSay]:=IntToStr(Gecici2.X);
  StringGrid2.Cells[2,DSay]:=IntToStr(Gecici2.Y);
  StringGrid2.Cells[3,DSay]:=IntToStr(DNokta.X);
  StringGrid2.Cells[4,DSay]:=IntToStr(DNokta.Y);
  StringGrid2.Cells[6,DSay]:=IntToStr(HXFark);
  StringGrid2.Cells[7,DSay]:=IntToStr(HYFark);
  inc(DSay);
  Gecici2:=Point(DNokta.X,DNokta.Y);
end;

Procedure TForm1.PozisyonAl(YPozX,YPozY:Integer);
begin
  if (Gecici2.X<>YPozX)or(Gecici2.Y<>YPozY)then begin
    if PozDur then YonMotorKontrol('Z',-100)
    else   YonMotorKontrol('Z',-160);
  end;
  YonMotorKontrol('X',YPozX-PozX);
  YonMotorKontrol('Y',YPozY-PozY);
  if PozDur then YonMotorKontrol('Z',100)
  else   YonMotorKontrol('Z',160);
  PozDur:=True;
end;

Procedure TForm1.SinirKontrol(Kontrol:ShortString);
begin
  if Dur then begin
    PortOut(888,0);
    PortOut(890,11);
    exit;
  end;
  if Kontrol='XG' then
    if PortIn(889) in PortOkuX then begin
       MotorD:=False;
       PozX:=1;
      end;
  if Kontrol='YG' then
    if PortIn(889) in PortOkuY then begin
```

```
      MotorD:=False;
      PozY:=1;
    end;
  if Kontrol='ZG' then
    if PortIn(889) in PortOkuZ then begin
       MotorD:=False;
       PozZ:=1;
      end;
  if Kontrol='XI' then
    if PozX>782 then MotorD:=False;
  if Kontrol='YI' then
    if PozY>482 then MotorD:=False;
  if Kontrol='ZI' then
    if PozZ>161 then MotorD:=False;
end;

procedure TForm1.YonMotorKontrol(Motor:ShortString;Adim:Smallint);
begin
  if Dur then begin
    PortOut(888,0);
    PortOut(890,11);
    exit;
  end;
  if Motor='X' then begin
    if Adim >0 then XIleri(Adim);
    if Adim <0 then Xgeri(Adim);
  end;
  if Motor='Y' then begin
    if Adim >0 then YIleri(Adim);
    if Adim <0 then Ygeri(Adim);
  end;
  if Motor='Z' then begin
    if Adim >0 then ZIleri(Adim);
    if Adim <0 then Zgeri(Adim);
  end; end;

Procedure TForm1.AdimAyarla(AdimX,AdimY,FAdimX,FAdimY:Integer);
begin
  YonMotorKontrol('X',AdimX+Sign(AdimX)*FAdimX);
  YonMotorKontrol('Y',AdimY+Sign(AdimY)*FAdimY);
end;

Procedure TForm1.XIleri(Adim:Smallint);
var i:Smallint;
begin
```

```
  Dur:=False;
  for i:=1 to Adim do begin
    if MotorD then begin
      PortOut(890,11);
      iX:=IfThen(iX=7,0,iX+1);
      PortCikis(888,PortX[iX]);
      if Dur then begin PortOut(888,0);exit;end;
      SinirKontrol('XI');
      Inc(PozX);
      StatusBar1.Panels.Items[21].Text:=IntToStr(PozX);
    end;
  end;
  MotorD:=True;
  PortOut(888,0);
end;

Procedure TForm1.XGeri(Adim:Smallint);
var i:Smallint;
begin
  Dur:=False;
  for i:=-1 downto Adim do begin
    if MotorD then begin
      PortOut(890,11);
      iX:=IfThen(iX=0,7,iX-1);
      PortCikis(888,PortX[iX]);
      if Dur then begin PortOut(888,0);exit;end;
      Dec(PozX);
      SinirKontrol('XG');
      StatusBar1.Panels.Items[21].Text:=IntToStr(PozX);
    end;
  end;
  MotorD:=True;
  PortOut(888,0);
end;

Procedure TForm1.YIleri(Adim:Smallint);
var i:Smallint;
begin
  Dur:=False;
  for i:=1 to Adim do begin
    if MotorD then begin
      PortOut(890,11);
      iY:=IfThen(iY=7,0,iY+1);
      PortCikis(888,PortY[iY]);
      if Dur then begin PortOut(888,0);exit;end;
```

```
      SinirKontrol('YI');
      Inc(PozY);
      StatusBar1.Panels.Items[23].Text:=IntToStr(PozY);
    end;
  end;
  MotorD:=True;
  PortOut(888,0);
end;

Procedure TForm1.YGeri(Adim:Smallint);
var i:Smallint;
begin
  Dur:=False;
  for i:=-1 downto Adim do begin
    if MotorD then begin
      PortOut(890,11);
      iY:=IfThen(iY=0,7,iY-1);
      PortCikis(888,PortY[iY]);
      if Dur then begin PortOut(888,0);exit;end;
      Dec(PozY);
      SinirKontrol('YG');
      StatusBar1.Panels.Items[23].Text:=IntToStr(PozY);
    end;
  end;
  MotorD:=True;
  PortOut(888,0);
end;

procedure TForm1.ZIleri(Adim:Smallint);
var i:Smallint;
begin
  Dur:=False;
  for i:=1 to Adim do begin
    if MotorD then begin
      PortOut(888,0);
      iZ:=IfThen(iZ=3,0,iZ+1);
      PortCikis(890,PortZ[iZ]);
      if Dur then begin PortOut(890,11);exit;end;
      SinirKontrol('ZI');
      Inc(PozZ);
      StatusBar1.Panels.Items[25].Text:=IntToStr(PozZ);
    end;
  end;
  MotorD:=True;
  PortOut(890,11); end;
```

```
procedure TForm1.ZGeri(Adim:Smallint);
var i:Smallint;
begin
  Dur:=False;
  for i:=1 downto Adim do begin
    if MotorD then begin
       PortOut(888,0);
      iZ:=IfThen(iZ=0,3,iZ-1);
      PortCikis(890,PortZ[iZ]);
      if Dur then begin PortOut(890,11);exit;end;
      Dec(PozZ);
      SinirKontrol('ZG');
      StatusBar1.Panels.Items[25].Text:=IntToStr(PozZ);
    end;
  end;
  MotorD:=True;
  PortOut(890,11);
end;

Procedure TForm1.PortCikis(PortNo,Port:Smallint);
begin
  Hiz:=ScrollBar1.Position;
  PortOut(PortNo,Port);
  if PortNo=888 then Renk(Port);
  if PortNo=890 then Renk(-Port);
  sleep(Hiz);
end;

procedure TForm1.Renk(port:integer);
begin
  case port of
    16:Shape1.Brush.Color:=clRed;
    32:Shape2.Brush.Color:=clRed;
    64:Shape3.Brush.Color:=clRed;
   128:Shape4.Brush.Color:=clRed;
    48:begin Shape1.Brush.Color:=clRed;
           Shape2.Brush.Color:=clRed;end;
    96:begin Shape2.Brush.Color:=clRed;
           Shape3.Brush.Color:=clRed;end;
   192:begin Shape3.Brush.Color:=clRed;
           Shape4.Brush.Color:=clRed;end;
   144:begin Shape4.Brush.Color:=clRed;
           Shape1.Brush.Color:=clRed;end;
     1:Shape5.Brush.Color:=clRed;
     2:Shape6.Brush.Color:=clRed;
```

```
     4:Shape7.Brush.Color:=clRed;
     8:Shape8.Brush.Color:=clRed;
     3:begin Shape5.Brush.Color:=clRed;
             Shape6.Brush.Color:=clRed;end;
     6:begin Shape6.Brush.Color:=clRed;
             Shape7.Brush.Color:=clRed;end;
    12:begin Shape7.Brush.Color:=clRed;
             Shape8.Brush.Color:=clRed;end;
     9:begin Shape8.Brush.Color:=clRed;
             Shape5.Brush.Color:=clRed;end;
   -10:Shape9.Brush.Color:=clRed;
    -9:Shape10.Brush.Color:=clRed;
   -15:Shape11.Brush.Color:=clRed;
    -3:Shape12.Brush.Color:=clRed;
  end;
  Application.ProcessMessages;
  Shape1.Brush.Color:=clWhite;
  Shape2.Brush.Color:=clWhite;
  Shape3.Brush.Color:=clWhite;
  Shape4.Brush.Color:=clWhite;
  Shape5.Brush.Color:=clWhite;
  Shape6.Brush.Color:=clWhite;
  Shape7.Brush.Color:=clWhite;
  Shape8.Brush.Color:=clWhite;
  Shape9.Brush.Color:=clWhite;
  Shape10.Brush.Color:=clWhite;
  Shape11.Brush.Color:=clWhite;
  Shape12.Brush.Color:=clWhite;
end;

procedure TForm1.FormMouseMove(Sender: TObject; Shift: TShiftState; X,
  Y: Integer);
var
  M:TPoint;
begin
  GetCursorPos(M);
  with StatusBar2.Panels do begin
  Items[1].Text:=IntToStr(M.X);
  Items[3].Text:=IntToStr(M.Y);
  end;end;

procedure TForm1.FormPaint(Sender: TObject);
begin
  Canvas.Pen.Width:=2;
  Canvas.MoveTo(10,35);
```

```
  Canvas.LineTo(800,35);
  Canvas.LineTo(800,525);
  Canvas.LineTo(10,525);
  Canvas.LineTo(10,35);
end;

procedure TForm1.ComboBox1Change(Sender: TObject);
begin
  Image1.Picture:=Nil;
  Grid;
  YenidenCiz;
end;

procedure TForm1.StringGrid1KeyPress(Sender: TObject; var Key: Char);
begin
 Basla:=True;
 Kayit:=False;
end;

procedure TForm1.Cizgi_MainMenu1Click(Sender: TObject);
begin Cizgi_ToolButton.Down:=True;end;

procedure TForm1.SCizgi_MainMenu1Click(Sender: TObject);
begin Pline_ToolButton.Down:=True;end;

procedure TForm1.Dortgen_MainMenu1Click(Sender: TObject);
begin Dortgen_ToolButton.Down:=True;end;

procedure TForm1.Kare_MainMenu1Click(Sender: TObject);
begin Kare_ToolButton.Down:=True;end;

procedure TForm1.Daire_MainMenu1Click(Sender: TObject);
begin Daire_ToolButton.Down:=True;end;

procedure TForm1.Elips_MainMenu1Click(Sender: TObject);
begin Elips_ToolButton.Down:=True;end;

procedure TForm1.Yay_MainMenu1Click(Sender: TObject);
begin Yay_ToolButton .Down:=True;end;

procedure TForm1.Cokgen_MainMenu1Click(Sender: TObject);
begin Cokgen_ToolButton.Down:=True;end;
End.
```

ÖZ GEÇMİS

1970 yılında Van'da doğdu. İlk, orta ve lise öğrenimini Van'da tamamladı. 1994 yılında Marmara Üniversitesi Teknik Eğitim Fakültesi Elektrik bölümünden mezun oldu. 1994-1996 yılları arasında Kayseri/Talas Atatürk Çok Programlı lisesinde elektrik bölümü öğretmeni olarak çalıştı. 1996 yılından bu yana YYÜ. Hakkâri MYO. Elektrik bölümünde Öğretim Görevlisi olarak çalışmaktadır. Evli ve iki çocuk babasıdır.