T.C.
SAKARYA ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ

BİR ROBOT KOLU MEKANİZMASINDA ADIM MOTORLARI VASITASIYLA, VERİLEN KOORDİNATLARA HAREKETİN GERÇEKLEŞTİRİLMESİ

YÜKSEK LİSANS TEZİ

Tek. Öğr. Savaş YILMAZ

Enstitü Anabilim Dalı : ELEKTRONİK VE BİLG. EĞT.

Tez Danışmanı : Yrd. Doç. Dr. Halil İbrahim ESKİKURT

Nisan 2006

T.C.
SAKARYA ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ

BİR ROBOT KOLU MEKANİZMASINDA ADIM MOTORLARI VASITASIYLA, VERİLEN KOORDİNATLARA HAREKETİN GERÇEKLEŞTİRİLMESİ

YÜKSEK LİSANS TEZİ

Tek. Öğr. Savaş YILMAZ

Enstitü Anabilim Dalı : ELEKTRONİK VE BİLG. EĞT.

Bu tez 11 / 04 /2006 tarihinde aşağıdaki jüri tarafından Oybirliği ile kabul edilmiştir.

Yrd.Doç.Dr. H.İbrahim ESKİKURT	**Yrd.Doç.Dr. İlyas ÇANKAYA**	**Doç.Dr. Recep KAZAN**
Jüri Başkanı	**Jüri Üyesi**	**Jüri Üyesi**

TEŞEKKÜR

Tezin hazırlanması aşamasında bana her türlü desteği veren danışman hocam sayın Yrd. Doç. Dr. Halil İbrahim ESKİKURT`a ve çalışmamda kullandığım adım motorların temin edilmesinde bana yardımcı olan arkadaşlarıma, baskı devre aşamasında yardımlarını eksik etmeyen Körfez Meslek Lisesi Elektronik Bölümü öğretmenlerine ve Fatih YAZICI arkadaşıma, mekanik sistemin yapılmasında yardımcı olan Sadi YILDIRIR arkadaşıma ve ev arkadaşlarıma teşekkürü bir borç bilirim.

Ayrıca çalışmalarımı gerçekleştirdiğim süre içinde maddi ve manevi desteklerini eksik etmeyen aileme de teşekkür ederim.

İÇİNDEKİLER

SİMGELER VE KISALTMALAR LİSTESİ

ALE	: Adres Latch Enable
BCD	: Binary Coded Decimal
DPTR	: Data Pointer
EPROM	: Erasable Programmable Read Only Memory
I/O	: Input / Output
Ns	: Nano Saniye
PC	: Program Counter
RAM	: Random Access Memory
ROM	: Read Only Memory
SFR	: Special Function Register
SP	: Stack Pointer
UART	: Universal Asynchronous Receiver / Transmitter

ŞEKİLLER LİSTESİ

TABLOLAR LİSTESİ

ÖZET

Anahtar Kelimeler: 8051 Mikro Denetleyici, Kartezyen Robot, 2 Boyutlu Robot, Adım Motoru, Otomasyon.

Endüstride ihtiyaç duyulan, zamandan tasarruf, maliyetten tasarruf ve daha hızlı iş yapılması gibi faktörler göz önünde bulundurulduğunda robotların kullanılmasının kaçınılmazlığı anlaşılmaktadır.

Robot teknolojisinde üç boyutlu robotlar olduğu gibi iki boyutlu robotlar da kullanılmaktadır. Bu robotların kullanım alanları ihtiyaca göre değişmektedir.

Bu uygulamada iki boyutlu dışarıdan kontrollü bir robot tasarlanmaya çalışılmıştır. Kontrol biriminde 8051 mikro denetleyici kullanılmıştır. Mekanik kısmın hareket mekanizmasında ise adım motorlar tercih edilmiştir. Koordinat girişi ise ikişer karakterli olmak üzere 7 parça göstergeler vasıtasıyla takip edilmektedir. Dışarıdan girilen 4 adet iki boyutlu koordinat, mikro denetleyici üzerinde kıyaslanarak, kalemin bulunduğu noktaya en yakından en uzağa doğru sıralanır. Daha sonra sıralanan koordinatlar kaleme en yakın noktadan başlamak üzere nokta koydurma işlemi yapılmaktadır. Bu uygulamanın, buna benzer uygulamalardan en büyük farkı bir bilgisayara bağımlılığı ortadan kaldırmasıdır.

AT MECANISM OF A ROBOT ARM, REALIZATION OF ACTION TO ENTERED COORDINATES BY MEANS OF STEP MOTORS

SUMMARY

Keywords : 8051 Micro Controller, Cartesian Robot, 2-D Robot, Step Motor, Automation.

When factors like saving of time, cost and doing a faster business kept in mind inevitability of using robots will be understood.

In Robot technology; 2-D robots are used as well as 3-D Robots. Use of these robots are changeable according to needs.

At this application a 2-D Robot which can manually be controlled is worked on to design. At control unit 8051 micro controllers were used. Step motors are preferred in mechanics part of action mechanism. Coordinate entrance is followed by 7 segments with double characters. 4 units of 2-D coordinates inputted will be compared on micro controller and arrange in order from the closest to the furthermost to where the pencil is. Then, ordered coordinates will do the dotting duty from the closest point to the pencil. The greatest difference of this application in a comparison to similar ones is dependence to a computer is removed.

BÖLÜM 1. GİRİŞ

Günümüzde teknolojinin çok hızlı bir şekilde gelişme kaydetmesiyle getirmiş olduğu yenilikler insan hayatının bir parçası olmuştur. Bu yenilikleri insanlara sunma ve bunlardan insanları haberdar etme bir zorunluluk haline gelmiştir. Dünyada iletişimin çok artması ile beraber insanlar değişik dünya pazarlarına yönelmiştir. Bir ürünü daha kaliteli ve ucuza imal etmek rekabet piyasasında bir zorunluluk haline gelmiştir. Bunun içinde otomasyon teknolojisini kullanarak üretim yapmak gerekmektedir [1].

İnsanlar fizikî yapılarından dolayı bedensel olarak bütün işleri yapma imkânına sahip değildir. Bundan dolayı gücünün yetmediği yerlerde kullanmak üzere değişik makineler yapmışlardır. İlk zamanlarda fonksiyonel olmayan bu ilkel makineler, teknolojinin gelişme süreci içerisinde insanlar tarafından geliştirilerek, insanın yeteneklerine yakın yeteneklere sahip olan makineler üretilmiştir. İlk zamanlarda insan yardımı ile çalışan bu makineler, zamanla daha da geliştirilerek ve çeşitli çevre birimlerini de beraberinde kullanarak insana ihtiyaç kalmadan otomatik olarak çalışır hale getirilmiştir.

Sanayi alanında kullanılmak üzere tasarlanmış birçok robot bulunmaktadır. Robotlar genellikle, daha kaliteli üretim yapmak ve üretim maliyetini düşürmek için kullanılmaktadırlar. Ayrıca kimyasal enerji, nükleer enerji, çok yüksek ısı, titreşim, yüksek ses v.b gibi insan sağlığının zarar görme riskinin olduğu işlerde ve insan elinin ulaşamayacağı yerlerde robotlar kullanılmaktadır.

Bugün robot kullanımı hayatımızın birçok alanına girmiş olup, özellikle insan sağlığını aşırı derecede tehdit eden iş kollarında; yüksek ısı, titreşim, kimyasal ve nükleer enerji ile çalışılan yerler gibi yerlerde kullanımı çok daha yaygındır [2].

Zamananımızda robotların büyük bir çoğunluğu endüstride kullanılmaktadır. Bunun sebebi ise robotların hassaslık ya da güç gerektiren işleri, büyük bir hızla hatasız olarak yerine getirebilmelerinden kaynaklanmaktadır. Bu sebepten dolayı robot teknolojilerini geliştirmede büyük şirketler (Sony, Honda…), üniversiteler ve teknoloji kurumlarıyla baş başa gitmektedirler. Robotlar, endüstriden başka volkanın kraterleri ve okyanusların derinlikleri gibi insanların çalışamayacağı yerlerde de sıklıkla kullanılmaktadırlar. İnsanların gidemeyeceği yerlere onlarca mini robot gönderilerek çeşitli araştırmalar gerçekleştirilmektedir. NASA da uzay araştırmalarında robotları sıklıkla kullanmaktadır. NASA'nın hedefi ise diğer gezegenlerde hayat arayacak robotlar yapmaktır. Daha birçok yerde robotlar kullanılmakta ve her geçen gün bu oran biraz daha fazlalaşmaktadır [3].

Yapılan bu çalışmada amaç herhangi bir düzenekte, dışarıdan girilen dört koordinatın sıralattırılarak en kısa yoldan, istenen herhangi bir işi, el değmeden basit bir şekilde gerçekleştirmektir. Tabi cihaza yaptırılacak iş ihtiyaçlara göre değişebilir. Bu çalışma yapılırken bir prototip olarak düşünüldü. Yapılması istenen iş ise iki boyutlu bir koordinat sisteminde, düzeneğin 4 koordinatına bir kalemle nokta koydurmaktır.

Adım motorların kullanıldığı bir çok çalışma mevcuttur. Örneğin Sakarya Üniversitesindeki bir yüksek lisans tezinde gerçeklerştirilen labirent robotu tasarımında hareket adım motorlarla sağlamıştır [21]. Hakkari Meslek Yüksek Okulunda yapılan bilgisayar kontrollü çizim cihazında hareket yine adım motorlarla sağlanmıştır [35]. 2003 yılı Elektrik - Elektronik - Bilgisayar Mühendisliği 10. Ulusal Kongresinde de sunulan bir çalışmada kartezyen robotun hareketleri adım motorlar vasıtasıyla gerçekleştirilmiştir [28]. Bu gibi daha bir çok örnek verilebilir.

Daha önceleri bu uygulamaya benzer uygulamalar yapılmıştır. Detay olarak aynı olmayabilir fakat işlev olarak aynı görevler yaptırılmaya çalışılmıştır. Burada yapılan çalışma ile diğer çalışmaların gerçekleştirilmesi safhalarında kullanılan yardımcı programlar, yazılım dilleri ve arabirim mikro denetleyicileri gibi farklar bulunmaktadır, fakat bunlar amaca göre değişik şekillerde kullanılabilir. Görünen en büyük fark ise cihazın kontrolünde görülmektedir. Sakarya Üniversitesinde bir yüksek lisans çalışmasında kartezyen robotun kontrolü PLC üzerinden yapılmakta ve

koordinatlarda bir değişiklik yapılmak istendiğinde bilgisayar kullanmak gerekmektedir [22]. Kocaeli Üniversitesinde yayınlanan bir makalede ise kartezyen robotun kontrolü bilgisayarın paralel portları üzerinden yapılmaktadır [28]. Gazi Üniversitesinde yapılan bir yüksek lisans çalışmasında ise gerçeklerştirilen robot kolunun kontrolü PIC mikro denetleyici vasıtasıyla yapılmaktadır. Robotun yaptığı iş standarttır. Farklı bir iş yaptırılmak istendiğinde bilgisayar vasıtasıyla programa müdahele etmek gerekmektedir [6]. Burada yapılan çalışmada ise bilgisayara olan bağımlılık ortadan kaldırılmaktadır. Kontrol ve dışarıdan bilgi girişi arabirim kartı üzerindeki girişlerden sağlanmaktadır.

Çalışmada üç adet adım motor kullanıldı. Düzenekte iki adım motor koordinatı takip etmek için, üçüncü motor ise nokta atmak için kullanılmaktadır.

Dışarıdan koordinat girişi için 4 adet 7 parçalı gösterge kullanılmaktadır. Koordinatların girilmesi için ise; X koordinatı için 2, Y koordinatı için 2 adet gösterge kullanılmıştır. Her koordinat için aşağı ve yukarı yönlü olmak üzere ikişer adet sayıcı görevli buton yerleştirilmiştir.

Motorların koordinata gitmesi için ise ayrıca bir buton konulmuştur. Bu butona basılmasıyla motorlar sırayla hareket etmeye başlarlar. Önce X koordinatı sonra Y koordinatı, sonrada nokta koymakla görevli motor harekete geçmektedir.

Bu çalışmada birinci arabirim olarak 8051 mikro denetleyici programlama kartı kullanılmıştır. Bu karta bağlı bulunan başka bir arabirim kartı ile de dışarıdan koordinat girilmesi ve motorların sürülmesi sağlanmıştır.

Çalışmada gerçekleştirilen arabirim kartında, göstergeleri sürmek için 4 adet 7447 BCD'den göstergeye kodlayıcı, adım motorları sürmek için ise 3 adet darlington transistor dizisine sahip ULN2003A entegresi kullanıldı.

Bölüm 2'de robotların tarihçesi, robot kavramı ve endüstriyel robotlar anlatılmıştır.

Bölüm 3'de robotlarla ilgili yapılmış çalışmalardan bir kısım verilmiştir.

Bölüm 4'de kontrol birimi olarak kullanılabilecek birimler, mikro denetleyiciler ve 8051 mikro denetleyicisinin tanımı, özellikleri ve avantajları gibi kısa bilgiler yer almaktadır.

Bölüm 5'de ise sistemin sanal ortamda tasarlanması sırasında yapılan işlemler anlatılmaktadır.

Bölüm 6'da ise sanal ortamda tasarlanın cihazın somutlaştırılması için yapılan işlemler anlatılmaktadır.

Bölüm 7'de de sonuçlar ve çalışmayla ilgili öneriler yer almaktadır.

BÖLÜM 2. ROBOT KAVRAMI

Robot, mekanik sistemler ve bunlarla ilişkili kontrol ve algılama sistemleriyle, bilgisayar algoritmalarına bağlı olarak davranan makinelerdir. Genel bir yaklaşım ile "robot yeniden programlanabilen, maddeleri, parçaları, aletleri, programlanmış hareketlerle yapılacak işe göre taşıyan veya işleyen çok fonksiyonlu makinelerdir" şeklinde tanımlanabilir.

2.1. Robotların İşlevleri ve Robot Kavramının Tarihteki Gelişimi

Robotlar; insanlar için tehlikeli sayılabilecek yerlerde kullanılabilirler. Örneğin uzayda, maden ocaklarında, su altında çalışan robotlar bulunmaktadır. İnsan sağlığı için tehlike arz eden radyoaktif maddelerin, zehirli kimyasal bileşiklerin ve hastalık yapıcı bakterilerin bulunduğu alanlarda robotlar faydalı olabilirler.

Ortaçağda Selçuklu Türklerinden Sükman boyundan Cizreli Ebul-Iz'in, makineler ve robotların yapımında suyun potansiyel ve kinetik enerjilerinden faydalandığına dair kaynaklar bulunmaktadır [49]. Avrupa'da 17. ve 18. yüzyıllarda kullanılan ilkel otomatlar birer mekanik harikasıydılar. O zamanlarda kilise ve katedrallerin tepesindeki devasa saatlerde insan, melek, şeytan gibi figürler bulunmakta ve bu figürler ellerindeki tokmağı, çana doğru giderek vurmaktaydılar. Vuruş sayısı o anki saati belirlemekteydi [3].

Bununla beraber kurulu düzenek tarafından miller ve kaldıraçlar yardımıyla kuşun kanatları, kafası ve gagası kontrol edilebilmekteydi. Vana ve pistonlar sayesinde kuş sesi çıkartılabilmekteydi. Çalışırken kafasını ve kanatlarını hareket ettirip, öterken de gagasını oynatabilmekteydi. Belirli bir sırayla kuşun hareketleri yapılmaktaydı ve bu sıra takibi için miller kullanılmaktaydı [3].

Şekil 2.1. Millerle yapılan ROM yapısına iyi bir örnek (Meillardet'in Otomatın'dan alınmadır.)

Jacques de Vaucanson 1738 yılında Paris'te otomasyon sisteminde, tasarladığı müzisyen kıyafeti giydirilmiş otomatik flütçü, dudaklarına yapışık flüte hava pompalarken parmaklarıyla da flütün deliklerini açıp kapatarak müzik yapabilmekteydi. Bu otomatın repertuarında 12 melodi vardı [51].

Şekil 2.2. Jacques de Vaucanson'un diğer çalışmalarını gösteren bir pul

19. yüzyılda otomasyon sistemleri içerisinde en karmaşık olanı, Henri Meillardet tarafından 1805 yılında Londra'da geliştirilen, resim yapabilme ve yazı yazabilme kabiliyeti ile birlikte geniş belleğe sahip olan otomasyon sistemidir (Şekil 2.3). Oluşturulan sistem bir gemi resmini aslına uygun, bütün detaylarıyla beş dakikada çizebilmekteydi [50].

Şekil 2.3. Meillardet 'in şiir yazan ve resim çizen otomatı

"Robot" kelimesi literatürde ilk defa 1917 yılında Karel Capek'in hikayesi olan Opilec'de geçmiştir. Fakat robot anlayışı asil kavram olarak 1921 yılında yine aynı yazarın Rossum's Universal Robots (R.U.R.) adlı tiyatro eserinde ortaya atılmıştır. (Şekil 2.4) Eserde robotlar Rossum ve oğlunun topluma hizmet için oluşturduğu insan görünüşlü mahluklardı [52]. Robot mâna olarak; sıkıcı, ağır, angarya iş anlamına gelmektedir ve Çek dilinde bir kelimedir [53].

Şekil 2.4. Rossum's Universal Robots 'un görünümü

Issac Asimov dünyada ilk olarak robotlarla ilgilenen bilim dalına "Robotic" ifadesini kullanan kişi'dir. Kelime "I,Robot" adlı kitapta ve 'Runaround' (1942) adlı hikayede yer almıştır [54].

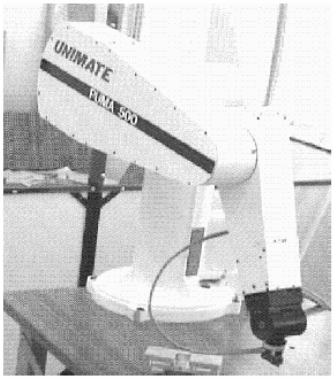
Issac Asimov'a göre robot kavramında insanlığın geleceği ile ilgili üç önemli kuram bulunmaktadır. (Daha sonradan 0. Kuramı eklemiştir);

0.kuram: Robotlar asla insan olgusuna zarar vermemelidir.

1.kuram: Robotlar asla insanlığa zarar vermemelidir. Aşağıdaki kuramlar tarafından bu kuramın aksi iddia edilemez.
2.kuram: Robotlar insanoğlundan aldığı emirleri yerine getirmelidir. Aşağıdaki kuramlar tarafından bu kuramın aksi iddia edilemez.
3.kuram: Robotlar kendi varlıklarını diğer kuramları bozmadan ellerinden geldikçe korumalıdırlar.

2.2. Robot Kavramının Gelişimi

Uzaktan kumanda ve sayısal kontrol kavramlarının gelişmesiyle robot ile ilgili çalışmalarda önemli gelişmeler meydana çıkmıştır. John Parson tarafından uzaktan kumandalı olarak yapılan makine 1940 yıllarında Amerika Birleşik Devletleri Hava Kuvvetleri tarafından, ardından da Atom Enerjisi Komisyonu tarafından kullanılmaya başlanmıştır. Endüstri alanında çeşitli uygulamalarda ve radyoaktif maddeler üzerine yapılan çalışmalarda kullanılmaya başlandı. Cyril Walter Kenward tarafından 1954'ün Mart ayında ilk endüstriyel robot denilebilecek bir mekanizma tasarlandı [3].



Şekil 2.5. Unimate 'in puma robotu

Robotların babası olarak adlandırılan parça aktarım robotu; Joseph F. Engelberger (Fizik Müh.) ile George C. Devol tarafından geliştirildi [10]. Geliştirilen sistemde en önemli özellik bilgilerin magnetik ortamlarda saklanmaları idi. 1949 yılındaki çalışmalar sonucunda "Unimate" adlı ilk robot firmasını kurdular. Amerika, Avrupa ve Japonya' da pek çok firma 1950'lerin başından itibaren özellikle 'robotic' üzerine çalışmaya başladılar. Çalışmalar sonucunda ilk robot uygulama dili 'WAVE'

Stanford Akademisi tarafından robotic bilimine kazandırıldı. Ultimate firması tarafından 1974'te geliştirilen 'VAL' programı "PUMA (Programmable Universal Machine for Assembly)" üzerinde uygulandı. PUMA'da temel olarak General Motors firmasının montaj hattı baz alınmıştı ve kısa eklemli bir robottu (Şekil 2.5). 1979 yılında 'SCARA' (Selective Compliance Arm for Robotic Assembly), Yamanashi Üniversitesi tarafından montaj amaçlı olarak geliştirildi ve geliştirilen sistem ticari olarak 1981 yılında piyasaya sürüldü [12, 48]. Robotlar 1990'lı yıllara gelindiğinde özellikle insanların rahatlıkla yapamayacağı işleri kusursuz yaparak, çok çeşitli alanlarda insanlığın yaşam sürecinde yerlerini aldılar. Artık bugün cerrahların hata yapmasını engelleyen hatta cerrahlık mesleğini ortadan kaldıracak kadar iddialı olan ameliyat robotları, insan kolunun yerine takılabilen yapay kol robotları (Şekil 2.7), robot köpek 'Aibo' (Şekil 2.8) [3], üretim hatlarında kullanılabilen robot kolları (Şekil 2.6), insan oğlunun inemediği derin sularda araştırma yapabilen ve hiçbir mola vermeden yıllarca çalışabilen montaj robotları robotlardaki gelişmelere en iyi örneklerdir.

Şekil 2.6. Otomobil üreten bantta bulunan robot kolları

Şekil 2.7. Biyonik robot kolu kavramına iyi bir örnek

Sony firması tarafından geliştirilen robot köpek "Aibo" ile karar verme, algılama, komut doğrultusunda istenilen işlemleri gerçekleştirme ve geniş açılı bir hafıza ile sahibini tanımaya kadar birçok özellik yer almaktadır [4].

Şekil 2.8. Sony firması tarafından yapılan Aibo isimli köpek robot

2.3. Endüstriyel Robotlar

ISO 8373 tarafından verilen sanayi robotu tanımı: Üç veya daha fazla programlanabilir ekseni olan, otomatik kontrollü, programlanabilir, çok amaçlı, bir yerde sabit duran veya tekerlekleri olan endüstriyel uygulamalarda kullanılan manipülatördür [1].

Bir robot, çeşitli işleri yerine getirmek üzere, malzeme, parça veya özel aletleri; değişken, programlanabilir hareketlerle taşımak üzere tasarlanmış, çok fonksiyonlu yeniden programlanabilir bir aygıttır. Robot uygulamaları başlıca elektronik, mekanik, otomotiv ve elektrik olmak üzere endüstrinin hemen her alanında görülebilmektedir [5].

Endüstride robot kullanımının başlıca nedenleri aşağıda görülmektedir:

1. İşçilik maliyetini azaltmak
2. Tehlikeli ve riskli yerlerde çalışanların yerini almak
3. Daha esnek bir üretim sistemi sağlamak
4. Daha tutarlı bir kalite kontrol sağlamak
5. Çıktı miktarını artırmak

6. Vasıflı işçilik sıkıntısını karşılayabilmek
7. Üç vardiya boyunca aralıksız çalışma kabiliyeti
8. İnsana göre daha fazla yük kaldırma kabiliyeti
9. İnsana göre daha çabuk sonuca ulaşma kabiliyeti
10. Usandırıcı ve tekrarlı işlerde yeterlilik
11. Tehlikeli ortamlarda çalışabilme kabiliyeti
12. İnsan hatalarını elimine etme
13. Kalite kontrol hatalarını minimuma indirme
14. Kendini hızla amorti etme
15. Yüksek hareket esnekliği
16. Yüksek kâr [2].

Yukarıdaki birçok faydalarının yanında şu sakıncaları robotlar için söylenebilir;

1. Düşünemez
2. Vision System, ile yalnızca kendisine öğretilen cisimleri görebilir
3. Programlanmadan çalışamaz
4. Kendisine öğretilenleri yapabildiğinden hareketleri kısıtlıdır
5. Yüksek yatırım maliyeti
6. Boşa geçen bakım ve onarım zamanları [2].

Endüstride kullanılan robotları değişik şekillerde sınıflandırmak mümkündür. Burada robot kolları hareket kabiliyetlerine göre sınıflandırıldı.

Tablo 2.1. Robot çeşitlerinin karşılaştırılması [8].

ROBOT TİPİ	EKLEM TİPLERİ	KULLANIM ALANLARI	KULLANIM SONUÇLARI
Kartezyen Robotlar	Prizmatik bel Prizmatik omuz Prizmatik dirsek	Demiryolu köprü inşaatları Büyük makine montaj hatları	Kinematik modelleri basittir. Rijit bir gövdeye sahiptir. Hareket analizleri basittir. Çalışması için büyük alanlar gerekir. Büyüklüğüne göre iş alanı küçüktür
Silindirik Robotlar	Dönebilen bel Prizmatik omuz Prizmatik dirsek	Büyük makine montaj sanayi Basit montaj-demontaj hatları	Kinematik modelleri basittir. Hareket analizi basittir. Güçlü hidrolik elemanlar kullanılır. İş alanları sınırlıdır. Tozlu ve ıslak ortamda çalışmaları zordur
Küresel Robotlar	Dönebilen bel Dönebilen omuz Prizmatik dirsek	Montaj sanayi Nükleer santraller	Büyük alanlara uzanabilirler Zeminden uzaktaki nesneleri tutabilirler.Kinematik modelleri karışıktır Hareket analizi zordur
Eklemli Robotlar	Dönebilen bel Dönebilen omuz Dönebilen dirsek	Otomobil montaj sanayi Otomobil boya sanayi Elektronik montaj sanayi Nükleer santrallerde Tıbbi araç-gereç yapım sanayi	Maksimum esnekliğe sahiptir. İş alanı robot büyüklüğü ile orantılıdır. Elektrik motorları kullanılabilinir Cisimleri altlarından tutabilir Kinematik yapıları karmaşıktır Hareket analizi zordur. Kolların rijitlik ayarı zordur

2.4. Endüstriyel Robot Kolları

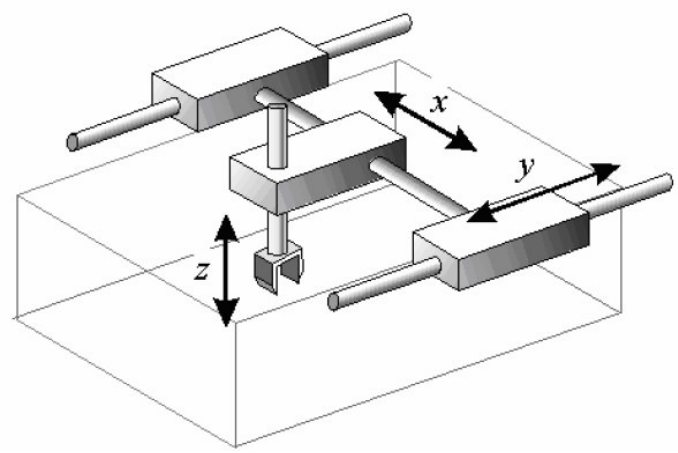
2.4.1. Kartezyen robot

X,Y,Z, eksenlerinde doğrusal olarak hareket etme yeteneğine sahip bu robot tipi sadece tutma ve taşıma yeteneğine sahiptirler. Basit bir yapıya sahip olduklarından dolayı hareketlerin planlanması basittir. Bu cins robotlarda; pozisyon hesaplamaları, robot uç elemanının bulunduğu pozisyon, mafsalların o anda olduğu yerde bulunduğundan dolayı çok basittir.

Şekil 2.9'da görüldüğü gibi çalışma alanları robotun yapısından daha küçüktür. Eğilme ve bükülme işlemlerini gerçekleştiremezler. Çalışma alanları dikdörtgen veya kare prizma şeklindedir. Diğer robot türlerine göre yük taşıma kapasitesi daha büyüktür. İnsan gücünün taşıma kapasitesinden fazla olan yüklerin taşınmasında kullanılır. Bu nedenle genellikle yükleme ve boşaltma işlerinde, fabrikalar da ağır yükleri taşımak amacı ile fabrikaların tavanlarına monte edilerek kullanımı yaygındır. Rtubetli ve ıslak ortamlarında kullanılabilirler.

Küçük güçtekiler pnömatik tahrik sistemine sahiptirler. Büyük güç gereken yerlerde hidrolik tahrikli olan kartezyen robotlar tercih edilir. Bunların yağ sızdırma gibi bir sorunları olduğundan dolayı temizliğin önem arz ettiği ortamlarda pnömatik tahrikli

olanlar tercih edilir. Pnömatik tahrikli robot tipinde basınçlı hava ve havanın kontrolüne ihtiyaç olduğu için yatırım maliyetleri daha ucuz olup işletim maliyetleri de düşüktür. Büyük güçte yapılan kartezyen robotların tahrik sistemleri hidrolik tahrik sistemleri veya elektrik motorları ile sağlanmaktadır [7].

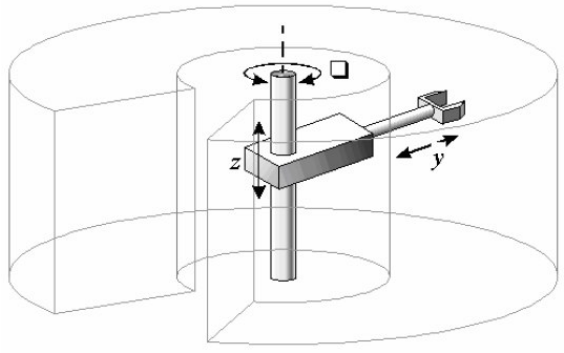


Şekil 2.9. Kartezyen robot kolun çalışma alanı

2.4.2. Silindirik robot kolları

Silindirik robot kolları da kendi etrafında dönebilen bir mafsal ve bunun üzerinde bulunan X,Y,Z düzleminde doğrusal hareket edebilen kollardan oluşmaktadır. Esnek olmayan silindirik bir koordinat sistemine sahiptirler (Şekil 2.10). Kartezyen robot kola göre hareket alanı daha geniştir.

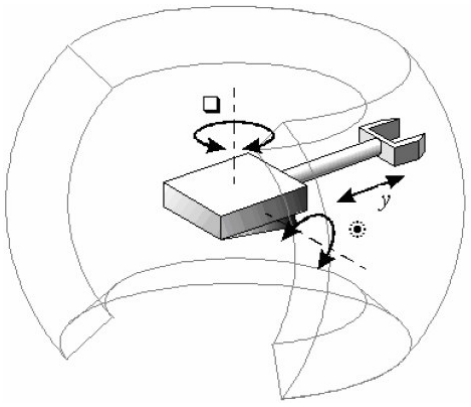
Çalışma alanındaki noktalara ulaşımı çok iyidir. Hareket kabiliyetinin azlığından dolayı programlanması basittir. Robot kolun çalışma alanı silindirik koordinat sisteminde hareket eden kolların uzunluğuna göre değişmektedir. Robotun kullanım alanı ve yük taşıma kapasitesine göre pnömatik, hidrolik veya elektrik tahrikli olarak kullanılmaktadır. Silindirik robot kolları nemli ve tozlu ortamlarda, deniz altı, uzay gözlem araçlarında ve nokta kaynağı işlerinde yaygın olarak kullanılmaktadır.



Şekil 2.10. Silindirik robot kolun çalışma alanı

2.4.3. Küresel robot kolları

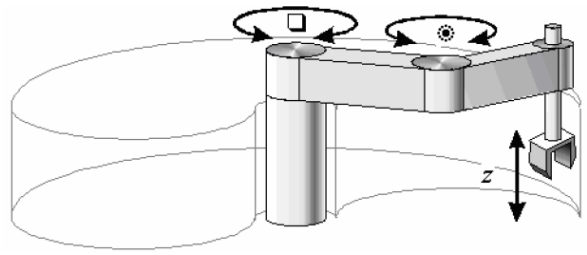
Bel, omuz ve dirsek mafsallarından oluşan bir sisteme sahiptirler. Omuz ve bel mafsalı kendi etrafında dönme hareketi yapabilirken, dirsek mafsalı kola uzama ve kısalma hareketi yaptırmaktadır. Hareket alanı şekil 2.11'de görüldüğü gibi silindirik bir koordinat sistemine sahiptir. Kol yapılarından dolayı eklemli robot kollarına benzemektedirler. Kkartezyen ve silindirik robot kollara göre kinematik yapıları daha karmaşıktır. Çalışma şeklinin zihinde canlandırılması zor olduğu için programlama ve kontrolü de kolay değildir. Çalışma alanının büyüklüğü kolların büyüklüğüne bağlıdır. Hidrolik tahrik sistemine sahip olan küresel robot kollar bükme, eğme işlerinde, kameralı izleme işlerinde kullanılmaktadır. Ayrıca sarkaç robot olarak da küçük bir moment ile hareketlerini devam ettiren bu robotlar, zamklama ve kaynak işlemlerinde kullanılmaktadırlar.



Şekil 2.11. Küresel robot kolun çalışma alanı

2.4.4. Scara robot kolu

İki eklem yerinde aşağı yukarı hareket edebilen pnömatik kol ve elektrik motorundan oluşmuştur. Eksenlerin kendi etrafında dönmesi eklemlerdeki elektrik motorları sayesinde sağlanmaktadır. Tutucu ağzın bulunduğu kol pnömatik tahrikli olup Z ekseninde hareket etme kabiliyetine sahiptir. Bu da robot kola esnek hareket imkânı sağlamaktadır. Konum ve hız performansı çok iyi olduğundan dolayı bu robot kolu en çok elektronik sanayinde, elektronik kartlara malzemelerin montajını gerçekleştirmek için kullanılmaktadır. Taşıma ve tutma işlerinde maliyetinin ucuz olmasından ve programlanmasının zor olmamasından dolayı şu anda sanayide en çok kullanılan robot kolu olmuştur. Bu robot kolu şekil 2.12'den de çalışma alanı görüldüğü gibi, kolun altında bulunan parçaların taşınmasında kullanılır.



Şekil 2.12. Scara Robot Kolun Hareket Alanı

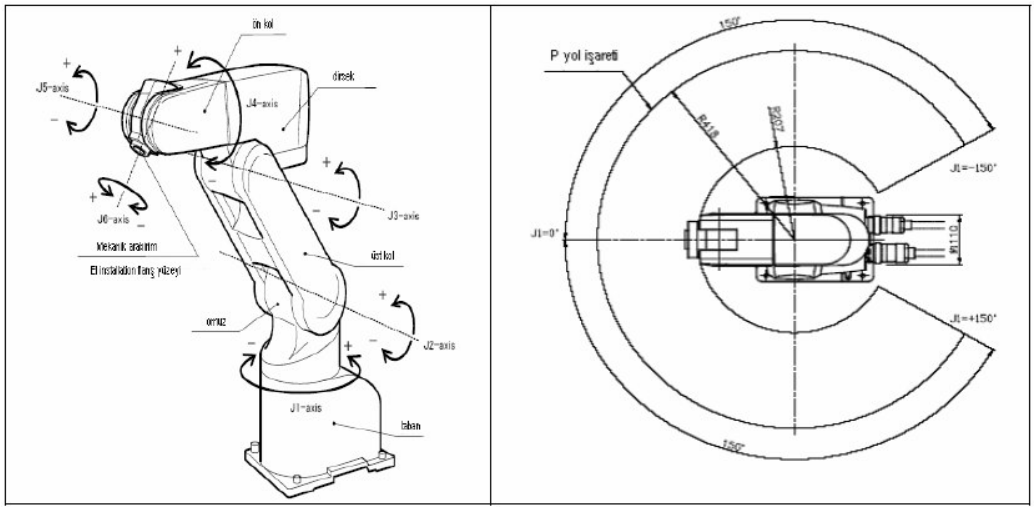
2.4.5. Mafsallı robot kolları

İnsan kolunun hareketlerini taklit etmeye en yakın robot koludur. Üretim sistemlerinde diğer kolların hareket kabiliyetlerinin sınırlı olmasından dolayı mafsal sayısı 5 veya 6 adet olan robot kollara ihtiyaç hasıl olmuştur. Bu tip robot kollarda her mafsal ayrı ayrı kontrol edilebilen servo motorlardan oluşmaktadır. Mafsallarda bulunan motorlar 12-24 V gerilim ile beslenmektedirler. Hareket esneklikleri en yüksek olan robot kollarıdır. Kol üzerinde bulunan her eklem şekil 2.13'de görüldüğü gibi X, Y, Z eksenlerinde üç boyutlu hareket yapabilmektedir. Çalışma alanı içerisinde tanımlanan bir noktaya en yakın yoldan ve kısa zamanda ulaşım imkânı tanımaktadır. Robotun hedef pozisyonlara yaklaşımı mafsal hareketi veya doğrusal X, Y, Z, koordinatları doğrultusunda hareket ederek gerçekleşmektedir.

Diğer robot türlerine göre karmaşık bir yapıya sahip olup, programlanması da diğerlerine göre zordur.

Her mafsal program içersinden sınırlandırılan belirlenmiş bir alan içersinde şekil 2.13'de görüldüğü gibi hareket edebilmektedir. Bu da robotun güvenli bir çalışma alan ve ortamı içerisinde bulunan diğer parçalara çarparak zarar vermesini önlemekte ve istenen noktaya, robotun daha hızlı ulaşmasını sağlamaktadır.

Robot kolun eksen sayısı tercihi yapılacak uygulamanın niteliğine göre yapılmalıdır. 3 eksenli robot kolu basit işlemlerin uygulanmasında yeterli olmakta iken daha karmaşık ve çok fonksiyonlu bir uygulama işleminde 3 eksenli robot kolu yeterli gelmemektedir. Uygulanan işlemler karmaşıklaştıkça mafsal sayısının arttırılması gerekmektedir. Robotun hareket serbestîyesini artırmak için mafsal sayısının arttırılması gerekmektedir.



Şekil 2.13. Robot kolu ve eksen hareketleri - Robotun X-X arası hareket alanı [8].

2.5. Robot Tahrik Sistemleri

Zamanımızdaki modern yapıya ulaşmış robotlarda tahrik sistemi olarak genellikle AC. Servo motorlar kullanılırken, sanayide kullanılan birçok robot kolunda, farklı tahrik sistemleri kullanılmaktadır [9].

2.5.1. Pnömatik

Maliyetleri oldukça düşük olan bu sistem birçok endüstriyel robotta tahrik sistemi olarak kullanılmaktadır. Ancak karmaşık bir kontrolleri vardır. Gelişmiş robotların tutucu kısımlarının tahrik edilmesinde yaygın olarak kullanılırken, basit yapılı robotlarda eksen hareketlerinin tahrikinde kullanılmaktadır. Hemen hemen bütün fabrikalarda basınçlı havanın bulunması kullanımını yaygınlaştırmaktadır.

2.5.2. Hidrolik

İlk zamanlarda çok kullanılan bir sistem olmasına rağmen bazı vazgeçilemeyen alanlar dışında yerini diğer yöntemlere bırakmaktadır. Yaygın olarak büyük güçlü robotlarda kullanılmaktadır. Çünkü hidrolikte elde edilen tahrik gücünü diğerlerinde elde etmemiz imkansızdır. Dezavantajları ise yavaş çalışmalarının yanında bulundukları ortama yağ sızdırmalarından dolayı kirletmeleridir.

2.5.3. Elektrik

- DC servo motorlar: Hız ve pozisyon kontrolünün geniş ölçülerde ve kolay yapılabildiği motorlar olduğu için kullanılmaktadırlar. Kurulum ve bakım masrafları diğerlerine göre çok daha fazladır. Bu dezavantajlarından dolayı yerini giderek diğer elektrik motorlarına bırakmaktadır.

- AC servo motor: Elektronik kontrolün gelişmesi ile birlikte bu motorlarda hız ve konum kontrolünde büyük ilerlemeler kaydedilmiştir. Bunun sonucu olarakta DC servo motorların yerini almaktadırlar. Maliyet açısından DC Servo motorlara göre daha ucuzdurlar, sessiz çalışma özellikleri vardır ve bakıma az ihtiyaç duyarlar.

- Adım motor : Maliyet olarak sürücü ünitelerinin ucuz olmasından dolayı tercih edilirler. Konum denetlemesinde daha hassas kontrol sağlarlar. Daha çok robot tutucularında kullanılmaktadırlar.

Tablo 2.2. Hidrolik, pnömatik ve elektrikli tahrik sistemlerinin karşılaştırılması [6].

	HİDROLİK SİSTEMLER	PNÖMATİK SİSTEMLER	ELEKTRİKİ SİSTEMLER
AVANTAJLARI	- Büyük yüklenme kapasitesine sahiptir. -Orta değerde süratlidir Yağın basıncı azalmadığından eklemler hareket ettrilmeden sabit tutulabilinir -Hassas kontrol edilebilme imkanına sahiptir.	-Hidrolik sistemlere göre ucuzdur. -Yüksek hız kabiliyeti sağlayabilir. -Akışkanlar ile çevre kirliliğine neden olmaz. -Laboratuar çalışmalarında kullanılabilinir.	- Hızlı ve hassastır. -Karmaşık kontrol tekniklerini uygulamaya elverişli bir yapıya sahiptir. - Kolay kullanımlı ve diğerlerine göre daha ucuzdur.
DEZAVANTAJLARI	- Hidrolik sistemler pahalıdır. - Gürültüye ve akışkanların sızması ile çevre kirliliğine neden olurlar. - Yüksek hızlı işleme döngülerine uygun değildirler.	- Havanın yağa göre sıkışabilirlik Özelliğinden dolayı basınç kaybına neden olur. -Gürültü kirliliği oluşabilir. - Hava yağa göre daha fazla sızma özelliğine sahiptir. -Sürekli bakım isteyen bir yapısı vardır.	- Dişli ve güç aktarma organlarına gereksinim gösterirler. - Güç sınırlaması vardır. - Oluşan elektrik arkı çeşitli sorunlar yaratabilir.

2.6. Robot Programlama

Robotların programlanmasında iki yöntem kullanılmaktadır. Bunlar öğretim yöntemi ve bir ara yüz programı ile programlamaktır. Gelişmiş robotlarda uygulanabilen öğretim yöntemi programcıya büyük kolaylık sağlarken, basit yapılı robotların programlanması ara yüz programı ile yapılmaktadır.

2.6.1. Öğretim yöntemi

Robot, bir öğretim kutusu sayesinde istenilen noktaya hareket ettirilip bu pozisyonlar hafızaya alınabilir. Ayrıca robotu programlamak için öğretim kutusu üzerindeki menüler kullanılabilir. bilgisayarın bulunmadığı yerlerde robotun bütün işlevleri ve programlanması öğretim kutusu kontrolü ile sağlanır. Özellikle pozisyonların belirlenmesi işleminde büyük kolaylık sağlamaktadır. Parça hangi konumdan alınacak veya hangi konuma bırakılacak ise bu konumlara gelip konumun

koordinatları direkt olarak hafızaya alınır ve program içersinde kullanılabilir. Ayrıca öğretim kutusu üzerinden robot koluna ait bütün kurma değerleri yapılmaktadır. Kullanıcı ile robot arasındaki iletişimi sağlayarak kontrolü kolaylaştırmaktadır.

2.6.2. Ara yüz programı ile programlama

Robotun gerçekleştireceği işlem robotun kendine ait olan bir ara yüz programı ile bilgisayar vasıtası ile programlanır ve robota yüklenir. Her firmanın kendi ürettiği bir ara yüz programı bulunmaktadır. Her marka robot için programlarda kullanılan komutlar farklı olmaktadır. Robotun çalışma alanı içindeki konum belirleme, hızı, hareket şekli, işlemleri ara yüz programından gerçekleştirilebilir.

BÖLÜM 3. ÖNCEKİ ÇALIŞMALAR (LİTERATÜR ÖZETİ)

Son yıllarda; robotlar ve mekatronik üzerine bilgi sunan birçok çalışma yapılmıştır. Bu çalışmalar kimi zaman robot kolu üzerine kimi zamanda mobil robotlar üzerinde olmuştur. Tez konusu robot kolu ile alakalı olduğu için robot kolu ile alakalı yapılmış bazı çalışmalar aşağıda sunulmuştur. Ayrıca yapılan tez çalışması bir kartezyen robot özelliği göstermektedir. Dolayısı ile bu robot tipi ile alakalı olan ve tez çalışmasındaki gibi adım motoru kullanılan bazı çalışmalar aşağıda sunulmuştur.

1949 tarihinden sonra dünya üzerinde özellikle Amerika, Avrupa ve Japonya'da pek çok firma robotik üzerine çalışmaya başlamıştır. Bu ilgi, gelişmeyi de beraberinde artırmıştır [11, 44, 45, 46, 47].

Takarobu ve arkadaşları uzaktan etkileşimli insan gibi hareket edebilen robot üzerine yaptıkları çalışmalarında; gerçek zamanlı gözün çalışması İtalya'dan robot kolunun çalışması ise Japonya'dan, sağlanmaktadır [13].

Chiaming Yen ve Wu-Jeng Li yayınladıkları makalelerinde pnömatik sistemler için web tabanlı bir eğitim öğretim sistemini anlatmaktadırlar. Açıklamış oldukları sistem; öğretim materyallerini, bir pnömatik laboratuar setini ve uzaktan veri toplama modüllerini içermektedir. HTML formatındaki bu öğretim malzemeleri yazılı hareketsiz ve hareketli resimler, doküman, bilgisayar destekli tasarım araçları ve simülasyon programlarından oluşmaktadır. Sistemin veri toplama modülü bilgisayarın giriş çıkış birimi vasıtasıyla gerçek deney cihazlarına bağlanabilmektedir. Böylece pnömatik öğretimini daha da iyileştirmek için simülasyon sonuçlarının uygulanması ve doğrulanmasını mümkün kılmıştır. Bu özelliği bakımından sistemin pnömatik laboratuar uygulamalarında ve pnömatik öğretiminde bilgisayar destekli sıralı kontrol tasarımında oldukça yardımcı olacağı ileri sürülmektedir. Ayrıca bu çalışma bilgisayar tabanlı sıralı kontrol, web tabanlı

sıralı kontrolü, internet vasıtasıyla uzaktan izleme ve PLC de içermektedir. Sistemin bütün kullanıcılar için diğer bir özelliği ise bir veri tabanı sunucusunu birlikte kullanabilmeleridir. Bununla birlikte bütün kullanıcılar aynı devreyi birlikte tasarlayabilmektedirler [14].

Gilbert Reyne klasik sistemlere göre deneylerde uzaktan kontrolün önemini ve avantajlarını, optik elektro mekanik sistemlere uygulanan elektro manyetik hareket sistemleri üzerine yaptıkları çalışmada anlatmışlardır. Bu amaçla üç farklı manyetik, optik, mikro eylemci ve sistem tanıtılmaktadır [15].

J. N. Liou, M. Jamshidi ve G. P. Star endüstriyel robotlarda adaptiv kuvvet kontrolü yapmayı amaçlamışlardır. Çalışmalarında otomatik montajda takım-iş parçası etkileşimini uygun bir kuvvet aralığında tutmayı hedeflemişlerdir [16].

Hong Daehie, Steven A. Velinsky ve Kazuo Yamazaki, uygulamalarında otobanların yapım ve bakım onarım işlerinde kullanılan bir mobil robotu ve bu robotun kontrol sistemini açıklamaktadırlar. Bu robotta Servo sistem kontrolü kullanılarak optimum kontrol gerçekleştirilmiştir [17].

İzmir Dokuz Eylül Üniversitesi'nde bir çalışmada internet üzerinden erişilebilecek mikro denetleyici tabanlı bir elektronik kartın tasarlanması ve gerçeklenmesi yapılmıştır. Uygulama olarak, internet üzerinden robot kolu kontrolü başarıyla gerçekleştirilmiştir [18].

Aynı üniversitede yayınlanan bir makalede değişik nesneleri tanımaya yönelik görüntü işleme sistemi ile bu nesneleri görüntü destekli ayırmak için kullanılan robot manipülatörü ile ilgili çalışmalar sunulmuştur. DC motorların üç boyutlu uzayda verilen bir yörüngeyi takip edebilmesi için C++ ile özel bir mafsal kontrol algoritması yazılmıştır. Görüntü tanımayla bağlantılı olarak robot kolun senkron çalışması ve değişik yolları takip edebilme yeteneği, 24 tanımlı nesne için test edilmiştir. Sonuç olarak, 5 ile 10 mm arasında bir kesinlik değeri ile yörünge takip edilebilmiş ve %95'lik bir nesne tanıma sonucuna ulaşılmıştır [19].

Diğer bir makalede, sabit bir bilgisayar tarafından RF modemler kullanılarak kontrol edilebilen bilgisayar destekli bir non-holonomik araç sunulmuştur. Mobil robot, programlanmış bir yolu ya da operatörce istenilen yolu takip edebilmektedir [20].

Sakarya üniversitesinde yapılan bir yüksek lisans projesinde ise bir labirent robotu tasarımı ve uygulaması yapılmıştır. Bu robotun yapısında 2 adet adım motor kullanılmıştır. Adım motorun tercih edilme sebebi daha kolay kontrol edilmesi olmuştur. Robot, 2 adet adım motora bağlı 2 adet tekerden oluştuğu için denge sorunu yaşanmış ve bu sorun ön ve arkaya bilyeler konarak giderilmiştir [21].

Aynı üniversitede yapılan diğer bir yüksek lisans projesinde mikro denetleyici kontrollü algılamalı örümcek robot tasarımı gerçekleştirilmiştir. Bu robot tasarımının çalışmasında da enerji problemi ön plana çıkmıştır. Robottaki 12 adet servo motor yaklaşık olarak 3 Amper akım çekmektedir. 4.5 Amperlik batarya, problemi kısmen çözse de çok fazla ağır olmasından dolayı denge problemi oluşturmuştur. Dış gövdede kullanılan epoxi malzemesi çok sert ve işlenmesi çok zor bir malzemedir. İstenilen şekle getirilebilmesi için elmas bıçakların kullanılması gerekmektedir. Bunun yanında çok sağlam bir malzemedir [4].

Başka bir yüksek lisans projesinde mekatronik sistemlerde internet tabanlı kontrol ve kartezyen robot üzerinde bir uygulama gerçekleştirilmiştir. Kontrol birimi olarak bir sunucu bilgisayara bağlı PLC kullanılmıştır. Kartezyen robota 3 ayrı renkte olan lastik topların dokuz ayrı noktaya, renklerin yerlerini de değiştirerek taşınması işlevini gerçekleştirecek bir program yüklenmiştir. Topların gideceği noktaların koordinatları programa girilmiştir ve istenen işlev gerçekleştirilmiştir [22].

Sakarya Üniversitesi'nde yapılan bir doktora projesinde, Uzun Menzilli Öngörülü Kontrol Algoritmaları sınıfına ait olan Genelleştirilmiş Öngörülü Kontrol (Generalized Predictive Control – GPC) ve Newton-Raphson uyarlamalı Yapay Sinir Ağlı Genelleştirilmiş Öngörülü Kontrol (Neural Generalized Predictive Control – NGPC) algoritmalarının her birisi Tek Giriş Tek Çıkış (Single Input Single Output – SISO) ve Çok Giriş Çok Çıkış (Multiple Inputs Multiple Outputs – MIMO) olmak üzere iki şekilde tasarlanarak üç eklemli bir robot koluna eklem esaslı yörünge

kontrolü için uygulanmıştır. Yörünge planlaması sinüzoidal ve kübik yörünge esaslarına göre belirlenmiştir. Sürtünme, bozucu etkileri, robot kolunun 10 kg'lik bir yük taşıması ve taşınan yükün taşıma esnasında düşmesi durumları da ayrıca ilave edilmiştir [23].

Başka bir yüksek lisans projesinde; yapılan simülasyon çalışmaları sonucunda altı serbestlik dereceli PUMA 560 robotunun önceden hesaplanmış dinamik parametreler altında; PD kontrol algoritması kullanılarak, hesaplanmış moment yöntemi metodu ile yörünge kontrolü yapılmıştır. Zamana bağlı olarak eklemlerin konum ve hız eğrileri elde edilmiştir [24].

Başka bir çalışmada üç eklemli bir SCARA robotu ele alınmış ve dinamiği yapay sinir ağları (YSA) ile modellenmiştir. Sonuç olarak YSA hedeflenen çıkışları müsaade edilebilecek çok küçük sapmalarla başarılı bir şekilde yakalamış ve iyi bir performans sergileyerek SCARA robotun modellenmesi problemine oldukça iyi cevap vererek çözüm üretebilmiştir [25].

Bir doktora tezinde ise üç eklemli bir robotik manipülatörün, görmeye dayalı kontrolü YSA kullanılarak yapılmıştır. Simülasyon programı kullanılmıştır [26].

Başka bir yüksek lisans projesinde bir labirent robotu tasarımı ve gerçekleştirilmesi yapılmıştır. Bu robotun da yapısında 2 adet adım motor kullanılmıştır. Robot 2 adet adım motora bağlı iki adet tekerden oluştuğu için denge sorunu yaşanmış ve bu sorun ön ve arkaya bilyeler konarak giderilmiştir. Proje 2003 de yapılan bir proje ile benzerlik göstermektedir [27].

Elektrik - Elektronik - Bilgisayar Mühendisliği 10. Ulusal Kongresi' nde yayınlanan bir bildiride bilgisayarlarla haberleşerek x-y düzleminde çizim yapabilen bir mekatronik sistem tasarımı sunulmuştur. Z ekseninde hareketi sağlayan sonlandırıcı eleman (kalem) ile sistem üç eksende hareket etmektedir. Üç eksende hareketiyle sistem kartezyen robot özelliği taşımaktadır. Bu uygulama üç kısım halinde incelenebilir. İki eksen, çizim yapılacak zemin ile rölelerden oluşan ve bir anlamda sistemin iskeletini oluşturan makine kısmı; sürücü devre ve mikro denetleyicinin

makine kısmıyla bağlantısını içeren elektronik kısmı; kullanıcının isteğine uygun (mekanik düzenin izin verdiği ölçüler dâhilinde) çizime ait koordinatları girmesine olanak sağlayan ve girilen koordinatları yorumlayarak iki eksendeki motorlara ve rölelere elektronik kart aracılığıyla gerekli sinyali gönderen kısmıdır [28].

Elektrik - Elektronik - Bilgisayar Mühendisliği 10. Ulusal Kongresi'nde yayınlanan diğer bir bildiride Festo tarafından üretilen hassas konumlama kontrolörü SPC200 yardımı ile geliştirilmiş pnömatik tahrikli kartezyen robot uygulaması açıklanmıştır [29].

Bakery Holdings LLC (Richmond, VA)'da her eksenin bağımsız olarak kontrol edildiği ve robot merkez hattı boyunca kuvvet uygulandığı; iki eksen, iki sürücü mekanizması ve iki kayıştan oluşan bir kartezyen robot tasarlanmıştır. Bu benzersiz dizayn çok eksenli hareket kontrolcüsünün gereksiz oluşunun söylenmesinden beri düşük maliyetli bir kontrol sistemi kullanımına izin vermektedir [30].

Queen's University'de yapılan bir tez çalışmasında bir kartezyen pnömatik robotun sürekli kayan kipli denetimi (SMC) incelenmiştir. SMC dizaynı ile alakalı genel literatür bilgileri verilmiş ve doğrusal ve doğrusal olmayan bir pnömatik robot sunulmuştur. Açık kapalı döngü testleri yürütülmüştür [31].

Massachusetts Teknoloji Enstitüsü Makine Mühendisliği'nde gerçekleştirilen bir doktora tezinde bir kartezyen robotun dinamik yapısal modelinde, işlem süresince konfigürasyon değişimlerinde meydana gelen sistemin titreşim tepki karakteristikleri öngörüleri için metotlar geliştirilmiştir [32].

Bir makalede, beş eksenli bir edubot robotta, ters kinematik hesaplamalar ve yörünge planlaması yapılmıştır. Ters kinematik probleminde, robotun uç noktasının gideceği yerin koordinatları (x, y, z) ve robot elinin başlangıç pozisyonuna göre açısı (φ) girdi olarak verilmiş ve eklem açılarının alabileceği değerler (θ_1, θ_2, θ_3, θ_4) hesaplanmıştır. Ayrıca bu çalışmada, endüstriyel robotların en önemli sorunlarından birisi olan "Yörünge Planlaması"na 5. dereceden zaman polinomları ile çözüm getirilmiştir [33].

Elektrik - Elektronik - Bilgisayar Mühendisliği 10. Ulusal Kongresi'nde yayınlanan bir çalışmada 3 serbestlik dereceli bir kartezyen robot kolunun donanımı değiştirilerek açık mimari bir yapıya dönüştürülmüştür. Kullanılan yazılım sayesinde, sistem bir defa kurulduktan sonra, başka araştırmacıların aynı düzen üzerinde çalışma yapabilmeleri için mevcut yazılımın öğrenilme sürecinin çok kısa olması sağlanmıştır [34].

Gazi Üniversitesi'nde bir yüksek lisans çalışmasında renge göre (kırmızı, yeşil, mavi) malzeme taşıyan robot kolu tasarımı ve uygulaması yapılmıştır. Bu çalışmada kullanılan kontrol birimi PIC mikro denetleyicisidir. Bu çalışmada gereken program bir defa kontrol birimine yüklenmekte ve daha sonra robot bu programa göre hareket etmektedir. Sonradan harici bir müdahale bulunmamaktadır [6].

Hakkâri Meslek Yüksek Okulu'nda yapılan bir çalışmada bilgisayar kontrollü bir çizim makinesi tasarlanmıştır. Bu çalışmada da tez konusu olan çalışmadaki gibi adım motorlar kullanılmıştır. Adım motorların sürülmesi ise yine aynı ULN2003A entegreleriyle gerçekleştirilmiştir. Kontrol ise bilgisayarın paralel portlarıyla gerçekleştirilmektedir [35].

BÖLÜM 4. SİSTEM KONTROLÜ

4.1. Mikroişlemci Nedir?

80386, 80486, PentiumII, PentiumIV isimleri günümüzde kullanılan bilgisayarların özelliklerinden bahsedilirken duyduğumuz birer mikroişlemci isimleridirler [36]. Bilgisayar programlarının yapmak istediği tüm işlemleri mikroişlemciler yerine getirebildiği için, çoğu zaman merkezi işlem birimi (CPU - Central Processing Unit) olarak da adlandırılırar. Şahsi bilgisayarlarda kullanıldığı gibi, bilgisayarla kontrol edilen üretim hatlarında sanayi tezgâhlarında, ve ev cihazlarında da kullanılmaktadırlar [37].

Bir mikroişlemci işlevini yerine getirebilmesi için aşağıdaki yardımcı birimlere ihtiyaç duyar:

- Giriş (Input) birimi
- Çıkış (Output) birimi
- Bellek (Memory) birimi [36].

CPU dışında, mikroişlemci tarafından ihtiyaç duyulan bu birimler, bilgisayarın ana kartı üzerinde bir yerde elektronik elemanlardan veya farklı entegrelerden oluşur. Birimler arasındaki bağlantı ise, kontrol yolu (control bus), adres yolu (address bus) veya veri yolu (data bus) olarak adlandırılan iletim hatları yardımıyla sağlanır [37].

4.2. Mikro Denetleyici Nedir?

Bir mikroişlemci ile birlikte kullanılma ihtiyacı bulunan giriş, çıkış ve bellek birimlerinin tek bir entegre içerisinde üretilmesi ile oluşan eleman, ‘mikro denetleyici’ (Microcontroller) olarak isimlendirilir [36]. Bilgisayar teknolojisi

gerektiren uygulamalarda kullanılmak üzere tasarlanmış olan mikro denetleyiciler, günümüzde kameralarda, cep telefonlarında, fotokopi, radyo, TV, otomobillerde, faks-modem cihazlarında ve bazı oyuncaklar gibi sayılamayacak kadar pek çok ev ve ofis cihazlarında kullanılmaktadır [36].

4.3. Neden Mikro Denetleyici?

Mikroişlemci ile kontrol edilecek bir sistemi kurmak için giriş-çıkış ve bellek birimlerine ihtiyaç duyulmakta ve bu birimler arasındaki iletişim için 'yol' (bus) olarak isimlendirilen hatlar gerekmektedir. Ayrıca bu birimleri yerleştirmek için baskılı devreye gereksinim duyulmaktadır [36].

Yukarıda sayılan birimleri içeren tek bir entegre (Mikro denetleyici) ve bir de devre kartı kullanmak, mikro denetleyici ile kontrol edilecek sistemde yeterlidir. Maliyet olarak, tek bir entegre kullanarak elektronik devreler üretmenin daha avantajlı olacağı bellidir. Programlama ve kullanım kolaylığı ikinci bir avantajıdır.

Yukarda açıklanan sebepler, bilgisayar kontrollü elektronik uygulamalarda mikro denetleyici kullanmaya olan ilginin artmasının haklılığını ortaya koymaktadır.

4.4. Mikro Denetleyicilerin Genel Özellikleri

Her mikroişlemci (CPU) firmasının ürettiği birkaç model mikro denetleyici bulunmakta ve bu denetleyicilerin mimarileri arasında farklılıklar olmasına rağmen genel hatları ile aynı görevleri yerine getirebilmektediler. Üretilen her mikro denetleyici entegresine bir isim ve özelliklerini birbirinden ayıran parça numarası verilmektedir.

Örneğin; Microchip firması mikrodenetleyicilerine PIC adını vermekte iken, parça numarası olarak da 12C508, 16C84, 16F84, 16F877, vb. gibi kodlamalar kullanmaktadır. Intel ise ürettiği mikro denetleyicilere MCS-51 ailesi adını vermekte ve genel olarak bu adla anılan mikro denetleyici ailesinde farklı özellikleri bulunan ürünleri birbirinden ayırt etmek için parça numarası olarak da 8031AH, 8051AH, 8751AHP, 8052AH, 80C51FA, vb. kodlamalar kullanmaktadır [38].

Bir çalışmaya başlamadan önce hangi üreticinin ürününün kullanılacağına, daha sonra da hangi denetleyicinin kullanılacağına karar vermek gerekmektedir. Bunun için mikro denetleyici gerektiren uygulamada ne gibi özelliklerin olması gerektiğinin önceden belirlenmesi gerekmektedir. Aşağıda verilen özellikler mikro denetleyicilerde bulunan genel özelliklerdir:

- Programlanabilir dijital paralel giriş/çıkış
- Programlanabilir analog giriş/çıkış
- Seri giriş/çıkış (senkron, asenkron ve cihaz denetimi gibi)
- Motor veya servo kontrol için pals sinyali çıkışı
- Harici giriş vasıtasıyla kesme
- Zamanlayıcı (Timer) vasıtasıyla kesme
- Harici bellek arabirimi
- Harici yol arabirimi (PC ISA gibi)
- Dahili bellek tipi seçenekleri (ROM, EPROM PROM ve EEPROM)
- Dahili RAM seçeneği
- Kayan nokta hesaplaması

Kontrol birimi olarak kullanılabilecek birçok birim vardır. Bunlardan bazıları PLC, PIC ya da 8051 gibi mikro denetleyicilerdir. Aşağıda bu kontrol birimleriyle alakalı kısa bilgiler verilmiştir.

4.5. PLC

Programlanabilir Lojik Kontrolörler (PLC) otomasyon devrelerinde zaman röleleri, yardımcı röleler, sayıcılar gibi kumanda elemanlarının yerine kullanılan mikroişlemci temelli cihazlardır. Bu cihazlarda; sıralama, zamanlama, sayma ve her türlü kombinasyonel ve ardışık lojik işlemler yazılımla gerçekleştirilir. Bu nedenle karmaşık otomasyon problemlerini daha çabuk ve güvenli bir şekilde çözmek mümkündür.

Endüstriyel otomasyon devrelerinde programlanabilir kontrolörlerin tercih edilmelerinin nedenleri şu şekilde sıralanabilir:

- Kumanda devresi yazılımla sağlandığından, kumanda devresini tasarlamak kontaklı (röleli) bir devrenin tasarımından daha kolaydır.
- Bütün kumanda fonksiyonları yazılımla gerçekleştirildiğinden, farklı bir uygulama için adaptasyon kolaydır.
- Kumanda devrelerine göre çok az yer kaplar.
- Güvenilirliği yüksek, bakımı kolaydır.
- Bilgisayarlarla ve diğer kontrolörlerle haberleşme olanağı vardır
- Arıza ihtimali düşüktür
- Kötü çevre koşullarında, özellikle tozlu ortamlarda, röleli kumanda devrelerine göre daha güvenilirdir [39].

4.6. PIC Mikro Denetleyicisinin Genel Özellikleri

Mikro denetleyicili bir sistemle denetimi gerektiren bir uygulamayı geliştirirken ilk olarak seçilecek mikro denetleyicinin tüm istekleri/ihtiyaçları yerine getirip getirmeyeceğine, daha sonra da maliyetinin düşüklüğüne bakmak gerekir [36].

Ayrıca, yapılacak uygulamanın devresini kurmadan önce seçilen mikro denetleyicinin desteklediği bir yazılım üzerinde simülasyon yapılıp, devrenin çalışma şekli hakkında bilgi edinilmelidir.

Mikro denetleyicide bulunması gerekli genel özellikleri göz önüne aldığımızda Microchip firmasının ürettiği PIC'lerin tercih edilmesinin birçok avantaja sahip olduğu görülür. PIC mikro denetleyicisinin tercih edilme sebeplerini aşağıdaki gibi sıralayabiliriz:

- Fiyatının oldukça ucuz olması
- Lojik uygulamalarının hızlı olması
- 8 bitlik mikro kontrolör olması ve bellek ve veri için ayrı yerleşik yolların kullanılması
- Veri ve belleğe hızlı olarak erişimin sağlanması

- PIC'e göre diğer mikro kontrolörlerde veri ve programı taşıyan bir tek yol bulunması, dolayısıyla PIC'in bu özelliği ile diğer mikro kontrolörlerden iki kat daha hızlı olması.
- Herhangi bir ek bellek veya giriş/çıkış elemanı gerektirmeden sadece 2 kondansatör ve bir direnç ile çalışabilmeleri
- Yüksek frekanslarda çalışabilme özelliği
- Standby durumunda çok düşük akım çekmesi
- Kesme kapasitesi ve 14 bit komut işleme hafızası
- Kod sıkıştırma özelliği ile aynı anda birçok işlem gerçekleştirebilmesi [40].

PIC, adını İngilizce'deki "Peripheral Interface Controller" (Çevresel Üniteler Denetleyici Arabirim) cümlesindeki kelimelerin baş harflerinden almış olan bir mikro denetleyicidir. PIC gerçekten de çevresel üniteler adı verilen lamba, motor, röle, ısı ve ışık algılayıcıyı gibi G/Ç elemanların denetimini çok hızlı olarak yapabilecek şekilde tasarlanmış bir entegredir. RISC (Reduced Instruction Set Computer) mimarisi adı verilen bir yöntem kullanılarak üretildiklerinden bir PIC'i programlamak için kullanılacak olan komutlar oldukça basit ve sayı olarak da azdır. 1980'lerin başından itibaren uygulanan bir tasarım yöntemi olan RISC mimarisindeki temel düşünce, daha basit ve daha az komut kullanılmasıdır. Örneğin PIC16F84 mikro denetleyicisi toplam 35 komut kullanılarak programlanabilmektedir.

4.7. 8051 Mikro Denetleyici

8051 Mikrodenetleyici ilk olarak Intel tarafından 1980 yılında üretilmiştir. Eski bir ürün olmasına rağmen, hem kendisi, hem de yapısı temel alınarak üretilmiş diğer işlemciler bugün geniş bir kullanım alanına sahiptir.

İlk olarak Intel tarafından üretilmesine rağmen bugün aralarında Intel'e ek olarak Atmel, Dallas Semiconductors ve Philips'in de bulunduğu onlarca üretici firma tarafından da üretilmektedir; kısaca çok kaynaklı bir mikro denetleyicidir.

Zamanla üzerindeki geliştirme çalışmalarının sonucu tek bir mikro denetleyici olmaktan çıkıp bir 8051 ailesi olarak anılan bir mikro denetleyici ailesi haline gelmiştir. Bu aileye 8031, 8032, 8051, 8052, 80151, 80251 ve XA serileri dahildir. Yukarıda da bahsedildiği gibi çok kaynaklı bir denetleyici olmasından ötürü, ona kaynak sağlayan firmalar tarafından bu aile elemanları da üretilmektedir [41].

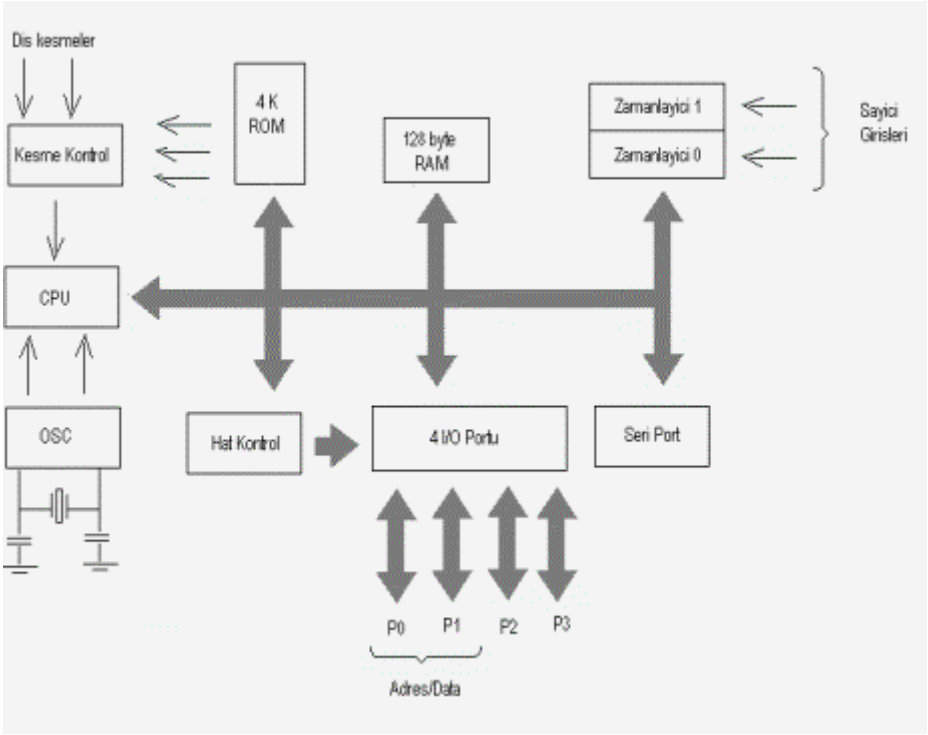
Yukarıda kontrol ünitesi olarak kullanılabilecek birimler hakkında kısaca bilgi verilmiştir. Bu bilgilere göre her birimde avantajının yüksek olduğu noktalar vardır. Fakat bu çalışma yapılırken sistemin maliyetinin en düşük seviyede tutulması, daha önceden tecrübe sahibi olunması ve elde mevcut bulunmasından dolayı 8051 mikro denetleyicisi tercih edilmiştir.

4.7.1. 8051 ailesinin temel özellikleri

8051 Intel firması tarafından üretilen MCS-51 ailesinin ilk mikro denetleyicisi olup MCS-51 ürünlerinin temel çekirdeğidir. 8051 çekirdeğinin ana özellikleri şunlardır:

- Kontrol uygulamalarına yönelik 8-bit CPU
- Yoğun boolean işlemeri yapabilme (tek-bit lojik işlemler) özelliği
- 64K Program Hafıza (Program Memory) adres alanı · 64K Veri Hafıza (Data Memory) adres alanı
- 4K tümdevre-üzeri (on-chip) program hafıza
- 128 byte tümdevre-üzeri veri RAM
- 32 tane iki yönlü adreslenebilir I/O hatları
- 2 tane 16-bit Zamanlayıcı/Sayıcı (Timer/Counter)
- Full duplex UART (Uıiversal Asynchronous Receiver Transmitter)
- İki öncelik seviyesine sahip 6-kaynak / 5-vektörlü kesme donanım yapısı
- ACC ve B saklayıcılarına ek olarak 8 adet “R” saklayıcıları (R0, R1, ..., R7).
- “R” saklayıcılarını üzerinde bulunduran 4 adet saklayıcı kümesi
- 8051 işlemcisinin çalışmasını kontrol eden Özel Fonksiyon Saklayıcıları (SFRs).
- 16 bitlik veriye erişimi sağlayan işaretçi (DPTR)

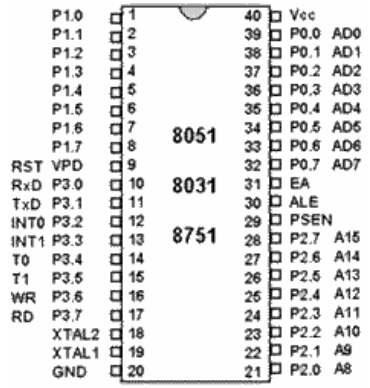
- Program Sayacı (PC), Yığın Göstergesi (SP)
- Üzerinde var olan iç belleklere ek olarak dış bellekler ekleyebilme [4].
- 1 makine çevrimlik bir komutu 100 ns. gibi bir sürede icra edebilme



Şekil 4.1. 4K ROM ve 128 byte iç hafızalı bir 8051 entegresinin blok yapısı

4.7.2. 8051 mikrodenetleyici mimarisi

8051 40 bacaklı bir entegredir. Dört giriş/çıkış birimi için 32 tane bacağa gereksinim vardır. Bu sebeple bacaklardan çoğu birden fazla fonksiyonu gerçekleştirebilmek üzere tasarlanmıştır. Aşağıda şekilde 8051 entegrenin bacak bağlantıları görülmektedir.



Şekil 4.2. 8051 serisi entegrelerin bacak bağlantıları

Giriş/çıkış 0 iki farklı amaç için kullanılabilecek bir 8 bit giriş/çıkış ünitesidir. Bu ünite ya iki yönlü (bidirectional) giriş/çıkış birimi olarak yada düşük değerli adres (AD0-AD7)/data yolu olarak kullanılabilir. Bir giriş/çıkış birimi olarak her bir bacak düşük (LOW) durumda iken, 8 tane LS TTL devrenin verdiği akımı üzerinden geçirebilir ve yüksek (HIGH) konumda ise dışarıdaki entegrelere 3.2 mA akım sürebilir. Adres ve data modunda, (AD0-AD7) hatları dış hafıza elemanlarına ulaşmak amacıyla kullanılır. ALE hattı kontrolü kullanılarak, AD0-AD7 hatları, adres veya data yolu olarak belirlenir. Yani AD0-AD7 hatları, ALE sinyali yardımıyla, A0-A7 ve D0-D7 hatları olarak ayrıştırılır (demultipleksing) [41].

4.7.3. 8051 avantajları

- Geniş Yelpaze: Pek çok üretici firma, orijinal 8051'e çeşitli ek özellikler katarak türev ürünler geliştirmiştir. Bütün bu ürünler için çeşitli yazılım ve uygulama geliştirme donanımları üreten firmaların da katkılarıyla 8051 bir endüstri standardı haline gelmiştir.

- Uyumluluk: 8051 türevleri, orijinal 8051 ile yazılım uyumlu olup yazılmış olan standart bir program başka bir firmanın ürünü olan işlemci üzerinde de çalışabilmektedir. Bu uyumluluğun sağladığı kolaylık ve esneklik, program geliştirme araçlarında, eğitimde ve yazılım desteğinde de bulunmaktadır.

- Hızlı ve Güçlü: 8051 çekirdek mimarisi kontrol uygulamaları için gayet uygun olup hızlı ve güçlüdür.

- Popüler: 8051 kullanıcıları için birçok kitap, teknik dokümanlar, yazılım ve donanım gereçleri, pek çok İnternet Web Sayfası mevcuttur. Ürün kolay bir şekilde bulunmakta ve desteklenmektedir.

- Sürekli Geliştirilme: 1980'lerden bugüne silikon ve tasarım olarak sürekli geliştirilen 8051'lerin hızları, işlem güçleri, on-chip çevre birimleri sayısı ve çeşitliliği artmıştır. Örneğin Dallas firması tarafından üretilen bir ürün 40 MHz'lik kristal ile 120 MHz'lik performansa sahip olma gibi özelliklere de sahiptir. Bu performans ile 1 makine çevrimlik bir komutu 100 ns. gibi bir sürede icra eder ki bu da birkaç yıl öncesine kadar sadece Sayısal İşaret İşleyicilerin yapabildiği bir işlem idi.

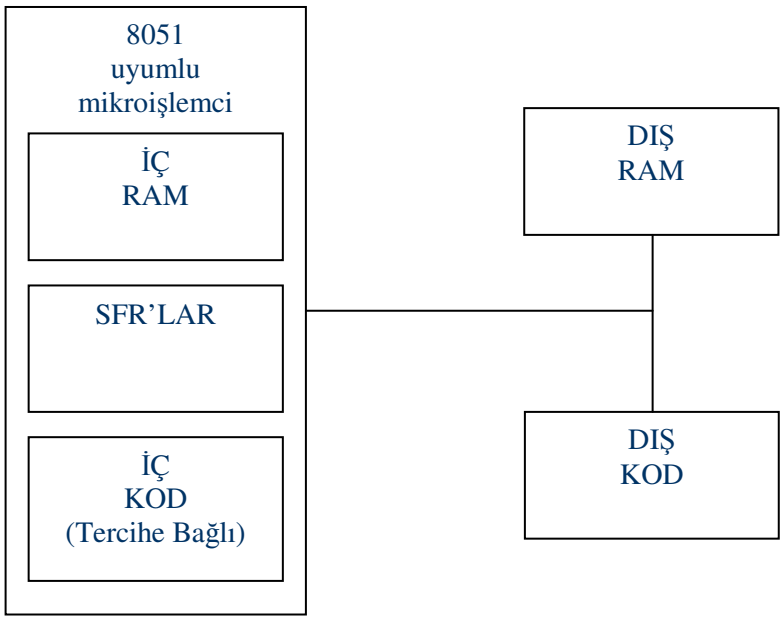
4.7.4. Eğitimde 8051

8051 Ailesi ürünler sıralanan avantajlara sahip olmalarının yanı sıra,

- Mikro kontrolörlerin temeli sayılmaları,
- Endüstride sıklıkla kullanılmaları,

sebebiyle eğitim amaçlı olarak da yaygın bir şekilde kullanılmaktadır.

4.7.5. Bellek yapısı



Şekil 4.3. 8051 bellek yapısı

4.7.6. Program akışı

8051 ilk çalışmaya başladığında Program Sayacı (PC) ilk değer olarak 0000h değerini almaktadır. Yani çalışmaya bu adresten başlayacaktır ve PC'nin değerinin değişmesine sebep olacak herhangi bir komut yürütülmediği sürece de sıralı bir şekilde çalışmaya devam edecektir. Bu değişimi sağlayan komutlar dallanma, çağırma ve kesme komutlarıdır.

Dallanma komutları bir koşul yerine geldiğinde gerçekleşebileceği gibi doğrudan başka bir kod bölümüne atlama şeklinde de gerçekleşebilir. Bu durumda programın çalışması oradan devam edecektir ve normal çalışma sürecinde geri dönülmeyecektir. Ayrıca alt programlara erişim alt programları çağırma şeklinde de gerçekleştirilebilir. Dallanmalardan farklı olarak alt programlar çalışmaları bittikten sonra RET komutu ile çalışma sürecini kendilerini çağıran programa iade ederler.

Kesmelerde ise durum daha farklıdır. Kesmede işlenen kesme hizmet programı, bir alt program olmasına rağmen programcıdan bağımsız olarak fiziksel bir koşul

gerçekleştiğinde mikroişlemci otomatik olarak bu alt programı işlemeye başlar. Alt programlarda olduğu gibi kesilmenin gerçekleştiği ana programa RET' ten ayrı olarak RETI komutu ile dönerler.

4.7.7. Sayaçların kullanımı

Sayaçlar, zaman hesaplaması veya olay sayımı işlemlerinde kullanılırlar. Mikro kontrolör üzerinde bulunan bir kristal ile işaret üretmektedir. Bu kristalin ürettiği her 12 işaret bir makine çevirimine denk gelmektedir. Her makine çeviriminde sayaç değerini 1 arttırmak imkân dahilinde olduğundan doğru değerler sayaçlara yüklendiklerinde zaman hesaplaması yapılabilmektedir. Ayrıca mikro kontrolörün ilgili girişleri kullanılarak olay sayma ve gözleme işlemleri de yapılabilir.

Her sayacın kendine ait 8 bitlik kısımlarını ifade eden THx (Yüksek anlamlı byte) ve TLx (Düşük anlamlı byte) saklayıcıları vardır. Bu saklayıcılara sıra ile değer yüklemekle sayaçlara saymaları gereken değerler yüklenir.

4.7.8. Kesmeler

Kesme yeteneği bize sıralı olan program akışının herhangi bir durumda iken, o andaki çalışmasını durdurup başka bir çalışma programı izlemesine ve o da bitince kaldığı yerden devam etmesine imkân verir. Bu çalışma programı Kesme Hizmet Programı olarak adlandırılır.

Ana programın durdurulmasına sebep olan olaylar sayacın taşması, seri porttan bir karakter okunması veya bir kaç dış olay olabilir. Bu olaylardan birisi gerçekleştiğinde 8051 ana programının çalışmasını geçici olarak durdurup bu hizmet programını çalıştırmaya koyulur. Bu hizmet programı, ana programın kesilmesine sebep olan olayın tipine göre değişiklik gösterir. Hizmet programının çalışması bittikten sonra kontrol tekrar ana programa geçer ve ana program kaldığı yerden devam eder.

8051'de ana programın kesilmesine yol açan bir kaç olay vardır. Bunlar :

- Sayaç 0'ın taşması, TF0
- Sayaç 1'in taşması TF1
- Seri bir karakterin alınması/yollanması RI/TI
- Dış kesme 0 IE0
- Dış kesme 1 IE1

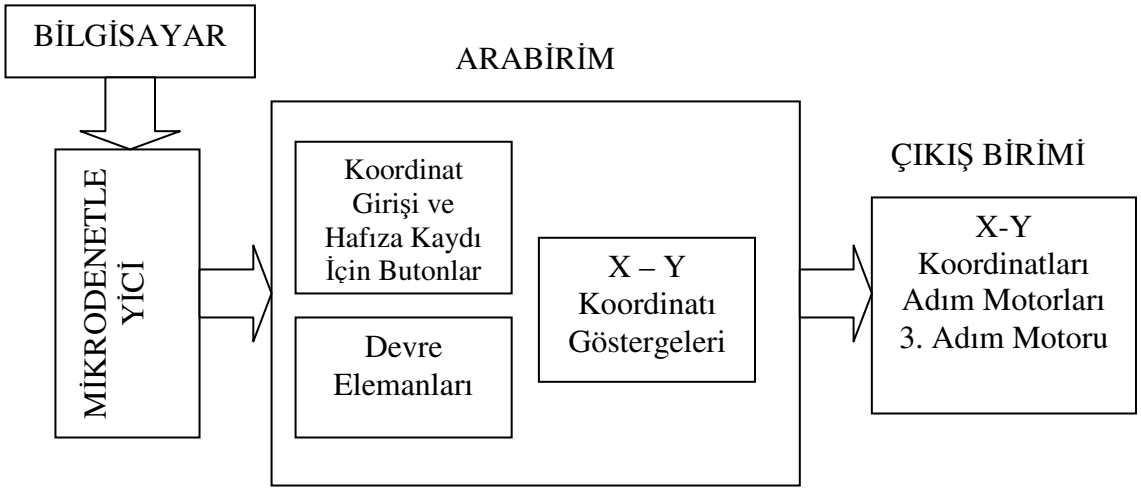
Bu olaylar gerçekleştiğinde hizmete sokulacak program ise kesme sebebine göre belleğin farklı bölgelerinde yer almaktadırlar. 8051 bu kesme tipine göre belleğin o ilgili yerine dallanıp çalışmasını oradan devam ettirir.

Ayrıca, kesme geldiğinde; kesme hizmet programı yürütülürken başka bir yerden kesme isteğinde bu kesmenin kabul edilip edilmeyeceğini kesmelerin öncelikleri belirleyecektir [42].

BÖLÜM 5. SİSTEMİN TASARLANMASI

5.1. Tasarım Süreci

Yapılacak proje, iki boyutlu bir koordinat düzleminde dışarıdan dijital olarak, el ile girilen 4 adet X ve Y koordinatının, mikro denetleyici vasıtasıyla kıyaslanarak, küçükten büyüğe doğru sıralanıp, mekanik bir sistem vasıtasıyla sırayla koordinatlara nokta koyulmasını amaçlamaktadır.

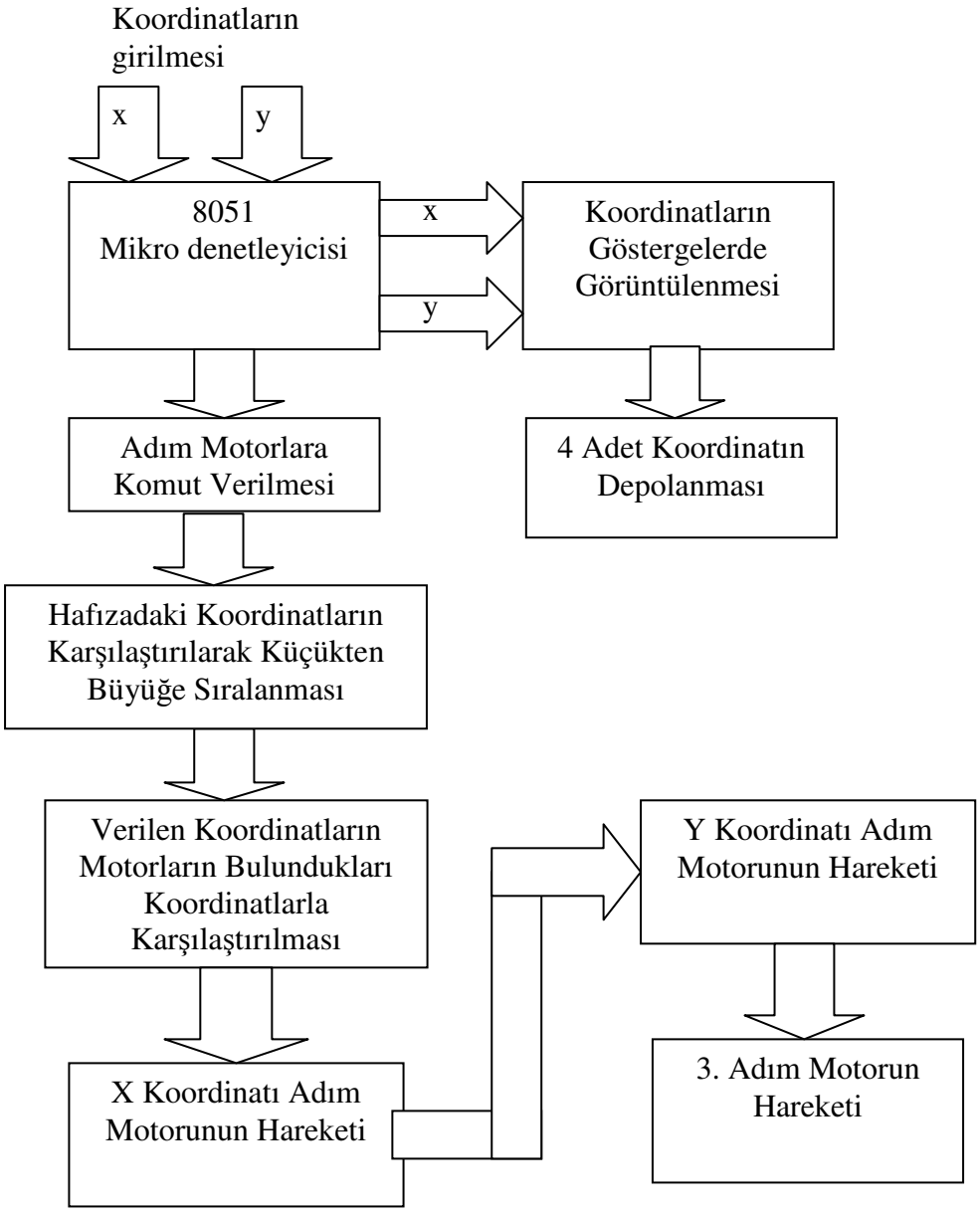


Şekil 5.1. Blok diyagram

Bu mekanik sistemin kontrolü 8051 mikro denetleyicisi üzerine yazılan bir program vasıtasıyla yapılacağı ve 8051 portlarının çıkışındaki "1" ve "0" lojik işaretleri kullanılacağından dolayı ve sistemde mümkün olduğunca en iyi hassasiyeti elde etmek amaç olduğundan dolayı koordinat bulma işlemi 2 adet adım motor kullanılarak yapılmaya karar verildi. Nokta koydurma işleminin ise yine bir adım motor yardımıyla yapılmasına karar verildi. Dışarıdan girilecek koordinatların

gösteriminin ikişer adet 7 parçalı gösterge vasıtasıyla yapılmasına karar verildi. Koordinatların girilmesi ise, her bir koordinat için aşağı ve yukarı yönde sayma olmak üzere toplam 4 adet buton kullanılarak yapılması tasarlandı. Girilen koordinatların hafızaya saklanması, kalemin tekrar başlangıç noktasına gelmesini sağlanması ve son olarak da motorların hareketinin başlamasını sağlamak için yine birer buton kullanılmasına karar verildi.

Projenin bilgisayar üzerindeki simülasyonu Proteus programında, yazılımı ise KEIL programında yapıldı.



Şekil 5.2. Uygulamanın çalışmasının blok diyagramı

Sistemi gerçekleştirmek için ilk ihtiyacı duyulan malzeme bir adet 8051 mikro denetleyici programlama kartı, daha sonra koordinat gösteriminin sağlanması için 4 adet 7 parçalı gösterge, yine bu göstergeleri 8051 portlarıyla sürmek için 4 adet BCD'den 7 parçalı göstergeye kodlayıcı 7447 entegresi, koordinat takibinde kullanmak için 2 adet adım motor, nokta koydurma işlemi için de bir adet adım motor, bunları lojik sinyallerle sürmek için 24V güç kaynağı ve 3 adet darlington transistor dizisine sahip ULN2003A entegresinin kullanımı kararlaştırıldı. Ayrıca göstergelerde netliği sağlamak amacı ile pull-up direnci olarak 8 adet 1K direnç, gösterge ve motor kontrolleri için de 5 adet buton kullanımı kararlaştırıldı. Ayriyeten göstergelerden girilen dört farklı koordinatın hafızada saklanması işleminin sağlanması için ve motorların tekrar başlangıç noktalarına dönmeleri için de birer tane buton kullanıldı.

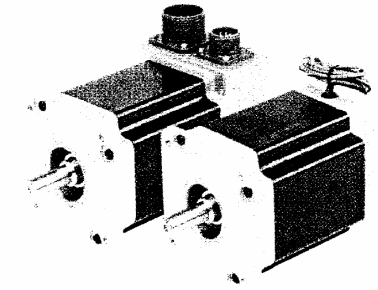
Malzeme listesi:

1 adet 8051 programlama kartı
4 adet 7 parçalı gösterge
24V güç kaynağı
4 adet 7447 entegre
3 adet ULN2003A entegre
3 adet adım motor
7 adet buton

5.2. Mekanik Sistemin Tasarımı

5.2.1. Adım motorlar

5.2.1.1. Giriş



Şekil 5.3. Adım motorlar

Adım motorların birkaç genel karakteristiği ve geniş bir kullanım alanı bulmalarının nedeni aşağıdaki özellikleri taşımasındandır.

1. Adım motorların açık çevrim davranışlarının ±1 adım doğruluk pozisyonuna sahip olmaları (rotorun açısal hızı yeterince küçük olduğundan hareket sırasında basamak kaybı olmaz.), yani kesin açısal mesafe tanımlanırsa motorun dönmesi uygun sayıda adımla (basamakla) kontrol edilebilir ve böylece mekanik sistemde milin hareketi yeterli ölçüde olur.

2. Adım motorların yüke yeterli momenti sağlayabilmeleri

3. Adım motorlar DC uyarmada geniş bir tutma torkuna sahiptirler. Yani adım motorların rotor hareketi sabitken (otomatik kilitleme) özelliği vardır. Bu durumda rotor sadece, terminal gerilimi zamanla değiştiği sürece hareket eder [55].

5.2.1.2. Adım motorların diğer motorlarla kıyaslanması

Adım motorların karakteristiği turu adımlara bölmesidir. Bu motorlarla çok basit ama çok hassas iş yapan makineler yapılabildiği gibi otomasyon alanında da çok hesaplı

çözüm sağlar. Adım motorlar 2faz, 3faz ve 5faz olarak da ayrılırlar ama standart olanı 2 faz'dır. Fakat bu motorlar fazla süratli dönemezler, hızlı pozisyonlama gerektiren uygulamalar için tercih edilmezler. Ama bir pantograf makinesinde adım motordan alınan verimlilik servo motordan alınamaz. Bir adım motorun bir turu sürücü marifetiyle 50.000 adıma bölünebilir. Bir servo motorda ise bu, enkoderin çözünürlüğüne bağlıdır. Servo sürücülerin de bölme kabiliyeti vardır. Ama bu, adım motor gibi kesin netice vermez yani tekrarlama hassasiyeti adım motorda çok yüksektir. Servoda ise enkoderin hassasiyeti kadardır. Ayrıca servo sürücüler kendi içlerinde yuvarlama ve düzeltme yaparlar bu da tabii ki yapılan işe yansır.

Servo motorlar 3 guruba ayrılır. BDC servo, BLDC servo, AC servo. Bunların içinde en kararlısı BDC servodur (fırçalı dc servo). Fakat fırçalardan dolayı çok uzun ömürlü değildirler. Buna karşılık adım motorlar kadar olmasa da ona yakın hassas iş yaparlar. Bu da tabii ki enkodere bağlıdır.

BLDC servo (fırçasız DC servo) motorlar diğerlerine göre oldukça uzun ömürlüdürler, ama fırçalı servo motorlar kadar hassas değildirler. Tekniği AC servo tekniği gibidir. Sadece kullanılan doğal mıknatıslar sayesinde AC servodan daha az enerji harcayarak muadili AC servodan daha fazla güç sağlarlar.

AC servo motorlar sıradan elektrik motorlarına benzer tek özelliği bir enkoderinin olması ve farklı bir tasarımı olmasıdır. Uzun ömürlüdür, az sorunludur. Genellikle hızlı pozisyonlamalarda kullanılır, ama hassas pozisyonlamada problemlidir. Çünkü fazla yaylanır. Yaklaşık 10 derece gibi yaylanır. Bu da hassas işlerde problem oluşturur. Bunu önlemek için kalkış ve duruş rampalarının uzun tutulmasını gerektirir. Bu da sistemin yavaşlamasına neden olur. Çok hassasiyet istemeyen ve hızlı çalışma gereken yerlerde ideal çözümdür. Ahşap işleme makineleri, plastik işleme makineleri, ambalaj makineleri gibi makinelerde ideal çözümdür.

Adım sürücüler ise en kolay yönetilebilen sürücüler olmalarının yanında en hesaplı sürücülerdir. Motor fazları genelde bir bobin resmiyle tanımlanır, ya da A-A bir faz B-B diğer faz olarak tanımlanır. Adım ve amper ayarları da genelde şematik ya da direnç değerleriyle tarif edilmişlerdir. Kısacası adım motor ve sürücüleri hem ekonomik hem de uygulaması kolaydır.

Servo sürücülerin hepsinin mantığı yaklaşık olarak aynıdır. Servo sürücülerin adım mantığıyla sürülenleri olduğu gibi analog sürülenleri çoğunluktadır. Bu sürücüleri analog sürmek için buna özel kontrol kartları gerektirir ve o karta özel yazılımlar gerekir bu da biraz maliyetlidir.

Adım motor mantığıyla sürülen servo sürücüler biraz pahalıdır ama bu sürücüleri özel bir kart olmadan bir bilgisayardan direkt sürmek bile mümkündür. Bu konuda çok ucuza birçok yazılım mevcuttur.

DC motorlar ise daha çok endüstride kullanılırlar. Motorun dairesel hızı, uygulanan voltajla doğru orantılıdır. Çıkış momenti ise bobin akım gücü ile doğru orantılıdır. Eğer motor hareketi hassas bir şekilde kontrol edilmek isteniyorsa geri besleme kullanılmalıdır.

Yukarıdaki özelliklerin yanında DC servo motor yerine adım motorlar seçilmesinin avantajlarını aşağıdaki gibi sıralayabiliriz.

- Adım motorların sayısal kontrollü sistemlerine uygun olması, dolayısıyla sayısal kontrol ve/veya bilgisayarlı uygulamaları mümkündür.

- Hata yalnız adım hatasıdır (Genellikle adım başına %5 den daha azdır.). Bu nedenle birimsizdir. Adım motorların pozisyonlarında mükemmel doğruluğa sahip olmaları ve daha önemlisi hataları bir yerde toplamasıdır.

- Motorda açık çevrim kontrolünün olması nedeniyle, takometre ve/veya encoderin motorda kullanılması gereksizdir. Buna bağlı olarak tasarımın maliyeti düşer. Geri besleme ile mil pozisyonunun tayin edilmesi elimine edilir.

- Sayısal işareti analog işarete çevirmek gerektiğinden, bir ara birim veya sayısal kompütür elimine edilebilir.

- Motor yapısının basit ve kuvvetli olması genellikle motorda iki duruş pozisyonu olması, motor bakımının kolay ve kullanım süresinin uzun olmasını sağlar. Bu nedenlerle maliyet düşmüş olur.

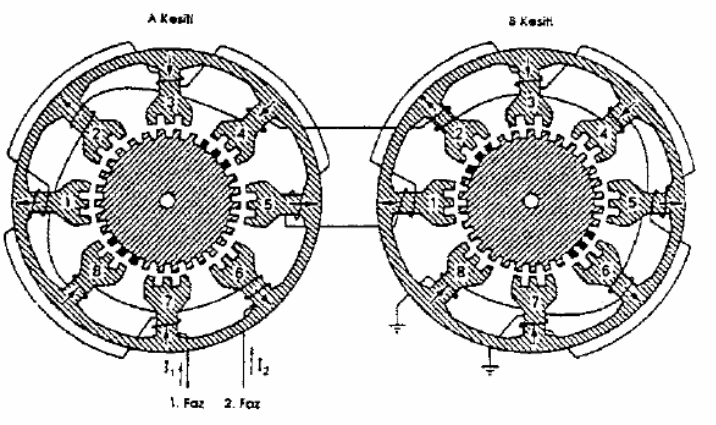
- Adım motorların ısınma gibi olumsuzluklardan gelen zararları azdır.

- Motor, bir güç kaynağı ve motor cevabını değerlendiren sürücü bir devre ile kontrol edilebilir.

Adım motorlar robot teknolojisinde sıkça kullanım alanı bulmuştur. Ayrıca maliyetinin düşük olması DC servo motorlara karşı bir üstünlüğüdür. Adım motorları kullanılmasının ikinci bir nedeni de tutma karakteristiğinin robotlarla bağdaşmasıdır. Motor sürücülerinde kullanılan genel güç kuvvetlendiricileri çıkışı kollektörden emitere kısa devre edilmiş bir transistördür. Bu da tüm DC geriliminin motor armatürüne uygulanmasına neden olur. Bir servo motor hareketi mekanik veya başka bir engelle karşılaşıncaya kadar devam eder ve sonuçta tehlikeli durum ortaya çıkabilir. Buna karşın, bir adım motora uygulanan gerilim ile adım hareketi veya tutma (durdurma) işlemi istendiğinde gerçekleşir. Yani adım motor durdurmak istendiğinde bir hareket olamayacak ve servo motorun yol açtığı olumsuzluklar görülmeyecektir.

5.2.1.3. Tipik yapısı

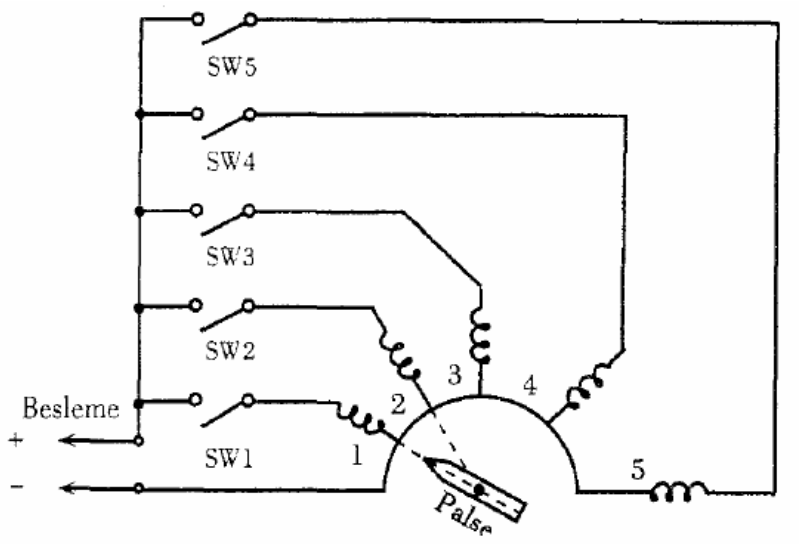
Adım motor statorunun birçok kutbu vardır. Bunların polaritesi elektronik anahtarlar yardımıyla değiştirilmektedir. Anahtarlama sonucunda statorun güney ve kuzey kutupları döndürülmektedir. Rotorun mıknatıslığı, sürekli bir mıknatıs veya dış uyarım metotlarıyla oluşturulabilmektedir. Böylece sürekli mıknatıs etkisi oluşmaktadır. Adımları vasıtasıyla ortalama stator alanı döner ve rotor da buna benzer adımlarla takip eder. Daha iyi bir seçicilik elde etmek için rotor ve stator üzerine küçük dişler yapılmaktadır. Bu dişler birbirleriyle temas etmemelidir. Fakat düşük redüktanslı bölümlerde işe yarayabilir [55].



Şekil 5.4. Tipik adım motor yapıları

5.2.1.4. Çalışma prensibi

Adım motora giriş pals uygulandığı zaman belli bir miktar döndükten sonra durur. Bu dönme sayısı motorun yapısına göre belli bir açı ile sınırlandırılmıştır. Adım motorda girişe uygulanan pals adedine bağlı olarak rotorun dönmesi değişmektedir. Girişe bir pals verildiğinde rotor tek bir adım hareket ettikten sonra durur. Daha çok miktarda pals uygulanınca pals adedi kadar adım hareketi yapar. Bütün adım motorlarının çalışması bu şekilde gerçekleşmektedir.



Şekil 5.5. Adım motorun prensip şeması

Adım motor bir daire içinde elektromanyetik alanların dönmesi ile açıklanabilir. Şekil 5.5'deki 1 nolu anahtar kapandığı zaman sabit mıknatıs kendiliğinden 1. elektromanyetik alan ile aynı hizaya gelecektir ve bundan sonra 1 nolu anahtar açılıp 2 nolu anahtar kapatılırsa sabit mıknatıs 2. elektromanyetik alanın karşısına gelecektir. Bu olaylar sırasıyla tekrarlanırsa daimi (sabit) mıknatıs yani rotor bir daire içinde düzgün şekilde döner [55].

5.2.1.5. Adım motorun çeşitleri

- Sabit mıknatıslı adım motorlar (PM)
- Değişken relüktanslı adım motorlar (VR)
  - Tek parçalı
  - Çok parçalı
- Hybrid adım motorlar
- Hidrolik adım motorlar
- Lineer adım motorlar

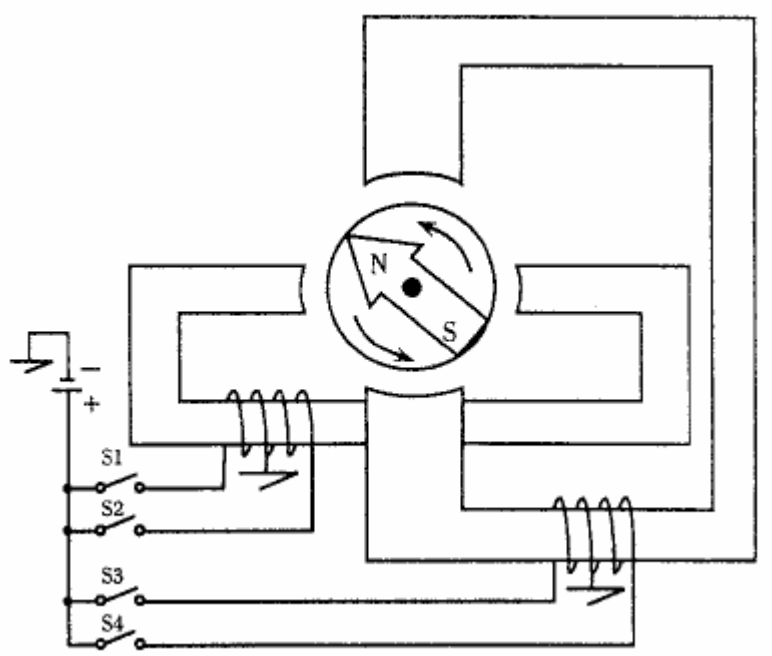
5.2.1.6. Orta uçlu sargılara sahip sabit mıknatıslı adım motor

Orta uçlu sargılara sahip bir PM adım motorun en basit kontrolü, şekil 5.6'da verilen devre ile gerçekleştirilebilir. Adım motorun çalışması için S1, S2, S3 ve S4 anahtarları üzerinden Faz1 ve Faz2 sargılarına sırası ile uygun faz ile gerilim uygulanmalıdır. Devrede kullanılan motorun 90^{o}'lik adımlarda dönmesini istersek tablo 5.1'de verilen dört değişik çalışma durumunu (kodlarını) art arda uygulamalıyız.

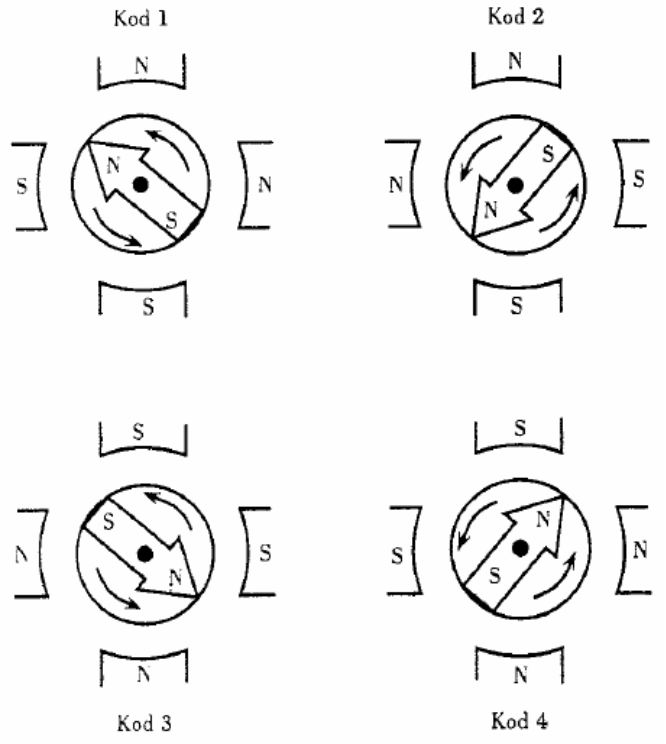
Tablo 5.1. İki fazlı orta uçlu sargılara sahip adım motor anahtarlama tablosu

Kod	S1	S3	S2	S4
1	0	0	1	1
2	1	0	0	1
3	1	1	0	0
4	0	1	1	0

Şekil 5.6'daki anahtarların dört değişik çalışma durumunu (kodunu) veren tablo 5.1 ve bu kodlara göre rotorun hareketleri adım adım çizilmiştir. Bu şekiller üzerinden S1, S2, S3 ve S4 anahtarlarının kapalı (1) ve açık (0) oluşuna göre rotorun iki kutup arasında 90^{o}' lik adımlarla ve saat ibresinin ters yönünde (CCW) nasıl döndüğü Şekil 5.7'de görülmektedir.



Şekil 5.6. PM adım motorun anahtarlarla kontrolü



Şekil 5.7. Anahtar durumlarına göre PM motorda rotor hareketleri

İlk adım yani Kod1 için S2 ve S4 anahtarları kapatılır. Faz 1 ve Faz 2 sargılarına uygulanan gerilim sonucu rotor, şekil 5.7'deki kod 1 çalışmasını tamamlar ve durur.

S4 anahtarı kapalıyken S2 açılıp S1 kapatılırsa rotor bu kez kod 2 çalışmasını tamamlar yani 90° döner ve durur. Kod3 çalışması için S1 anahtarı kapalıyken S4 açılıp S3 kapatılır. Aynı şekilde Kod4 çalışması için ise S3 kapalıyken S1 açılıp S2 kapatılmalıdır. Anahtarlar bu sırayla değiştirilmeye devam edildiğinde rotor da dönmeye devam edecektir. Adım motorun çalışma durumları değiştirilmeye devam edildiği sürece buna bağlı olarak rotor da dönmeye devam edecektir. Adım motorun çalışma durumlarının değişmesinde sadece bir anahtarın değiştiğine dikkat ediniz. Bu durum, rotorun eşit adımlarla ve aynı yönde dönmesini sağlar [55].

5.2.1.7. Adım motorların özellikleri

- Hata yalnız adım hatasıdır.
- Motor bakımı kolaydır.
- Tasarım maliyeti ucuzdur.

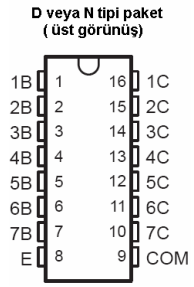
- Otomatik kilitleme özelliğine sahiptir.
- Yüke yeterli momenti sağlar.
- Isınma gibi olumsuzluklardan meydana gelen zararlar azdır.
- Hızı, programlama yoluyla ayarlanabilir.
- Mikro bilgisayarlarla kolaylıkla kontrol edilebilir.
- Çalışma sırasında hızı sabit kalır, değişmez.
- Kullanım ömrü uzundur [55].

5.2.2. Motor kontrolü

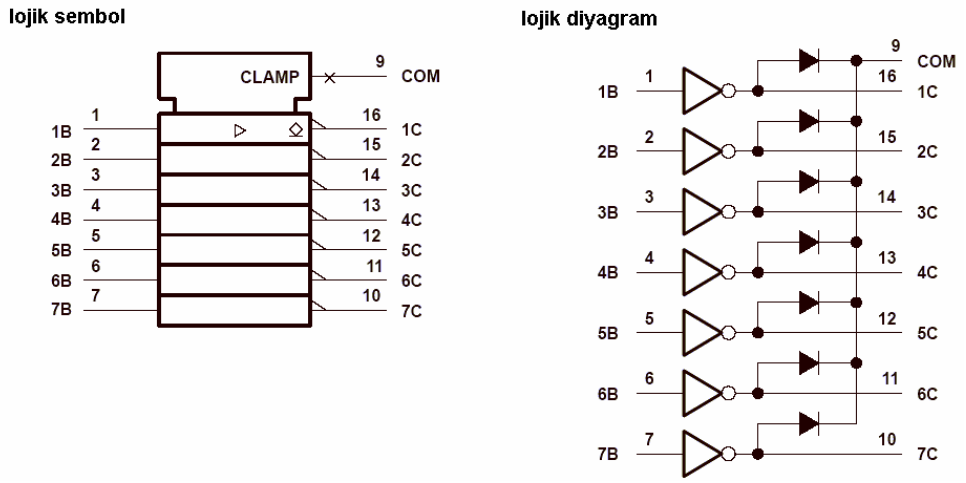
Motor kontrolü için ULN2003A entegresi kullanılmıştır.

5.2.2.1. ULN2003A genel özellikleri

500 mA ile Değerlenmiş Kollektör Akımı (Tek Çıkış)
Yüksek Voltaj Çıkışları 50V
Çıkış Sınırlama Diyodları
Çeşitli Lojik Tipleriyle Uyumlu Girişler
Anahtarlama Sürücü Uygulamaları
Sprague ULN2001A Serileriyle Değiştirilebilir Tasarım



Şekil 5.8. Entegre üst görünüşü

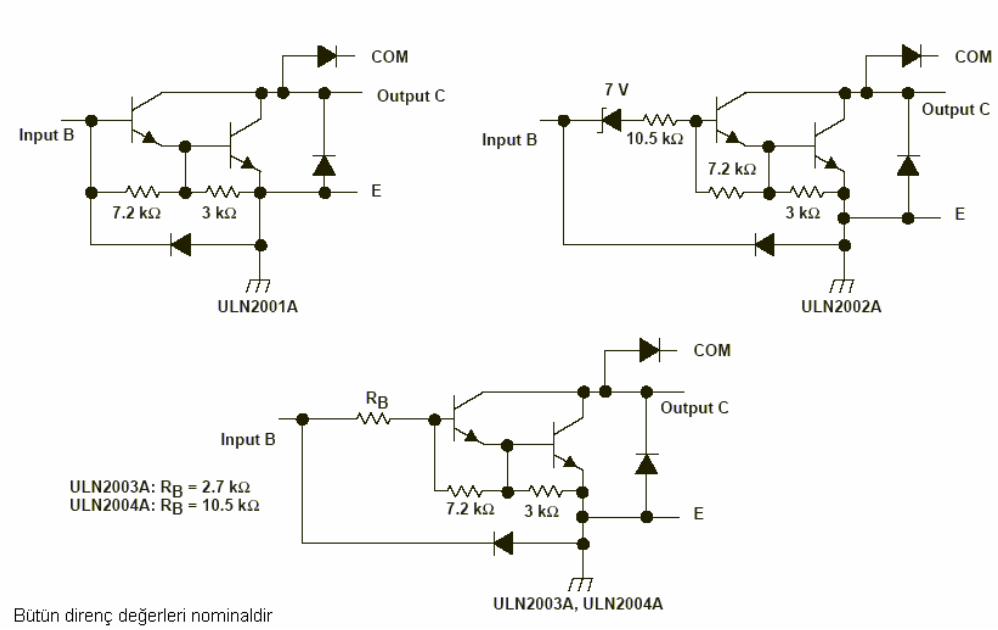


Şekil 5.9. Lojik sembol ve lojik diyagram

5.2.2.2. Tanım

ULN2003A entegresi yüksek voltaj, yüksek akım darlington transistör dizisinden oluşmuş tek parça bir cihazdır. Anahtarlama için ortak katod sınırlanmış diyodlu yüksek voltaj çıkış özelliğine sahip yedi adet npn darlington çiftinin her biri endüktif yüklenir. Bir darlington çiftinin kollektör akım sınırı 500mA'dir. Darlington çiftleri daha yüksek bir akım kapasitesi için paralellenebilir. Uygulamalar, anahtarlama sürücüleri, çekiç sürücüleri, lamba sürücüleri, gösterge sürücüleri, (LED ve gaz boşaltma), hat sürücüleri ve lojik tamponları içerir [43].

5.2.2.3. Darlington çiftleri için şemalar



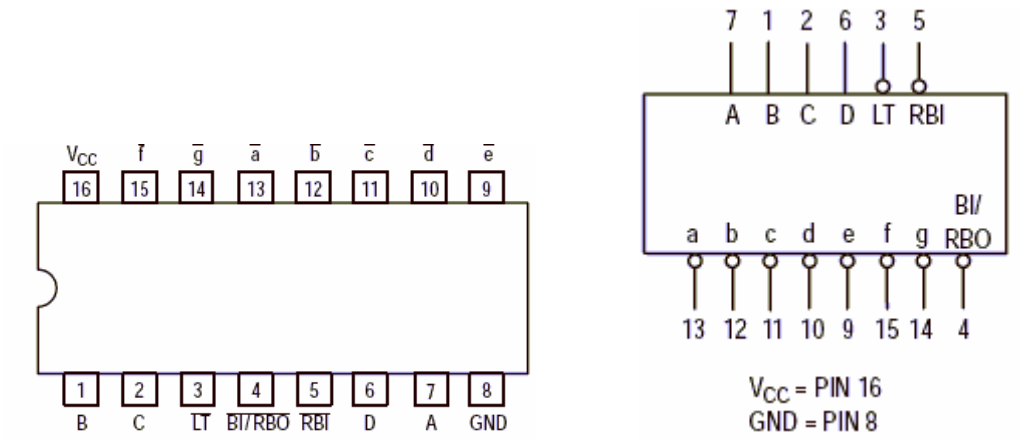
Şekil 5.10. Darlington çiftleri

5.2.3. Gösterge ünitesi

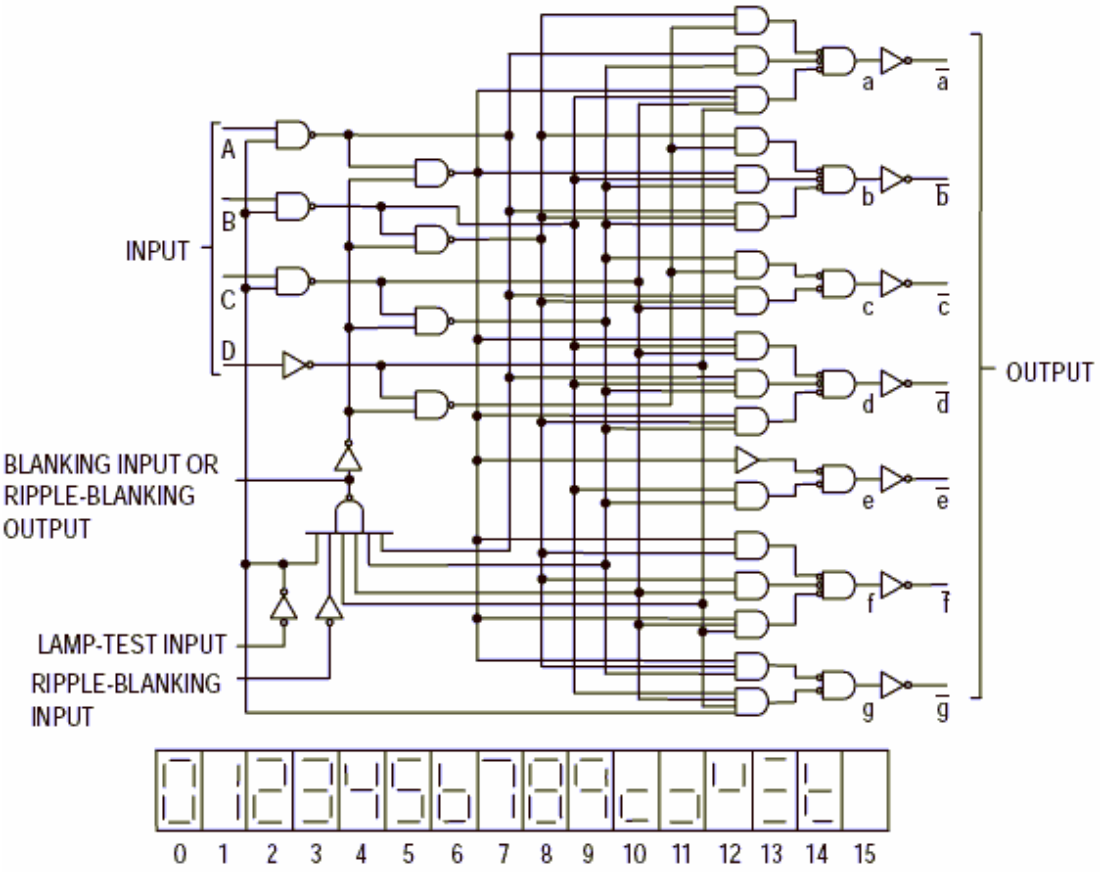
Gösterge ünitesi olarak SN74LS47 entegresi kullanılmıştır.

SN74LS47 entegresi NAND kapıları, giriş tamponları ve yedi adet AND-OR-INVERT kapısı içeren, BCD' den 7 parça göstergeye kodlama yapan düşük güçlü schottky'dir. Göstergeleri çalıştırmak için yüksek bantlı akım gerekir. BCD veri girişinin uygun bir şekilde yedi parçaya kodlanması için girişte yedi NAND kapısı ve bir sürücü, çıkışta ise yedi adet AND-OR-INVERT kapısı kullanılmıştır. Kalan NAND kapısı ve üç giriş tamponu, lamba testini, silme girişini, tepe silme çıktısı ve tepe silme girişini sağlarlar. Devreler, yardımcı girişlerin durumuna bağlı olarak, 4 bit BCD veriyi kabul ederler ve 7 parça göstericiyi sürmek için bu veri kodunu çözerler. Yardımcı girişlerde gerekli şartların gerektirdiği pozitif çıkış seviyeleri doğruluk tablosunda gösterilmiştir. SN74LS47 entegresinin çıkış yapısı yüksek voltaj gerektiren 7 parça göstergeler için dizayn edilmiştir. Bu çıkışlar en fazla 250 µA ters akıma sahip 15V'a dayanabilir. 24 mA'e kadar akım gerektirebilen göstergeler,

yüksek performanslı çıkış transistörlerine sahip olan SN74LS47 entegresiyle direkt olarak sürülebilir. BCD olarak girilen verinin dokuzun üstünde olması halinde farklı şekillerle ya da harflerle ifade edilir [43].



Şekil 5.11. 7447 lojik sembolleri



Şekil 5.12. 7447 lojik iç yapısı

Tablo 5.2. Doğruluk tablosu

	INPUTS							OUTPUTS							
DECIMAL OR FUNCTION	$\overline{LT}$	$\overline{RBI}$	D	C	B	A	$\overline{BI/RBO}$	$\overline{a}$	$\overline{b}$	$\overline{c}$	$\overline{d}$	$\overline{e}$	$\overline{f}$	$\overline{g}$	NOTE
0	H	H	L	L	L	L	H	L	L	L	L	L	L	H	A
1	H	X	L	L	L	H	H	H	L	L	H	H	H	H	A
2	H	X	L	L	H	L	H	L	L	H	L	L	H	L	
3	H	X	L	L	H	H	H	L	L	L	L	H	H	L	
4	H	X	L	H	L	L	H	H	L	L	H	H	L	L	
5	H	X	L	H	L	H	H	L	H	L	L	H	L	L	
6	H	X	L	H	H	L	H	H	H	L	L	L	L	L	
7	H	X	L	H	H	H	H	L	L	L	H	H	H	H	
8	H	X	H	L	L	L	H	L	L	L	L	L	L	L	
9	H	X	H	L	L	H	H	L	L	L	H	H	L	L	
10	H	X	H	L	H	L	H	H	H	H	L	L	H	L	
11	H	X	H	L	H	H	H	H	H	L	L	H	H	L	
12	H	X	H	H	L	L	H	H	L	H	H	H	L	L	
13	H	X	H	H	L	H	H	L	H	H	L	H	L	L	
14	H	X	H	H	H	L	H	H	H	H	L	L	L	L	
15	H	X	H	H	H	H	H	H	H	H	H	H	H	H	
$\overline{BI}$	X	X	X	X	X	X	L	H	H	H	H	H	H	H	B
$\overline{RBI}$	H	L	L	L	L	L	L	H	H	H	H	H	H	H	C
$\overline{LT}$	L	X	X	X	X	X	H	L	L	L	L	L	L	L	D

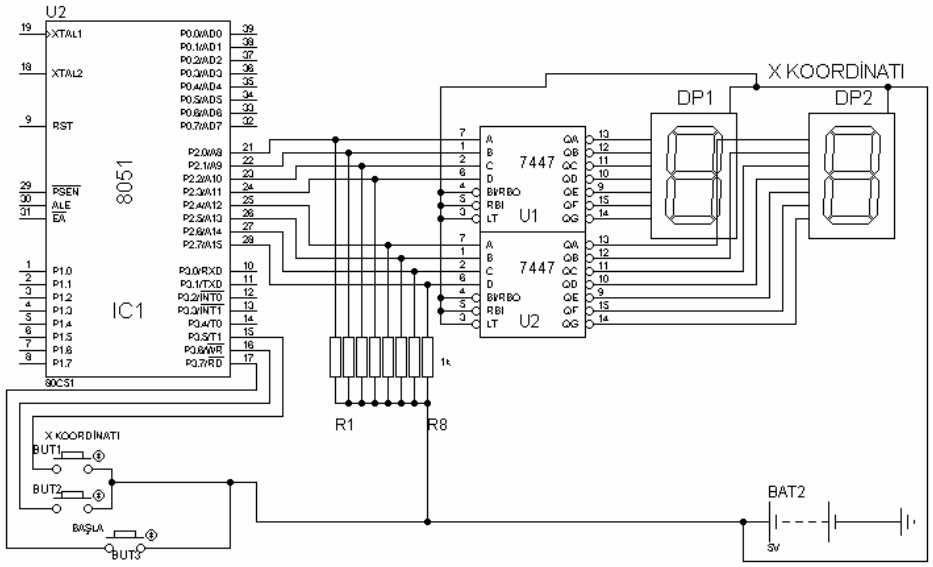
H = HIGH Voltage Level
L = LOW Voltage Level
X = Immaterial

5.3. Sistem Kartının Tasarımı

5.3.1. Motorun X ve Y koordinatlarında hareketi

5.3.1.1. X Koordinatının gerçeklenmesi

- Bilgisayar ortamında sayıcının tasarlanarak yazılımının yapılması : Projede iki karakterli bir koordinat girişi istenildi. Ve aşağıdaki şemadaki gibi bir düzenek tasarlandı. Aşağıdaki düzenekte görüldüğü gibi BUT1 butonuyla yukarı ve BUT2 butonuyla da aşağı sayma işlemi gerçekleştirilmek istenildi. R1–R8 dirençleri ise daha net bir görüntü elde etmek için pull-up direnci olarak kullanıldı. Tasarlanan bu düzeneğin çalışması için de şemanın altındaki program yazıldı.



Şekil 5.13. Sayıcı devre şeması

- Sayıcı için yazılan program :

```
#include <AT89X51.H>

#define X10SIFIR P2_3=0,P2_2=0,P2_1=0,P2_0=0;
#define X10BIR   P2_3=0,P2_2=0,P2_1=0,P2_0=1;
#define X10IKI   P2_3=0,P2_2=0,P2_1=1,P2_0=0;
#define X10UC    P2_3=0,P2_2=0,P2_1=1,P2_0=1;
#define X10DORT  P2_3=0,P2_2=1,P2_1=0,P2_0=0;
#define X10BES   P2_3=0,P2_2=1,P2_1=0,P2_0=1;
#define X10ALTI  P2_3=0,P2_2=1,P2_1=1,P2_0=0;
#define X10YEDI  P2_3=0,P2_2=1,P2_1=1,P2_0=1;
#define X10SEKIZ P2_3=1,P2_2=0,P2_1=0,P2_0=0;
#define X10DOKUZ P2_3=1,P2_2=0,P2_1=0,P2_0=1;

#define X1SIFIR P2_7=0,P2_6=0,P2_5=0,P2_4=0;
#define X1BIR   P2_7=0,P2_6=0,P2_5=0,P2_4=1;
#define X1IKI   P2_7=0,P2_6=0,P2_5=1,P2_4=0;
#define X1UC    P2_7=0,P2_6=0,P2_5=1,P2_4=1;
#define X1DORT  P2_7=0,P2_6=1,P2_5=0,P2_4=0;
```

```
#define X1BES   P2_7=0,P2_6=1,P2_5=0,P2_4=1;
#define X1ALTI  P2_7=0,P2_6=1,P2_5=1,P2_4=0;
#define X1YEDI  P2_7=0,P2_6=1,P2_5=1,P2_4=1;
#define X1SEKIZ P2_7=1,P2_6=0,P2_5=0,P2_4=0;
#define X1DOKUZ P2_7=1,P2_6=0,P2_5=0,P2_4=1;

void bekle(void){
        unsigned int sayac= 5669;
        while(sayac--);}

main(){
unsigned char Xsayac1=0,Xsayac2=0;
P2=0;
while(1){
P3_6=0,P3_5=0;
        if(P3_5==1){
        if(Xsayac1==10) Xsayac1=1;
        else if(Xsayac1==9) Xsayac1=0;
        else Xsayac1++;
                switch (Xsayac1){
                        case 0: X1SIFIR;
                           if(Xsayac2==10) Xsayac2=1;
                            else if(Xsayac2==9) Xsayac2=0;
                            else Xsayac2++;
                                switch(Xsayac2){
                                        case 0: X10SIFIR; break;
                                        case 1: X10BIR; break;
                                        case 2: X10IKI; break;
                                        case 3: X10UC; break;
                                        case 4: X10DORT; break;
                                        case 5: X10BES; break;
                                        case 6: X10ALTI; break;
                                        case 7: X10YEDI; break;
```

```
                                    case 8: X10SEKIZ; break;
                                    case 9: X10DOKUZ; Xsayac2=-1; break;}
break;

                        case 1: X1BIR; break;
                        case 2: X1IKI; break;
                        case 3: X1UC; break;
                        case 4: X1DORT; break;
                        case 5: X1BES; break;
                        case 6: X1ALTI; break;
                        case 7: X1YEDI; break;
                        case 8: X1SEKIZ; break;
                        case 9: X1DOKUZ; Xsayac1=-1; break;} bekle();}

if(P3_6==1){
                if(Xsayac1==0) Xsayac1=9;
                else if(Xsayac1==-1) Xsayac1=8;
                else Xsayac1--;
                switch (Xsayac1){
                        case 9: X1DOKUZ;
                           if(Xsayac2==0) Xsayac2=9;
                           else if(Xsayac1==-1) Xsayac1=8;
                           else Xsayac2--;
                                switch(Xsayac2){
                                case 9: X10DOKUZ; break;
                                case 8: X10SEKIZ; break;
                                case 7: X10YEDI; break;
                                case 6: X10ALTI; break;
                                case 5: X10BES; break;
                                case 4: X10DORT; break;
                                case 3: X10UC; break;
                                case 2: X10IKI; break;
```

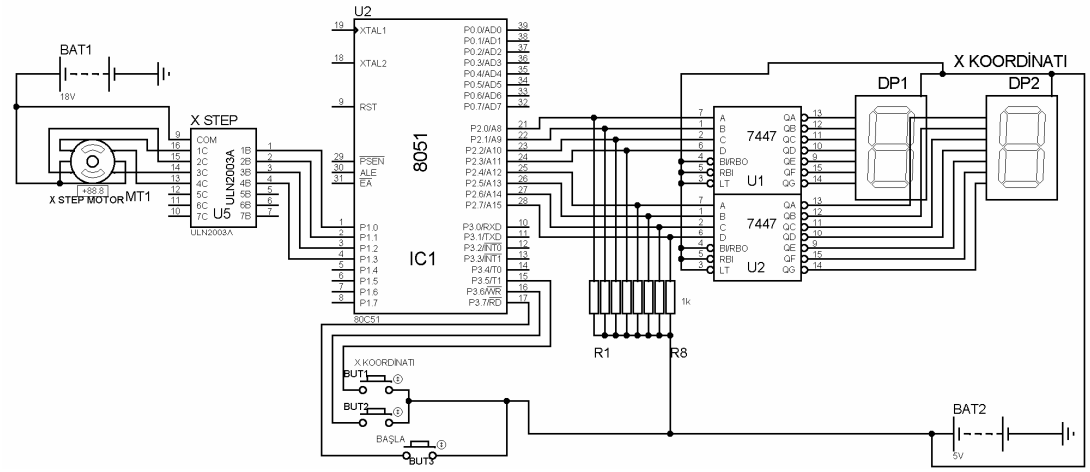
```
                    case 1: X10BIR; break;
                    case 0: X10SIFIR; Xsayac2=10; break;} break;

            case 8: X1SEKIZ; break;
            case 7: X1YEDI; break;
            case 6: X1ALTI; break;
            case 5: X1BES; break;
            case 4: X1DORT; break;
            case 3: X1UC; break;
            case 2: X1IKI; break;
            case 1: X1BIR; break;
            case 0: X1SIFIR; Xsayac1=10; break;} bekle();
                                                    }}}
```

Bu yazılımda ;

```
void bekle(void){
        unsigned int sayac= 5669;
        while(sayac--);}
```

satırları, göstergelerdeki iki sayı arasındaki bekleme zamanını ayarlamaktadır. #define komutuyla tanımlanan kısımlar ise 8051 portlarının çıkışında istediğimiz karakterleri elde etmek için portlara BCD kodları göndermek için kullanılmıştır. X koordinatı gösterimi için P2 portu seçildi. Portun ilk 4 ucu göstergelerin onlar basamağı için, son 4 ucu da birler basamağı için kullanılmıştır.

- Sayıcı ile bir adım motorun irtibatlandırılması ve motorun bulunduğu noktanın program hafızasında tekrar kullanım için saklanması: İlk önce adım motorun nasıl sürüleceği tasarlandı. Bunun için ULN2003A entegresi kullanılmaya karar verildi. Bu adım motorun sürülmesi için Port1'in ilk dört ucunun kullanılmasına karar verildi. Ve aşağıdaki şemadaki gibi bir bağlantı yapıldı.



Şekil 5.14. X koordinatı için devre şeması

Yukarıdaki gibi devreye eklenen, adım motor sürülmesi işleminin yazılıma yansımasında ise, yine yukarıdaki, göstergelerin göstereceği rakamların #define ile tanımlanması gibi burada da adım motorun atacağı adımlar #define XADIM1.... gibi tanımlamalarla tanımlanmıştır.

Ve motorun o an için bulunduğu konumunun sonraki değer atamalarında tekrar kullanılabilmesi için aşağıdaki komut satırları eklendi.

Motorun ileri doğru hareketi için;

Xdurum=Xyeni-Xeski;Xeski=Xyeni;

Motorun geriye doğru hareketi için;

Xdurum=Xeski-Xyeni;Xeski=Xyeni;

komutları kullanılmıştır. Burada Xeski, motorun o anki konumunu; Xyeni, yeni girilen değeri; Xdurum ise motorun yapacağı hareket sınırını gösterir.

Motorun harekete başlaması için gereken komut, Port3'ün 7. bacağına bağlanan buton vasıtasıyla verilmektedir. Adım motorun eklenmesiyle programın son hali aşağıdaki gibi olmuştur.

- X koordinatının çalışması için gereken yazılım :

```
#include <AT89X51.H>

#define X10SIFIR P2_3=0,P2_2=0,P2_1=0,P2_0=0;
#define X10BIR   P2_3=0,P2_2=0,P2_1=0,P2_0=1;
#define X10IKI   P2_3=0,P2_2=0,P2_1=1,P2_0=0;
#define X10UC    P2_3=0,P2_2=0,P2_1=1,P2_0=1;
#define X10DORT  P2_3=0,P2_2=1,P2_1=0,P2_0=0;
#define X10BES   P2_3=0,P2_2=1,P2_1=0,P2_0=1;
#define X10ALTI  P2_3=0,P2_2=1,P2_1=1,P2_0=0;
#define X10YEDI  P2_3=0,P2_2=1,P2_1=1,P2_0=1;
#define X10SEKIZ P2_3=1,P2_2=0,P2_1=0,P2_0=0;
#define X10DOKUZ P2_3=1,P2_2=0,P2_1=0,P2_0=1;

#define X1SIFIR P2_7=0,P2_6=0,P2_5=0,P2_4=0;
#define X1BIR   P2_7=0,P2_6=0,P2_5=0,P2_4=1;
#define X1IKI   P2_7=0,P2_6=0,P2_5=1,P2_4=0;
#define X1UC    P2_7=0,P2_6=0,P2_5=1,P2_4=1;
#define X1DORT  P2_7=0,P2_6=1,P2_5=0,P2_4=0;
#define X1BES   P2_7=0,P2_6=1,P2_5=0,P2_4=1;
#define X1ALTI  P2_7=0,P2_6=1,P2_5=1,P2_4=0;
#define X1YEDI  P2_7=0,P2_6=1,P2_5=1,P2_4=1;
#define X1SEKIZ P2_7=1,P2_6=0,P2_5=0,P2_4=0;
#define X1DOKUZ P2_7=1,P2_6=0,P2_5=0,P2_4=1;

#define XADIM1 P1_3=1,P1_2=0,P1_1=0,P1_0=0;
#define XADIM2 P1_3=0,P1_2=1,P1_1=0,P1_0=0;
#define XADIM3 P1_3=0,P1_2=0,P1_1=1,P1_0=0;
#define XADIM4 P1_3=0,P1_2=0,P1_1=0,P1_0=1;

void bekle(void){
        unsigned int sayac= 5669;
```

```
        while(sayac--);}

void motorbekle(void){
        unsigned int sayac=9567;
        while(sayac--);}

main(){
unsigned char Xsayac1=0,Xsayac2=0,Xyeni=0,Xeski=0,Xdurum=0;
P2=0;
while(1){
P3=0;
if(P3_5==1){
        if(Xsayac1==10) Xsayac1=1;
        else if(Xsayac1==9) Xsayac1=0;
        else Xsayac1++;
        switch (Xsayac1){
        case 0: X1SIFIR;
           if(Xsayac2==10) Xsayac2=1;
           else if(Xsayac2==9) Xsayac2=0;
           else Xsayac2++;
                switch(Xsayac2){
                        case 0: X10SIFIR; break;
                        case 1: X10BIR; break;
                        case 2: X10IKI; break;
                        case 3: X10UC; break;
                        case 4: X10DORT; break;
                        case 5: X10BES; break;
                        case 6: X10ALTI; break;
                        case 7: X10YEDI; break;
                        case 8: X10SEKIZ; break;
                        case 9: X10DOKUZ; Xsayac2=-1; break;} break;

        case 1: X1BIR; break;
```

```
        case 2: X1IKI; break;
        case 3: X1UC; break;
        case 4: X1DORT; break;
        case 5: X1BES; break;
        case 6: X1ALTI; break;
        case 7: X1YEDI; break;
        case 8: X1SEKIZ; break;
        case 9: X1DOKUZ; Xsayac1=-1; break;} bekle();
                if(Xsayac1==-1 && Xsayac2==-1) Xyeni=99;
                else if(Xsayac1==-1 && Xsayac2==10) Xyeni=Xsayac1;
                else if(Xsayac1==10 && Xsayac2==-1) Xyeni=90;
                else if(Xsayac1==10 && Xsayac2==10) Xyeni=0;
                else if(Xsayac1==-1) Xyeni=(10*Xsayac2)+9;
                else if(Xsayac1==10) Xyeni=10*Xsayac2;
                else if(Xsayac2==-1) Xyeni=Xsayac1+90;
                else if(Xsayac2==10) Xyeni=Xsayac1;
                else  Xyeni=(10*Xsayac2)+Xsayac1;}

if(P3_6==1){
if(Xsayac1==0) Xsayac1=9;
else if(Xsayac1==-1) Xsayac1=8;
else Xsayac1--;
switch (Xsayac1){
        case 9: X1DOKUZ;
           if(Xsayac2==0) Xsayac2=9;
           else if(Xsayac1==-1) Xsayac1=8;
            else Xsayac2--;
                switch(Xsayac2){
                        case 9: X10DOKUZ; break;
                        case 8: X10SEKIZ; break;
                        case 7: X10YEDI; break;
                        case 6: X10ALTI; break;
                        case 5: X10BES; break;
```

```
                    case 4: X10DORT; break;
                    case 3: X10UC; break;
                    case 2: X10IKI; break;
                    case 1: X10BIR; break;
                    case 0: X10SIFIR; Xsayac2=10; break;} break;

        case 8: X1SEKIZ; break;
        case 7: X1YEDI; break;
        case 6: X1ALTI; break;
        case 5: X1BES; break;
        case 4: X1DORT; break;
        case 3: X1UC; break;
        case 2: X1IKI; break;
        case 1: X1BIR; break;
        case 0: X1SIFIR; Xsayac1=10; break;} bekle();
                if(Xsayac1==-1 && Xsayac2==-1) Xyeni=99;
                else if(Xsayac1==-1 && Xsayac2==10) Xyeni=Xsayac1;
                else if(Xsayac1==10 && Xsayac2==-1) Xyeni=90;
                else if(Xsayac1==10 && Xsayac2==10) Xyeni=0;
                else if(Xsayac1==-1) Xyeni=(10*Xsayac2)+9;
                else if(Xsayac1==10) Xyeni=10*Xsayac2;
                else if(Xsayac2==-1) Xyeni=Xsayac1+90;
                else if(Xsayac2==10) Xyeni=Xsayac1;
                else  Xyeni=(10*Xsayac2)+Xsayac1;}

if(P3_7==1){
if(Xyeni>Xeski){
        Xdurum=Xyeni-Xeski;Xeski=Xyeni;
if(Xkonum==2)goto XIA4; else goto XIA2;
        while(1){
XIA2:           XADIM1
                motorbekle();
                XADIM2
```

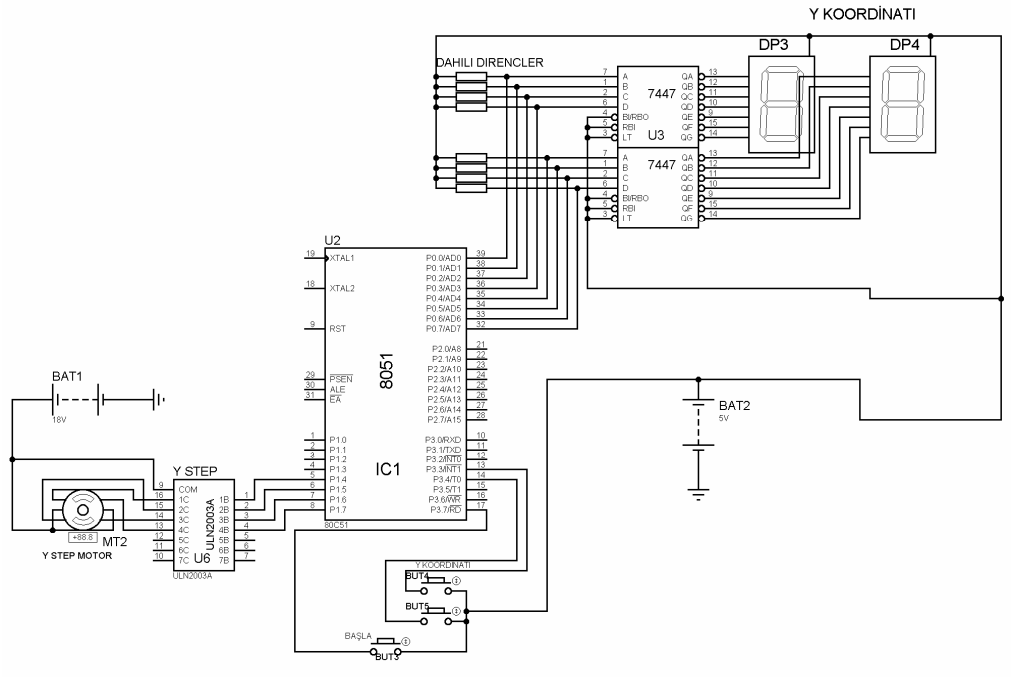
```
                motorbekle();
Xdurum--;if(Xdurum==0){Xkonum=2; break;}
XIA4:           XADIM3
                motorbekle();
                XADIM4
                motorbekle();
Xdurum--;if(Xdurum==0){Xkonum=4; break;}}}
P1=0;
if(Xyeni<Xeski){
        Xdurum=Xeski-Xyeni;Xeski=Xyeni;
if(Xkonum==2)goto XGA4; else goto XGA2;
        while(1){
XGA2:           XADIM3
                motorbekle();
                XADIM2
                motorbekle();
Xdurum--;if(Xdurum==0){Xkonum=2; break;}
XGA4:           XADIM1
                motorbekle();
                XADIM4
                motorbekle();
Xdurum--;if(Xdurum==0){Xkonum=4; break;}}}P1=0;}
        }
}
```

- Adım motorun sayıcıyla paralel olarak çalışmasının doğrulanması : Yukarıdaki işlemlerin yapılmasından sonra motorun çeşitli değerlerde ileri ve geri dönmesi için komutlar verildi. Adım motorun verilen bütün komutları eksiksiz bir şekilde yerine getirdiği görüldü.

5.3.1.2. Y koordinatının gerçeklenmesi

Y koordinatının gerçeklenmesi için yapılan işlemler X koordinatı için yapılan işlemlerin aynısıdır. Fakat aralarındaki tek fark portların farklı uçlarının kullanılmasıdır.

Göstergeler için Port0, adım motor için ise Port1'in son dört ucu kullanılmaktadır. Aşağıda Y koordinatı için tasarlanmış devre şeması ve yazılımı bulunmaktadır.



Şekil 5.15. Y koordinatı için devre şeması

- Y koordinatı için gereken yazılım:

```
#include <AT89X51.H>

#define Y10SIFIR P0_3=0,P0_2=0,P0_1=0,P0_0=0;
#define Y10BIR   P0_3=0,P0_2=0,P0_1=0,P0_0=1;
#define Y10IKI   P0_3=0,P0_2=0,P0_1=1,P0_0=0;
#define Y10UC    P0_3=0,P0_2=0,P0_1=1,P0_0=1;
```

```
#define Y10DORT  P0_3=0,P0_2=1,P0_1=0,P0_0=0;
#define Y10BES   P0_3=0,P0_2=1,P0_1=0,P0_0=1;
#define Y10ALTI  P0_3=0,P0_2=1,P0_1=1,P0_0=0;
#define Y10YEDI  P0_3=0,P0_2=1,P0_1=1,P0_0=1;
#define Y10SEKIZ P0_3=1,P0_2=0,P0_1=0,P0_0=0;
#define Y10DOKUZ P0_3=1,P0_2=0,P0_1=0,P0_0=1;

#define Y1SIFIR P0_7=0,P0_6=0,P0_5=0,P0_4=0;
#define Y1BIR   P0_7=0,P0_6=0,P0_5=0,P0_4=1;
#define Y1IKI   P0_7=0,P0_6=0,P0_5=1,P0_4=0;
#define Y1UC    P0_7=0,P0_6=0,P0_5=1,P0_4=1;
#define Y1DORT  P0_7=0,P0_6=1,P0_5=0,P0_4=0;
#define Y1BES   P0_7=0,P0_6=1,P0_5=0,P0_4=1;
#define Y1ALTI  P0_7=0,P0_6=1,P0_5=1,P0_4=0;
#define Y1YEDI  P0_7=0,P0_6=1,P0_5=1,P0_4=1;
#define Y1SEKIZ P0_7=1,P0_6=0,P0_5=0,P0_4=0;
#define Y1DOKUZ P0_7=1,P0_6=0,P0_5=0,P0_4=1;

#define YADIM1 P1_7=1,P1_6=0,P1_5=0,P1_4=0;
#define YADIM2 P1_7=0,P1_6=1,P1_5=0,P1_4=0;
#define YADIM3 P1_7=0,P1_6=0,P1_5=1,P1_4=0;
#define YADIM4 P1_7=0,P1_6=0,P1_5=0,P1_4=1;

void bekle(void){
        unsigned int sayac= 5669;
        while(sayac--);}

void motorbekle(void){
        unsigned int sayac=9567;
        while(sayac--);}

main(){
unsigned char Ysayac1=0,Ysayac2=0,Yyeni=0,Yeski=0,Ydurum=0, Ygor, Ykonum;
```

```
P0=0;
while(1){
P3=0;
if(P3_3==1){
        if(Ysayac1==10) Ysayac1=1;
        else if(Ysayac1==9) Ysayac1=0;
        else Ysayac1++;
        switch (Ysayac1){
        case 0: Y1SIFIR;
           if(Ysayac2==10) Ysayac2=1;
           else if(Ysayac2==9) Ysayac2=0;
           else Ysayac2++;
                switch(Ysayac2){
                        case 0: Y10SIFIR; break;
                        case 1: Y10BIR; break;
                        case 2: Y10IKI; break;
                        case 3: Y10UC; break;
                        case 4: Y10DORT; break;
                        case 5: Y10BES; break;
                        case 6: Y10ALTI; break;
                        case 7: Y10YEDI; break;
                        case 8: Y10SEKIZ; break;
                        case 9: Y10DOKUZ; Ysayac2=-1; break;} break;
        case 1: Y1BIR; break;
        case 2: Y1IKI; break;
        case 3: Y1UC; break;
        case 4: Y1DORT; break;
        case 5: Y1BES; break;
        case 6: Y1ALTI; break;
        case 7: Y1YEDI; break;
        case 8: Y1SEKIZ; break;
        case 9: Y1DOKUZ; Ysayac1=-1; break;} bekle();
                if(Ysayac1==-1 && Ysayac2==-1) Yyeni=99;
```

```
                else if(Ysayac1==-1 && Ysayac2==10) Yyeni =Ysayac1;
                else if(Ysayac1==10 && Ysayac2==-1) Yyeni =90;
                else if(Ysayac1==10 && Ysayac2==10) Yyeni =0;
                else if(Ysayac1==-1) Yyeni =(10*Ysayac2)+9;
                else if(Ysayac1==10) Yyeni =10*Ysayac2;
                else if(Ysayac2==-1) Yyeni =Ysayac1+90;
                else if(Ysayac2==10) Yyeni =Ysayac1;
                else  Yyeni =(10*Ysayac2)+Ysayac1;}

if(P3_4==1){
if(Ysayac1==0) Ysayac1=9;
else if(Ysayac1==-1) Ysayac1=8;
else Ysayac1--;
        switch (Ysayac1){
        case 9: Y1DOKUZ;
           if(Ysayac2==0) Ysayac2=9;
           else if(Ysayac1==-1) Ysayac1=8;
           else Ysayac2--;
                        switch(Ysayac2){
                        case 9: Y10DOKUZ; break;
                        case 8: Y10SEKIZ; break;
                        case 7: Y10YEDI; break;
                        case 6: Y10ALTI; break;
                        case 5: Y10BES; break;
                        case 4: Y10DORT; break;
                        case 3: Y10UC; break;
                        case 2: Y10IKI; break;
                        case 1: Y10BIR; break;
                        case 0: Y10SIFIR; Ysayac2=10; break;} break;
        case 8: Y1SEKIZ; break;
        case 7: Y1YEDI; break;
        case 6: Y1ALTI; break;
        case 5: Y1BES; break;
```

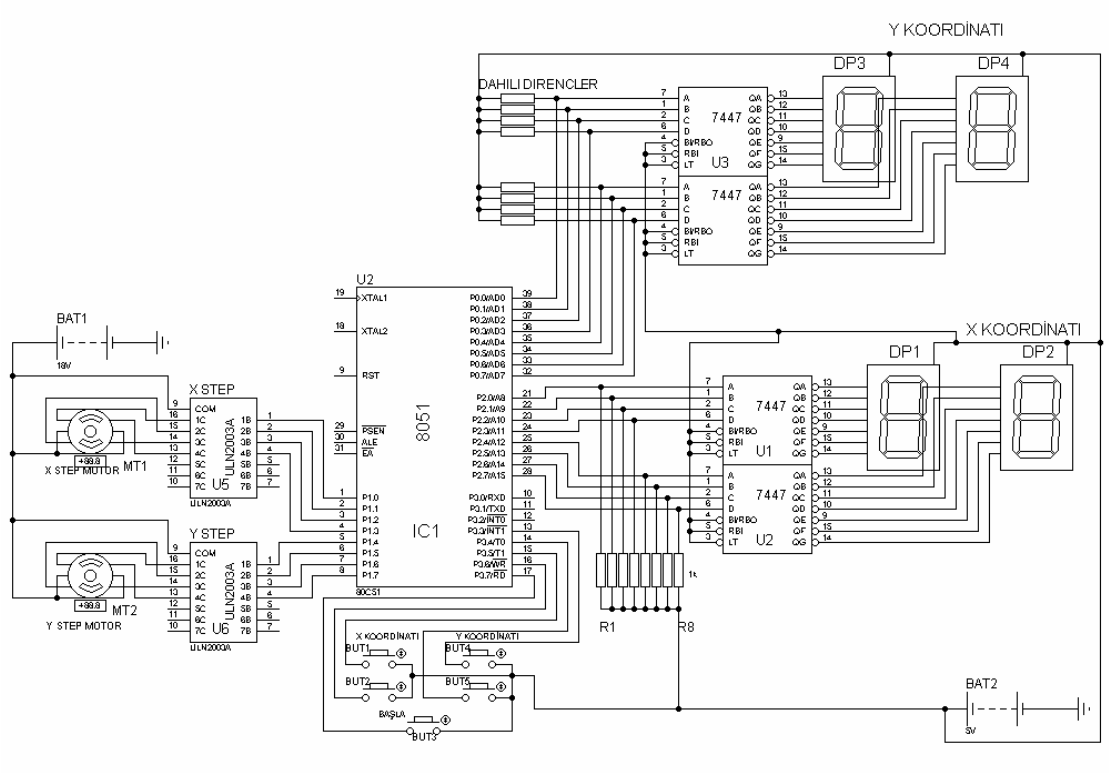
```
        case 4: Y1DORT; break;
        case 3: Y1UC; break;
        case 2: Y1IKI; break;
        case 1: Y1BIR; break;
        case 0: Y1SIFIR; Ysayac1=10; break;} bekle();
                if(Ysayac1==-1 && Ysayac2==-1) Ygor=99;
                else if(Ysayac1==-1 && Ysayac2==10) Yyeni =Ysayac1;
                else if(Ysayac1==10 && Ysayac2==-1) Yyeni =90;
                else if(Ysayac1==10 && Ysayac2==10) Yyeni =0;
                else if(Ysayac1==-1) Yyeni =(10*Ysayac2)+9;
                else if(Ysayac1==10) Yyeni =10*Ysayac2;
                else if(Ysayac2==-1) Yyeni =Ysayac1+90;
                else if(Ysayac2==10) Yyeni =Ysayac1;
                else  Yyeni =(10*Ysayac2)+Ysayac1;}

if(P3_7==1){
if(Yyeni>Yeski){
Ydurum=Yyeni-Yeski;Yeski=Yyeni;
if(Ykonum==2)goto YIA4; else goto YIA2;
while(1){
YIA2:           YADIM1
                motorbekle();
                YADIM2
                motorbekle();
Ydurum--;if(Ydurum==0){Ykonum=2; break;}
YIA4:           YADIM3
                motorbekle();
                YADIM4
                motorbekle();
Ydurum--;if(Ydurum==0){Ykonum=4; break;}}}
P1=0;
if(Yyeni<Yeski){
Ydurum=Yeski-Yyeni;Yeski=Yyeni;
```

```
if(Ykonum==2)goto YGA4; else goto YGA2;
while(1){
YGA2:       YADIM3
            motorbekle();
            YADIM2
            motorbekle();
Ydurum--;if(Ydurum==0){Ykonum=2; break;}
YGA4:       YADIM1
            motorbekle();
            YADIM4
            motorbekle();
Ydurum--;if(Ydurum==0){Ykonum=4; break;}}}P1=0;}
}}
```

5.3.1.3. X ve Y koordinatlarının tek devrede gerçeklenmesi

X ve Y koordinatları teker teker gerçekleştirildikten sonra yapılacak iş ikisini bir anda gerçekleştirebilecek bir düzeneğin ve yazılımın gerçekleştirilmesidir. İki sistemin birleştirilmesi için devre şemalarında hiçbir değişiklik yapmadan birleştirilmesi yeterli oldu.

Aşağıdaki şemada bu durum görülmektedir.



Şekil 5.16. X ve Y koordinatları için devre şeması

- X ve Y koordinatlarının birlikte çalıştıkları yazılım:

```
#include <AT89X51.H>

#define X10SIFIR P2_3=0,P2_2=0,P2_1=0,P2_0=0;
#define X10BIR   P2_3=0,P2_2=0,P2_1=0,P2_0=1;
#define X10IKI   P2_3=0,P2_2=0,P2_1=1,P2_0=0;
#define X10UC    P2_3=0,P2_2=0,P2_1=1,P2_0=1;
#define X10DORT  P2_3=0,P2_2=1,P2_1=0,P2_0=0;
#define X10BES   P2_3=0,P2_2=1,P2_1=0,P2_0=1;
#define X10ALTI  P2_3=0,P2_2=1,P2_1=1,P2_0=0;
#define X10YEDI  P2_3=0,P2_2=1,P2_1=1,P2_0=1;
#define X10SEKIZ P2_3=1,P2_2=0,P2_1=0,P2_0=0;
#define X10DOKUZ P2_3=1,P2_2=0,P2_1=0,P2_0=1;
```

```
#define X1SIFIR P2_7=0,P2_6=0,P2_5=0,P2_4=0;
#define X1BIR   P2_7=0,P2_6=0,P2_5=0,P2_4=1;
#define X1IKI   P2_7=0,P2_6=0,P2_5=1,P2_4=0;
#define X1UC    P2_7=0,P2_6=0,P2_5=1,P2_4=1;
#define X1DORT  P2_7=0,P2_6=1,P2_5=0,P2_4=0;
#define X1BES   P2_7=0,P2_6=1,P2_5=0,P2_4=1;
#define X1ALTI  P2_7=0,P2_6=1,P2_5=1,P2_4=0;
#define X1YEDI  P2_7=0,P2_6=1,P2_5=1,P2_4=1;
#define X1SEKIZ P2_7=1,P2_6=0,P2_5=0,P2_4=0;
#define X1DOKUZ P2_7=1,P2_6=0,P2_5=0,P2_4=1;

#define XADIM1 P1_3=1,P1_2=0,P1_1=0,P1_0=0;
#define XADIM2 P1_3=0,P1_2=1,P1_1=0,P1_0=0;
#define XADIM3 P1_3=0,P1_2=0,P1_1=1,P1_0=0;
#define XADIM4 P1_3=0,P1_2=0,P1_1=0,P1_0=1;

#define Y10SIFIR P0_3=0,P0_2=0,P0_1=0,P0_0=0;
#define Y10BIR   P0_3=0,P0_2=0,P0_1=0,P0_0=1;
#define Y10IKI   P0_3=0,P0_2=0,P0_1=1,P0_0=0;
#define Y10UC    P0_3=0,P0_2=0,P0_1=1,P0_0=1;
#define Y10DORT  P0_3=0,P0_2=1,P0_1=0,P0_0=0;
#define Y10BES   P0_3=0,P0_2=1,P0_1=0,P0_0=1;
#define Y10ALTI  P0_3=0,P0_2=1,P0_1=1,P0_0=0;
#define Y10YEDI  P0_3=0,P0_2=1,P0_1=1,P0_0=1;
#define Y10SEKIZ P0_3=1,P0_2=0,P0_1=0,P0_0=0;
#define Y10DOKUZ P0_3=1,P0_2=0,P0_1=0,P0_0=1;

#define Y1SIFIR P0_7=0,P0_6=0,P0_5=0,P0_4=0;
#define Y1BIR   P0_7=0,P0_6=0,P0_5=0,P0_4=1;
#define Y1IKI   P0_7=0,P0_6=0,P0_5=1,P0_4=0;
#define Y1UC    P0_7=0,P0_6=0,P0_5=1,P0_4=1;
#define Y1DORT  P0_7=0,P0_6=1,P0_5=0,P0_4=0;
```

```
#define Y1BES   P0_7=0,P0_6=1,P0_5=0,P0_4=1;
#define Y1ALTI  P0_7=0,P0_6=1,P0_5=1,P0_4=0;
#define Y1YEDI  P0_7=0,P0_6=1,P0_5=1,P0_4=1;
#define Y1SEKIZ P0_7=1,P0_6=0,P0_5=0,P0_4=0;
#define Y1DOKUZ P0_7=1,P0_6=0,P0_5=0,P0_4=1;

#define YADIM1 P1_7=1,P1_6=0,P1_5=0,P1_4=0;
#define YADIM2 P1_7=0,P1_6=1,P1_5=0,P1_4=0;
#define YADIM3 P1_7=0,P1_6=0,P1_5=1,P1_4=0;
#define YADIM4 P1_7=0,P1_6=0,P1_5=0,P1_4=1;

void bekle(void){
        unsigned int sayac= 5669;
        while(sayac--);}

void motorbekle(void){
        unsigned int sayac=9567;
        while(sayac--);}

main(){
unsigned
char Xsayac1=0,Xsayac2=0,Xyeni=0,Xeski=0,Xdurum=0,Ysayac1=0,Ysayac2=0,
Yyeni=0,Yeski=0,Ydurum=0,Xkonum,Ykonum;
unsigned int s,d;
P2=0,P0=0;
while(1){
P3=0;

if(P3_5==1){
if(Xsayac1==10) Xsayac1=1;
else if(Xsayac1==9) Xsayac1=0;
else Xsayac1++;
        switch (Xsayac1){
```

```
case 0: X1SIFIR;
   if(Xsayac2==10) Xsayac2=1;
  else if(Xsayac2==9) Xsayac2=0;
  else Xsayac2++;
        switch(Xsayac2){
              case 0: X10SIFIR; break;
              case 1: X10BIR; break;
              case 2: X10IKI; break;
              case 3: X10UC; break;
              case 4: X10DORT; break;
              case 5: X10BES; break;
              case 6: X10ALTI; break;
              case 7: X10YEDI; break;
              case 8: X10SEKIZ; break;
              case 9: X10DOKUZ; Xsayac2=-1; break;} break;
case 1: X1BIR; break;
case 2: X1IKI; break;
case 3: X1UC; break;
case 4: X1DORT; break;
case 5: X1BES; break;
case 6: X1ALTI; break;
case 7: X1YEDI; break;
case 8: X1SEKIZ; break;
case 9: X1DOKUZ; Xsayac1=-1; break;} bekle();
        if(Xsayac1==-1 && Xsayac2==-1) Xyeni=99;
        else if(Xsayac1==-1 && Xsayac2==10) Xyeni =Xsayac1;
        else if(Xsayac1==10 && Xsayac2==-1) Xyeni =90;
        else if(Xsayac1==10 && Xsayac2==10) Xyeni =0;
        else if(Xsayac1==-1) Xyeni =(10*Xsayac2)+9;
        else if(Xsayac1==10) Xyeni =10*Xsayac2;
        else if(Xsayac2==-1) Xyeni =Xsayac1+90;
        else if(Xsayac2==10) Xyeni =Xsayac1;
        else  Xyeni =(10*Xsayac2)+Xsayac1;}
```

```
if(P3_6==1){
if(Xsayac1==0) Xsayac1=9;
else if(Xsayac1==-1) Xsayac1=8;
else Xsayac1--;
        switch (Xsayac1){
        case 9: X1DOKUZ;
           if(Xsayac2==0) Xsayac2=9;
          else if(Xsayac1==-1) Xsayac1=8;
          else Xsayac2--;
                switch(Xsayac2){
                        case 9: X10DOKUZ; break;
                        case 8: X10SEKIZ; break;
                        case 7: X10YEDI; break;
                        case 6: X10ALTI; break;
                        case 5: X10BES; break;
                        case 4: X10DORT; break;
                        case 3: X10UC; break;
                        case 2: X10IKI; break;
                        case 1: X10BIR; break;
                        case 0: X10SIFIR; Xsayac2=10; break;} break;
        case 8: X1SEKIZ; break;
        case 7: X1YEDI; break;
        case 6: X1ALTI; break;
        case 5: X1BES; break;
        case 4: X1DORT; break;
        case 3: X1UC; break;
        case 2: X1IKI; break;
        case 1: X1BIR; break;
        case 0: X1SIFIR; Xsayac1=10; break;} bekle();
                if(Xsayac1==-1 && Xsayac2==-1) Xyeni=99;
                else if(Xsayac1==-1 && Xsayac2==10) Xyeni =Xsayac1;
                else if(Xsayac1==10 && Xsayac2==-1) Xyeni =90;
```

```
                else if(Xsayac1==10 && Xsayac2==10) Xyeni =0;
                else if(Xsayac1==-1) Xyeni =(10*Xsayac2)+9;
                else if(Xsayac1==10) Xyeni =10*Xsayac2;
                else if(Xsayac2==-1) Xyeni =Xsayac1+90;
                else if(Xsayac2==10) Xyeni =Xsayac1;
                else  Xyeni =(10*Xsayac2)+Xsayac1;}

if(P3_3==1){
if(Ysayac1==10) Ysayac1=1;
else if(Ysayac1==9) Ysayac1=0;
else Ysayac1++;
        switch (Ysayac1){
        case 0: Y1SIFIR;
           if(Ysayac2==10) Ysayac2=1;
          else if(Ysayac2==9) Ysayac2=0;
          else Ysayac2++;
                switch(Ysayac2){
                        case 0: Y10SIFIR; break;
                        case 1: Y10BIR; break;
                        case 2: Y10IKI; break;
                        case 3: Y10UC; break;
                        case 4: Y10DORT; break;
                        case 5: Y10BES; break;
                        case 6: Y10ALTI; break;
                        case 7: Y10YEDI; break;
                        case 8: Y10SEKIZ; break;
                        case 9: Y10DOKUZ; Ysayac2=-1; break;} break;
        case 1: Y1BIR; break;
        case 2: Y1IKI; break;
        case 3: Y1UC; break;
        case 4: Y1DORT; break;
        case 5: Y1BES; break;
        case 6: Y1ALTI; break;
```

```
        case 7: Y1YEDI; break;
        case 8: Y1SEKIZ; break;
        case 9: Y1DOKUZ; Ysayac1=-1; break;} bekle();
                if(Ysayac1==-1 && Ysayac2==-1) Yyeni=99;
                else if(Ysayac1==-1 && Ysayac2==10) Yyeni =Ysayac1;
                else if(Ysayac1==10 && Ysayac2==-1) Yyeni =90;
                else if(Ysayac1==10 && Ysayac2==10) Yyeni =0;
                else if(Ysayac1==-1) Yyeni =(10*Ysayac2)+9;
                else if(Ysayac1==10) Yyeni =10*Ysayac2;
                else if(Ysayac2==-1) Yyeni =Ysayac1+90;
                else if(Ysayac2==10) Yyeni =Ysayac1;
                else  Yyeni =(10*Ysayac2)+Ysayac1;}

if(P3_4==1){
if(Ysayac1==0) Ysayac1=9;
else if(Ysayac1==-1) Ysayac1=8;
else Ysayac1--;
        switch (Ysayac1){
        case 9: Y1DOKUZ;
           if(Ysayac2==0) Ysayac2=9;
          else if(Ysayac1==-1) Ysayac1=8;
          else Ysayac2--;
                switch(Ysayac2){
                        case 9: Y10DOKUZ; break;
                        case 8: Y10SEKIZ; break;
                        case 7: Y10YEDI; break;
                        case 6: Y10ALTI; break;
                        case 5: Y10BES; break;
                        case 4: Y10DORT; break;
                        case 3: Y10UC; break;
                        case 2: Y10IKI; break;
                        case 1: Y10BIR; break;
                        case 0: Y10SIFIR; Ysayac2=10; break;} break;
```

```
        case 8: Y1SEKIZ; break;
        case 7: Y1YEDI; break;
        case 6: Y1ALTI; break;
        case 5: Y1BES; break;
        case 4: Y1DORT; break;
        case 3: Y1UC; break;
        case 2: Y1IKI; break;
        case 1: Y1BIR; break;
        case 0: Y1SIFIR; Ysayac1=10; break;} bekle();
                if(Ysayac1==-1 && Ysayac2==-1) Yyeni=99;
                else if(Ysayac1==-1 && Ysayac2==10) Yyeni =Ysayac1;
                else if(Ysayac1==10 && Ysayac2==-1) Yyeni =90;
                else if(Ysayac1==10 && Ysayac2==10) Yyeni =0;
                else if(Ysayac1==-1) Yyeni =(10*Ysayac2)+9;
                else if(Ysayac1==10) Yyeni =10*Ysayac2;
                else if(Ysayac2==-1) Yyeni =Ysayac1+90;
                else if(Ysayac2==10) Yyeni =Ysayac1;
                else  Yyeni =(10*Ysayac2)+Ysayac1;}

if(P3_7==1){
if(Xyeni>Xeski){
Xdurum=Xyeni-Xeski;Xeski=Xyeni;
if(Xkonum==2)goto XIA4; else goto XIA2;
while(1){
XIA2:           XADIM1
                motorbekle();
                XADIM2
                motorbekle();
Xdurum--;if(Xdurum==0){Xkonum=2; break;}
XIA4:           XADIM3
                motorbekle();
                XADIM4
```

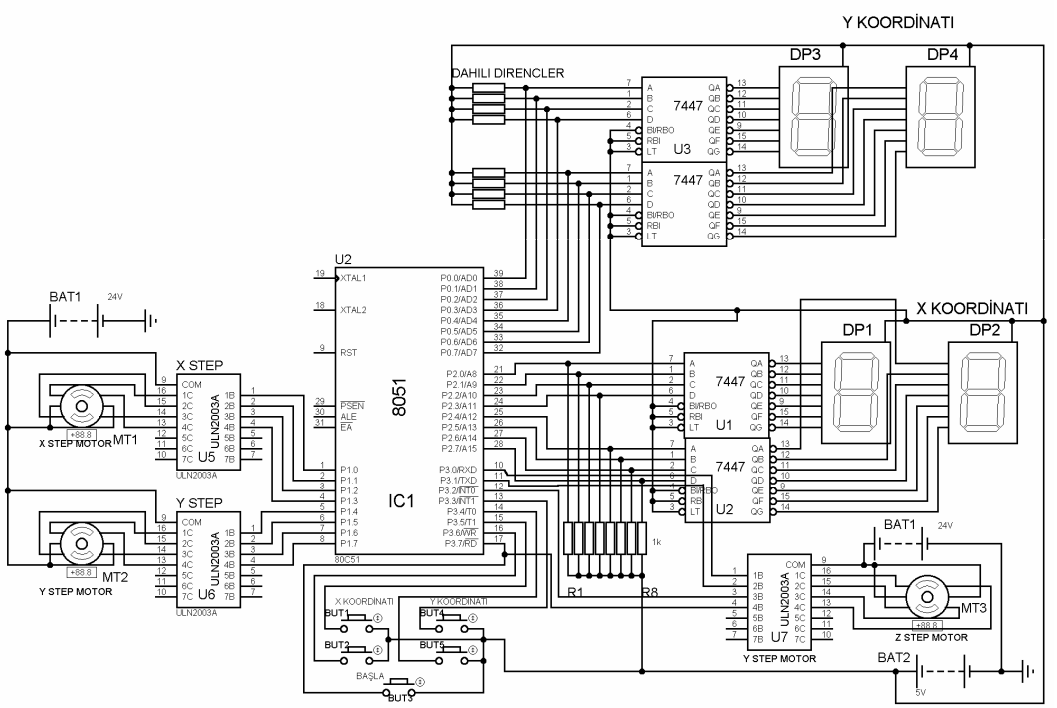
```
                motorbekle();
Xdurum--;if(Xdurum==0){Xkonum=4; break;}}}
P1=0;
if(Xyeni<Xeski){
Xdurum=Xeski-Xyeni;Xeski=Xyeni;
if(Xkonum==2)goto XGA4; else goto XGA2;
while(1){
XGA2:       XADIM3
                motorbekle();
                XADIM2
                motorbekle();
Xdurum--;if(Xdurum==0){Xkonum=2; break;}
XGA4:       XADIM1
                motorbekle();
                XADIM4
                motorbekle();
Xdurum--;if(Xdurum==0){Xkonum=4; break;}}}
P1=0;

if(Yyeni>Yeski){
Ydurum=Yyeni-Yeski;Yeski=Yyeni;
if(Ykonum==2)goto YIA4; else goto YIA2;
while(1){
YIA2:       YADIM1
                motorbekle();
                YADIM2
                motorbekle();
Ydurum--;if(Ydurum==0){Ykonum=2; break;}
YIA4:       YADIM3
                motorbekle();
                YADIM4
                motorbekle();
Ydurum--;if(Ydurum==0){Ykonum=4; break;}}}
```

```
P1=0;
if(Yyeni<Yeski){
Ydurum=Yeski-Yyeni;Yeski=Yyeni;
if(Ykonum==2)goto YGA4; else goto YGA2;
while(1){
YGA2:       YADIM3
            motorbekle();
            YADIM2
            motorbekle();
Ydurum--;if(Ydurum==0){Ykonum=2; break;}
YGA4:       YADIM1
            motorbekle();
            YADIM4
            motorbekle();
Ydurum--;if(Ydurum==0){Ykonum=4; break;}}}P1=0;}
        }}
```

5.3.2. X ve Y koordinatları girildikten sonra koordinata nokta koydurulması işleminin tasarlanması

Adım motorların X ve Y koordinatlarına gitmesi işlemi tamamlandı. Daha sonra başka bir adım motorun bir miktar ileri ve hemen arkasından da bir miktar geri gelmesiyle, motora bağlı bulunan bir kalemin düzeneğe doğru aşağı ve yukarı hareketleriyle nokta koyması işleminin yaptırılmasına sıra geldi. Bu işlem 8051' in port3'ünün 0, 1, 2 ve port yetersizliğinden dolayı başlatma butonu için kullanılan 7 numaralı uçlar kullanılarak yapıldı. Burada adım motora bağlı kalemin düzeneğe doğru gitme mesafesi 290*4=1160 adım ile geri dönüşü ise yine aynı 290*4=1160 adım ile sağlanmıştır. Adım sayılarının çok fazla olmasının nedeni, sistemde kullanılan sonsuz vida dişlisi mekanizması mantığıdır.

Motorun devreye bağlanmış şekli aşağıdaki gibidir.



Şekil 5.17. X ve Y koordinatlarına nokta koydurmak için devre şeması

Yazılımdaki farklılık ise sadece yazılım başına ve en sonuna aşağıdaki satırların eklenmesiyle sağlanmıştır.

Başına;

```
void motorbekle2(void){
        unsigned int sayac=500;
        while(sayac--);}

void motorbekle3(void){
        unsigned int sayac=50300;
        while(sayac--);}
```

Sonuna;

```
if(Yyeni<Yeski){
Ydurum=Yeski-Yyeni;Yeski=Yyeni;
```

```
if(Ykonum==2)goto YGA4; else goto YGA2;
while(1){
YGA2:       YADIM3
            motorbekle();
            YADIM2
            motorbekle();
Ydurum--;if(Ydurum==0){Ykonum=2; break;}
YGA4:       YADIM1
            motorbekle();
            YADIM4
            motorbekle();
Ydurum--;if(Ydurum==0){Ykonum=4; break;}}}P1=0;
while(s<290){
P3_0=1,P3_1=0,P3_7=0,P3_2=0;motorbekle2();
P3_0=0,P3_1=1,P3_7=0,P3_2=0;motorbekle2();
P3_0=0,P3_1=0,P3_7=1,P3_2=0;motorbekle2();
P3_0=0,P3_1=0,P3_7=0,P3_2=1;motorbekle2();
s=s+1;}
motorbekle3();
while(d<290){
P3_0=0,P3_1=0,P3_7=1,P3_2=0;motorbekle2();
P3_0=0,P3_1=1,P3_7=0,P3_2=0;motorbekle2();
P3_0=1,P3_1=0,P3_7=0,P3_2=0;motorbekle2();
P3_0=0,P3_1=0,P3_7=0,P3_2=1;motorbekle2();
d=d+1;}
s=0,d=0;
P3=0;}
}}
```

5.3.3. Dört koordinatın hafızaya depolanması işlemi

Koordinatların hafızaya alınması işlemi için dört koordinat kapasitesine sahip bir dizi oluşturuldu.

```
Char xilk[4]={0},yilk[4]={0},a=0;
```

Dizi yukarıdaki komut satırı ile oluşturuldu. Daha sonra butonlarla koordinatlar her girildiğinde, Port3' ün "0" numaralı ucuna bağlı bulunan bir başka butona basılarak koordinatlar hafızaya aldırıldı. Bu şekilde 4 adet koordinat hafızaya aldırılmış oldu. Bu işlemi yapmak için gerekli komut satırı aşağıda verilmiştir.

```
if(P3_0==1){
xilk[a]=Xgor;
yilk[a]=Ygor;a=a+1;motorbekle3(),motorbekle3();
}
```

Buradaki dizide dört adet depolama istendiği için dizi 4 elemanlı seçilmiştir. Eğer ki daha fazla istenirse dizideki eleman sayısı artırılarak depolanan koordinat sayısı artırılabilir.

5.3.4. Hafızaya depolanan koordinatların sıralanması işlemi

Hafızadaki koordinatların sıralanması işlemi, koordinatların birbirlerine olan uzaklıklarına göre gerçekleştirildi. Bu uzaklıkları bulmak için ise aşağıdaki formül kullanıldı.

Mesafe = $\sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2}$

İlk olarak (0, 0) noktasına en yakın olan koordinat bulundu. Daha sonra bulunan bu koordinat temel alınarak diğer koordinatların bu noktaya olan uzaklıkları bulunarak tekrar karşılaştırıldı. Yine en yakın olan nokta bulunduktan sonra kalan noktaların bu noktaya olan uzaklıkları bulunarak karşılaştırma yapıldı ve noktalar küçükten büyüğe doğru sıralanmış oldu.

Başlat komutunu veren butona basılmasıyla sıralanan koordinatlar, sırayla uygulanacak duruma gelmiş oldu. Kalem en son noktayı koyduktan sonra kaldığı

yerde bekleyecektir. Daha sonra başlat butonuna tekrar basıldığında kalemin tekrar başa dönüp aynı noktaları teker teker işlemesi zaman kaybına neden olacağından dolayı tekrar bir sıralama işlemi yaptırıldı. Yani bu sıralama işlemi, başlat butonuna basılmasıyla gerçekleşmektedir. Bunun sebebi ise kalem nerde bulunursa bulunsun tekrar noktalama işlemine geçtiğinde kendine en yakın olandan başlamasının istenmesidir. Sıralama işleminin gerçekleştiği program satırları aşağıda verilmiştir. Bu işlemde kullanılan değişkenler ilk önce yukarıda bahsi geçtiği gibi göstergeler de görünen değerler Xgor ve Ygor değişkenlerine atanmaktadır. Sonra bunlar hafızaya aldırma işleminde Xilk[4] ve Yilk[4] değişkenlerine atanmaktadır. Bu değişkenler de sıralama işlemine tabi tutulduktan sonra da Xson[4] ve Yson[4] değişkenleri olarak son halini almaktadırlar.

Komut satırları ise şöyledir:

```
mesafe[0] = sqrt( (xilk[0]-Xeski)*(xilk[0]-Xeski)  +  (yilk[0]-Yeski)*(yilk[0]-Yeski)
); sayilar[0]=mesafe[0];
mesafe[1] = sqrt( (xilk[1]-Xeski)*(xilk[1]-Xeski)  +  (yilk[1]-Yeski)*(yilk[1]-Yeski)
); sayilar[1]=mesafe[1];
mesafe[2] = sqrt( (xilk[2]-Xeski)*(xilk[2]-Xeski)  +  (yilk[2]-Yeski)*(yilk[2]-Yeski)
); sayilar[2]=mesafe[2];
mesafe[3] = sqrt( (xilk[3]-Xeski)*(xilk[3]-Xeski)  +  (yilk[3]-Yeski)*(yilk[3]-Yeski)
); sayilar[3]=mesafe[3];

    for( buyuk=0; buyuk<3 ; buyuk++)
    {  for( kucuk = buyuk+1; kucuk<4; kucuk++)
            {  if( sayilar[buyuk] > sayilar[kucuk] )
                {     temp   = sayilar[buyuk];
                      sayilar[buyuk] = sayilar[kucuk];
                      sayilar[kucuk] = temp;}}}

if(sayilar[0]==mesafe[0]){ xson[0]=xilk[0],yson[0]=yilk[0];
                mesafe[1]  =  sqrt(  (xilk[1]-xilk[0])*(xilk[1]-xilk[0])  +  (yilk[1]-
yilk[0])*(yilk[1]-yilk[0]) ); sayilar[1]=mesafe[1];
```

```
                mesafe[2] = sqrt( (xilk[2]-xilk[0])*(xilk[2]-xilk[0]) + (yilk[2]-
yilk[0])*(yilk[2]-yilk[0]) ); sayilar[2]=mesafe[2];
                mesafe[3] = sqrt( (xilk[3]-xilk[0])*(xilk[3]-xilk[0]) + (yilk[3]-
yilk[0])*(yilk[3]-yilk[0]) ); sayilar[3]=mesafe[3];

for( buyuk=1; buyuk<3 ; buyuk++)
{   for( kucuk = buyuk+1; kucuk<4; kucuk++)
            {   if( sayilar[buyuk] > sayilar[kucuk] )
                {    temp   = sayilar[buyuk];
                     sayilar[buyuk] = sayilar[kucuk];
                     sayilar[kucuk] = temp;}}}

if(sayilar[1]==mesafe[1]){ xson[1]=xilk[1],yson[1]=yilk[1];
        mesafe[2] = sqrt( (xilk[2]-xilk[1])*(xilk[2]-xilk[1]) + (yilk[2]-
yilk[1])*(yilk[2]-yilk[1]) );
        mesafe[3] = sqrt( (xilk[3]-xilk[1])*(xilk[3]-xilk[1]) + (yilk[3]-
yilk[1])*(yilk[3]-yilk[1]) );

if(mesafe[2]<mesafe[3]){xson[2]=xilk[2],yson[2]=yilk[2],xson[3]=xilk[3],yson[3]=
yilk[3];}
else {xson[2]=xilk[3],yson[2]=yilk[3],xson[3]=xilk[2],yson[3]=yilk[2];}}

else if(sayilar[1]==mesafe[2]) {xson[1]=xilk[2],yson[1]=yilk[2];
        mesafe[1] = sqrt( (xilk[1]-xilk[2])*(xilk[1]-xilk[2]) + (yilk[1]-
yilk[2])*(yilk[1]-yilk[2]) );
        mesafe[3] = sqrt( (xilk[3]-xilk[2])*(xilk[3]-xilk[2]) + (yilk[3]-
yilk[2])*(yilk[3]-yilk[2]) );
if(mesafe[1]<mesafe[3]){xson[2]=xilk[1],yson[2]=yilk[1],xson[3]=xilk[3],yson[3]=y
ilk[3];}
else {xson[2]=xilk[3],yson[2]=yilk[3],xson[2]=xilk[1],yson[2]=yilk[1];}}

else if(sayilar[1]==mesafe[3]){ xson[1]=xilk[3],yson[1]=yilk[3];
```

```
        mesafe[2] = sqrt( (xilk[2]-xilk[3])*(xilk[2]-xilk[3]) + (yilk[2]-
yilk[3])*(yilk[2]-yilk[3]) );
        mesafe[1] = sqrt( (xilk[1]-xilk[3])*(xilk[1]-xilk[3]) + (yilk[1]-
yilk[3])*(yilk[1]-yilk[3]) );
if(mesafe[1]<mesafe[2]){xson[2]=xilk[1],yson[2]=yilk[1],xson[3]=xilk[2],yson[3]=y
ilk[2];}
else {xson[2]=xilk[2],yson[2]=yilk[2],xson[3]=xilk[1],yson[3]=yilk[1];}}}

else if(sayilar[0]==mesafe[1]) {xson[0]=xilk[1],yson[0]=yilk[1];
        mesafe[0] = sqrt( (xilk[1]-xilk[0])*(xilk[1]-xilk[0]) + (yilk[1]-
yilk[0])*(yilk[1]-yilk[0]) ); sayilar[1]=mesafe[0];
        mesafe[2] = sqrt( (xilk[2]-xilk[1])*(xilk[2]-xilk[1]) + (yilk[2]-
yilk[1])*(yilk[2]-yilk[1]) ); sayilar[2]=mesafe[2];
        mesafe[3] = sqrt( (xilk[3]-xilk[1])*(xilk[3]-xilk[1]) + (yilk[3]-
yilk[1])*(yilk[3]-yilk[1]) ); sayilar[3]=mesafe[3];

for( buyuk=1; buyuk<3 ; buyuk++)
{   for( kucuk = buyuk+1; kucuk<4; kucuk++)
{   if( sayilar[buyuk] > sayilar[kucuk] )
{     temp   = sayilar[buyuk];
sayilar[buyuk] = sayilar[kucuk];
sayilar[kucuk] = temp;}}}

if(sayilar[1]==mesafe[0]){ xson[1]=xilk[0],yson[1]=yilk[0];
        mesafe[2] = sqrt( (xilk[2]-xilk[0])*(xilk[2]-xilk[0]) + (yilk[2]-
yilk[0])*(yilk[2]-yilk[0]) );
        mesafe[3] = sqrt( (xilk[3]-xilk[0])*(xilk[3]-xilk[0]) + (yilk[3]-
yilk[0])*(yilk[3]-yilk[0]) );

if(mesafe[2]<mesafe[3]){xson[2]=xilk[2],yson[2]=yilk[2],xson[3]=xilk[3],yson[3]=y
ilk[3];}
else {xson[2]=xilk[3],yson[2]=yilk[3],xson[3]=xilk[2],yson[3]=yilk[2];}}
```

```
else if(sayilar[1]==mesafe[2]) {xson[1]=xilk[2],yson[1]=yilk[2];
        mesafe[0] = sqrt( (xilk[0]-xilk[2])*(xilk[0]-xilk[2]) + (yilk[0]-
yilk[2])*(yilk[0]-yilk[2]) );
        mesafe[3] = sqrt( (xilk[3]-xilk[2])*(xilk[3]-xilk[2]) + (yilk[3]-
yilk[2])*(yilk[3]-yilk[2]) );
if(mesafe[0]<mesafe[3]){xson[2]=xilk[0],yson[2]=yilk[0],xson[3]=xilk[3],yson[3]=y
ilk[3];}
else {xson[2]=xilk[3],yson[2]=yilk[3],xson[3]=xilk[0],yson[3]=yilk[0];}}

else if(sayilar[1]==mesafe[3]){ xson[1]=xilk[3],yson[1]=yilk[3];
        mesafe[2] = sqrt( (xilk[2]-xilk[3])*(xilk[2]-xilk[3]) + (yilk[2]-
yilk[3])*(yilk[2]-yilk[3]) );
        mesafe[1] = sqrt( (xilk[1]-xilk[3])*(xilk[1]-xilk[3]) + (yilk[1]-
yilk[3])*(yilk[1]-yilk[3]) );
if(mesafe[1]<mesafe[2]){xson[2]=xilk[1],yson[2]=yilk[1],xson[3]=xilk[2],yson[3]=y
ilk[2];}
else {xson[2]=xilk[2],yson[2]=yilk[2],xson[3]=xilk[1],yson[3]=yilk[1];}}}

else if(sayilar[0]==mesafe[2]){ xson[0]=xilk[2],yson[0]=yilk[2];
        mesafe[0] = sqrt( (xilk[2]-xilk[0])*(xilk[2]-xilk[0]) + (yilk[2]-
yilk[0])*(yilk[2]-yilk[0]) ); sayilar[1]=mesafe[0];
        mesafe[1] = sqrt( (xilk[2]-xilk[1])*(xilk[2]-xilk[1]) + (yilk[2]-
yilk[1])*(yilk[2]-yilk[1]) ); sayilar[2]=mesafe[1];
        mesafe[3] = sqrt( (xilk[3]-xilk[2])*(xilk[3]-xilk[2]) + (yilk[3]-
yilk[2])*(yilk[3]-yilk[2]) ); sayilar[3]=mesafe[3];

for( buyuk=1; buyuk<3 ; buyuk++)
{   for( kucuk = buyuk+1; kucuk<4; kucuk++)
{   if( sayilar[buyuk] > sayilar[kucuk] )
{     temp   = sayilar[buyuk];
sayilar[buyuk] = sayilar[kucuk];
sayilar[kucuk] = temp;}}}
```

```
if(sayilar[1]==mesafe[0]){ xson[1]=xilk[0],yson[1]=yilk[0];
        mesafe[1] = sqrt( (xilk[1]-xilk[0])*(xilk[1]-xilk[0]) + (yilk[1]-
yilk[0])*(yilk[1]-yilk[0]) );
        mesafe[3] = sqrt( (xilk[3]-xilk[0])*(xilk[3]-xilk[0]) + (yilk[3]-
yilk[0])*(yilk[3]-yilk[0]) );

if(mesafe[1]<mesafe[3]){xson[2]=xilk[1],yson[2]=yilk[1],xson[3]=xilk[3],yson[3]=y
ilk[3];}
else {xson[2]=xilk[3],yson[2]=yilk[3],xson[3]=xilk[1],yson[3]=yilk[1];}}

else if(sayilar[1]==mesafe[2]) {xson[1]=xilk[2],yson[1]=yilk[2];
        mesafe[0] = sqrt( (xilk[0]-xilk[1])*(xilk[0]-xilk[1]) + (yilk[0]-
yilk[1])*(yilk[0]-yilk[1]) );
        mesafe[3] = sqrt( (xilk[3]-xilk[1])*(xilk[3]-xilk[1]) + (yilk[3]-
yilk[1])*(yilk[3]-yilk[1]) );

if(mesafe[0]<mesafe[3]){xson[2]=xilk[0],yson[2]=yilk[0],xson[3]=xilk[3],yson[3]=y
ilk[3];}
else {xson[2]=xilk[3],yson[2]=yilk[3],xson[3]=xilk[0],yson[3]=yilk[0];}}

else if(sayilar[1]==mesafe[3]){ xson[1]=xilk[3],yson[1]=yilk[3];
        mesafe[0] = sqrt( (xilk[0]-xilk[3])*(xilk[0]-xilk[3]) + (yilk[0]-
yilk[3])*(yilk[0]-yilk[3]) );
        mesafe[1] = sqrt( (xilk[1]-xilk[3])*(xilk[1]-xilk[3]) + (yilk[1]-
yilk[3])*(yilk[1]-yilk[3]) );
if(mesafe[1]<mesafe[0]){xson[2]=xilk[1],yson[2]=yilk[1],xson[3]=xilk[0],yson[3]=y
ilk[0];}
else {xson[2]=xilk[0],yson[2]=yilk[0],xson[3]=xilk[1],yson[3]=yilk[1];}}}

else {xson[0]=xilk[3],yson[0]=yilk[3];
        mesafe[0] = sqrt( (xilk[3]-xilk[0])*(xilk[3]-xilk[0]) + (yilk[3]-
yilk[0])*(yilk[3]-yilk[0]) ); sayilar[1]=mesafe[0];
```

```
        mesafe[1] = sqrt( (xilk[3]-xilk[1])*(xilk[3]-xilk[1]) + (yilk[3]-
yilk[1])*(yilk[3]-yilk[1]) ); sayilar[2]=mesafe[1];
        mesafe[2] = sqrt( (xilk[3]-xilk[2])*(xilk[3]-xilk[2]) + (yilk[3]-
yilk[2])*(yilk[3]-yilk[2]) ); sayilar[3]=mesafe[2];

for( buyuk=1; buyuk<3 ; buyuk++)
{   for( kucuk = buyuk+1; kucuk<4; kucuk++)
{   if( sayilar[buyuk] > sayilar[kucuk] )
{     temp   = sayilar[buyuk];
sayilar[buyuk] = sayilar[kucuk];
sayilar[kucuk] = temp;}}}

if(sayilar[1]==mesafe[0]){ xson[1]=xilk[0],yson[1]=yilk[0];
        mesafe[1] = sqrt( (xilk[1]-xilk[0])*(xilk[1]-xilk[0]) + (yilk[1]-
yilk[0])*(yilk[1]-yilk[0]) );
        mesafe[2] = sqrt( (xilk[2]-xilk[0])*(xilk[2]-xilk[0]) + (yilk[2]-
yilk[0])*(yilk[2]-yilk[0]) );

if(mesafe[1]<mesafe[2]){xson[2]=xilk[1],yson[2]=yilk[1],xson[3]=xilk[2],yson[3]=y
ilk[2];}
else {xson[2]=xilk[2],yson[2]=yilk[2],xson[3]=xilk[1],yson[3]=yilk[1];}}

else if(sayilar[1]==mesafe[1]) {xson[1]=xilk[1],yson[1]=yilk[1];
        mesafe[0] = sqrt( (xilk[0]-xilk[1])*(xilk[0]-xilk[1]) + (yilk[0]-
yilk[1])*(yilk[0]-yilk[1]) );
        mesafe[2] = sqrt( (xilk[2]-xilk[1])*(xilk[2]-xilk[1]) + (yilk[2]-
yilk[1])*(yilk[2]-yilk[1]) );
if(mesafe[0]<mesafe[2]){xson[2]=xilk[0],yson[2]=yilk[0],xson[3]=xilk[2],yson[3]=y
ilk[2];}
else {xson[2]=xilk[2],yson[2]=yilk[2],xson[3]=xilk[0],yson[3]=yilk[0];}}

else if(sayilar[1]==mesafe[2]){ xson[1]=xilk[2],yson[1]=yilk[2];
```

```
        mesafe[0] = sqrt( (xilk[0]-xilk[2])*(xilk[0]-xilk[2]) + (yilk[0]-
yilk[2])*(yilk[0]-yilk[2]) );
        mesafe[1] = sqrt( (xilk[1]-xilk[2])*(xilk[1]-xilk[2]) + (yilk[1]-
yilk[2])*(yilk[1]-yilk[2]) );
if(mesafe[1]<mesafe[0]){xson[2]=xilk[1],yson[2]=yilk[1],xson[3]=xilk[0],yson[3]=y
ilk[0];}
else {xson[2]=xilk[0],yson[2]=yilk[0],xson[3]=xilk[1],yson[3]=yilk[1];}}}
```

5.3.5. Motorların başlangıç noktasına geri döndürülmesi işleminin gerçekleştirilmesi

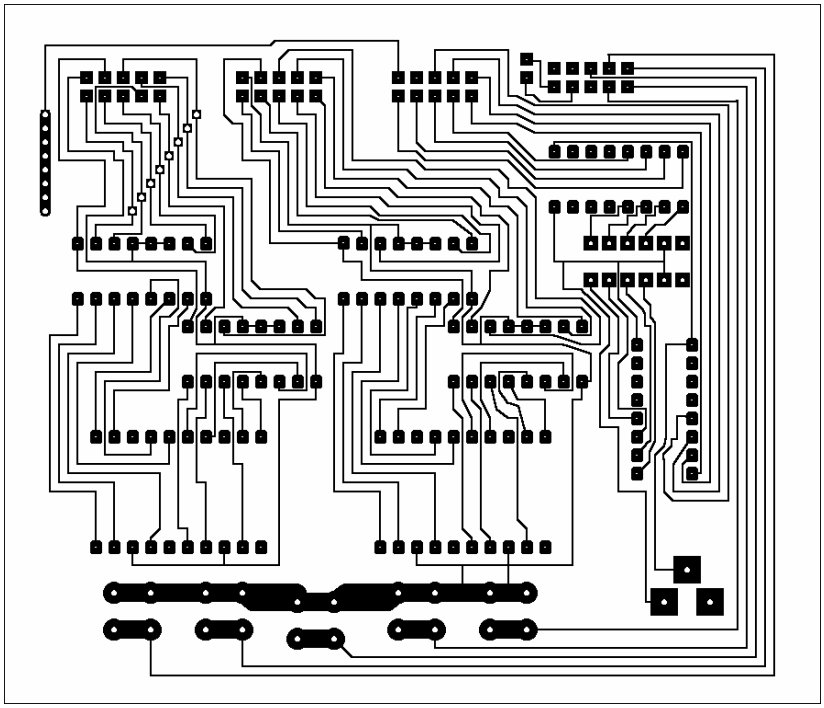
Motorların bulunduğu konum Xeski ve Yeski değişkenleriyle hafızada tutulmaktadır. Yeni gelen koordinat değerleri ise Xyeni ve Yyeni değişkenleriyle gelmektedir. Eğer yeni bir koordinat değeri gelirse Xyeni, Yyeni değerleri Xeski, Yeski değerleriyle karşılaştırılarak gereken hareket gerçekleştirilir. Motorların tekrar (0, 0) noktasına gelmeleri için program içinde, dışarıdan bir müdahale ile, Xyeni ve Yyeni değişkenlerine (0, 0) koordinat değeri atanarak motorlara gönderilir. Bu değer Xeski ve Yeski değerleriyle karşılaştırılarak gereken hareket yapılır.

5.3.6. Sistemin bilgisayar ortamında doğru olarak çalışmasının doğrulanması

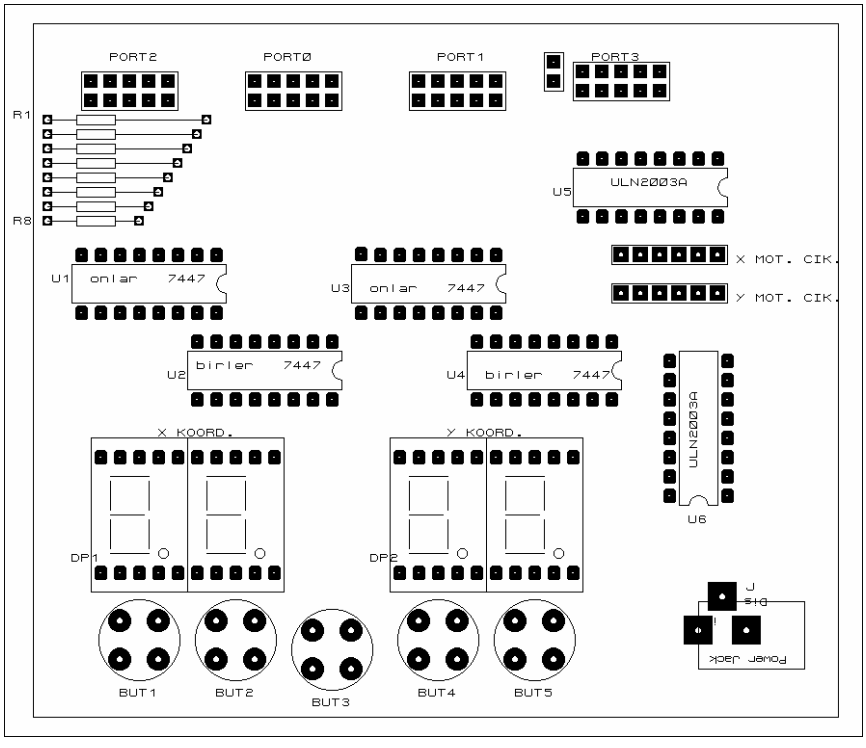
Yukarıdaki işlemlerin yapılmasından sonra girilen 4 adet koordinatın depolanması işlemi yapıldı. Başlat butonuna basılmasıyla birlikte koordinatların sıralanması işlemi ve motorların da bu sıraya göre hareketlerini gerçekleştirmesi gözlendi. Sanal ortamda sistem istenen şekilde çalıştırıldı.

5.3.7. Devre şemasının baskı devresinin çıkartılması

Devre şemasının baskı devresinin çizilmesi işlemi ISIS programında yapıldı. Çizilen baskı devre şeması ve devrenin üst görünüşü aşağıda görülmektedir.

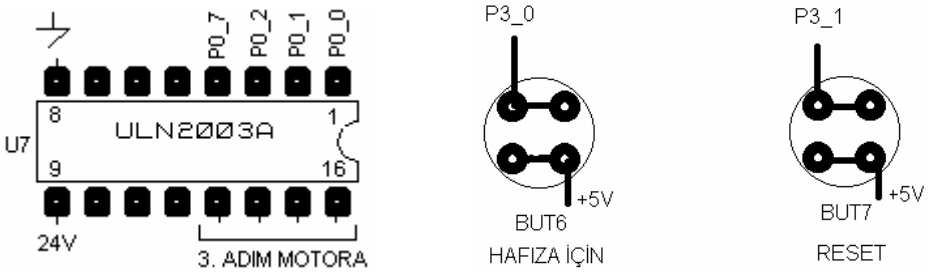


Şekil 5.18. Baskı devre şeması



Şekil 5.19. Devrenin üstten görünüşü

Devreye daha sonradan da eklenen birimler olduğundan dolayı bu ekler aşağıda gösterilmiştir. Bu birimler 4 koordinatın hafızada saklanması, kalemin başlangıca getirilmesi, nokta koydurma işleminin bir adım motora yaptırılması işlemleridir.



Şekil 5.20. Sonradan eklenen kısım

BÖLÜM 6. SİSTEMİN GERÇEKLENMESİ

6.1. Baskı Devrenin Çıkartılması

Baskı devre şemasını plaket üzerine çıkartmak için bir çok yöntem vardır. Burada kullanılan yöntem ise ütü yöntemi olmuştur. Ütü yönteminin bir çok avantajı vardır. En büyük avantajı maliyetinin düşük olmasına karşılık çok kaliteli bir sonuç alınmasıdır. Bu yöntemde ilk yapılacak işlem bilgisayarda çizilen şemanın bir lazer yazıcı vasıtasıyla, piyasada reklam kağıdı diye geçen üzeri çok ince bir naylonla kaplı kuşe kağıdına benzeyen bir kağıt üzerine çıktı almak oldu. Elde edilen çıktı uygun ebatlardaki bir plaketin üzerine kapatıldı. Sonra plaket sert ve düz bir zemine konup daha önceden iyice ısıtılan ütü ile bastırılarak ütülendi. Bir miktar ütüledikten sonra plaket alınıp bir musluğun altında su ile ıslatıldı. Sonra akan suyun altında parmaklarla plaket üzerindeki kağıda yavaş yavaş sürtülerek kağıt tabaka kaldırıldı. Bu işlemler bittikten sonra artık baskı devre şeması plaket üzerine çıkmış oldu. Plaket belirli oranlarda yapılan tuz ruhu peridrol karışıma atıldı. Plaket karışım içendeyken kap yavaş yavaş sallandı. Bu esnada plaket üzerindeki çizim dışında kalan bölgeler asit vasıtasıyla yavaş yavaş buharlaştırıldı. Buharlaşma işlemi bittikten sonra plaket su ile yıkandı ve üzeri reçine tabakası ile kaplandı. Devre plaketi artık hazır hale getirilmiş oldu.

6.2. Adım Motorların Karta Bağlanması

Malzemelerin kart üzerine lehimlenmesinden sonra mekanik bir sistemde çalışacak olan adım motorlar için kart üzerinde ayrılmış olan bölümlere motorların bağlanması işlemine sıra geldi. Adım motorların üzerinde bulunan 5 adet ucun bağlanması belli bir sıraya göre yapılmak zorundadır. Bu kabloların sırasını bulmak için bazı işlemlerin yapılması gerektir. Bu işlemler şu şekilde olmaktadır. Adım motorun kablolarından bir veya iki tanesi ortaktır (Vmotor). Yapılan işlem basit olarak bu

ortak kabloya sürekli +24 Volt göndermek ve diğer uçları ise belli bir sırada toprağa göndererek bir adım hareketi elde etmektir. Adım motor sürücüsü olarak ULN2003A entegresi kullanıldı. Sürücü devresi olarak kullanılan ULN2003A içerisinde 7 adet NPN transistör ve dahili diod barındırmaktadır. Haliyle bizi transistör bacaklarıyla uğraşmaktan kurtarmaktadır. Beş kablolu adım motorunun kablolarından bir tanesi Vmotor dediğimiz ortak kablodur. Önemli olan bu kablonun hangisi olduğunu bulmaktır. Bunun için avometre ohm ölçere getirildi ve kabloların uçları ikişer ikişer ölçüldü. Tüm uçlar ile arasındaki direnç aynı olan kablo Vmotor kablosudur. Diğer 4 kablo ise motor bobinlerine gitmektedir. Bu 4 kablonun da bir sırası vardır. Bu sırayı da deneme yanılma yöntemiyle bulmak mümkündür. Uçları bulmak için şöyle bir yöntem izlenildi. +24V ortak uca verildi. Sonra uçlardan bir tanesi devamlı olmak üzere "–" ye bağlandı. Devamlı olarak bağlanan bu uç 1 numaralı uç olarak kabul edildi. Ardından diğer uçlar sıra ile "–" uca bağlandı. Uçlar "–" ye temas ettiği anda ileri yönlü hareket edecekmiş gibi yapan uç 2 numaralı uç, geri yönlü hareket edecekmiş gibi görünen uç 4 numaralı uç, hiçbir hareket gözlenmeyen uç ise 3 numaralı uç olarak işaretlendi. Motorların karta bağlanması da bu sıraya göre yapıldı.

6.3. Yazılımın Yüklenmesi

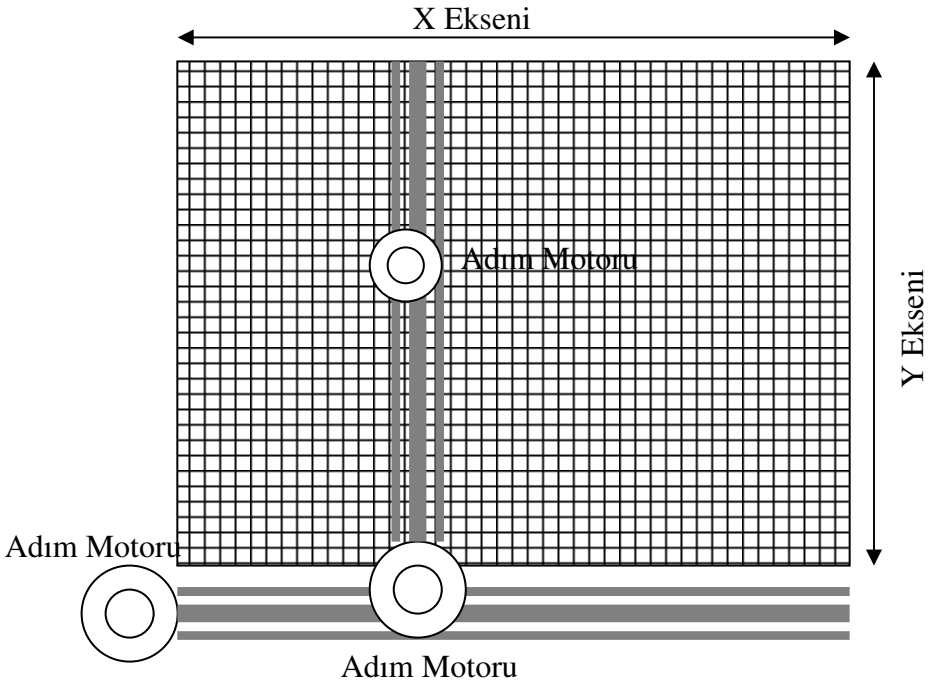
Kart üzerinde yapılacak işlemler tamamlandıktan sonra geriye kalan iş kartı 8051 programlama kartı ile irtibatlandırmaya geldi. Devre kartı ve programlama kartı üzerindeki eş numaralı portlar 10 hatta sahip kablolarla bağlandı. 8051'in bulunduğu kart RS232 kablosu ile bilgisayara bağlandı. 8051'in PSEN ucu açılarak RESET tuşuna basıldı. Daha sonra ATMEL firmasına ait FLIP programı vasıtasıyla daha önceden üretilen programın HEX kodu 8051'e yüklendi. Yükleme işleminden sonra 8051'in PSEN ucu tekrar kapatıldı ve RESET tuşuna basıldı. RESET tuşuna basılmasıyla birlikte göstergeler çalışmaya başladı. Sayma butonlarıyla göstergelere değerler girildi, 4 adet koordinat belleğe kaydedildi ve START butonuna basıldı. Motorlar istendiği gibi çalıştı.

6.4. Mekanik Sistemin Kurulması

Mekanik sistemin kurulmasında kullanılan parçalar mümkün olduğunca elimiz altında bulunan ya da kolay temin edebileceğimiz malzemelerden oluşturulmuştur. Bu parçalar satın alınan HP690C model iki adet bozuk yazıcıdan sağlanmıştır. Yani sistem kurulurken parçaların ağırlığı, sürtünmeleri gibi etkenler önceden hesap edilerek tasarıma katılmamıştır. Bu etkenler mekanik sistem kurulduktan sonra çeşitli müdahalelerle en aza indirilmeye çalışılmıştır. Bu müdahaleler sürtünme olan yerlerin yağlanması ya da sürtünmenin en aza indirilmesi için tekerlekli bir mekanizmanın sağlanması gibi işlemler olmuştur.

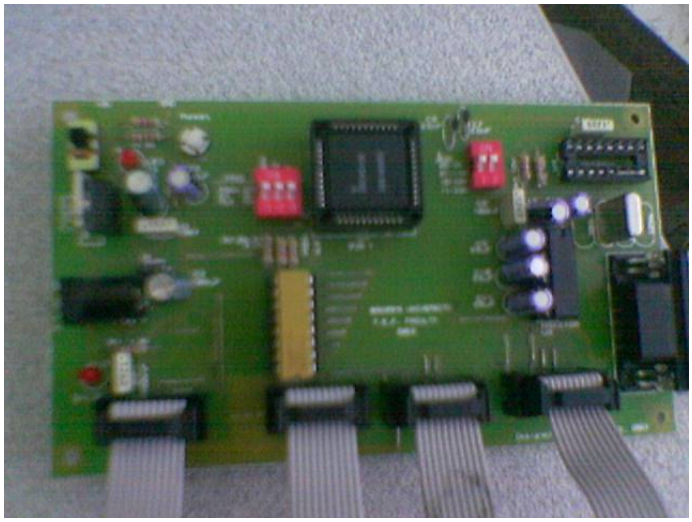
6.5. Motorların Bir Düzenek Üzerine Yerleştirilmesi

Motorların düzeneğe yerleştirilmesi aşağıdaki şekildeki gibi olmuştur. Sistemin fotoğrafları da düzenek şeklinin altında yer almaktadır.



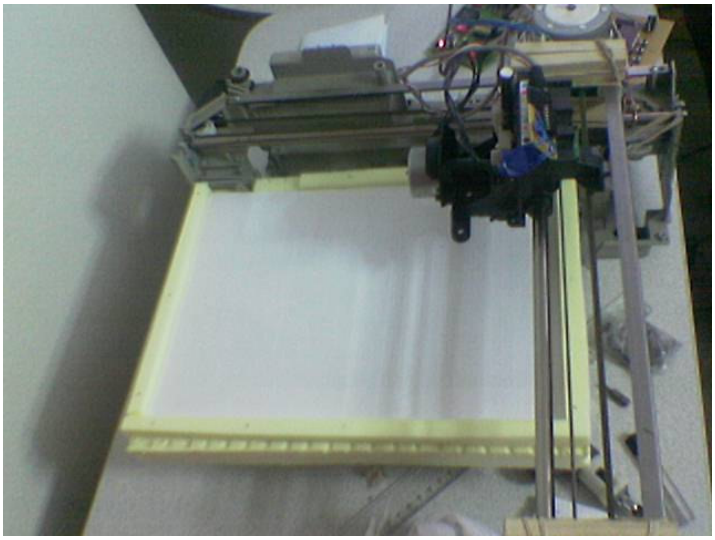
Şekil 6.1. Mekanik sistemin görünümü

6.6. Sistemin Kurulmuş Halinin Fotoğrafları

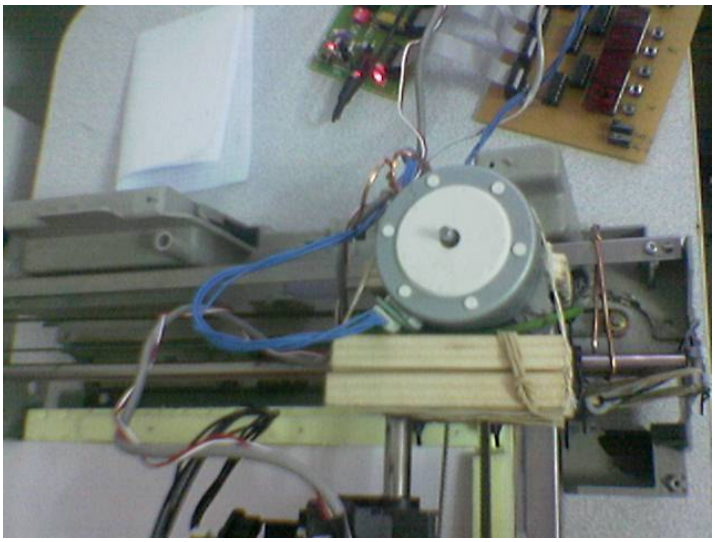


Şekil 6.2. 8051 programlama kartı

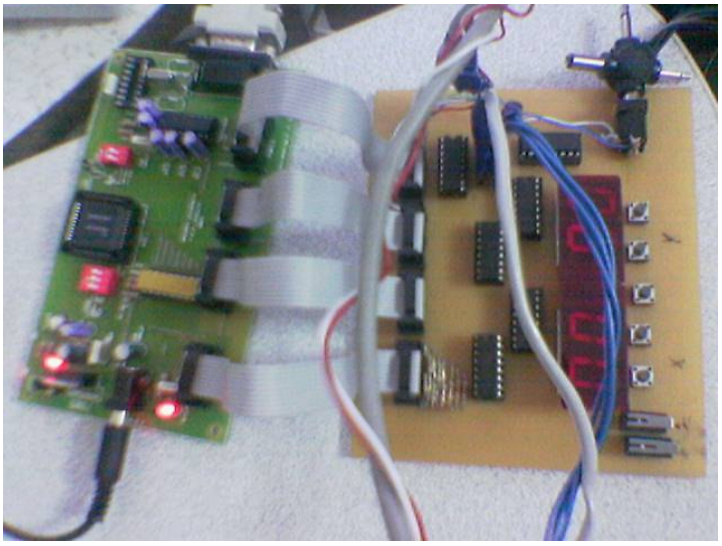
Şekil 6.3. Tasarlanan arayüz kartı



Şekil 6.4. Mekanik sistem



Şekil 6.5. Mekanik sistem



Şekil 6.6. 8051 programlama kartı ve arayüz kartı

BÖLÜM 7. SONUÇLAR VE ÖNERİLER

Endüstriyel otomasyon sistemleri hızla gelişmekte olup, birçok iş sahasında insan gücünün yerini almaktadır. Dünyada hızlı bir gelişme sağlayan otomasyon sistemleri, sağladıkları faydalardan dolayı endüstrinin birçok alanında kullanımı yaygınlaşmıştır. Ülkemiz, otomasyon sistemlerini ve robot teknolojisini kullanma konusunda dünyada ki gelişmiş ülkelerin oldukça gerisinde kalmıştır.

Robot, kullanılan sistemlerde üretim kalitesini artırmakla kalmayıp bunun yanında verimliliği de artırmaktadır. Üretim sektörünün rekabet ortamındaki vazgeçilmez iki unsur, kaliteli ve ucuza imalat yapmaktır. Bu sağlandığı takdirde diğer üreticilerle rekabet etme imkânı bulunmaktadır. Robotlu üretim sistemleri; üretimde insan kaynaklı hataları ortadan kaldırdığı ve kuruluş maliyetinin yüksek olması yanı sıra sonradan kendini amorti edip karlılık sağlayan bir sistem olduğu için üretim sistemlerinde yaygın olarak kullanılmaya başlanmıştır. Türkiye'de robotlu sistemler çok pahalı olarak algılandığı ve bilinmediği için yaygınlaşamamıştır. Robotlu üretim sistemlerinin geliştirilmesi için; bu sistemlerin hızla tanıtımının yapılıp, gerekli olan eğitimlerin yaygın bir şekilde verilmesi gerekmektedir [7].

Yapılan çalışma, dışarıdan el ile giriş amacıyla ikişer karakterlik dijital gösterge özelliğine sahiptir. Göstergelerin kullanımı BCD kodlarının çözülmesiyle gerçekleştirilmiştir. Ayrıca X ve Y koordinatlarına ulaşmak için kullanılan 2 adet adım motoru mevcuttur. Son olarak nokta koydurma işleminde de bir adet adım motoru kullanılmıştır.

Sistemin gerçekleştirilmesinde çeşitli problemlerle karşılaşılmıştır. Bunlardan bir tanesi baskı devre kartındaki bazı kopuk olan yolların tespit edilmesi olmuştur. Göstergelerden bir tanesinin yanlış çalışması üzerine 7447 entegrelerinden bir tanesinin beslemesinin verilemediği tespit edilmiş ve arızalı bölgeye müdahale

edilerek hata düzeltilmiştir. Diğer bir problem ise adım motorları sürmek için kullanılan ULN2003A entegrelerinin aktif hale gelmeleri için 8051'den gelen gerilimin yetersiz kalması olmuştur. Bunun giderilmesi için ise kartta fazla akım çekmekte olan göstergelere, belirli değerlerde dirençler bağlanarak 8051 port çıkışının gerilim seviyesi arttırılması işlemi yapılmıştır.

Çalışmada göstergelerde görünen her bir sayı değeri için motorlara 2 adım attırılmıştır. 99*2=198 adımlık bir mesafe kullanılmıştır. Bu mesafe 13,4 cm' ye tekabül etmektedir. Yani 179,56 cm^2' lik bir alan kullanılmıştır. Cihazın hassasiyeti ise her bir sayı değeri için 1,34 mm'dir. Bu hassasiyet değeri 2 adımlık bir mekanizma kullanılarak gerçekleştirilmiştir. Bu adım sayısı 1'e, hatta yarım adıma düşürülerek çok daha hassas bir değer elde edilebilir. Adım sayısı 1'e düşürüldüğünde 1,34/2=0,67 mm'lik bir hassasiyet, yarım adıma düşürüldüğünde ise 1,34/4=0,335 mm'lik bir hassasiyet elde edilecektir.

Fakat hassasiyeti artırmak için bir sayı değerine karşılık gelen adım sayısını azaltmak, kullanılan alanı daraltmaktadır. Bunun nedeni, elle girebileceğimiz sayı değerinin 99 ile sınırlı olmasıdır.

Hafızadaki 4 koordinatın küçükten büyüğe doğru sıralanması işlemi zaman tasarrufunda önemli bir role sahiptir.

Hafızaya 4 koordinatın depolanması işlemi program dâhilinde bulunan dizi sayısının artırılması ile mikro denetleyicinin bellek kapasitesi dâhilinde çok daha fazla miktarlara çıkarılabilir. Fakat koordinatların sıralanması işlemi için daha fazla komut satırı yazma gereksinimi ortaya çıkacaktır.

Cihaz üzerinde kullanılan malzemelerin özel bir şekilde tasarlanarak özel olarak üretilmesi çalışmanın performansının daha üst düzeye çıkarılmasında etkili olacaktır.

Burada yapılan uygulamayla ilgili olarak ise nokta koyma başlığı yerine daha farklı cihazlar koyularak amaca hizmet ettirilebilir. Tabi ki bu cihaz değişimi sırasında mekanik aksamda bazı değişiklikler de yapmak gerekecektir. Mesela buradaki kalem

yerine mekanik değişiklikler de yapılarak küçük bir plaket delme matkabı konarak belirli koordinatlarda delme işlemi yapılabilir. Ya da elektronik malzemelerin bir plaket üzerine lehimlenmesi işleminde de kullanılabilir.

Çalışmanın daha kullanışlı hale getirilmesi için ise algılama özelliğine sahip bir kalem ve bir düzlem kullanılarak, düzlem üzerinde işaretlenen noktaların cihaz vasıtasıyla depolanıp, uygulama alanına aynen işlenmesi önerisi yapılabilir.

KAYNAKLAR

[1] Altınay Robotik ve Otomasyon A.Ş., "Dünya Robot Nüfusu-1997", Makine Magazin Dergisi., Sayı: 23, Mart 1998, s. 89-94

[2] De Silva, D.,"Reactions to Robots, Engineering". April, 1987, s. 227-230

[3] http://dijitalbilgi.tripod.com/robot.htm

[4] ŞAKRAKER Onur, Mikro Denetleyici Kontrollü Algılamalı Örümcek Robot, Sakarya Üniversitesi, Yüksek Lisans Tezi,

[5] Browne et al.1998 Browne, I.W.A., et al. 1998 MNRAS 293, 257, Interferometer phase calibration sources - II

[6] ÇİÇEK Serdar, Renge Göre (Kırmızı, Yeşil, Mavi) Malzeme Taşıyan Robot Kolu Tasarımı Ve Uygulaması, Gazi Üniversitesi, Yüksek Lisans Tezi, Şubat 2006

[7] ÇENGELCİ B., ÇİMEN H., "Endüstriyel Robotlar 2005", Makine Teknolojileri Elektronik Dergisi,2005

[8] Nakamura Y., "Advanced Robotics Redundancy and Optimization" Mitsubishi Industrial Robot,"Standart Specifications Manual"(Mitsubishi Electric Corporation), Nagoya, Japan.

[9] Shımon Y., "Endustrial Robotics" 605 Thirt Avenue, New York, N.Y. 10158–0012, 1999

[10] http://www.jsautomation.com/default.htm

[11] http://www.ira.uka.de/I32/wave/wave.html,

[12] http://www.sankyo.com/scaras.htm

[13] TAKAROBU ve ARKADAŞLARI, "Remote interaction between human and humanoid robot", J. Of. Intelligent and Robotic Systems, 25, s. 371 – 385, 1999.

[14] CHIAMING YEN VE WU-JENG LI, 'Web-based learning and instruction support system for pneumatics', Computers & Education Volume 41, Issue 2, Pages 107-120, September 2003

[15] REYNE G., "Electromagnetic actuation for MOEMS, examples, advantages and drawbacks of MAGMAS, Journal of Magnetism and Magnetic Materials", Volumes 242-245, Part 2, Pages 1119-1125, April 2002.

[16] J. N. LIOU, M. JAMSHIDI VE G. P. STAR, 'Adaptive edge-following force control of an industrial robot', Robotics and Computer-Integrated Manufacturing, Volume 6, Issue 4, 1989, Pages 331-337

[17] HONG DAEHIE, STEVEN A. VELİNSKY VE KAZUO YAMAZAKİ, 'Tethered mobile robot for automating highway maintenance operations', Robotics and Computer-Integrated Manufacturing Volume 13, Issue 4, March 1997, Pages 297-307

[18] YARIM M. Ali, "ROBOT CONTROL OVER INTERNET USING TCP/IP PROTOCOL", Yüksek Lisans Tezi, İzmir Dokuz Eylül Üniversitesi, Temmuz 2004.

[19] AYBERK Alper, "VISUAL OBJECT RECOGNITION AND ASSORTMENT BY A ROBOT MANIPULATOR", İzmir Dokuz Eylül Üniversitesi, Ağustos 2001.

[20] GÖREN Aytaç, "REMOTE CONTROL OF A NON-HOLONOMIC VEHICLE VIA COMPUTER", İzmir Dokuz Eylül Üniversitesi, Ağustos 2001.

[21] ÇETİNER Ali, "Labirent Robotu Tasarımı ve Uygulaması", Yüksek Lisans Tezi, Sakarya Üniversitesi, Haziran 2003

[22] ÇALIŞKAN Adem, "Mekatronik Sistemlerde İnternet Tabanlı Kontrol ve Kartezyen Robot Üzerinde Bir Uygulama", Yüksek Lisans Tezi, Sakarya Üniversitesi, Şubat 2004.

[23] TEMURTAŞ Hasan, "Üç Eklemli Bir Robot Kolunun Yapay Sinir Ağlı Genelleştirilmiş Öngörülü Kontrol ile Eklem Esaslı Yörünge Kontrolü", Doktora Tezi, Sakarya Üniversitesi, Mart 2004.

[24] BOSTAN Bülent, "Puma 560 Robotunun Hesaplanmış Moment Metoduyla Kontrolü", Yüksek Lisans Tezi, Sakarya Üniversitesi, Haziran 2004.

[25] TİRYAKİ Aysun Eğrisöğüt, KAZAN Recep, "Scara Robot Dinamiğinin Yapay Sinir Ağları Kullanarak Modellenmesi", Mühendis ve Makine Cilt: 46 Sayı: 550, 2005

[26] KÖKER Raşit, "Üç Eklemli Bir Robot Kolunun Görmeye Dayalı Olarak Model Tabanlı Veri Kontrolü", Doktora Tezi, Sakarya Üniversitesi, Mayıs 2002.

[27] YAĞLI Oktay, "Labirent Robotu Tasarımı ve Gerçekleştirilmesi", Yüksek Lisans Tezi, Sakarya Üniversitesi, Haziran 2005.

[28] YAZICI Şermin, KILIVAN Murat, ERTUNÇ H. Metin, EROL M. Coşkun, "Bilgisayar Kontrollü Kartezyen Robot Tasarımı", Elektrik - Elektronik - Bilgisayar Mühendisliği 10. Ulusal Kongresi, 2003.

[29] BERKAY Ahmet, ŞEKER Murat, ESİN E. Murat, "Pnömatik Robot Uygulaması", Elektrik - Elektronik - Bilgisayar Mühendisliği 10. Ulusal Kongresi, 2003.

[30] Lomerson, Jr.; Roland (Bradenton, FL); Parnel; Geoffrey James (Midlothian, VA); Dye; Richard Douglas (Chesterfield, VA); Prill; James (Richmond, VA), Bakery Holdings LLC (Richmond, VA), June 24, 2002.

[31] XIA Yang, "Continuous Sliding Mode Control of a Cartesian Pneumatic Robot", Queen's University, Kingston, Ontario Canada, October 2001.

[32] REYNOSO A. Garcia, "Structural Dynamics Model of a Cartesian Robot", Mechanical Engineering at the Massachusetts Institute Of Technology, Doctor of Science, October 1985.

[33] TONBUL T. Selcen, SARITAŞ Müzeyyen, "Beş Eksenli Bir Edubot Robot Kolunda Ters Kinematik Hesaplamalar ve Yörünge Planlaması", Gazi Üniv. Müh. Mim. Fak. Der. Cilt 18, No 1, 145-167, 2003.

[34] ÇETİN A. Emre, ADLI M. Arif, "Mekatronikte Donanım ve Yazılımın Robot Kontrolündeki Önemi", Elektrik – Elektronik - Bilgisayar Mühendisliği 10. Ulusal Kongresi, 2003.

[35] http://www.teknokent.hakkari.gov.tr/academics/plotter.php

[36] Altınbaşak Orhan, Mikrodenetleyiciler ve PIC Programlama, İstanbul, Eylül 2000

[37] Özkan, Turhan, Mikro İşlemciler, Mikro Bilgisayarlar, 1993

[38] http://mezire2.topcities.com/elektronik/picders1.htm

[39] http://www.elektrotekno.com/about73.html

[40] http://www.angelfire.com/ak4/aga/pic2.htm

[41] http://www.ume.tubitak.gov.tr

[42] http://www.ieee.itu.edu.tr

[43] http://www.datasheetcatalog.net/

[44] http://oop.rosweb.ru/Other/58.html,

[45] http://www.mame.mu.oz.au/~phr/telerobot/App_III.html

[46] http://www.ar2.com/puma.html

[47] http://www.sankyo.com/scaras.htm

[48] http://bilisim.omuegitim.edu.tr

[49] http://www.mihrace.net/myazdir.asp?id=ebuliz&sayfa=1

[50] www.engr.ucr.edu/~mtaylor/ Bmorrow%20individual%20report.doc

[51] BARUTÇUOĞLU E. Itır, "Robotların Tarihcesi", Boğaziçi Üniversitesi, Temmuz 2001

[52] http://prag-matik.blogspot.com/ 2005_05_01_prag-matik_archive.html

[53] www.sualti.net/izer/html/tarihce.html

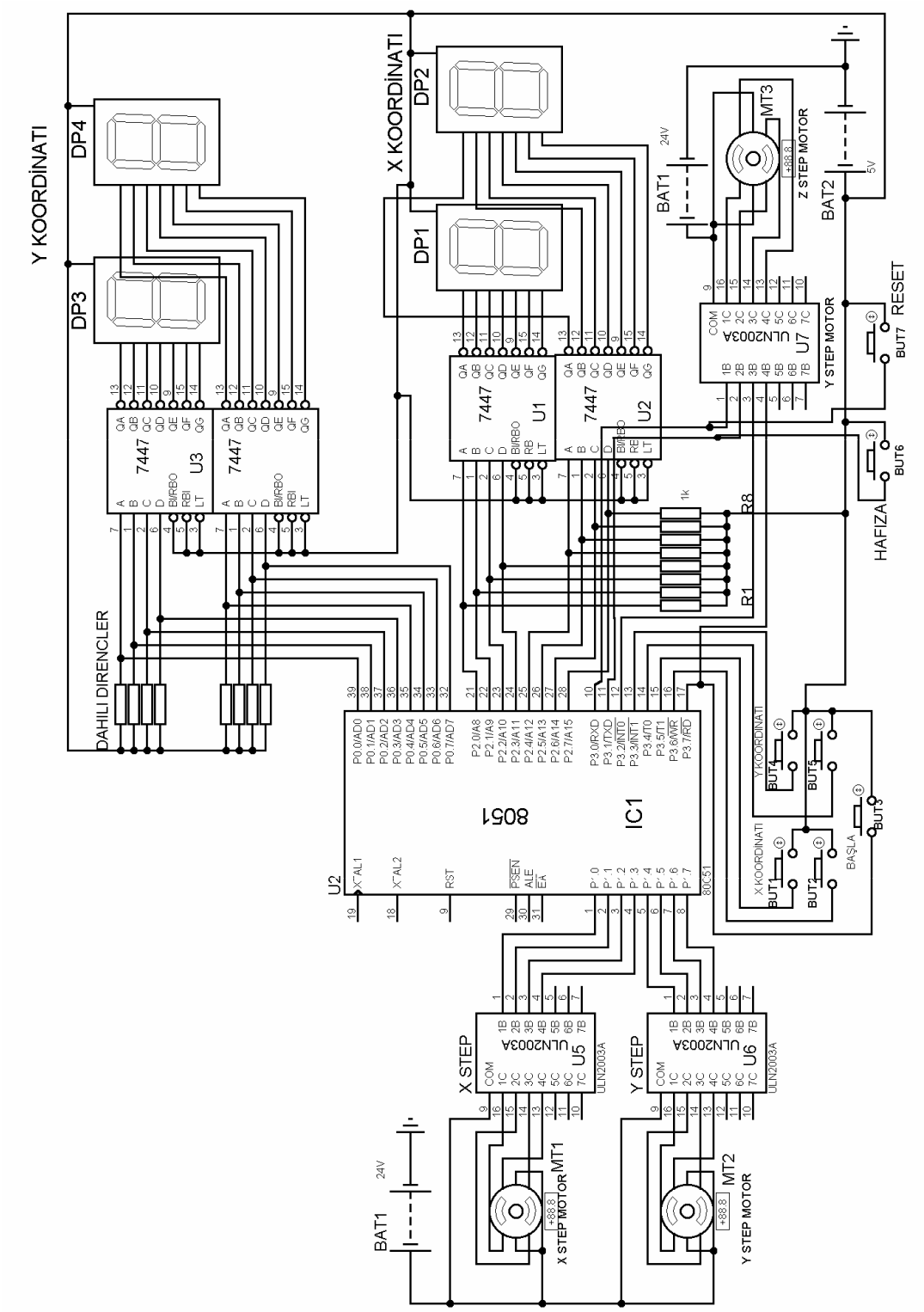
[54] http://bilimkurgu2000.com/asp/Yazar.asp?inNo=235

[55] http://myo.mersin.edu.tr/UZAK/TP/EndElo/elt-204/elt-204.htm

EKLER

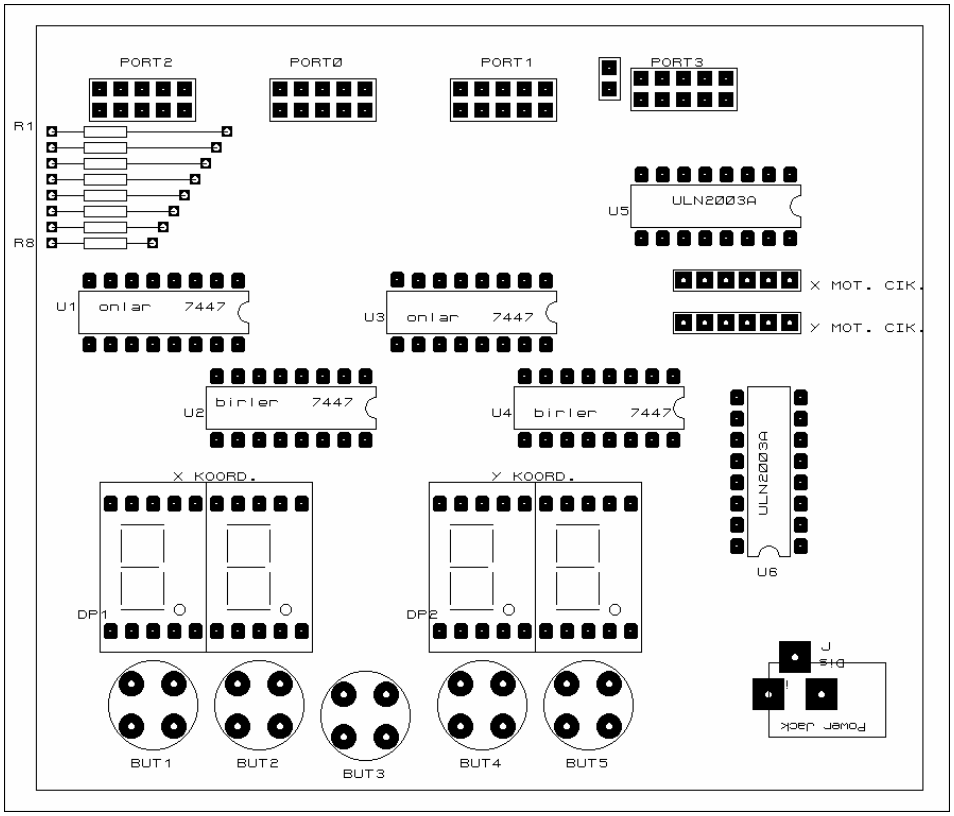
EK-A. Uygulanan Devrenin Şematik Diyagramları

A.1. Devre açık şeması



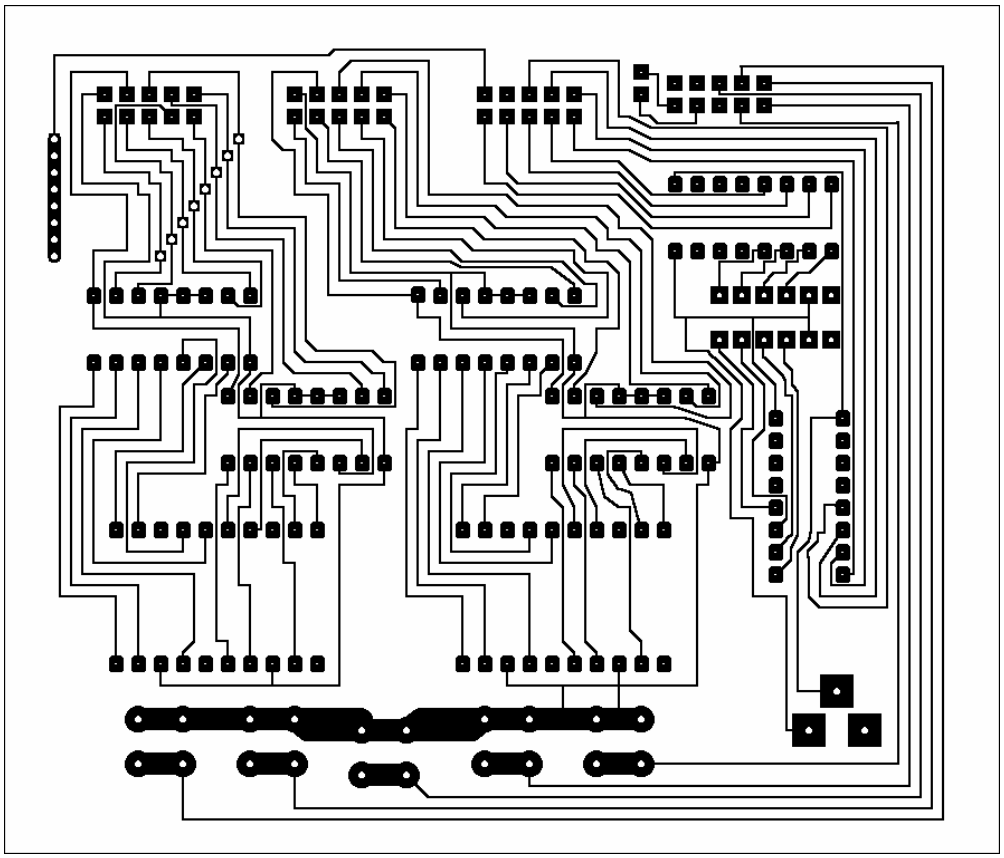
Şekil E.1. Devrenin açık şeması

A.2. Devre plaketi üst görünüşü

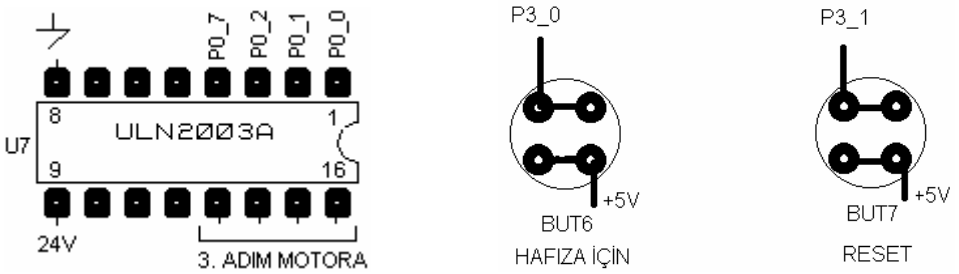


Şekil E.2. Devre plaketi üst görünüşü

A.3. Devre plaket baskı şeması



Şekil E.3. Devre plaketi baskı şeması



Şekil E.4. Sonradan eklenen birimler

EK-B Devre Malzeme Listeleri

IC1	8051 (Programlama Kartı)
U1, U2, U3, U4	SN74LS47N
U5, U6, U7	ULN2003A
DP1, DP2, DP3, DP4	7 Parça Gösterge
BAT1	24V Güç kaynağı
BAT2	5V Güç kaynağı
BUT1, BUT2, BUT3, BUT4, BUT5, BUT6, BUT7	Buton
MT1, MT2	13 ohm 7,5^{o} Adım Motor
MT3	30 ohm 7,5^{o} Adım Motor
R1, ……, R8	1K

EK-C Kontrol Programı

```
#include <AT89X51.H>
#include <MATH.H>

#define X10SIFIR P2_3=0,P2_2=0,P2_1=0,P2_0=0;
#define X10BIR   P2_3=0,P2_2=0,P2_1=0,P2_0=1;
#define X10IKI   P2_3=0,P2_2=0,P2_1=1,P2_0=0;
#define X10UC    P2_3=0,P2_2=0,P2_1=1,P2_0=1;
#define X10DORT  P2_3=0,P2_2=1,P2_1=0,P2_0=0;
#define X10BES   P2_3=0,P2_2=1,P2_1=0,P2_0=1;
#define X10ALTI  P2_3=0,P2_2=1,P2_1=1,P2_0=0;
#define X10YEDI  P2_3=0,P2_2=1,P2_1=1,P2_0=1;
#define X10SEKIZ P2_3=1,P2_2=0,P2_1=0,P2_0=0;
#define X10DOKUZ P2_3=1,P2_2=0,P2_1=0,P2_0=1;

#define X1SIFIR P2_7=0,P2_6=0,P2_5=0,P2_4=0;
#define X1BIR   P2_7=0,P2_6=0,P2_5=0,P2_4=1;
#define X1IKI   P2_7=0,P2_6=0,P2_5=1,P2_4=0;
#define X1UC    P2_7=0,P2_6=0,P2_5=1,P2_4=1;
#define X1DORT  P2_7=0,P2_6=1,P2_5=0,P2_4=0;
#define X1BES   P2_7=0,P2_6=1,P2_5=0,P2_4=1;
#define X1ALTI  P2_7=0,P2_6=1,P2_5=1,P2_4=0;
#define X1YEDI  P2_7=0,P2_6=1,P2_5=1,P2_4=1;
#define X1SEKIZ P2_7=1,P2_6=0,P2_5=0,P2_4=0;
#define X1DOKUZ P2_7=1,P2_6=0,P2_5=0,P2_4=1;

#define XADIM1 P1_3=1,P1_2=0,P1_1=0,P1_0=0;
#define XADIM2 P1_3=0,P1_2=1,P1_1=0,P1_0=0;
#define XADIM3 P1_3=0,P1_2=0,P1_1=1,P1_0=0;
#define XADIM4 P1_3=0,P1_2=0,P1_1=0,P1_0=1;

#define Y10SIFIR P0_3=0,P0_2=0,P0_1=0,P0_0=0;
```

```
#define Y10BIR   P0_3=0,P0_2=0,P0_1=0,P0_0=1;
#define Y10IKI   P0_3=0,P0_2=0,P0_1=1,P0_0=0;
#define Y10UC    P0_3=0,P0_2=0,P0_1=1,P0_0=1;
#define Y10DORT  P0_3=0,P0_2=1,P0_1=0,P0_0=0;
#define Y10BES   P0_3=0,P0_2=1,P0_1=0,P0_0=1;
#define Y10ALTI  P0_3=0,P0_2=1,P0_1=1,P0_0=0;
#define Y10YEDI  P0_3=0,P0_2=1,P0_1=1,P0_0=1;
#define Y10SEKIZ P0_3=1,P0_2=0,P0_1=0,P0_0=0;
#define Y10DOKUZ P0_3=1,P0_2=0,P0_1=0,P0_0=1;

#define Y1SIFIR P0_7=0,P0_6=0,P0_5=0,P0_4=0;
#define Y1BIR   P0_7=0,P0_6=0,P0_5=0,P0_4=1;
#define Y1IKI   P0_7=0,P0_6=0,P0_5=1,P0_4=0;
#define Y1UC    P0_7=0,P0_6=0,P0_5=1,P0_4=1;
#define Y1DORT  P0_7=0,P0_6=1,P0_5=0,P0_4=0;
#define Y1BES   P0_7=0,P0_6=1,P0_5=0,P0_4=1;
#define Y1ALTI  P0_7=0,P0_6=1,P0_5=1,P0_4=0;
#define Y1YEDI  P0_7=0,P0_6=1,P0_5=1,P0_4=1;
#define Y1SEKIZ P0_7=1,P0_6=0,P0_5=0,P0_4=0;
#define Y1DOKUZ P0_7=1,P0_6=0,P0_5=0,P0_4=1;

#define YADIM1 P1_7=1,P1_6=0,P1_5=0,P1_4=0;
#define YADIM2 P1_7=0,P1_6=1,P1_5=0,P1_4=0;
#define YADIM3 P1_7=0,P1_6=0,P1_5=1,P1_4=0;
#define YADIM4 P1_7=0,P1_6=0,P1_5=0,P1_4=1;

void bekle(void){
        unsigned int sayac=5669;
        while(sayac--);}

void motorbekle(void){
        unsigned int sayac=9567;
        while(sayac--);}
```

```
void motorbekle2(void){
        unsigned int sayac=500;
        while(sayac--);}

void motorbekle3(void){
        unsigned int sayac=50300;
        while(sayac--);}

main(){
char
Xsayac1=0,Xsayac2=0,Xyeni=0,Xeski=0,Xdurum=0,Ysayac1=0,Ysayac2=0,Yyeni=
0,Yeski=0,Ydurum=0,
Xkonum,Ykonum,Xgor,Ygor,a=0,b=0,xilk[4]={0},yilk[4]={0},xson[4]={0},yson[4]
={0},buyuk = 0, kucuk = 0;
float mesafe[4] = {0},temp = 0,sayilar[4]={0};
unsigned int s,d;
P2=0,P0=0,P1=0;

while(1){
P3=0;
if(P3_5==1){
if(Xsayac1==10) Xsayac1=1; else if(Xsayac1==9) Xsayac1=0; else Xsayac1++;
switch (Xsayac1){
        case 0: X1SIFIR;
                if(Xsayac2==10) Xsayac2=1; else if(Xsayac2==9) Xsayac2=0; else
                Xsayac2++;
                switch(Xsayac2){
                case 0: X10SIFIR; break;
                case 1: X10BIR; break;
                case 2: X10IKI; break;
                case 3: X10UC; break;
                case 4: X10DORT; break;
```

```
                case 5: X10BES; break;
                case 6: X10ALTI; break;
                case 7: X10YEDI; break;
                case 8: X10SEKIZ; break;
                case 9: X10DOKUZ; Xsayac2=-1; break;} break;
        case 1: X1BIR; break;
        case 2: X1IKI; break;
        case 3: X1UC; break;
        case 4: X1DORT; break;
        case 5: X1BES; break;
        case 6: X1ALTI; break;
        case 7: X1YEDI; break;
        case 8: X1SEKIZ; break;
        case 9: X1DOKUZ; Xsayac1=-1; break;} bekle();
                if(Xsayac1==-1 && Xsayac2==-1) Xgor=99;
                else if(Xsayac1==-1 && Xsayac2==10) Xgor=Xsayac1;
                else if(Xsayac1==10 && Xsayac2==-1) Xgor=90;
                else if(Xsayac1==10 && Xsayac2==10) Xgor=0;
                else if(Xsayac1==-1) Xgor=(10*Xsayac2)+9;
                else if(Xsayac1==10) Xgor=10*Xsayac2;
                else if(Xsayac2==-1) Xgor=Xsayac1+90;
                else if(Xsayac2==10) Xgor=Xsayac1;
                else  Xgor=(10*Xsayac2)+Xsayac1;}

if(P3_6==1){
if(Xsayac1==0) Xsayac1=9; else if(Xsayac1==-1) Xsayac1=8; else Xsayac1--;
        switch (Xsayac1){
        case 9: X1DOKUZ;
if(Xsayac2==0) Xsayac2=9; else if(Xsayac1==-1) Xsayac1=8; else Xsayac2--;
                switch(Xsayac2){
                case 9: X10DOKUZ; break;
                case 8: X10SEKIZ; break;
                case 7: X10YEDI; break;
```

```
                case 6: X10ALTI; break;
                case 5: X10BES; break;
                case 4: X10DORT; break;
                case 3: X10UC; break;
                case 2: X10IKI; break;
                case 1: X10BIR; break;
        case 0: X10SIFIR; Xsayac2=10; break;} break;
        case 8: X1SEKIZ; break;
        case 7: X1YEDI; break;
        case 6: X1ALTI; break;
        case 5: X1BES; break;
        case 4: X1DORT; break;
        case 3: X1UC; break;
        case 2: X1IKI; break;
        case 1: X1BIR; break;
        case 0: X1SIFIR; Xsayac1=10; break;} bekle();
                if(Xsayac1==-1 && Xsayac2==-1) Xgor=99;
                else if(Xsayac1==-1 && Xsayac2==10) Xgor=Xsayac1;
                else if(Xsayac1==10 && Xsayac2==-1) Xgor=90;
                else if(Xsayac1==10 && Xsayac2==10) Xgor=0;
                else if(Xsayac1==-1) Xgor=(10*Xsayac2)+9;
                else if(Xsayac1==10) Xgor=10*Xsayac2;
                else if(Xsayac2==-1) Xgor=Xsayac1+90;
                else if(Xsayac2==10) Xgor=Xsayac1;
                else  Xgor=(10*Xsayac2)+Xsayac1;}

if(P3_3==1){
if(Ysayac1==10) Ysayac1=1; else if(Ysayac1==9) Ysayac1=0; else Ysayac1++;
        switch (Ysayac1){
        case 0: Y1SIFIR;
if(Ysayac2==10) Ysayac2=1; else if(Ysayac2==9) Ysayac2=0; else Ysayac2++;
                switch(Ysayac2){
                case 0: Y10SIFIR; break;
```

```
                case 1: Y10BIR; break;
                case 2: Y10IKI; break;
                case 3: Y10UC; break;
                case 4: Y10DORT; break;
                case 5: Y10BES; break;
                case 6: Y10ALTI; break;
                case 7: Y10YEDI; break;
                case 8: Y10SEKIZ; break;
                case 9: Y10DOKUZ; Ysayac2=-1; break;} break;
        case 1: Y1BIR; break;
        case 2: Y1IKI; break;
        case 3: Y1UC; break;
        case 4: Y1DORT; break;
        case 5: Y1BES; break;
        case 6: Y1ALTI; break;
        case 7: Y1YEDI; break;
        case 8: Y1SEKIZ; break;
        case 9: Y1DOKUZ; Ysayac1=-1; break;} bekle();
                if(Ysayac1==-1 && Ysayac2==-1) Ygor=99;
                else if(Ysayac1==-1 && Ysayac2==10) Ygor=Ysayac1;
                else if(Ysayac1==10 && Ysayac2==-1) Ygor=90;
                else if(Ysayac1==10 && Ysayac2==10) Ygor=0;
                else if(Ysayac1==-1) Ygor=(10*Ysayac2)+9;
                else if(Ysayac1==10) Ygor=10*Ysayac2;
                else if(Ysayac2==-1) Ygor=Ysayac1+90;
                else if(Ysayac2==10) Ygor=Ysayac1;
                else  Ygor=(10*Ysayac2)+Ysayac1;}

if(P3_4==1){
if(Ysayac1==0) Ysayac1=9; else if(Ysayac1==-1) Ysayac1=8; else Ysayac1--;
        switch (Ysayac1){
        case 9: Y1DOKUZ;
if(Ysayac2==0) Ysayac2=9; else if(Ysayac1==-1) Ysayac1=8; else Ysayac2--;
```

```
                switch(Ysayac2){
                case 9: Y10DOKUZ; break;
                case 8: Y10SEKIZ; break;
                case 7: Y10YEDI; break;
                case 6: Y10ALTI; break;
                case 5: Y10BES; break;
                case 4: Y10DORT; break;
                case 3: Y10UC; break;
                case 2: Y10IKI; break;
                case 1: Y10BIR; break;
                case 0: Y10SIFIR; Ysayac2=10; break;} break;
        case 8: Y1SEKIZ; break;
        case 7: Y1YEDI; break;
        case 6: Y1ALTI; break;
        case 5: Y1BES; break;
        case 4: Y1DORT; break;
        case 3: Y1UC; break;
        case 2: Y1IKI; break;
        case 1: Y1BIR; break;
        case 0: Y1SIFIR; Ysayac1=10; break;} bekle();
                if(Ysayac1==-1 && Ysayac2==-1) Ygor=99;
                else if(Ysayac1==-1 && Ysayac2==10) Ygor=Ysayac1;
                else if(Ysayac1==10 && Ysayac2==-1) Ygor=90;
                else if(Ysayac1==10 && Ysayac2==10) Ygor=0;
                else if(Ysayac1==-1) Ygor=(10*Ysayac2)+9;
                else if(Ysayac1==10) Ygor=10*Ysayac2;
                else if(Ysayac2==-1) Ygor=Ysayac1+90;
                else if(Ysayac2==10) Ygor=Ysayac1;
                else  Ygor=(10*Ysayac2)+Ysayac1;}

if(P3_0==1){
xilk[a]=Xgor;
yilk[a]=Ygor;a=a+1;motorbekle3(),motorbekle3();
```

```
}

if(P3_7==1){

mesafe[0] = sqrt( (xilk[0]-Xeski)*(xilk[0]-Xeski)  +  (yilk[0]-Yeski)*(yilk[0]-Yeski)
); sayilar[0]=mesafe[0];
mesafe[1] = sqrt( (xilk[1]-Xeski)*(xilk[1]-Xeski)  +  (yilk[1]-Yeski)*(yilk[1]-Yeski)
); sayilar[1]=mesafe[1];
mesafe[2] = sqrt( (xilk[2]-Xeski)*(xilk[2]-Xeski)  +  (yilk[2]-Yeski)*(yilk[2]-Yeski)
); sayilar[2]=mesafe[2];
mesafe[3] = sqrt( (xilk[3]-Xeski)*(xilk[3]-Xeski)  +  (yilk[3]-Yeski)*(yilk[3]-Yeski)
); sayilar[3]=mesafe[3];

    for( buyuk=0; buyuk<3 ; buyuk++)
    {  for( kucuk = buyuk+1; kucuk<4; kucuk++)
            {  if( sayilar[buyuk] > sayilar[kucuk] )
                {     temp   = sayilar[buyuk];
                    sayilar[buyuk] = sayilar[kucuk];
                    sayilar[kucuk] = temp;}}}

if(sayilar[0]==mesafe[0]){ xson[0]=xilk[0],yson[0]=yilk[0];
                mesafe[1] = sqrt( (xilk[1]-xilk[0])*(xilk[1]-xilk[0]) + (yilk[1]-
yilk[0])*(yilk[1]-yilk[0]) ); sayilar[1]=mesafe[1];
                mesafe[2] = sqrt( (xilk[2]-xilk[0])*(xilk[2]-xilk[0]) + (yilk[2]-
yilk[0])*(yilk[2]-yilk[0]) ); sayilar[2]=mesafe[2];
                mesafe[3] = sqrt( (xilk[3]-xilk[0])*(xilk[3]-xilk[0]) + (yilk[3]-
yilk[0])*(yilk[3]-yilk[0]) ); sayilar[3]=mesafe[3];

for( buyuk=1; buyuk<3 ; buyuk++)
{   for( kucuk = buyuk+1; kucuk<4; kucuk++)
            {   if( sayilar[buyuk] > sayilar[kucuk] )
                {     temp   = sayilar[buyuk];
                    sayilar[buyuk] = sayilar[kucuk];
```

```
                    sayilar[kucuk] = temp;}}}

if(sayilar[1]==mesafe[1]){ xson[1]=xilk[1],yson[1]=yilk[1];
        mesafe[2] = sqrt( (xilk[2]-xilk[1])*(xilk[2]-xilk[1]) + (yilk[2]-
yilk[1])*(yilk[2]-yilk[1]) );
        mesafe[3] = sqrt( (xilk[3]-xilk[1])*(xilk[3]-xilk[1]) + (yilk[3]-
yilk[1])*(yilk[3]-yilk[1]) );

if(mesafe[2]<mesafe[3]){xson[2]=xilk[2],yson[2]=yilk[2],xson[3]=xilk[3],yson[3]=
yilk[3];}
else {xson[2]=xilk[3],yson[2]=yilk[3],xson[3]=xilk[2],yson[3]=yilk[2];}}

else if(sayilar[1]==mesafe[2]) {xson[1]=xilk[2],yson[1]=yilk[2];
        mesafe[1] = sqrt( (xilk[1]-xilk[2])*(xilk[1]-xilk[2]) + (yilk[1]-
yilk[2])*(yilk[1]-yilk[2]) );
        mesafe[3] = sqrt( (xilk[3]-xilk[2])*(xilk[3]-xilk[2]) + (yilk[3]-
yilk[2])*(yilk[3]-yilk[2]) );
if(mesafe[1]<mesafe[3]){xson[2]=xilk[1],yson[2]=yilk[1],xson[3]=xilk[3],yson[3]=y
ilk[3];}
else {xson[2]=xilk[3],yson[2]=yilk[3],xson[2]=xilk[1],yson[2]=yilk[1];}}

else if(sayilar[1]==mesafe[3]){ xson[1]=xilk[3],yson[1]=yilk[3];
        mesafe[2] = sqrt( (xilk[2]-xilk[3])*(xilk[2]-xilk[3]) + (yilk[2]-
yilk[3])*(yilk[2]-yilk[3]) );
        mesafe[1] = sqrt( (xilk[1]-xilk[3])*(xilk[1]-xilk[3]) + (yilk[1]-
yilk[3])*(yilk[1]-yilk[3]) );
if(mesafe[1]<mesafe[2]){xson[2]=xilk[1],yson[2]=yilk[1],xson[3]=xilk[2],yson[3]=y
ilk[2];}
else {xson[2]=xilk[2],yson[2]=yilk[2],xson[3]=xilk[1],yson[3]=yilk[1];}}}

else if(sayilar[0]==mesafe[1]) {xson[0]=xilk[1],yson[0]=yilk[1];
        mesafe[0] = sqrt( (xilk[1]-xilk[0])*(xilk[1]-xilk[0]) + (yilk[1]-
yilk[0])*(yilk[1]-yilk[0]) ); sayilar[1]=mesafe[0];
```

```
        mesafe[2] = sqrt( (xilk[2]-xilk[1])*(xilk[2]-xilk[1]) + (yilk[2]-
yilk[1])*(yilk[2]-yilk[1]) ); sayilar[2]=mesafe[2];
        mesafe[3] = sqrt( (xilk[3]-xilk[1])*(xilk[3]-xilk[1]) + (yilk[3]-
yilk[1])*(yilk[3]-yilk[1]) ); sayilar[3]=mesafe[3];

for( buyuk=1; buyuk<3 ; buyuk++)
{   for( kucuk = buyuk+1; kucuk<4; kucuk++)
{   if( sayilar[buyuk] > sayilar[kucuk] )
{     temp   = sayilar[buyuk];
sayilar[buyuk] = sayilar[kucuk];
sayilar[kucuk] = temp;}}}

if(sayilar[1]==mesafe[0]){ xson[1]=xilk[0],yson[1]=yilk[0];
        mesafe[2] = sqrt( (xilk[2]-xilk[0])*(xilk[2]-xilk[0]) + (yilk[2]-
yilk[0])*(yilk[2]-yilk[0]) );
        mesafe[3] = sqrt( (xilk[3]-xilk[0])*(xilk[3]-xilk[0]) + (yilk[3]-
yilk[0])*(yilk[3]-yilk[0]) );

if(mesafe[2]<mesafe[3]){xson[2]=xilk[2],yson[2]=yilk[2],xson[3]=xilk[3],yson[3]=y
ilk[3];}
else {xson[2]=xilk[3],yson[2]=yilk[3],xson[3]=xilk[2],yson[3]=yilk[2];}}

else if(sayilar[1]==mesafe[2]) {xson[1]=xilk[2],yson[1]=yilk[2];
        mesafe[0] = sqrt( (xilk[0]-xilk[2])*(xilk[0]-xilk[2]) + (yilk[0]-
yilk[2])*(yilk[0]-yilk[2]) );
        mesafe[3] = sqrt( (xilk[3]-xilk[2])*(xilk[3]-xilk[2]) + (yilk[3]-
yilk[2])*(yilk[3]-yilk[2]) );
if(mesafe[0]<mesafe[3]){xson[2]=xilk[0],yson[2]=yilk[0],xson[3]=xilk[3],yson[3]=y
ilk[3];}
else {xson[2]=xilk[3],yson[2]=yilk[3],xson[3]=xilk[0],yson[3]=yilk[0];}}

else if(sayilar[1]==mesafe[3]){ xson[1]=xilk[3],yson[1]=yilk[3];
```

```
        mesafe[2] = sqrt( (xilk[2]-xilk[3])*(xilk[2]-xilk[3]) + (yilk[2]-
yilk[3])*(yilk[2]-yilk[3]) );
        mesafe[1] = sqrt( (xilk[1]-xilk[3])*(xilk[1]-xilk[3]) + (yilk[1]-
yilk[3])*(yilk[1]-yilk[3]) );
if(mesafe[1]<mesafe[2]){xson[2]=xilk[1],yson[2]=yilk[1],xson[3]=xilk[2],yson[3]=y
ilk[2];}
else {xson[2]=xilk[2],yson[2]=yilk[2],xson[3]=xilk[1],yson[3]=yilk[1];}}}

else if(sayilar[0]==mesafe[2]){ xson[0]=xilk[2],yson[0]=yilk[2];
        mesafe[0] = sqrt( (xilk[2]-xilk[0])*(xilk[2]-xilk[0]) + (yilk[2]-
yilk[0])*(yilk[2]-yilk[0]) ); sayilar[1]=mesafe[0];
        mesafe[1] = sqrt( (xilk[2]-xilk[1])*(xilk[2]-xilk[1]) + (yilk[2]-
yilk[1])*(yilk[2]-yilk[1]) ); sayilar[2]=mesafe[1];
        mesafe[3] = sqrt( (xilk[3]-xilk[2])*(xilk[3]-xilk[2]) + (yilk[3]-
yilk[2])*(yilk[3]-yilk[2]) ); sayilar[3]=mesafe[3];

for( buyuk=1; buyuk<3 ; buyuk++)
{   for( kucuk = buyuk+1; kucuk<4; kucuk++)
{   if( sayilar[buyuk] > sayilar[kucuk] )
{     temp   = sayilar[buyuk];
sayilar[buyuk] = sayilar[kucuk];
sayilar[kucuk] = temp;}}}

if(sayilar[1]==mesafe[0]){ xson[1]=xilk[0],yson[1]=yilk[0];
        mesafe[1] = sqrt( (xilk[1]-xilk[0])*(xilk[1]-xilk[0]) + (yilk[1]-
yilk[0])*(yilk[1]-yilk[0]) );
        mesafe[3] = sqrt( (xilk[3]-xilk[0])*(xilk[3]-xilk[0]) + (yilk[3]-
yilk[0])*(yilk[3]-yilk[0]) );

if(mesafe[1]<mesafe[3]){xson[2]=xilk[1],yson[2]=yilk[1],xson[3]=xilk[3],yson[3]=y
ilk[3];}
else {xson[2]=xilk[3],yson[2]=yilk[3],xson[3]=xilk[1],yson[3]=yilk[1];}}
```

```
else if(sayilar[1]==mesafe[2]) {xson[1]=xilk[2],yson[1]=yilk[2];
        mesafe[0] = sqrt( (xilk[0]-xilk[1])*(xilk[0]-xilk[1]) + (yilk[0]-
yilk[1])*(yilk[0]-yilk[1]) );
        mesafe[3] = sqrt( (xilk[3]-xilk[1])*(xilk[3]-xilk[1]) + (yilk[3]-
yilk[1])*(yilk[3]-yilk[1]) );

if(mesafe[0]<mesafe[3]){xson[2]=xilk[0],yson[2]=yilk[0],xson[3]=xilk[3],yson[3]=y
ilk[3];}
else {xson[2]=xilk[3],yson[2]=yilk[3],xson[3]=xilk[0],yson[3]=yilk[0];}}

else if(sayilar[1]==mesafe[3]){ xson[1]=xilk[3],yson[1]=yilk[3];
        mesafe[0] = sqrt( (xilk[0]-xilk[3])*(xilk[0]-xilk[3]) + (yilk[0]-
yilk[3])*(yilk[0]-yilk[3]) );
        mesafe[1] = sqrt( (xilk[1]-xilk[3])*(xilk[1]-xilk[3]) + (yilk[1]-
yilk[3])*(yilk[1]-yilk[3]) );
if(mesafe[1]<mesafe[0]){xson[2]=xilk[1],yson[2]=yilk[1],xson[3]=xilk[0],yson[3]=y
ilk[0];}
else {xson[2]=xilk[0],yson[2]=yilk[0],xson[3]=xilk[1],yson[3]=yilk[1];}}}

else {xson[0]=xilk[3],yson[0]=yilk[3];
        mesafe[0] = sqrt( (xilk[3]-xilk[0])*(xilk[3]-xilk[0]) + (yilk[3]-
yilk[0])*(yilk[3]-yilk[0]) ); sayilar[1]=mesafe[0];
        mesafe[1] = sqrt( (xilk[3]-xilk[1])*(xilk[3]-xilk[1]) + (yilk[3]-
yilk[1])*(yilk[3]-yilk[1]) ); sayilar[2]=mesafe[1];
        mesafe[2] = sqrt( (xilk[3]-xilk[2])*(xilk[3]-xilk[2]) + (yilk[3]-
yilk[2])*(yilk[3]-yilk[2]) ); sayilar[3]=mesafe[2];

for( buyuk=1; buyuk<3 ; buyuk++)
{   for( kucuk = buyuk+1; kucuk<4; kucuk++)
{   if( sayilar[buyuk] > sayilar[kucuk] )
{     temp   = sayilar[buyuk];
sayilar[buyuk] = sayilar[kucuk];
sayilar[kucuk] = temp;}}}
```

```
if(sayilar[1]==mesafe[0]){ xson[1]=xilk[0],yson[1]=yilk[0];
        mesafe[1] = sqrt( (xilk[1]-xilk[0])*(xilk[1]-xilk[0]) + (yilk[1]-
yilk[0])*(yilk[1]-yilk[0]) );
        mesafe[2] = sqrt( (xilk[2]-xilk[0])*(xilk[2]-xilk[0]) + (yilk[2]-
yilk[0])*(yilk[2]-yilk[0]) );

if(mesafe[1]<mesafe[2]){xson[2]=xilk[1],yson[2]=yilk[1],xson[3]=xilk[2],yson[3]=y
ilk[2];}
else {xson[2]=xilk[2],yson[2]=yilk[2],xson[3]=xilk[1],yson[3]=yilk[1];}}

else if(sayilar[1]==mesafe[1]) {xson[1]=xilk[1],yson[1]=yilk[1];
        mesafe[0] = sqrt( (xilk[0]-xilk[1])*(xilk[0]-xilk[1]) + (yilk[0]-
yilk[1])*(yilk[0]-yilk[1]) );
        mesafe[2] = sqrt( (xilk[2]-xilk[1])*(xilk[2]-xilk[1]) + (yilk[2]-
yilk[1])*(yilk[2]-yilk[1]) );
if(mesafe[0]<mesafe[2]){xson[2]=xilk[0],yson[2]=yilk[0],xson[3]=xilk[2],yson[3]=y
ilk[2];}
else {xson[2]=xilk[2],yson[2]=yilk[2],xson[3]=xilk[0],yson[3]=yilk[0];}}

else if(sayilar[1]==mesafe[2]){ xson[1]=xilk[2],yson[1]=yilk[2];
        mesafe[0] = sqrt( (xilk[0]-xilk[2])*(xilk[0]-xilk[2]) + (yilk[0]-
yilk[2])*(yilk[0]-yilk[2]) );
        mesafe[1] = sqrt( (xilk[1]-xilk[2])*(xilk[1]-xilk[2]) + (yilk[1]-
yilk[2])*(yilk[1]-yilk[2]) );
if(mesafe[1]<mesafe[0]){xson[2]=xilk[1],yson[2]=yilk[1],xson[3]=xilk[0],yson[3]=y
ilk[0];}
else {xson[2]=xilk[0],yson[2]=yilk[0],xson[3]=xilk[1],yson[3]=yilk[1];}}}

for(b=0;b<4;b++){
Xyeni=xson[b];
Yyeni=yson[b];
```

```
if(Xyeni>Xeski){
Xdurum=Xyeni-Xeski;Xeski=Xyeni;

if(Xkonum==2)goto XIA4; else goto XIA2;
while(1){
XIA2:                   XADIM1
                        motorbekle();
                        XADIM2
                        motorbekle();
Xdurum--;if(Xdurum==0){Xkonum=2; break;}
XIA4:                   XADIM3
                        motorbekle();
                        XADIM4
                        motorbekle();
Xdurum--;if(Xdurum==0){Xkonum=4; break;}}}
P1=0;
if(Xyeni<Xeski){
Xdurum=Xeski-Xyeni;Xeski=Xyeni;

if(Xkonum==2)goto XGA4; else goto XGA2;
while(1){
XGA2:                   XADIM3
                        motorbekle();
                        XADIM2
                        motorbekle();
Xdurum--;if(Xdurum==0){Xkonum=2; break;}
XGA4:                   XADIM1
                        motorbekle();
                        XADIM4
                        motorbekle();
Xdurum--;if(Xdurum==0){Xkonum=4; break;}}}
P1=0;
```

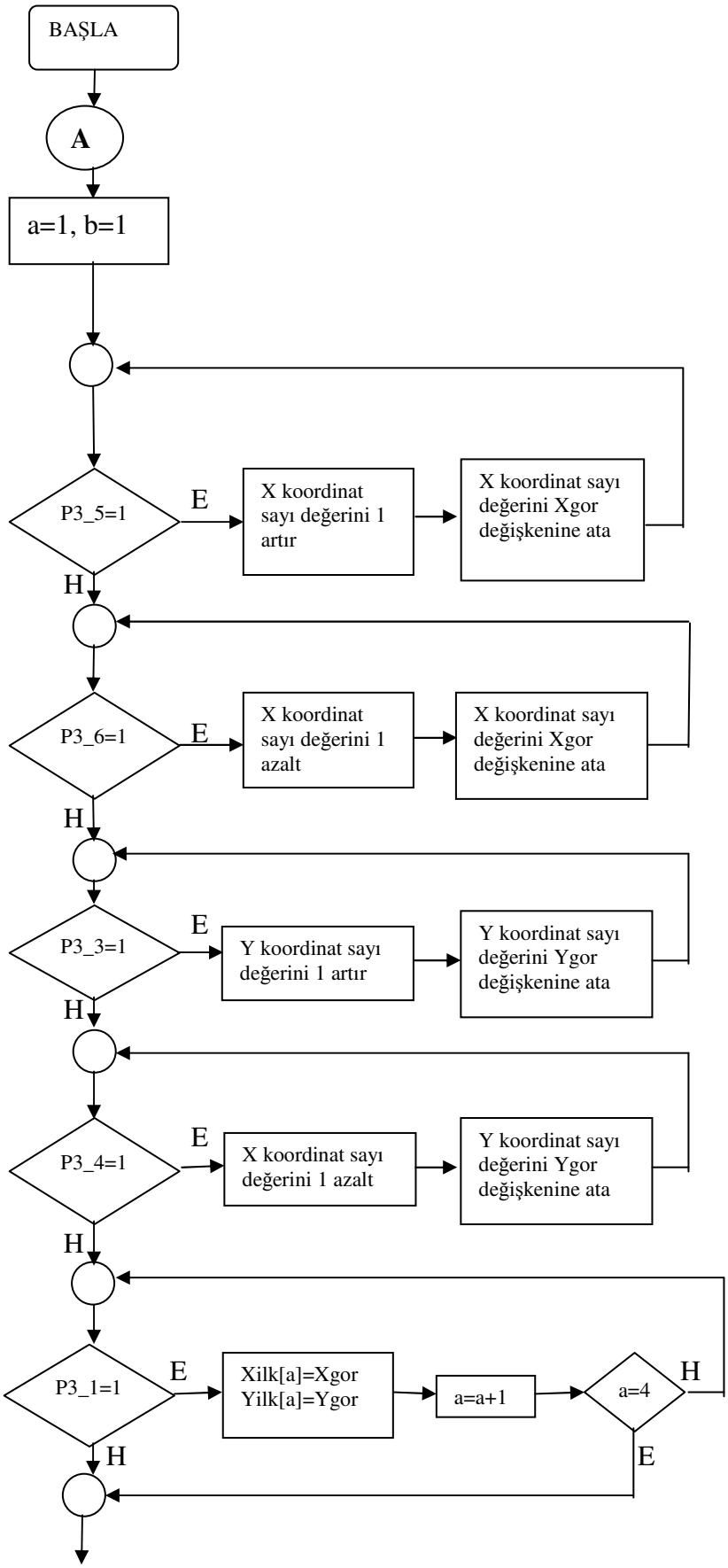
```
if(Yyeni>Yeski){
Ydurum=Yyeni-Yeski;Yeski=Yyeni;

if(Ykonum==2)goto YIA4; else goto YIA2;
while(1){
YIA2:                   YADIM1
                        motorbekle();
                        YADIM2
                        motorbekle();
Ydurum--;if(Ydurum==0){Ykonum=2; break;}
YIA4:                   YADIM3
                        motorbekle();
                        YADIM4
                        motorbekle();
Ydurum--;if(Ydurum==0){Ykonum=4; break;}}}
P1=0;

if(Yyeni<Yeski){
Ydurum=Yeski-Yyeni;Yeski=Yyeni;

if(Ykonum==2)goto YGA4; else goto YGA2;
while(1){
YGA2:                   YADIM3
                        motorbekle();
                        YADIM2
                        motorbekle();
Ydurum--;if(Ydurum==0){Ykonum=2; break;}
YGA4:                   YADIM1
                        motorbekle();
                        YADIM4
                        motorbekle();
Ydurum--;if(Ydurum==0){Ykonum=4; break;}}}
P1=0;
```

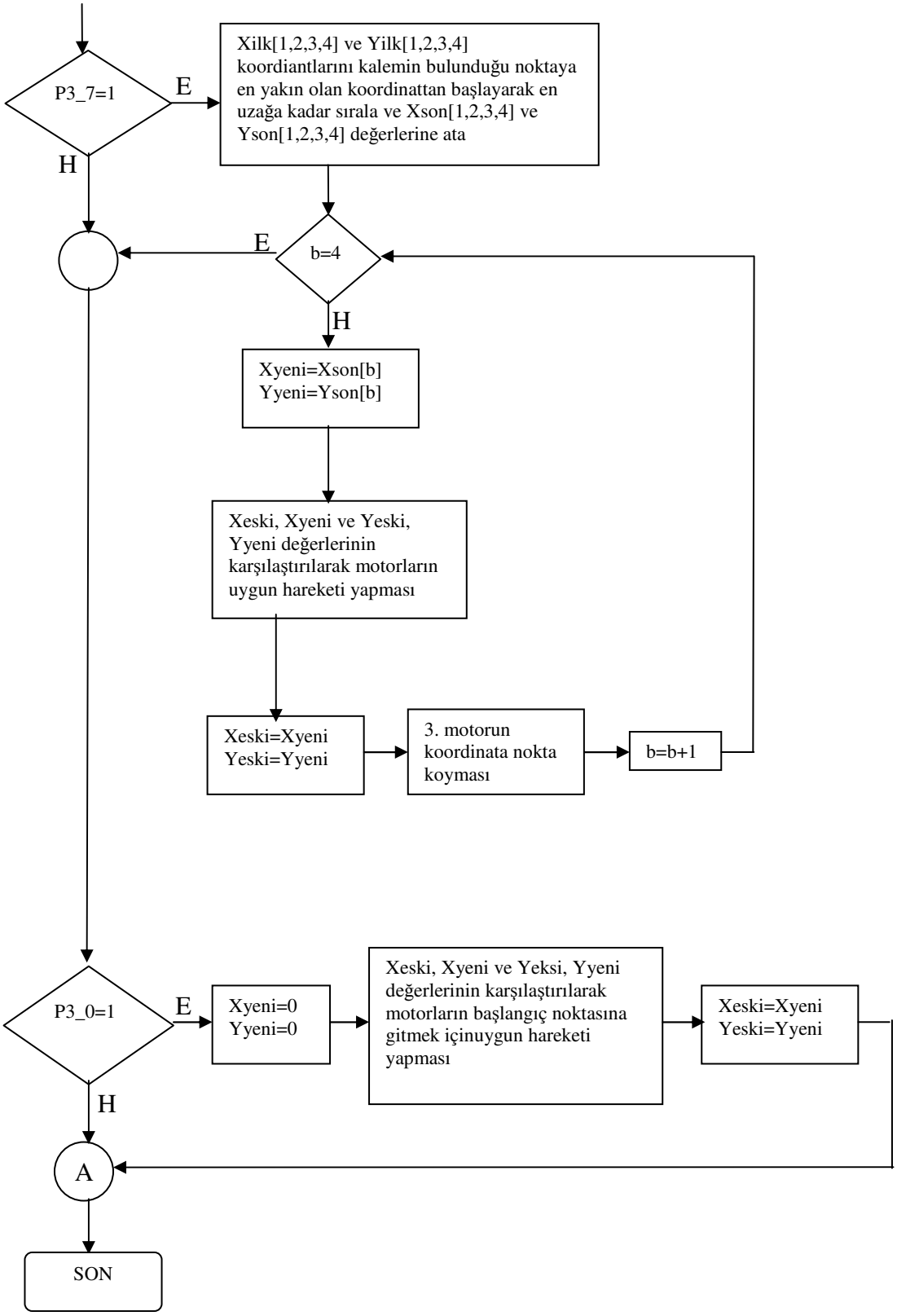
```
while(s<290){
P3_0=1,P3_1=0,P3_7=0,P3_2=0;motorbekle2();
P3_0=0,P3_1=1,P3_7=0,P3_2=0;motorbekle2();
P3_0=0,P3_1=0,P3_7=1,P3_2=0;motorbekle2();
P3_0=0,P3_1=0,P3_7=0,P3_2=1;motorbekle2();
s=s+1;}
motorbekle3();
while(d<290){
P3_0=0,P3_1=0,P3_7=1,P3_2=0;motorbekle2();
P3_0=0,P3_1=1,P3_7=0,P3_2=0;motorbekle2();
P3_0=1,P3_1=0,P3_7=0,P3_2=0;motorbekle2();
P3_0=0,P3_1=0,P3_7=0,P3_2=1;motorbekle2();
d=d+1;}
s=0,d=0;
P3=0;}b=0;}

if(P3_1==1){
Xyeni=0,Yyeni=0;

if(Xyeni>Xeski){
Xdurum=Xyeni-Xeski;Xeski=Xyeni;

if(Xkonum==2)goto XIA04; else goto XIA02;
while(1){
XIA02:                  XADIM1
                        motorbekle();
                        XADIM2
                        motorbekle();
Xdurum--;if(Xdurum==0){Xkonum=2; break;}
XIA04:                  XADIM3
                        motorbekle();
                        XADIM4
```

```
                    motorbekle();
Xdurum--;if(Xdurum==0){Xkonum=4; break;}}}
P1=0;
if(Xyeni<Xeski){
Xdurum=Xeski-Xyeni;Xeski=Xyeni;

if(Xkonum==2)goto XGA04; else goto XGA02;
while(1){
XGA02:              XADIM3
                    motorbekle();
                    XADIM2
                    motorbekle();
Xdurum--;if(Xdurum==0){Xkonum=2; break;}
XGA04:              XADIM1
                    motorbekle();
                    XADIM4
                    motorbekle();
Xdurum--;if(Xdurum==0){Xkonum=4; break;}}}
P1=0;

if(Yyeni>Yeski){
Ydurum=Yyeni-Yeski;Yeski=Yyeni;

if(Ykonum==2)goto YIA04; else goto YIA02;
while(1){
YIA02:              YADIM1
                    motorbekle();
                    YADIM2
                    motorbekle();
Ydurum--;if(Ydurum==0){Ykonum=2; break;}
YIA04:              YADIM3
                    motorbekle();
                    YADIM4
```

```
                    motorbekle();
Ydurum--;if(Ydurum==0){Ykonum=4; break;}}}
P1=0;
if(Yyeni<Yeski){
Ydurum=Yeski-Yyeni;Yeski=Yyeni;

if(Ykonum==2)goto YGA04; else goto YGA02;
while(1){
YGA02:              YADIM3
                    motorbekle();
                    YADIM2
                    motorbekle();
Ydurum--;if(Ydurum==0){Ykonum=2; break;}
YGA04:              YADIM1
                    motorbekle();
                    YADIM4
                    motorbekle();
Ydurum--;if(Ydurum==0){Ykonum=4; break;}}}P1=0;}
}}
```

EK-D Sistem Algoritması





Şekil E.5. Sistem akış diyagramı

ÖZGEÇMİŞ

1981 yılında ISPARTA'da doğdu. İlk ve orta öğrenimini Isparta'da tamamladı. 1999 yılında Isparta Anadolu Meslek Lisesi'nden mezun oldu. Aynı yıl Sakarya Üniversitesi Teknik Eğitim Fakültesi Elektronik ve Bilgisayar Öğretmenliği Bölümü'ne girdi. 2003 yılında bu bölümü bitirdi. Mezuniyetten sonra 2003/2004 öğretim yılında Sakarya Üniversitesi Fen Bilimleri Enstitüsü Elektronik ve Bilgisayar Eğitimi A.B.D.'de yüksek lisans eğitimine başladı ve 2006 Nisan ayında eğitimini tamamladı. 2006 Şubat ayında Kocaeli Kandıra Safalı İlk Öğretim Okulunda Bilgisayar Öğretmeni olarak göreve başladı. Halen aynı okulda görevine devam etmektedir.