

ULTRASONİK MESAFE ALGILAYICILARI İLE MOBİL ROBOT KONTROLÜ

ZEKİ MURAT BAYRAM

**YÜKSEK LİSANS TEZİ**

**Elektrik – Elektronik Mühendisliği Anabilim Dalı**

**Haziran 2006**

ULTRASONİK MESAFE ALGILAYICILARI İLE MOBİL ROBOT KONTROLÜ

ZEKİ MURAT BAYRAM

Osmangazi Üniversitesi  
Fen Bilimleri Enstitüsü  
Lisansüstü Yönetmeliği Uyarınca  
Elektrik – Elektronik Mühendisliği Anabilim Dalı  
Kontrol ve Kumanda Bilim Dalında  
YÜKSEK LİSANS TEZİ  
Olarak Hazırlanmıştır

Danışman: Doç. Dr. Osman PARLAKTUNA

Haziran 2006

Zeki Murat BAYRAM' in YÜKSEK LİSANS tezi olarak hazırladığı “ULTRASONİK MESAFE ALGILAYICILARI İLE MOBİL ROBOT KONTROLÜ” başlıklı bu çalışma, jürimizce lisansüstü yönetmeliğinin ilgili maddeleri uyarınca değerlendirilerek kabul edilmiştir.

Üye : Doç. Dr. Osman PARLAKTUNA

Üye : Prof. Dr. Abdurrahman KARAMANCIOĞLU

Üye : Yrd. Doç. Dr. Ayşe KAHVECİOĞLU

Fen Bilimleri Enstitüsü Yönetim Kurulu'nun ..... tarih ve ..... sayılı kararıyla onaylanmıştır.

Prof. Dr. Abdurrahman KARAMANCIOĞLU

Enstitü Müdürü

## ÖZET

Bu çalışmada ultrasonik mesafe algılayıcılardan aldığı bilgileri kullanarak önündeki, yanlarındaki ve arkasındaki engelleri algılayan ve bu bilgileri kullanarak yönünü bulan bir mobil robot uygulaması üzerinde çalışılmıştır. Öncelikle robotlar hakkında genel bilgi verilmiş, daha sonra üzerinde çalışılan mobil robotun çalışma mantığı, donanımı, kartlar ve kullanılan elemanlar anlatılmıştır. En son aşamada ise amaca göre oluşturulan yazılım ve tasarlanan kartlar sunulmuştur.

## **SUMMARY**

In this research, the control of mobile robot by using ultrasonic sensors is applied. First the general information about robots are explained and working of mobile robot, its hardware, PCBs and components used are evaluated. Finally, software and circuits are presented.

## TEŐEKKÜR

Görüş ve önerileriyle çalışmama yön veren değerli hocalarım Doç. Dr. Osman PARLAKTUNA' ya ve her zaman yanımda olduklarını bildiğim canım aileme teşekkür ediyorum.

Zeki Murat BAYRAM

## İÇİNDEKİLER

### Sayfa

<b>ÖZET .....</b>	<b>iv</b>
<b>SUMMARY .....</b>	<b>v</b>
<b>TEŞEKKÜR .....</b>	<b>vi</b>
<b>ŞEKİLLER DİZİNİ.....</b>	<b>ix</b>
<b>1. GİRİŞ VE AMAÇ .....</b>	<b>1</b>
<b>2. ROBOT NEDİR?.....</b>	<b>3</b>
<b>3. MOBİL ROBOTUN DONANIMI.....</b>	<b>5</b>
3.1. Ultrasonik Mesafe Algılayıcısı.....	5
3.2. Adım Motoru ve Sürülmesi.....	9
3.2.1. Adım motoru çeşitleri.....	11
3.2.1.1. Değişken relüktanslı adım motorlar.....	11
3.2.1.2 Sabit mıknatıslı adım motorlar.....	12
3.2.1.2.1. Tek kutuplu adım motor.....	12
3.2.1.2.2 Çift kutuplu adım motor.....	13
3.2.2. Adım motorlara ait önemli parametreler.....	15
3.2.2.1. Çözünürlük.....	15
3.2.2.2. Doğruluk.....	15
3.2.2.3. Tutma momenti.....	15
3.2.2.4. Tek adım tepkisi.....	16
3.2.2.5. Sürekli rejimde maksimum yük momenti eğrisi.....	16
3.2.3. Adım motorlarının denetimi ve sürülmesi.....	16
3.3. Pic Mikrodenetleyicisi.....	20
<b>4. BASKI DEVRE YAPIMI.....</b>	<b>31</b>
<b>5. SONUÇ VE ÖNERİLER.....</b>	<b>33</b>

## **İÇİNDEKİLER (devam)**

<b>6.KAYNAKLAR DİZİNİ.....</b>	<b>34</b>
<b>EK-1 MOBİL ROBOTUN YAZILIMI.....</b>	<b>35</b>
<b>EK-2 TASARLANAN DEVRELER.....</b>	<b>49</b>
<b>EK-3 MOBİL ROBOTA AİT FOTOĞRAFLAR.....</b>	<b>53</b>

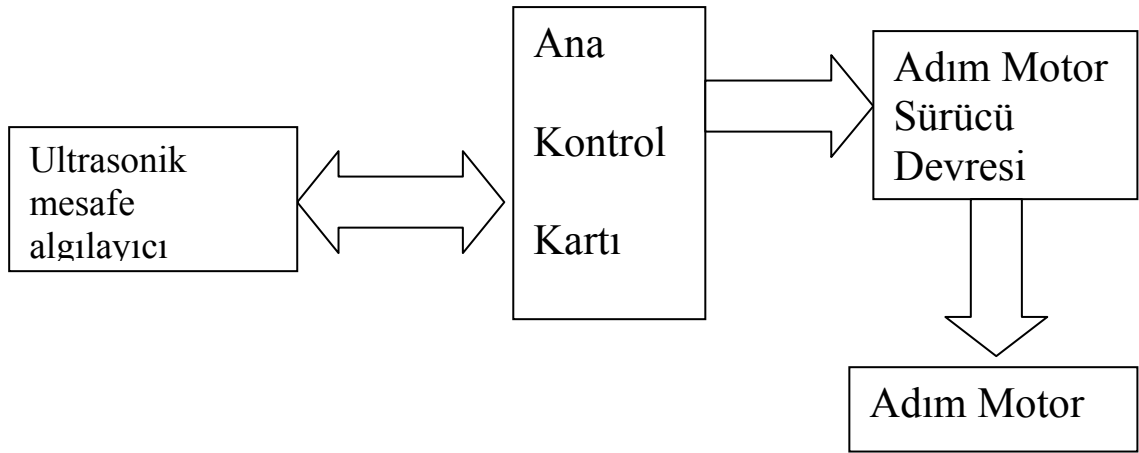


## ŞEKİLLER DİZİNİ

<u>Sekil</u>	<u>Sayfa</u>
1.1. Mobil Robotun Basitleştirilmiş Blok Diyagramı.....	1
3.1.1. SRF04 Ultrasonik Mesafe Algılayıcı .....	5
3.1.2. Algılayıcıların Yerleşimi.....	6
3.1.3. Algılayıcı Bağlantı Bacakları.....	7
3.1.4. Zaman Diyagramı.....	7
3.1.5. Algılayıcının Dalga Yapısı .....	8
3.2.1. Adım Motorun Genel Çalışma Mantığı.....	10
3.2.2. Yarım-Adım Yöntemi .....	11
3.2.3. Değişken Relüktanslı Adım Motorunun Sürülmesi.....	12
3.2.4. Tek Kutuplu Adım Motor Sargısı.....	12
3.2.5. Tam-Adım ve Yarım-Adım Döngüsü.....	13
3.2.6. Çift Kutuplu Adım Motorun Sürülmesi.....	14
3.2.7. Örnek H-Köprü Devresi.....	15
3.2.8. Açık Döngülü Denetim.....	17
3.2.9. Adım Motorunun Kapalı Döngülü Denetimi.....	18
3.2.10. Lojik Sıralayıcının Motora Bağlanması.....	18
3.2.11. UCN5804 Bağlantı Şeması.....	19
3.3.1. PIC16F877 bağlantı Pinleri.....	21
4.1. Bastır ve ayır Kağıdı.....	31
4.2. Bakır Plakanın Temizlenmesi.....	31
4.3. Devrenin Ütü ile Yapıştırılması.....	32
4.4. Asite Atılacak Plaka.....	32
EK-2 Anakart Devresi.....	49
EK-2 Motor Sürücü Devresi.....	50
EK-2 Sensör Devresi.....	51
EK-2 Güç Devresi.....	52

## 1. GİRİŞ VE AMAÇ

Bu çalışmanın amacı, serbest dolanım yapan ve yol üzerindeki engelleri algılayıp bu engellere çarpmadan yoluna devam eden bir mobil robot tasarlanması ve gerçekleştirilmesidir. Daha önceden hazır olan mobil robot platformunda 2 adet adım motor ve 4 adet ultrasonik mesafe algılayıcısı bulunmaktadır. Adım motorlardan bir tanesi ileri ve geri hareketi sağlamakta, diğeri ise robotun engellerden kurtulma yönüne göre sağa ve sola dönebilmektedir. Robot üzerine değişik amaçlı 3 adet elektronik kart tasarlanıp monte edilmiştir. Ana kartımızda mobil robotun tüm fonksiyonlarını kontrol eden ve derleyen PIC16F877 mikro denetleyicisi kullanılmıştır. Diğerkartımız adım motor sürücü kartı olup ana karttan aldığı bilgiler çerçevesinde motorları yönetmektedir. Son kartımız ise güç kartı olup 5 V ve 12 V'un sağlandığı karttır.



Şekil 1.1: Mobil Robotun Basitleştirilmiş Blok Diyagramı

Ana kontrol kartında kullanılan mikro denetleyici için geliştirilen yazılım Ek-A'da verilmiştir. Yazılım MPLAB IDE programı kullanılarak oluşturulmuş ve kodlar PICC derleyicisi ile derlenmiştir.

Tasarlanan 3 kartta Protel DXP benzetim programı ile önce test edilmiş ve sorunsuz çalıştığına karar verilip baskı devreye aktarılmıştır.

Tez çalışmamın ikinci bölümünde robotlarla ilgili genel bilgilere ve robot tarihçesinden bahsedilmiştir. Projenin kapsamı, öngörülen malzemelerin özellikleri, kullanımı ve donanım ile ilgili tüm konular üçüncü bölümde yer almaktadır. Dördüncü bölümde ise ilgili devrelerin tasarlanması ve baskı devre yapımı ile ilgili bilgiler verilmektedir. Beşinci ve son bölümde, sonuçlar, karşılaşılan zorluklar ve öneriler verilmiştir.

## 2. ROBOT NEDİR?

Bir robot, bu konuda çalışmalarıyla tanınan Maja Mataric'in yaptığı tanıma göre, ortamdan topladığı verileri dünyası hakkında sahip olduğu bilgiyle sentezleyerek, anlamlı ve amaçlarına yönelik bir şekilde hareket edebilen ve bunu güvenli bir biçimde yapabilen bir makinedir. Bu tanıma girebilmesi için öncelikle robot diye adlandırdığımız makinenin fiziksel bir varlığının olması gereklidir. Yani sadece bir bilgisayar benzetimi değil, gerçek bir ortamla iletişim halinde olan bir makine kastedilmektedir. [7]

Amerikan Robot Enstitüsüne göre ise robot, eşyaları, parçaları, aletleri ve özel aygıtları programlanmış şekilde hareket ettirebilen ve yeniden programlanabilen çok fonksiyonlu bir cihazdır. İnsana benzeyen ama bazı yönleriyle insandan eksik olan varlıklar aslında çok eski bir düşüncedir. Bu düşünce, ortaya çıkışından beri insandan daha aşağı olan bu varlıkların insana hizmet için var olduğu varsayımıyla birlikte yürümüştür.

İnsanlığın kendine yardımcı olacak mekanizmalar düşünmesi, tahminlerimizden çok daha eskilere dayanmaktadır. M.Ö 800'de, Homeros, İlyada adlı eserinde verilen görevleri yerine getirebilen hareketli üç ayaklılardan bahseder. M.Ö. 350'de Aristo, bir eserinde; "Eğer her araç kendi işini görebilseydi, insan eline ihtiyaç duymadan mekik kendi dokuyabilse, lir kendi çalabilseydi, yöneticilerin elemanlara ihtiyacı kalmazdı." Diyerek ilk otomasyon fikrini ortaya atmıştır.13.yy ise Eb-ül-İz-el-Cezeri adlı bir Arap otomatlar hakkında bir kitap yazdı. Kitapta 300'e yakın otomatik mekanizmanın yanı sıra, çamaşır teknesini doldurup boşaltabilen otomatik bir Arap kadını resmediliyordu. Robot teknolojisi, ismini Çek oyun yazarı Karel Capek'in "Rossum'un Evrensel Robotları (1921)"<sup>(1)</sup> oyununa borçludur. Yazar, angarya-zorunlu iş anlamındaki "robata" kelimesi ile işçi anlamına gelen "robotnik" kelimelerini birleştirerek "robotic" kelimesini türetmiştir.

Isaac Asimov ise yazdığı bilimkurgu romanlarla “robot” fikrinin önceliğini yapmıştır. Yazarın kitaplarında bahsettiği 3 kanun Asimov kuralları olarak bilinmektedir. [6]

1. Bir robot insana zarar vermez ve bir insanın zarar görmesine izin vermez.
2. Bir robot birinci kanuna aykırı olmadığı sürece insanlar tarafından verilen tüm emirlere itaat eder.
3. bir robot birinci ve ikinci kanuna aykırı bir durum olmadığı sürece kendi varlığını korur.

Bu şekilde bilimkurgu dünyasında az çok oturmuş bir robot modeli çıkar karşımıza. Bu model öylesine etkili olmuştur ki, günümüzde robot araştırmaları sadece endüstride kullanımı hedefleyen fonksiyonel tasarımların ötesinde insansı ve hayvansı davranış diye adlandırılacak özelliklere sahip robotlar üzerinde yoğunlaşmaktadır.

Robotlar üzerinde yapılan araştırmaların son yıllarda yoğunlaşmasının temel sebebi bu konuda üretilmiş olan bilgi birikiminin daha güvenilir tasarımlara daha çabuk ulaşılmasını sağlamasıdır. Endüstride robotlar hızlı ve görece hatasız çalışmalarıyla verimi artırırken, insanların çalışmasını kısıtlayan organik faktörlerden etkilenmemeleri onlara bazı işlerde önemli avantajlar sağlamaktalar.

Örnek vermek gerekirse, elektronik devrelere pirinç tanesi büyüklüğündeki yüzlerce parçanın yerleştirilmesi insanlarca yapıldığında yorucu, zahmetli, sıkıcı ve hata oranı yüksek bir iştir. Oysa bir robot kolu bu işi süratli ve hatasız bir biçimde gerçekleştiriyor.

Rutin işlerin yanı sıra insanların uzun süreler çalışamayacakları ortamlarda da robotlara iş düşüyor. Deniz dibinde yüksek basınç altında, yüksek sıcaklıkta, Mars yüzeyi gibi elverişsiz ortamlarda, hatta insan vücudunun sığamayacağı genişlikteki havalandırma borularında robotlar kullanılıyor. [7]

### 3. MOBİL ROBOTUN DONANIMI

Robotlar pratikte çok çeşitli uygulama alanında kullanılmaktadır ve boyutları da görevlerine göre çok farklı özellikler göstermektedir. Fakat temel özelliklerini göz önüne alırsak algılayıcılardan, elektro-mekanik hareket sistemlerinden ve programlanabilir işlem birimlerinden oluştukları söylenebilir.

#### 3.1. Ultrasonik Mesafe Algılayıcısı

Bir mobil robot bir yerden diğer bir yere başarılı bir şekilde yol alabilmek için etrafındaki engeller hakkında bir takım bilgileri edinmek zorundadır. Bu bilgiler uygun algılayıcıların yardımıyla elde edilebilir. Gerçek zamanlı engelden sakınma, mobil robot sistemlerinin başarılı uygulamaları için anahtar rol oynar. Tüm mobil robotlar, engelleri tespit eden ve robotu durduran ilkel algoritmalarından, engelleri aşabilen karmaşık algoritmalara kadar değişen birtakım özellikler içerirler. Tüm bunları yapabilmeleri için mobil robotlar ilkel algılayıcılardan karmaşık algılayıcılara kadar bir çok algılayıcı içerebilirler. [3]

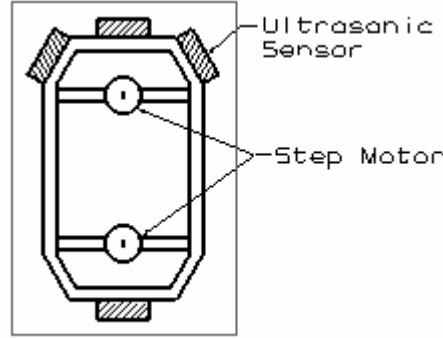
Projede engel bilgilerine Devantech SRF04 marka ultrasonik mesafe algılayıcılar yardımıyla ulaşıldı. Önde, arkada, sağ ve sol ön çaprazlarda olmak üzere 4 adet algılayıcı konumlandırılmıştır. Algılayıcının çalışması, iç yapısı ve genel özellikleri ile ilgili bilgi vermek gerekirse;



Şekil 3.1.1. SRF04 ultrasonik mesafe algılayıcı

Şekil 3.1.1.'de gösterilen Devantech SRF04 ultrasonik mesafe algılayıcı 3 cm. ile 3 m. arasındaki mesafelerin algılanmasında kullanılır. Algılayıcı, ultrasonik darbeler gönderir ve engellere çarpıp gelen ses darbelerini tekrar geri alır. Bu işlem süresince geçen zaman bilinir ve bu zamana göre engelin ne kadar uzaklıkta olduğu basitçe hesaplanır. Kısaca kontrol etmemiz gereken sadece sinyalin çıktıktan sonra geri gelinceye kadar geçen süredir.

4 adet algılayıcı ön sağ, ön sol, öne ve arka tarafa konumlandırılmıştır. Yerleşim şekil 3.1.2 de verilmiştir. Engel algılamak için dört adet algılayıcıya birden enerji verilmemekte aksine algılayıcılara sıra ile enerji verilerek alıcılardan gelen tepkiler değerlendirilmektedir.

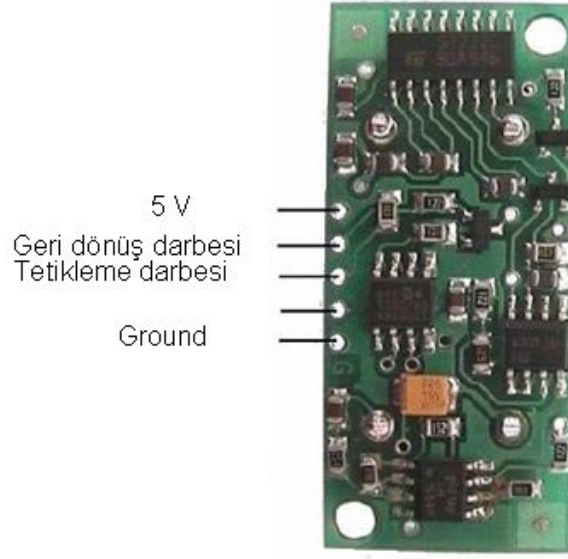


Şekil 3.1.2 Algılayıcıların yerleşimi

Algılayıcının bazı genel özellikleri ise şu şekildedir.

- 5 V giriş voltajı
- 30 mA akım
- 40 kHz frekans
- Mesafe aralığı 3 cm. – 3 m.
- Minimum 10µs. tetikleme darbesi gerekmektedir.
- 43 mm x 20 mm x 17 mm boyutlarındadır.

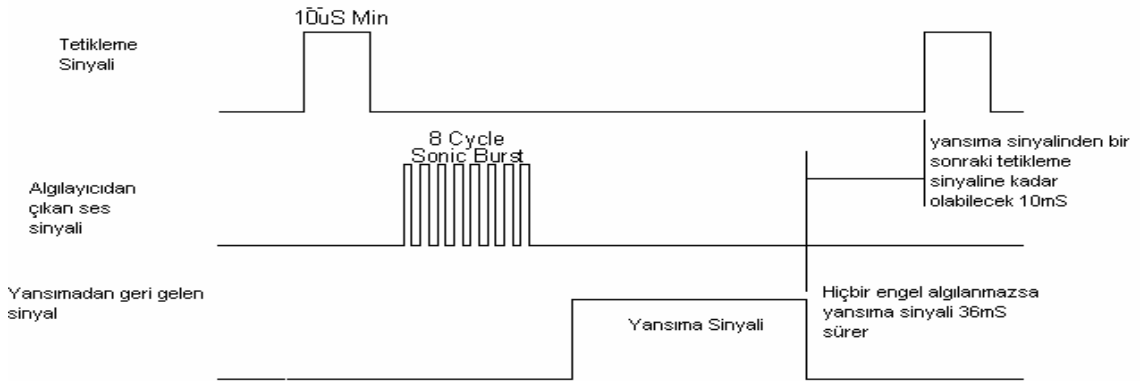
Algılayıcının bağlantıları aşağıdaki gibidir.



Şekil 3.1.3. Algılayıcı Bağlantı Bacakları

SRF04'ün çalışması için tetikleme girişine  $10\mu\text{s}$ 'lik bir darbe sinyali göndermemiz yeterli olmaktadır. Algılayıcı 40 kHz.'de 8 döngülü bir ses sinyali yaymaya başlar ve yansımaları almaya hazır durumda bekler. Cisimden çarpıp geri dönen sesin dalga genişliği cismin uzaklığıyla orantılıdır. Eğer algılayıcı hiçbir cisim algılamazsa 36 ms. sonra yankı beklemeyi bırakır. Aşağıda SRF04'ün zaman diyagramını görebilirsiniz.

### SRF04 ZAMAN DİYAGRAMI



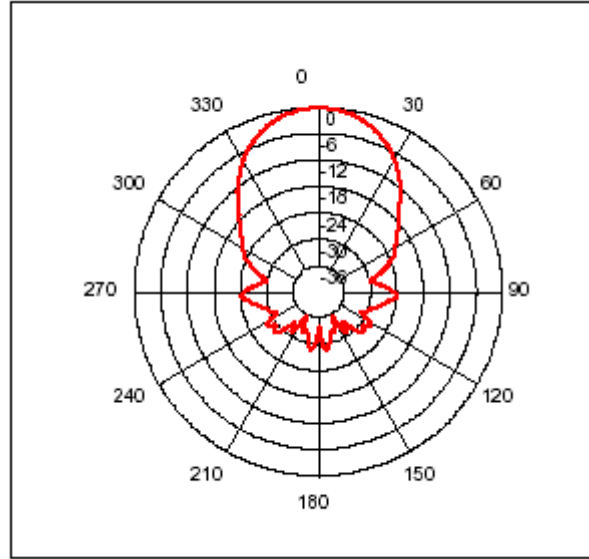
Şekil 3.1.4. Zaman Diyagramı



Eğer darbe sinyelimiz  $\mu\text{s}$  türünde ise bu değer 58'e bölerek cm cinsinden, 148'e bölerek ise inch cinsinden uzaklığı bulabiliriz.

Kullanılan 4 adet algılayıcıyı aynı anda tetiklemek bazı hatalı okumalara neden olabilmektedir, bu nedenle her algılayıcıyı 65 ms.'lik aralıklarla tetikleme yapmamız en sağlıklı sonuçları almamızı sağlayacaktır. Bu süreden daha önce algılayıcıyı tetiklememiz engelle çarpacak ses dalgası ile engelden gelecek ses dalgasının karışmasına neden olabilir.

SRF04 ultrasonik mesafe algılayıcının dalga yapısı aşağıdaki şekildedir. Algılayıcı konik bir dalga modeline sahiptir. Ayrıca, algılayıcının etkili olduğu açı değeri yaklaşık olarak  $55^\circ$  civarlarındadır.



Şekil 3.1.5. Algılayıcının dalga yapısı

Ultrasonik algılayıcıların mobil robotlarda kullanımı ile ilgili bazı avantaj ve dezavantajları bulunmaktadır.

Kolay adapte edilebilir olması, darbelere, toza ve nemli ortamlara karşı dayanıklı olması en belirgin avantajlarıdır.

Işık algılayıcılarına göre daha yavaş olması, alıcı ve vericinin yan yana olması nedeniyle kör aralık diye adlandırabileceğimiz frekanslarda atlamaların olabilmesi, çalışırken 2.5 A çekmesi, aktif değilken 150 mA akım gerekliliği ilk başta söylenebilecek dezavantajlarıdır.

### **3.2.Adım Motoru ve Sürülmesi**

Çok hassas sinyallerle sürülen ve açısal konumunu adımlar halinde değiştiren motorlara adım motorlar denir. Adından da anlaşılacağı gibi adım motorlar belirli adımlarla hareket ederler. Bu adımların her uyarımda ne kadar olacağı adım motorun adım açısına bağlıdır.

Adım motorlarının hangi yöne doğru döneceği, devir sayısı, dönüş hızı gibi değerler mikroişlemci veya bilgisayar yardımı ile kontrol edilebilir. Sonuç olarak adım motorlarının hızı, dönüş yönü ve konumu her zaman bilinmektedir. Bu özelliklerinden dolayı adım motorları çok hassas konum kontrolü istenen yerlerde çok kullanılırlar. Adım motorlarının kullanıldıkları yerlere örnek olarak, endüstriyel kontrol teknolojisi içerisinde bulunan bazı sistemler, robot sistemleri, takım tezgâhlarının ayarlama ve ölçmeleri verilebilir. Ayrıca, adım motorları konumlandırma sistemlerinde ve büro makineleri ile teknolojisi alanında da kullanma alanı bulmaktadır.

Adım motorlarının bu kadar çok kullanılma alanı bulmasının nedeni bu motorların bazı avantajlara sahip olmasıdır. Bu avantajlar aşağıdaki gibi sıralanabilir.

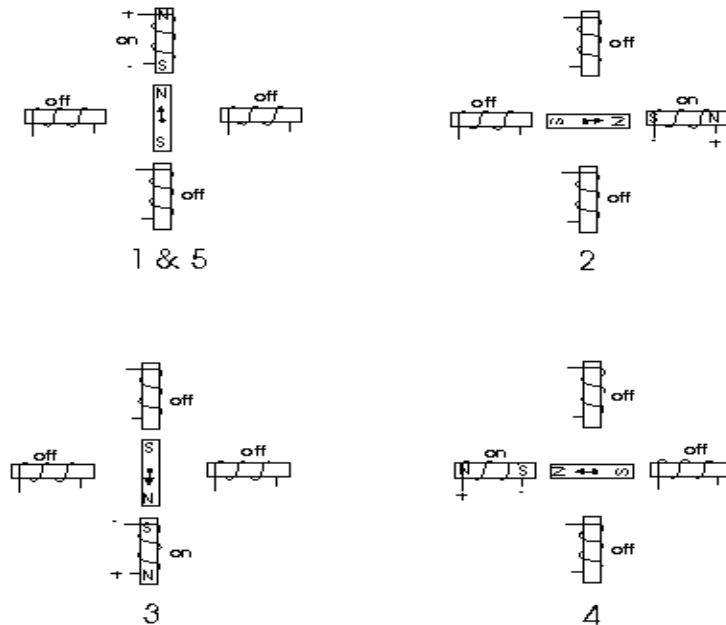
- Geri beslemeye ihtiyaç göstermezler. Açık döngülü olarak kontrol edilebilirler.
- Motorun hareketlerinde konum hatası yoktur.
- Sayısal olarak kontrol edilebildiklerinden bilgisayar veya mikroişlemci gibi elemanlarla kontrol edilebilirler.
- Mekanik yapısı basit olduğundan bakım gerektirmezler.
- Herhangi bir hasara yol açmadan defalarca çalıştırılabilirler.

Adım motorlarının bu avantajları yanında bazı dezavantajları da aşağıdaki şekilde sıralanabilir.

- Adım açıları sabit olduğundan hareketleri sürekli değil darbelidir.
- Sürtünme kaynaklı yükler, açık döngülü kontrolde konum hatası meydana getirirler.
- Elde edilebilecek güç ve moment sınırlıdır. [8]

Adım motorun kablolarından bir veya iki tanesi ortaktır (vMotor). Yaptığımız işlem basit olarak bu ortak kabloya sürekli +12 Volt göndermek ve diğer uçları ise belli bir sırada toprağa göndererek bir adım hareketi elde etmek. Bunu iki farklı şekilde yapabiliriz. Bunlardan bir tanesi 12 Voltluk bir röle devresine benzer bir devrenin 4 transistörlüsü ile bu sinyalleri göndermek. Bir diğer yolu ise içerisinde bu transistörleri hali hazırda bulunduran bir entegre kullanmak.

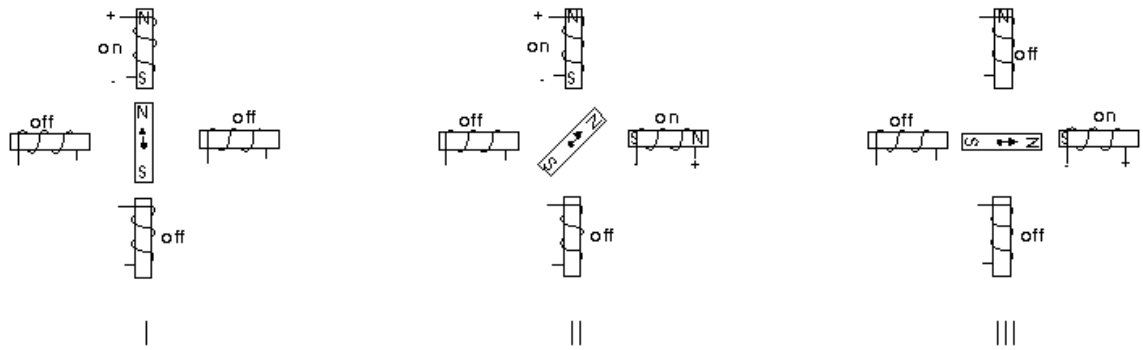
Adım motorlar rotor ve stator kısımlarından oluşmaktadır. Şekil 3.2.1 de adım motorun bir tam turunu görebiliriz. Birinci pozisyonda en üstteki kutup aktif olur. Rotorun saat yönünde 90 dönmesiyle üstteki kutup pasif ve sağdaki kutup aktif konuma geçer. Bu dönme işlemi rotor ilk pozisyonuna gelene kadar devam eder.



Şekil 3.2.1. Adım Motorun Genel Çalışma Mantığı

Yukarıdaki genel anlatımda ki adım motorun adım açısı  $90^0$ 'dir. Gerçekte, birçok uygulama için kullanışlı bir motor değildir. Adım motorun hassasiyeti her darbede döndüğü açığa bağlıdır. Örneğin, eğer motorumuz  $5^0$ 'lik hassasiyette ise rotor her darbede  $5^0$  döner ve 1 tam turu 72 adımda tamamlar.

Yarım adım atma (half-stepping) yöntemiyle motorun hassasiyetini iki kat fazlasına çıkarabiliriz. Normalde yaptığımız gibi tek kutbu aktif yapma yerine rotor hangi kutuplar arasındaysa bu iki kutbu aktif yapmamız lazım. Şekil 3.2.2 de görüldüğü gibi, ilk durumda üstteki kutup aktif durumdadır. Sonraki durumda üstteki ve sağdaki kutupları aktif yaparak rotorun ikisinin arasında kalması sağlanabilir. Son durumda ise üstteki kutup pasif duruma geçiyor ve sadece sağdaki kutup aktif durumda kalıyor.



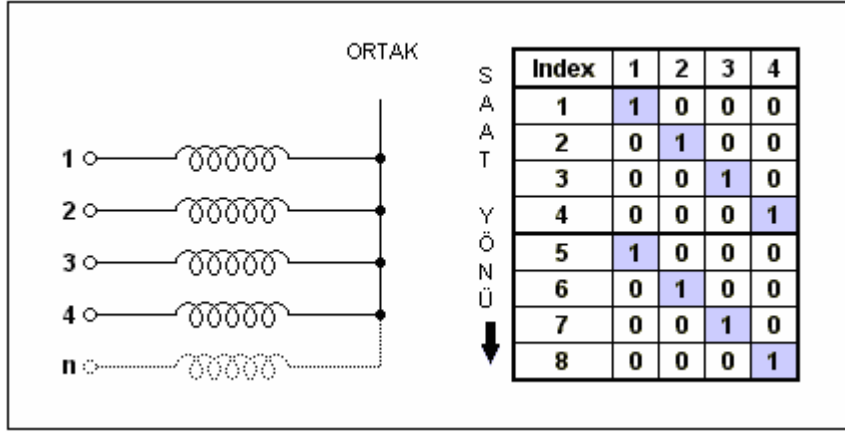
Şekil 3.2.2. Yarım-Adım Yöntemi

### 3.2.1. Adım motor çeşitleri

Adım motorları çalışmaları ve iç yapılarına göre sınıflandırılmaktadır.

#### 3.2.1.1. Değişken relüktanslı adım motorlar

Bu tip motorlar, adım motor çeşitleri içinde kontrolü en kolay olan motor tipidir. Her sarımı sırayla enerjilendirmek motoru sürmemize yeterlidir. Motor şaftını elle döndürdüğümüzde rahatça hareket eder ve adımları hissedemeyiz, bu nedenle bu tip motoru DC motora benzetebiliriz. Aşağıdaki şekil 3.2.3'te bobin ve motoru sürme yapısını görebiliriz.



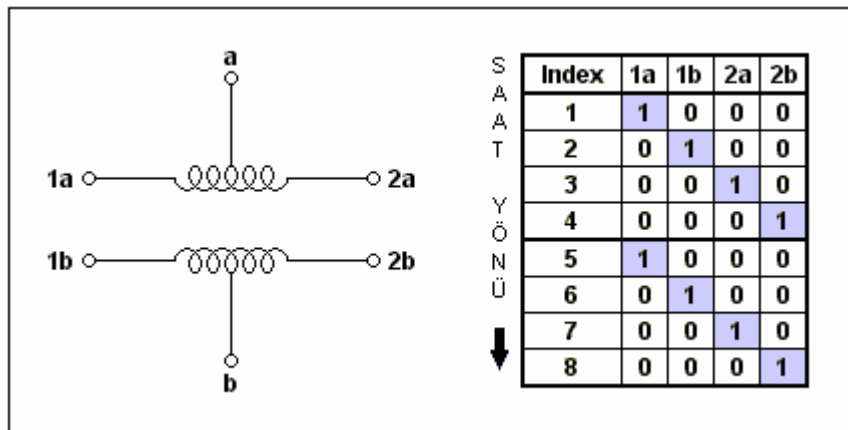
Şekil 3.2.3. Değişken relüktanslı adım motorun sürülmesi

### 3.2.1.2. Sabit mıknatıslı adım motorlar

Sabit mıknatıslı robotları ise genel olarak “tek kutuplu” ve “çift kutuplu” olarak 2’ye ayırabiliriz.

#### 3.2.1.2.1 Tek kutuplu adım motor

Tek kutuplu motorlar kontrolü kolay motorlardır. Bu tip motorun kullanılması sadece bobin voltajını açıp, kapatmak yoluyla olduğundan oldukça kolaydır. Tek kutuplu adım motoru diğerlerinden farklı olan sarımından ayırt edebiliriz. Her bobin 2’ye bölüdüğünden dolayı, bobin sayısının 2 katı kadar faz vardır. Aşağıdaki şekilde örnek sarımları görebiliriz.



Şekil 3.2.4. tek kutuplu adım motor sarımsı

Standart sürme düzenine ilave olarak, yüksek tork ve yarım-adım sürme düzenleride mümkündür. Yüksek torkta, 2 sargı aynı anda aktif edilir. Bu 2 sargı kombinasyonu normal haldekinden 1.5 kat daha fazla tork sağlamaktadır. Yarım-adım ise kombine bir düzendir. İlk olarak 1 sargı aktif edilir, ardından 2 tanesi ve tekrar 1 tanesi...vb. şekilde devam etmektedir. Böylelikle adım sayısını 2'ye çıkartmış oluruz. Şekil 3.2.5.'te tam-adım ve yarım-adım olarak sürülme sırası verilmiştir.

S A A T	Y Ö N Ü	Index	1a	1b	2a	2b
		1	1	0	0	1
2	1	1	0	0		
3	0	1	1	0		
4	0	0	1	1		
5	1	0	0	1		
6	1	1	0	0		
7	0	1	1	0		
8	0	0	1	1		

Bir Başka Tam Adım Düzeni

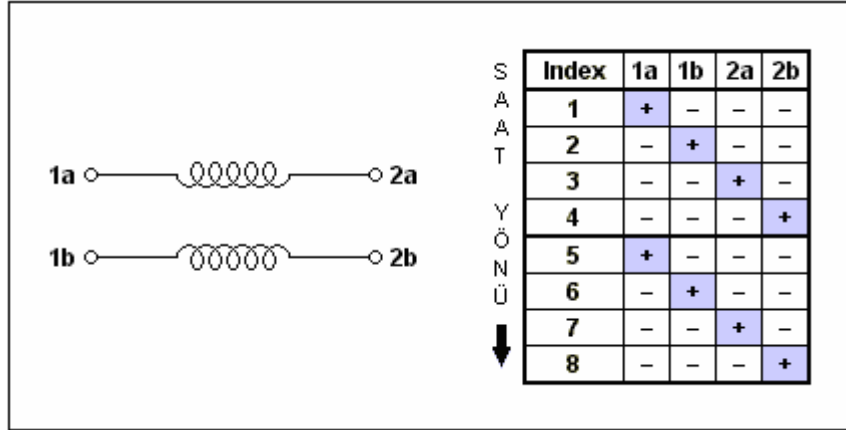
S A A T	Y Ö N Ü	Index	1a	1b	2a	2b
		1	1	0	0	0
2	1	1	0	0		
3	0	1	1	0		
4	0	1	1	0		
5	0	0	1	1		
6	0	0	1	1		
7	0	0	0	1		
8	1	0	0	1		
9	1	0	0	0		
10	1	1	0	0		
11	0	1	0	0		
12	0	1	1	0		
13	0	0	1	0		
14	0	0	1	1		
15	0	0	0	1		
16	1	0	0	1		

Yarım Adım Düzeni

Şekil 3.2.5. Tam-adım ve Yarım-adım döngüsü

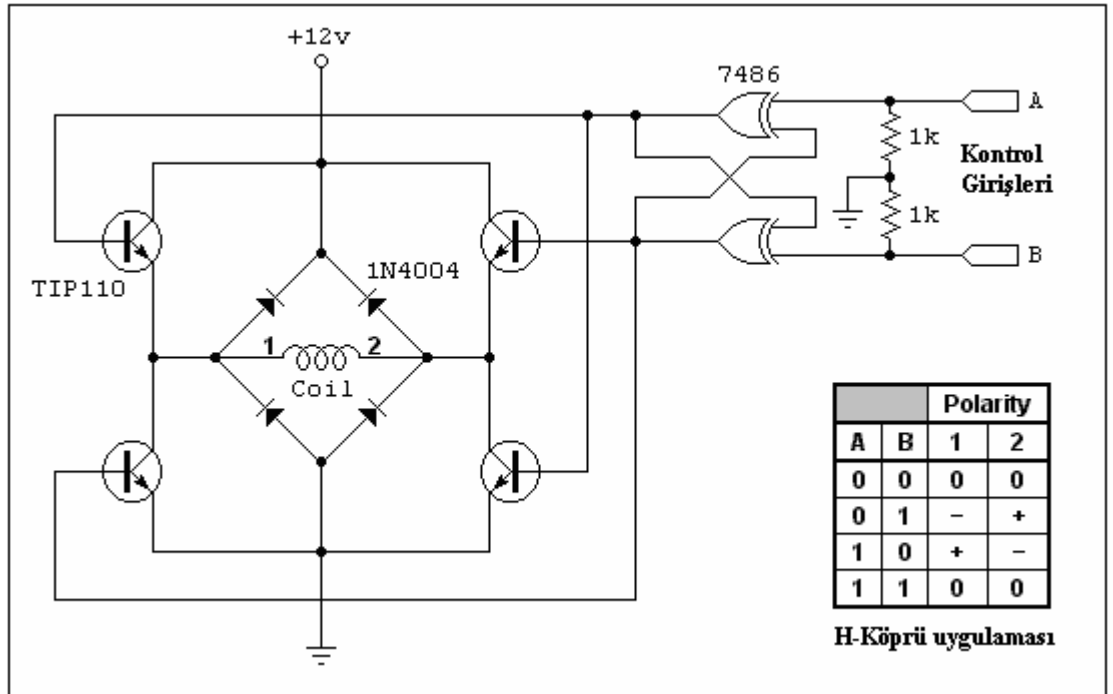
### 3.2.1.2.2 Çift kutuplu adım motor

Tek kutuplu adım motorların aksine, çift kutuplu adım motorları sürmek daha karmaşık devreler gerektirmektedir. Çift kutuplu motorlar tek kutuplu motorlara göre daha fazla tork sağlayabilirler.



Şekil 3.2.6. Çift kutuplu adım motorun sürülmesi

Çift kutuplu motorun sürülmesindeki zorluk her sarımı ters kutuplama yapmayı gerektirecek devrenin olmasıdır. Şekil 3.2.7. de görülen ve H-köprüsü olarak bilinen bu devre çift kutuplu adım motorların sürülmesinde kullanılır. Motorun her sargısının kendi H-köprü sürücü devresi vardır.



Şekil 3.2.7. Örnek H-köprü devresi

## **3.2.2. Adım Motorlara Ait Önemli Parametreler**

### **3.2.2.1. Çözünürlük**

Çözünürlük; bir devirdeki adım sayısı veya dönen motorlar için adım açısı (derece), lineer motorlar için ise adım uzunluğu (mm) olarak tanımlanır. Bu sabit değer, üretim sırasında tespit edilen bir büyüklüktür. Bir adım motorunun adım büyüklüğü, çeşitli kontrol düzenleri ile değiştirilebilir. Yarım adım çalışmada adım büyüklüğü normal değerinin (çözünürlüğünün) yarısına indirilir.

### **3.2.2.2. Doğruluk**

Bir adım motorunun adım konumu, tasarım ve üretim sırasında bir araya getirilen birçok parçanın boyutları ile belirlenir. Bu parçaların boyutlarındaki toleranslar ve dahili sürtünmeler adımların nominal denge konumlarında da toleranslara neden olurlar. Bu durum adım motorunun doğruluğu olarak isimlendirilir ve belli bir konumdaki maksimum açısal hatanın nominal tek adım değerinin yüzdesi olarak ifade edilmiş halidir. Klasik adım motorlarında bu hata % 0.1 ile % 0.5 arasında değişmektedir. Sürtünme momenti veya kuvveti nedeniyle oluşan konum hataları bu doğrulukla ilgisi olmayan, daha az veya çok olabilen rasgele hatalardır. Ancak her iki tip hata toplanarak sistemin toplam hatası elde edilir.

### **3.2.2.3. Tutma momenti**

Tutma momenti, bir adım motorunun en temel moment karakteristiğidir. Tutma momenti eğrisi, motorun ürettiği tutma momentinin rotor konumuna bağlı olarak değişimini veren eğridir. Eğrinin merkezi motorun bir fazının uyartılmış olduğu durumda rotorun kararlı adım konumuna karşılık düşer. Bu eğri, rotor adım pozisyonundan uzaklaştırılırsa, motorda indüklenerek olan ve rotoru sıfır momentli adım pozisyonuna geri getirmeye çalışan momentin (tutma momenti) yönünü ve miktarını verir. Tutma momenti eğrisi, motorun tüm rotor konumları ve statik uyarma koşullarındaki ani momentini tam olarak tanımlamak için gereklidir. Diğer moment karakteristikleri (statik ve dinamik) bu eğri temel alınarak elde edilebilir.



#### **3.2.2.4. Tek adım tepkisi**

Motor fazlarından biri uyarılmış durumdaysa motor kararlı bir adım konumundadır. Bu fazın uyarıtımı kesilip yeni bir faz uyarılırsa motor bir adım atacaktır. Rotor konumunun zamana göre bu deęişimi tek adım tepkisi olarak tanımlanır. Tek adım tepkisi, motorun adım hareketinin hızını, tepkinin aşım ve salınım miktarını, adım açısının hassaslığını veren önemli bir karakteristiktir. Adım motorlarından maksimum performans elde edebilmek için tek adım tepkisindeki aşım ve salınımların azaltılması ve yerleşme zamanının kısaltılması gerekmektedir. Bu nedenle tek adım tepkisinin iyileştirilmesi adım motorlarının kontrolünde çok büyük öneme sahiptir.

#### **3.2.2.5. Sürekli rejimde maksimum yük momenti eğrisi**

Sürekli rejimde maksimum yük momenti/ hız eğrisi herhangi bir sabit dönüş hızında, rotor hareketinin giriş darbe dizisiyle olan eşleşmesini bozmadan ve rotorun durmasına neden olmadan sürekli halde motor miline uygulanabilecek maksimum yük momentini verir. Bu moment aynı zamanda, söz konusu hızda motorda meydana gelecek maksimum moment anlamına da gelmektedir. Klasik motorlarda bu eğriye karşılık gelebilecek bir karakteristik yoktur. Maksimum yük momenti eğrisi çalışma noktalarını göstermediği gibi bir transfer fonksiyonu eğrisi de değildir. Sadece, çalışma bölgesini sınırlar. Bu eğrinin sınırladığı bölge içinde herhangi bir noktada motor giriş darbe dizilerini kaybetmeden ve durma tehlikesi olmadan ilgili hız ve yük momenti ile çalışır. Sınırların dışına çıkıldığında bu durum deęişebilir. [8]

#### **3.2.3. Adım Motorlarının Denetimi ve Sürülmesi**

Şekil 3.2.8.'te açık döngü denetim için blok diyagramı görülmektedir. Sayısal kontrol sinyalleri denetleyici tarafından üretilir ve sürücü devre tarafından yükseltilip adım motorunun sargılarına uygulanır. Eğer denetleyici olarak mikroişlemci veya bilgisayar kullanılırsa bu elemanların getirdiği esnekliklerden dolayı aynı denetleyici ile farklı adım motorları kontrol edilebilir. Kontrol edilecek adım motorları 3, 4 veya daha farklı faz sayısına sahip olabilir. Ayrıca kullanılacak uyarıtım metodu için tek-fazlı, iki-

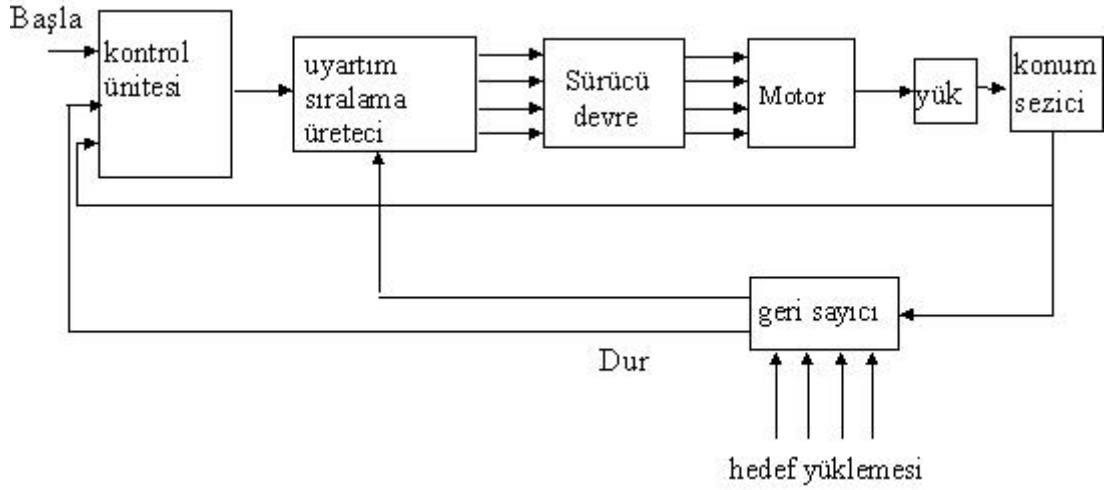
fazlı veya yarım adım uyarımlarından herhangi biri seçilebilir. Bu uyarım metotlarından hangisinin kullanılacağı daha önce de açıklandığı gibi motorun kullanılacağı sisteme bağlıdır.



Şekil 3.2.8. Açık döngülü denetim

Denetleyici tasarlanırken motorun cinsi ve yükün durumu göz önünde bulundurulmalıdır. Bu sırada meydana gelen sınırlamalar kalıcı veya geçici durum sınırlamaları olabilir. Açık döngülü denetimde motorun konumu bilinmediğinden dolayı motorun gönderilen bütün adım komutlarını yerine getirdiği varsayılmaktadır. Eğer uyarım hızı çok yüksek ise, motor adım komutlarından bir kısmını yerine getiremeyebilir. Bu durumda kalıcı bir hata meydana gelir. Bu tür hataların meydana gelmemesi için motor yükünün en büyük olduğu durum göz önüne alınarak hata yapılmayan en yüksek hız belirlenip, bu hızın üzerindeki hızlarda uyarım yapılmamalıdır.

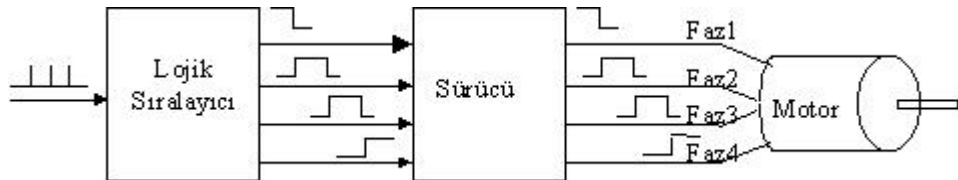
Kapalı döngü sistemlerde ani rotor konumu sezilerek denetim birimine iletilir. Her adım komutu için bir önceki komutun gerçekleştirildiği adım bilgisi alınarak uygulanır. Bu nedenle motor ile denetleyici arasında herhangi bir adım kaybı olmaz. Kapalı döngü denetime bir örnek Şekil 3.2.9.'da gösterilmiştir.



Şekil 3.2.9. Adım motorunun kapalı döngülü denetimi

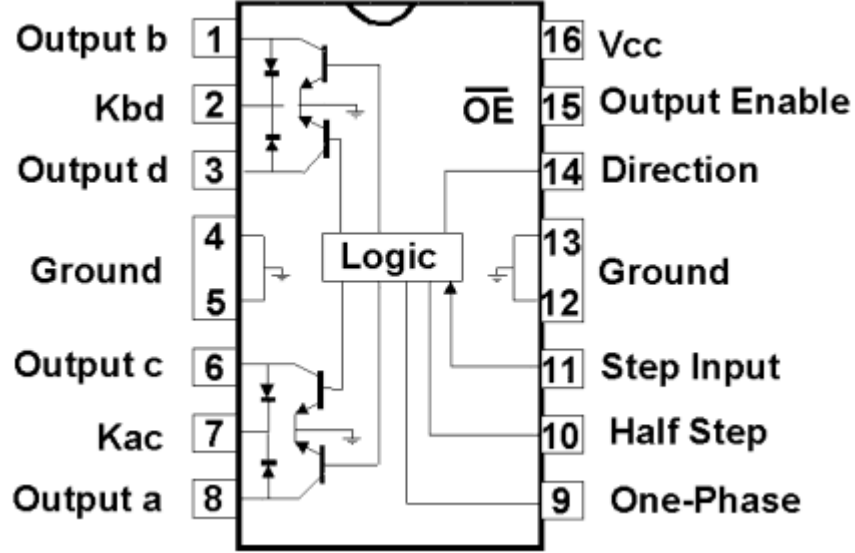
İlk olarak geri sayıcıya hedef konum yüklenir. Daha sonra başla komutu verilerek adım komutlarının sıralayıcıya uygulanması sağlanır. Adım komutlarına bağlı olarak motor adım hareketi yapmaya başlar. İlk adım tamamlanınca, konum sezici geri sayıcıyı ve denetim birimlerini uyarır ve geri sayıcı değeri bir azalır. Eğer bu denetim açık döngülü yapılırsa, geri sayıcı adım komutlarının sayısını yine saklar fakat komutun uygulanıp uygulanmadığı bilinmez. Konum sezici, denetim birimine yeni adım komutu üretimi için sinyal gönderir. Ağır yükler için adım komutları arası sürenin daha büyük olması nedeniyle adım komutlarının ardarda gelmesi istenmez. Yüke göre hız ayarlaması yapılır ve motor hedef konuma gelene kadar bu olaylar tekrarlanır. Adım motoru hedef konuma gelince denetim birimi dur komutu ile uyarılarak yeni adım komutu üretilmesi engellenir. Kapalı döngü sistemi, adım motorunu yük durumunu da göz önüne alarak uyarım sürelerini ayarlar ve en uygun hız profilinde çalıştırır.

Şekil 3.2.10.' da bir adım motoru için gerekli olan sürücü devrenin blok diyagramı gösterilmiştir.



Şekil 3.2.10. Lojik sıralayıcının motora bağlantısı

Motor sürücü devresinde UCN5804 entegresi kullanarak devre daha basitleşti ve programında sadeleşmesi sağlandı.



Şekil 3.2.11. UCN5804 bağlantı şeması

Bazı genel özellikleri şunlardır;

- 1.5 A Maximum Çıkış Akımı
- 50 V Çıkış Voltajı
- Power-ON Reset
- 7 V Giriş Voltajı
- $-20^{\circ}\text{C} - 80^{\circ}\text{C}$  Çalışma Sıcaklığı

Motor kontrol kartında UCN5804'ün çıkış uçları devamlı akım çekiyorlar. UCN5804'ler fazla akım kaldırmadığı için motorları direk bunlardan sürmek yerine çıkışlarına transistörler eklenerek motor sürülmüştür. Transistör de emitterden kolektöre akım geçmesi için base'in 1 olması gerektiği için UCN5804'lerin 0 olan çıkışları 74HC14 ile terslenmiş ve transistörün base'ine giriş olarak uygulanmıştır. Devredeki  $2.2\text{k}\Omega$  olan dirençler 74HC14 girişlerini UCN5804'den 0 gelene kadar 1 yapmak için kullanılmıştır.

### 3.3. Pic Mikrodenetleyicisi

PIC serisi mikroişlemciler MICROCHIP firması tarafından geliştirilmiş ve üretim amacı çok fonksiyonlu lojik uygulamalarının hızlı ve ucuz bir mikroişlemci ve yazılım ile karşılanmasıdır. PIC'in kelime anlamı (Peripheral Interface Controller) giriş çıkış işlemcisidir. İlk olarak 1994 yılında 16 bitlik ve 32 bitlik büyük işlemcilerin giriş ve çıkışlarındaki yükü azaltmak ve denetlemek amacıyla ihtiyaç üzerine geliştirilmiştir.

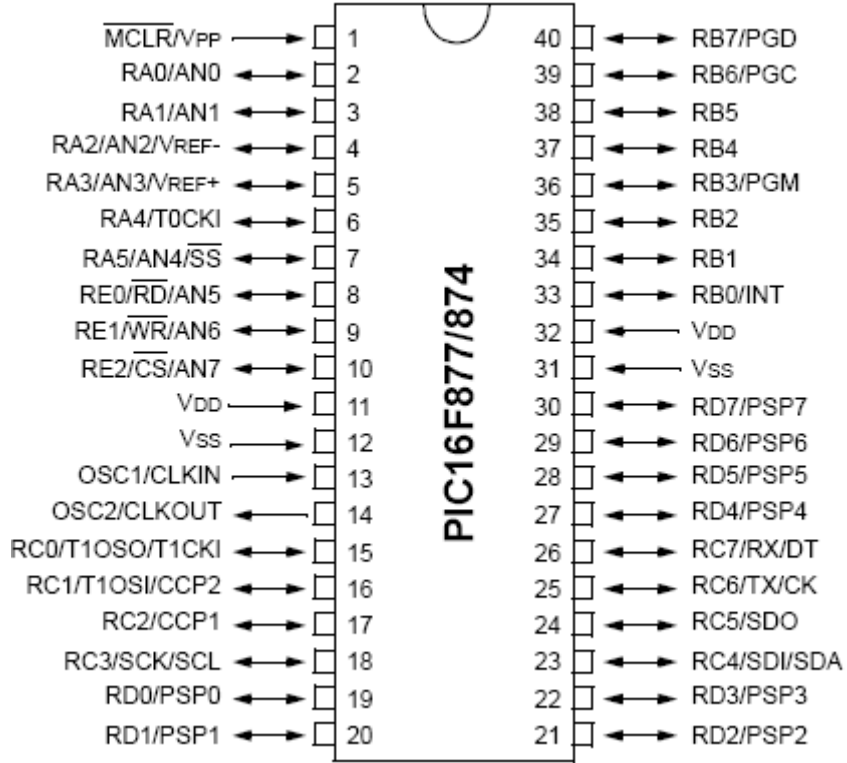
Mikro denetleyiciler, mikroişlemcilerin çalışması için gereken çevre elemanlarını (RAM, EEPROM, Portlar vs.) tek bir yonga içerisinde barındırırlar. Bir mikro denetleyici küçük boyutlu, düşük güç tüketimli, ucuz bir mikroişlemci ile yeterli derecede sinyal işleyebilen ayrı devrelerin birleşiminden oluşur. Bir mikro denetleyici karmaşık lojik fonksiyonları tek bir yongada toplayıp sunan bir cihazdır denilebilir. Genelde başlı başına bir sistem olmayıp, başka bir sistemi yönetmek amacıyla kullanılır.

Mikro denetleyicilerde klasik mikroşlemcili sistemlerden farklı olarak program belleği, RAM, I/O ve CPU bir aradadır.[3]

Bir mikro denetleyici kabaca, bilgisayar içerisinde bulunması gereken temel bileşenlerden RAM, I/O gibi bileşenleri içerisinde barındıran tek bir yonga mahiyetindedir. Böyle olunca bilgisayar kontrolü gerektiren elektronik uygulamalarda mikro denetleyici kullanma eğilimi artmaktadır. Çünkü bir mikroşlemci ile kontrol edilecek bir sistemi kurmak için, CPU, RAM, I/O ve bu birimler arasındaki veri alış-verişini sağlayacak baskılı devreyi kurmak gerekmektedir. Oysa aynı iş için bir mikro denetleyici ve küçük bir devre kartı kullanmak yeterli olmaktadır. Bu da maliyet ve kolaylık anlamında mikro denetleyicinin seçiminde etkili olmaktadır.[2]

Bu tez çalışmasında gerçekleştirilen mobil robotta kullanılan mikro denetleyici Microchip firmasının üretmiş olduğu PIC16F877'dir. Bu mikro denetleyicinin özelliklerini özet olarak aşağıdaki gibi sıralayabiliriz.

- Yüksek Performanslı CPU
- Sadece 35 adet komut kelimesi
- 14 adet dahili/harici kesme kaynağı
- Sekiz seviyeli derin donanım yığını
- İki bacakla Devre İçi Seri Programlama (ICSP)
- Üç adet zamanlama modülü
- İki adet PWM modülü
- 10-bit 8 kanallı analog dijital çevirici
- Seri giriş / çıkış



Şekil 3.3.1. PIC16F877 Bağlantı Pinleri

Mikro denetleyici çip içerisinde 3 adet zamanlayıcı, 8 kanallı analog dijital çevirici, kullanıcıya ayrılmış EEPROM hafıza bölgesi ve bir takım seri haberleşme modülleri kullanıma hazır olarak bulunmaktadır. Bu özellikler ise mikro denetleyici kullanımını cazip hale getirmiştir.

PIC 16F877'de iki ayrı bellek kullanır:

- Data Eeprom Hafıza (Program çalışırken değişken kaydı için kullanılır)
- Flash Program Hafıza ( PIC programlanırken kullanılır )

EEPROM bellek kullanımını bilmemiz için önce yazmaçların ne olduğunu bilmemiz gerekmektedir. Yazmaçlar da bellek birimleridir. Program çalışma esnasında kaydedilir.

PIC 16F877A'nın belleği flash program belleği ve EEPROM veri belleği olmak üzere iki ayrı bloktan oluşur. Esasında bu tür uygulamalarda yani belleğin iç bellek olması denetleyici önemli bir miktarda hızlandırmaktadır. Harici bellek kullanan işlemciler mikro denetleyicilere göre daha yavaştır. EEPROM belleğe doğrudan erişemeyiz. EEPROM belleğe erişebilmemiz için 6 adet yazmacın kullanımını bilmemiz gerekmektedir.

EECON1	EECON2	EEDATA	EEDATH	EEADR	EEADRH
--------	--------	--------	--------	-------	--------

EEDATA yazmacı EEPROM belleğe yazılacak veriyi saklar. EEADR yazmacı EEPROM belleğin adresini tutar. EECON1 ve EECON2 yazmaçları kontrol yazmaçlarıdır. EEDATH yazmacı program belleğe kayıt yapılacağı zaman kullanılır. EEDATH ve EEDATA yazmaç çifti birleşerek 14 bit uzunluğundaki veriyi (komut kodu) saklar. 13 bit uzunluğundaki program bellek adresini ise EEADRH ve EEADR yazmaç çifti saklar.

EEADR ve EEADRH yazmaçları 256 byte uzunluğunda EEPROM bellek ya da 8K'lık program belleği adreslenebilir. Eğer veri yazmacı adreslenecekse (1 byte ile adreslenebilir) sadece EEADR yazmacı LSByte ile adreslenir. Eğer program hafızayı adresleyecekse bu durumda 2 yazmacı da ihtiyacımız olacak. MSByte adres değeri EEADRH yazmacına, LSByte adres değeri de EEADR yazmacına yazılır. Böylece 2 yazmaç ta kayıtlı birer byte'lık veri toplamda bize 8K'lık program hafızayı adreslememizi sağlar.

EECON1 ve EECON2 yazmaçları ise;

R/W-*	U-0	U-0	U-0	R/W-*	R/W-0	R/S-0	R/S-0
EEPGD	-	-	-	WRERR	WREN	WR	RD

EECON1 bellek erişim için kullanılan kontrol yazmacıdır.

EEPGD : Program ya da veri EEPROM bellek seçim biti.

1 : Program belleğe erişim

0 : Data belleğe erişim

WRERR : EEPROM hata biti

1 : Yazma işleminin tamamlanamaması (sıfırlama ya da kesme ile)

0 : Yazma işlemi sorunsuzca tamamlanmıştır

WREN : EEPROM yazma aktif biti

1 : Yazılabilir

0 : Yazılamaz

WR : Yazma kontrol biti

1 : Yazma sinyali gönderir. Yazma işlemini gösterir.

0 : Yazma işlemi bittiğini gösterir. Sadece donanım bu biti sıfırlayabilir.

RD : Okuma kontrol biti

1 : Okuma sinyali gönderir. Okuma işlemini gösterir. Yazılım tarafından sıfırlanamaz.

0 : Okuma işlemi tamamlanınca kendiliğinden sıfırlanır.

WR ve RD bitleri yazılım tarafından sıfırlanamazlar. Bu kontrol bitleri yalnızca donanım tarafından yazma ya da okuma işlemi bittiğinde sıfırlanırlar.

Yazma işlemi tamamladığı zaman PIR2 yazmacında EEIF bayrağı aktif olur. Bayrak yazılım ile sıfırlanmalıdır.

EPPROM bellekten veri okumak için önce EEADR yazmacına okunacak bellek bölgesinin adresi girilmez. Hatırlayacağınız gibi EEPROM belleğe ulaşmak için tek



baytlık bir adres bilgisi yetiyordu. Daha sonra EECON1 yazmacının EEPGD kontrol biti sıfırlanmalıdır. Böylece okuma işleminin EEPROM bellekten olacağını seçmiş oluruz. Bu işlemden sonra gene EECON1 yazmacının RD kontrol biti set edilmelidir. Bu işlemden sonra bellek bölgesindeki veri EEDATA yazmacına aktarılır. EEDATA yazmacından okuma işlemi yapılarak bellek okunmuş olur.

EEPROM belleğe veri yazmak için önce EEADR yazmacına bellek adresi ve EEDATA yazmacına da veri değeri yazdırılmalıdır. Daha sonra yazma sinyali gönderilerek yazma işlemi tamamlanır. Yazma sinyali gönderme kısmı okuma sinyalin den daha karmaşıktır.

Önce EECON1 yazmacının WREN biti set edilmelidir. Sonra INTCON yazmacının GIE biti silinmelidir. Bu işlem yazma işlemi sırasında kesmeleri engelleyecektir. Kullanılmasa da olur ama kullanılması tavsiye edilir. Daha sonra EECON2 yazmacına 55H ve AAH değerleri yazılmalıdır. En son olarak EECON1 yazmacının WR biti set edilerek yazma işlemi tamamlanır.

Yazma işlemi tamamlandıktan sonra (WR biti donanım tarafından sıfırlanır) kesmeler aktif edilmeli (INTCON yazmacının GIE biti set edilmelidir) ve EECON 1 yazmacının WREN biti silinmelidir.

A **bağlantı kapısı (portu)** 6 bacadan oluşur. Çift yönlü haberleşmeyi sağlayabilir. TRISA yazmacı bağlantı kapısı A'nın bacalarının giriş mi yoksa çıkış mı olacağını belirler. Örneğin TRISA 00H değerinde olursa tüm bağlantı kapısı A bacaları çıkış olarak atanır. TRISA yazmacının hangi biti sıfır ise bağlantı kapısı A nın o biti çıkış olur. TRISA yazmacının hangi biti 1 ise bağlantı kapısı A nın o biti giriş olarak atanır. Böylece istediğimiz bacaları giriş, istediklerimizi de çıkış olarak ayarlayabiliriz.

RA5...RA0 bağlantı kapısı A bacaları sırasıyla 7...2 bacalarına karşılık gelir. Bağlantı kapısı A dan yapılan okuma ve yazma işlemleri bu bacaların sayısal durumunu okuma ve değiştirmedir.

Her bacak başka bir görev için de seçilebilir. RA4 bacağı TOCKI için de kullanılır. Yani TIMER0 sayıcısı için harici bir saat işlevi de görebilir. RA3 ve RA2 bacakları analog giriş referans voltajı için kullanılırken diğer bağlantı kapısı A bacakları analog giriş için kullanılır. Eğer herhangi bir bacak analog giriş için kullanılacaksa TRISA yazmaç biti sıfırlanmalıdır. Yani analog giriş için kullanılacak bacak TRISA tarafından giriş olarak atanmalıdır.

<u>İsim</u>	<u>Bit sırası</u>	<u>Buffer</u>	<u>Fonksiyon</u>
RA0/AN0	0	TTL	I/O, Analog giriş
RA1/AN1	1	TTL	I/O, Analog giriş
RA2/AN2/VREF- /CVREF	2	TTL	I/O, Analog giriş, VREF-, CVREF
RA3/AN3/VREF+	3	TTL	I/O, Analog giriş, VREF+
RA4/TOCKI/C1OUT	4	ST	I/O, Timer0 için harici saat girişi, karşılaştırıcı çıkışı, <b>open-drain</b> çıkış
RA5/AN4/SS'/C2OUT	5	TTL	I/O, Analog giriş, senkron seri bağlantı kapısı için <b>bağımlı(slave)</b> seçici, karşılaştırıcı çıkışı

Bağlantı kapısı B çift yönlü giriş/ çıkış özellikli 8 bacedan oluşan bir **bağlantı kapısıdır**. TRISB kullanımı bağlantı kapısı A ile aynıdır. Yani TRISB de 0 yapılan bit PORT B de karşılığı olan bacağı çıkış yapar. TRISB de 1 olan bit bağlantı kapısı B de onun karşılığı olan bacağı giriş yapar. Bu bağlantı kapısının 3 bacağı aynı zamanda PIC programlamada kullanılır: RB3/PGM, RB6/PGC, RB7/PGD.

<u>İsim</u>	<u>Bit #</u>	<u>Buffer</u>	<u>Fonksiyon</u>
RB0/INT	bit 0	TTL/ST	I/O, harici kesme girişı
RB1	bit 1	TTL	I/O,
RB2	bit 2	TTL	I/O,
RB3/PGM	bit 3	TTL	I/O, LVP modunda programlama bacađı
RB4	bit 4	TTL	I/O, deđişim kesmesi
RB5	bit 5	TTL	I/O, deđişim kesmesi
RB6/PGC	bit 6	TTL/ST	I/O, deđişim kesmesi, seri programlama saat sinyali
RB7/PGD	bit 7	TTL/ST	I/O, deđişim kesmesi, seri programlama veri sinyali

Bađlantı kapısı C 8 bit uzunluđunda ve çift yönlü iletişim sađlayabilen **bir kapıdır**. TRISC yazmacı TRISA ve TRISB yazmacı gibi kullanılır. Yani TRISC yazmacı bađlantı kapısı B bacaklarının veri iletişim yönünü belirler. Bađlantı kapısı C de çoklayıcı ile çoklu özelliđe sahip olabiliyor. Her bacak giriş/çıkış işleminden başka görevlerde de kullanılabilir.

<u>İsim</u>	<u>Bit #</u>	<u>Buffer</u>	<u>Fonksiyon</u>
RC0/T1OSO/T1CKI	bit 0	ST	I/O, Zamanlayıcı1 salıngaç çıkışı, Zamanlayıcı1 saat girişi
RC1/T1OSI/CCP2	bit 1	ST	I/O, Zamanlayıcı1 salıngaç girişi, Capture2 girişi/ Karşılaştırıcı2 çıkışı/ PWM2 çıkış
RC2/CCP1	bit 2	ST	I/O, Capture1 girişi/ Karşılaştırıcı1 çıkışı/ PWM1 çıkış
RC3/SCK/SCL	bit 3	ST	I/O, EEPROM yada SPI için senkron seri saat
RC4/SDI/SDA	bit 4	ST	I/O, SPI modda SPI veri ya da EEPROM modda I/O veri
RC5/SDO	bit 5	ST	I/O, senkron seri bağlantı kapısı data çıkışı
RC6/TX/CK	bit 6	ST	I/O, USART asenkron veri aktarımı, senkron saat
RC7/RX/DT	bit 7	ST	I/O, USART asenkron alıcı, senkron data

Bağlantı kapısı D 8 bit genişliğinde, çift yönlü veri akışı sağlayabilen paralel bağımlı (**slave**) **bağlantı kapısı** olarak ayarlanabilen **bir kapıdır**. TRISE 4. biti PSPMODE biti içindir. Bu kontrol biti bağlantı kapısı D'nin paralel **bağımlı bağlantı kapısı** olup olmayacağını ayarlar.

<u>İsim</u>	<u>Bit #</u>	<u>Buffer</u>	<u>Fonksiyon</u>
RD0/PSP0	bit 0	ST/TTL	I/O, Paralel bağımlı kapı
RD1/PSP1	bit 1	ST/TTL	I/O, Paralel bağımlı kapı
RD2/PSP2	bit 2	ST/TTL	I/O, Paralel bağımlı kapı
RD3/PSP3	bit 3	ST/TTL	I/O, Paralel bağımlı kapı
RD4/PSP4	bit 4	ST/TTL	I/O, Paralel bağımlı kapı
RD5/PSP5	bit 5	ST/TTL	I/O, Paralel bağımlı kapı
RD6/PSP6	bit 6	ST/TTL	I/O, Paralel bağımlı kapı
RD7/PSP7	bit 7	ST/TTL	I/O, Paralel bağımlı kapı

ST : Schmitt Trigger, TTL : TTL giriş

ST Giriş/Çıkış modda iken, TTL paralel **bağımlı kapı** modunda iken geçerlidir.

PORT E 3 bit genişliğinde, her bacağına farklı veri akış yönü atanabilen, Schmitt Trigger giriş tamponlarına sahip bir porttur.

<u>İsim</u>	<u>Bit #</u>	<u>Buffer</u>	<u>Fonksiyon</u>
RE0/RD'/AN5	bit 0	ST/TTL	I/O, Paralel bağımlı kapı modda iken okuma kontrol biti (RD'), Analog giriş. RD' 0 değerinde okuma işlemi gerçekleştirir. Bağlantı kapısı D yazmaç değeri harici aygıta aktarılır.
RE1/WR'/AN6	bit 1	ST/TTL	I/O, Paralel bağımlı kapı modda yazma kontrol biti, Analog giriş. WR' 0 değerinde yazma işlemi gerçekleştirir. Bağlantı kapısı D girişindeki bilgi bağlantı kapısı D yazmacına aktarılır.
RE2/CS'/AN7	bit 2	ST/TTL	I/O, Paralel bağımlı kapı modda aygıt seçme kontrol biti, Analog giriş. CS' 1 durumunda aygıt seçilmemiş, 0 durumunda aygıt seçilmiş olur.

Zaman geciktirmeleri için genelde kullandığımız yöntem yazılımda döngüler yaratmaktır. Donanım sayıcı ve zamanlayıcıları bu işi yazılıma gerek kalmadan donanımsal olarak yapar. Her sayıcı ve zamanlayıcı modülünün kendisine ait bir yazmacı vardır. Sayma işlemleri bu yazmaçlarla yapılır. Genelde saymaya 00H değerinden başlayarak birer birer artar. PIC 16F877A 4 adet zamanlayıcı içerir. Bunlar Timer0, Timer1, Timer2 ve WDT.

Timer0 flash program belleğin 01H adresinde kayıtlı özel bir yazmaçtır.

- 8 bittir. Yani 0-255 arası sayma yapar
- Yazılabilir ve okunabilir
- Bit programlanabilen ön ölçekleme (frekans bölme sayısı) değeri
- Sayı artışı için iç ya da dış saat kaynağı seçebilme
- FFH tan 00H değerine geçişte kesme oluşturma
- Harici saat kaynağı için tetikleme kenarı seçebilme

Esasen sayıcı ve zamanlayıcı aynı şeylerdir. Sadece sinyal kaynakları farklıdır. Timer0 saat kaynağı dâhili komut çevrimi ise zamanlayıcı harici saat kaynağı ise sayıcı deriz.

Zamanlayıcı modu OPTION\_REG yazmacının TOCS kontrol biti ile seçilir. TOCS'un set olması zamanlayıcı, reset olması sayıcı modunu seçer. Sayıcı modda harici saat sinyalinin tetikleme kenarı TOSE kontrol biti ile seçilir. Ön ölçekleme değeri PSA kontrol biti ile ya Timer0 ya da WDT için atanır. Timer0 yazmacı yazılabilir olduğu için sayma işlemine herhangi bir sayıdan başlanılabilir. Böylesi bir sistemin bize avantajı yazılımdan bağımsız bir sayma ve zamanlama gerçekleştiği için adım motor gibi aygıtların rahatça hız kontrollerini yapabilmemizdir. PIC 16F877 3 adet zamanlayıcı içerdiği için 3 farklı adım motoru 3 farklı hızda sürebiliriz.

Timer1 modülü iki adet 8 bitlik yazmaç kullanarak 16 bitlik sayma yapabilen bir sayıcıdır. TMR1 yazmacı TMR1H:TMR1L yazmaç çiftinden oluşur. Her iki yazmaçta okunabilir ve yazılabilir. 0000h değerinden FFFFh değerine sayma yapar ve tekrar 0000h değerine dönerek kesme oluşturur. Bu kesme bayrağı PIR1 yazmacının TMR1IF (bit 0) bitidir. Bu kesme PIE1 yazmacının TMR1IE (bit 0) kontrol biti ile aktif ya da pasif edilir. Timer1 sayıcının bir de kendi T1CON yazmacı vardır.

Timer2 8 bitlik yazmacıya sahip, 2 adet frekans bölücüye sahip (ön ölçekleyici, arka ölçekleyici) bir sayıcıdır. Timer2 diğer sayıcılardan farklı olarak sayma işlemini FFh'ta bitirmek zorunda değildir. PR2 yazıcı periyot değerini saklar ve sayma işlemi bu değerden sonra 00h değerine döner. Diğer fark ise harici saat sinyal girişi olmamasıdır.

PIC 16F877 8 adet analog giriře sahiptir. Analog çevirme 10 bitlik hassasiyete sahiptir. Bu da hassas denebilecek bir deęer.  $2^{10}$  adet farklı deęerde çözümlene yapabildiđi demek. Bunların yanında A/D modülü referans voltaj deęerleri için alçak ve yüksek deęerlere sahiptir. Bu referans deęerleri  $V_{DD}$ ,  $V_{SS}$  ya da RA2, RA3 deęerleridir. A/D modül 4 adet yazmacı sahiptir. Bunlar:

- A/D sonucu üst 8 bit yazmacı (ADRESH)
- A/D sonucu alt 8 bit yazmacı (ADRESL)
- A/D kontrol yazmacı 0 (ADCON0)
- A/D kontrol yazmacı 1 (ADCON1)

#### 4. BASKI DEVRE YAPIMI

Gerekli olan 3 kartta Protel DXP programı ile tasarlanmış ve “bastır ve ayır” yöntemi ile baskı devreler hazırlanmıştır. Baskı devreleri hazırlama yöntemine kısaca değinmek gerekirse;

İlk olarak elektronik ortamda çizilen devre bastır ve ayır transfer filmine çıktı alınır ya da bu filme fotokopisi çekilir.



Şekil 4.1. Bastır ve ayır kağıdı

Kâğıt, devre boyutlarında kesilir ve birebir boyuta getirilir. Ardından kullanacağımız bakır plakayı da gerekli boyutlarda keseriz. Bakır plakayı çelik telle iyice temizledikten sonra sabunla yıkayabiliriz.



Şekil 4.2. Bakır plakanın temizlenmesi

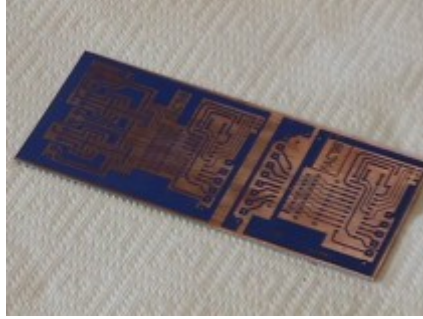
Ardından devreyi bakır plakanın üzerine koyarak ütümeye başlarız. Ütünün sıcaklık ayarı bu iş için oldukça önemlidir. Başlangıç sıcaklığı olarak 45<sup>0</sup>C ile 90<sup>0</sup>C arası bir sıcaklık önerilmektedir. Ütü işlemi plakanın kalınlığına göre 1.5 dk. ile 10 dk. arasında değişebilir.





Şekil 4.3. Devrenin Ütü ile yapıştırılması

Ütü işleminden sonra soğuk su ile plaka yıkanır ve plaka üzerindeki film ayrılır. Plaka daha sonra aside batırılmadan önce tekrar yıkanır.



Şekil 4.4. Asite atılacak plaka

Plaka ardından ferro-klorür asite atılır ve bir süre bekletilir. Ardından çıkarılan plaka kuruduktan sonra çelik telle gereksiz yollar temizlenir ve en son istediğimiz devreye sahip plakayı elde edebiliriz.

## 5. SONUÇ VE ÖNERİLER

Bu çalışmanın amacı, yol üzerindeki engellere çarpmadan ilerleyebilen bir mobil robot tasarlamak ve gerçekleştirmektir.

Mobil robotumuzda 2 adet adım motor bulunmaktadır. Öndeki motor engellerden kurtulmak amacıyla robota sağ ve sol hareketi sağlamaktadır, arkadaki robot ise robotun ileriye hareketini sağlamaktadır. Mobil robotun engelleri algılaması kullanılan 4 adet ultrasonik mesafe algılayıcısı yardımıyla sağlanmıştır. Robotun bütün fonksiyonları ise ana kartımızda kullanılan PIC 16F877 ile kontrol edilmektedir. Ayrıca güç kaynağı olarak 12V 1.3 Ah'luk şarj edilebilir akü kullanıldı.

Ultrasonik mesafe algılayıcılarının özelliklerinden dolayı araba platformundan aşağıdaki engelleri ya da olmayan yolları algılamamaktadır, bunun için ileride robot üzerine bir kamera sistemi monte edilebilir.

Robotun mekanik yapısı gayet iyidir bunu engellere karşı verdiği tepkilerden anlayabilmekteyiz, yalnız üzerine daha çok eleman konulacaksa platform sağ ve sol yanlardan eğilme olmaması için desteklenmelidir.

## 6. KAYNAKLAR DİZİNİ

- [1] Fu, K. S., Gonzalez, R. C., Lee C. S. G., 1987. *ROBOTICS : Control, Sensing, Vision, and Intelligence*. McGraw-Hill Book Company.
- [2] Altınbaşak O., 2000, ”*Mikrodenetleyiciler ve Pic Programlama*”, Altaş Yayınları, İstanbul
- [3] Yalçın, M.K., 2003, Mobil Robot Uygulaması, T.C. Niğde Üniversitesi Fen Bilimleri Enstitüsü, 102s.
- [4] <http://www.antrak.org.tr/gazete/052001/adnan.htm>
- [5] <http://ww1.microchip.com/downloads/en/DeviceDoc/30292c.pdf>
- [6] [http://robot.metu.edu.tr/dosya/robot\\_nedir.pdf](http://robot.metu.edu.tr/dosya/robot_nedir.pdf)
- [7] <http://robot.cmpe.boun.edu.tr>
- [8] [http://www.obitet.gazi.edu.tr/obitet/elektrik\\_elektronik/adim\\_motorlari.htm](http://www.obitet.gazi.edu.tr/obitet/elektrik_elektronik/adim_motorlari.htm)
- [9] [http://webelektronik.com/tr/index.php?option=com\\_content&task=view&id=14&Itemid=41](http://webelektronik.com/tr/index.php?option=com_content&task=view&id=14&Itemid=41)

## EK-1 MOBİL ROBOTUN YAZILIMI

```
#include <pic.h>
```

```
#include <delay.c>
```

```
__CONFIG(0x3f32);
```

```
//SRF04'LER için Echo girişleri
```

```
#define echo1 RB0 //pin 33
```

```
#define echo2 RB2 //pin 35
```

```
#define echo3 RB4 //pin 37
```

```
#define echo4 RB6 //pin 39
```

```
#define echo5 RA0 //pin 2
```

```
//SRF04'ler için Trigger çıkışları
```

```
#define trig1 RB1 //pin 34
```

```
#define trig2 RB3 //pin 36
```

```
#define trig3 RB5 //pin 38
```

```
#define trig4 RB7 //pin 40
```

```
#define trig5 RA1 //pin 3
```

```
//Sürücü adım motoru için adım, yön, aç/kapa
```

```
#define drive_step RD0 //pin 19
```

```
#define drive_dir RD1 //pin 20
```

```
#define drive_en RD2 //pin 21
```

```
//Yön adım motoru için adım, yön, aç/kapa
```

```
#define steer_step RD3 //pin 22
```

```
#define steer_dir RD4 //pin 27
```

```
#define steer_en RD5 //pin 28
```

```
#define on 1
```

```
#define off 0
```

```
void setup(void);
```

```
//SRF04 data toplama fonksiyonlari
```

```
unsigned int get_srf04_1(void); //Sol ön SRF04
```

```
unsigned int get_srf04_2(void); //Orta ön SRF04
```

```
unsigned int get_srf04_3(void); //Sag ön SRF04
```

```
unsigned int get_srf04_4(void); //Yedek SRF04
```

```
unsigned int get_srf04_5(void); //Arka SRF04
```

```
void main(void)
```

```
{  
  
unsigned int range1, range2, range3, range5; //range4 -->(ters);  
  
const unsigned int limit = 10; //10cm mesafe siniri  
  
    DelayS(5);  
  
    setup();  
  
while(1)  
    {  
  
        range1 = get_srf04_1();  
  
        TMR1H = 0; //52ms bekle --  $256*256*0.2*4 = 52.429ms$   
  
        TMR1L = 0;  
  
        T1CON = 0x21;  
  
        TMR1IF = 0;  
  
        while(!TMR1IF);  
  
        TMR1ON = 0;  
  
        if((range1/72)<limit)  
            {  
  
                steer_dir = off; //sag - tam dönüş için 2 tane 250ms  
  
                steer_en = off; //aç  
  
                DelayMs(250);  
  
                DelayMs(250);  
  
                steer_en = on; //kapat
```

```
DelayS(1); //biraz bekle
```

```
steer_dir = on; //sol - ortaya geri dön
```

```
steer_en = off; //on
```

```
DelayMs(250);
```

```
DelayMs(250);
```

```
steer_en = on; //off
```

```
}
```

```
else
```

```
{
```

```
steer_en = on; //off
```

```
}
```

```
////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////
```

```
range3 = get_srf04_3();
```

```
TMR1H = 0;
```

```
TMR1L = 0;
```

```
T1CON = 0x21;
```

```
TMR1IF = 0;
```

```
while(!TMR1IF);
```

```
TMR1ON = 0;
```

```
if((range3/72)<limit)
```

```
{
```

```
steer_dir = on; //sol
```

```

steer_en = off; //açik
DelayMs(250);
DelayMs(250);
steer_en = on; //kapali
DelayS(1);
steer_dir = off; //sag - ortaya geri dön
steer_en = off; //açik
DelayMs(250);
DelayMs(250);
steer_en = on; //kapali
}
else
{
steer_en = on; //kapali
}

```

////////////////////////////////////

```

range2 = get_srf04_2();
range5 = get_srf04_5();

```

```

TMR1H = 0;

```

```

TMR1L = 0;

```

```

T1CON = 0x21;

```



```

TMR1IF = 0;

while(!TMR1IF);

TMR1ON = 0;

if(((range2/72)<limit) && ((range5/72)>limit))
    {
        steer_dir = on; //sol

        drive_dir = off; //geri

        steer_en = off; //açik

        DelayMs(250);

        DelayMs(250);

        steer_en = on; //kapali

        DelayS(1);

        steer_dir = off; //sag - ortaya geri dön

        drive_dir = on; //ileri

        steer_en = off; //açik

        DelayMs(250);

        DelayMs(250);

        steer_en = on; //kapali

    }

else if(((range2/72)<limit) && ((range5/72)<limit))

    {

        steer_dir = off; //sag

        drive_dir = off; //geri

        steer_en = off; //açik

```

```
    DelayMs(250);  
    DelayMs(250);  
    steer_en = on; //kapali  
    DelayS(3); //3 saniye bekle  
    steer_dir = on; //sol - ortaya geri dön  
    drive_dir = on; //ileri  
    steer_en = off; //açık  
    DelayMs(250);  
    DelayMs(250);  
    steer_en = on; //kapali  
    }  
else  
    {  
    steer_en = on; //kapali  
    }  
}  
}
```

```
/**
*****
*****/
```

```
unsigned int get_srf04_1()
```

```
{
```

```
    TMR1H = 0xff;    // zamanlayiciyi 10us darbe için hazırla -
(0.05*0.2*4*256=10.24us)
```

```
    TMR1L = -14;    // 1/20 = 0.05
```

```
    T1CON = 0x21;    // 1:4 prescale ve çalışıyor
```

```
    TMR1IF = 0;
```

```
    trig1 = 1;    // Trigger darbesini baslat
```

```
    while(!TMR1IF); // 10uS bekle
```

```
    trig1 = 0;    // Trigger darbesini durdur
```

```
    TMR1ON = 0;    // zamanlayiciyi durdur
```

```
    TMR1H = 0;    // Echo darbesini ölçmek için zamanlayiciyi hazırla
```

```
    TMR1L = 0;
```

```
    T1CON = 0x20;    // 1:4 prescale ama henüz çalışmıyor
```

```
    TMR1IF = 0;
```

```
    while(!echo1 && !TMR1IF); // Echo darbesinin başlamasını bekle (high
olmasını)
```

```

TMR1ON = 1;           // darbeyi ölçmek için zamanlayiciyi çalistir

while(echo1 && !TMR1IF); // Echo darbesinin bitmesini bekle (low olmasını)

TMR1ON = 0;           // zamanlayiciyi durdur

return (TMR1H<<8)+TMR1L; // TMR1H:TMR1L darbenin süresini 0.8uS
birimler halinde içerir
}

```

```

unsigned int get_srf04_2(void)

```

```

{
    TMR1H = 0xff;
    TMR1L = -14;
    T1CON = 0x21;
    TMR1IF = 0;
    trig2 = 1;
    while(!TMR1IF);
    trig2 = 0;
    TMR1ON = 0;
    TMR1H = 0;
    TMR1L = 0;
    T1CON = 0x20;
    TMR1IF = 0;
    while(!echo2 && !TMR1IF);
    TMR1ON = 1;
    while(echo2 && !TMR1IF);
}

```

```
TMR1ON = 0;

return (TMR1H<<8)+TMR1L;

}
```

```
unsigned int get_srf04_3(void)
```

```
{

    TMR1H = 0xff;

    TMR1L = -14;

    T1CON = 0x21;

    TMR1IF = 0;

    trig3 = 1;

    while(!TMR1IF);

    trig3 = 0;

    TMR1ON = 0;

    TMR1H = 0;

    TMR1L = 0;

    T1CON = 0x20;

    TMR1IF = 0;

    while(!echo3 && !TMR1IF);

    TMR1ON = 1;

    while(echo3 && !TMR1IF);

    TMR1ON = 0;

    return (TMR1H<<8)+TMR1L;

}
```

```
unsigned int get_srf04_4(void)
{
    TMR1H = 0xff;
    TMR1L = -14;
    T1CON = 0x21;
    TMR1IF = 0;
    trig4 = 1;
    while(!TMR1IF);
    trig4 = 0;
    TMR1ON = 0;
    TMR1H = 0;
    TMR1L = 0;
    T1CON = 0x20;
    TMR1IF = 0;
    while(!echo4 && !TMR1IF);
    TMR1ON = 1;
    while(echo4 && !TMR1IF);
    TMR1ON = 0;
    return (TMR1H<<8)+TMR1L;
}
```

```
unsigned int get_srf04_5(void)
{
    TMR1H = 0xff;
    TMR1L = -14;
```

```
T1CON = 0x21;

TMR1IF = 0;

trig5 = 1;

while(!TMR1IF);

trig5 = 0;

TMR1ON = 0;

TMR1H = 0;

TMR1L = 0;

T1CON = 0x20;

TMR1IF = 0;

while(!echo5 && !TMR1IF);

TMR1ON = 1;

while(echo5 && !TMR1IF);

TMR1ON = 0;

return (TMR1H<<8)+TMR1L;

}
```

```
void setup(void)
```

```
{
```

```
ADCON1 = 0x07;
```

```
ADCON0 = 0x00;
```

```
CCP1CON = 0x00;
```

```
CMCON = 0x07;
```

```
TRISA = 0x01;
```

```
PORTA = 0x01;
```

```
TRISB = 0x55;
```

```
PORTB = 0x55;
```

```
TRISD = 0x00;
```

```
PORTD = 0x00;
```

```
T0CS=0;
```

```
PSA=0; //TMR0 seç
```

```
PS0=1; //prescaler -> 256
```

```
PS1=1;
```

```
PS2=1;
```

```
TMR0=0xD0; //0xD0 ->208'den 255'e say -> 48
```

```
T0IE=1; //zamanlayiciyi aç
```

```
T0IF=0; //tasma yok
```

```
ei(); //global kesmeleri aç
```

```
drive_dir = on;
```

```
drive_en = off; //açık
```

```
steer_dir = on;
```

```
steer_en = on; //kapalı
```

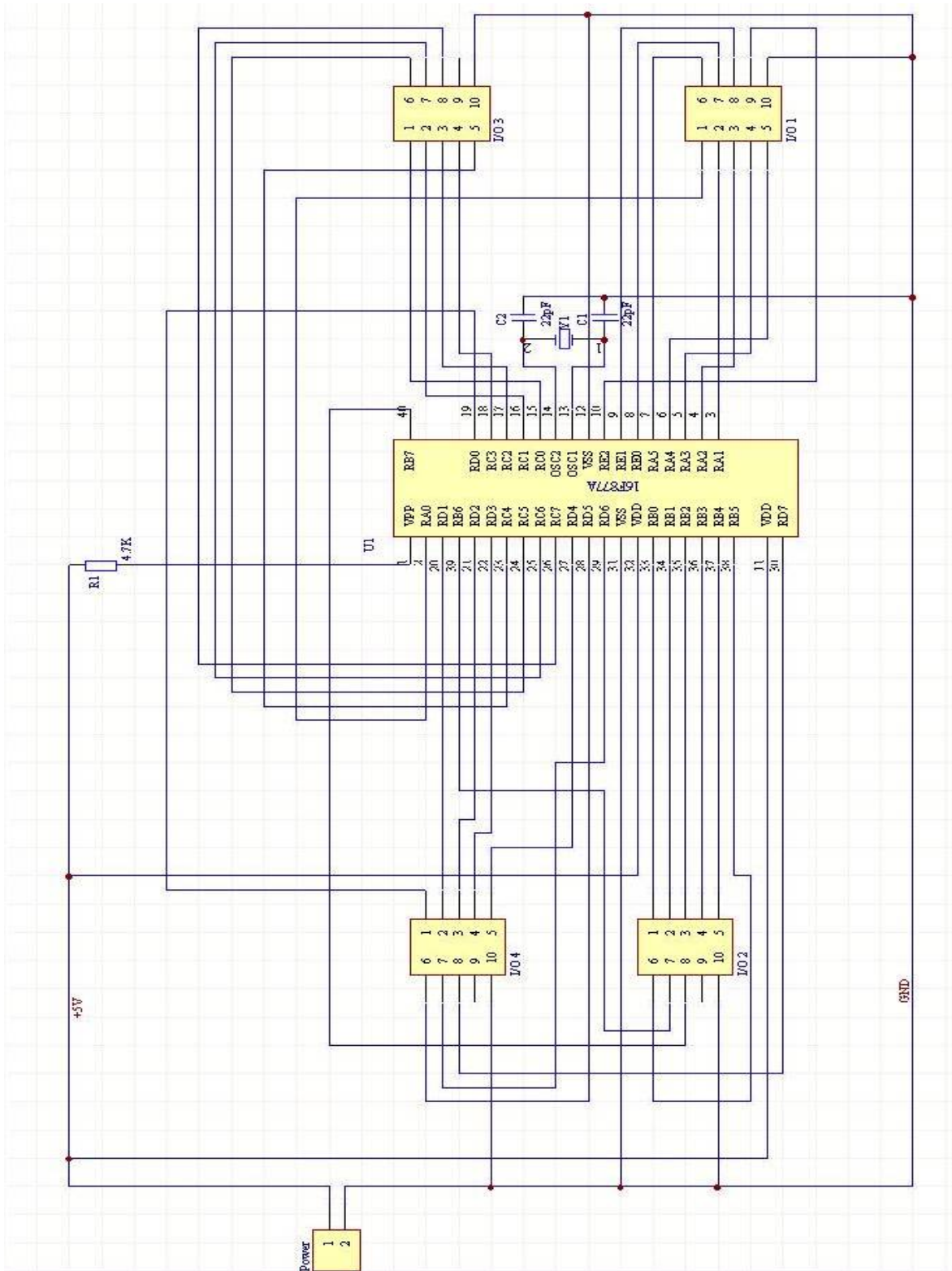
```
}
```



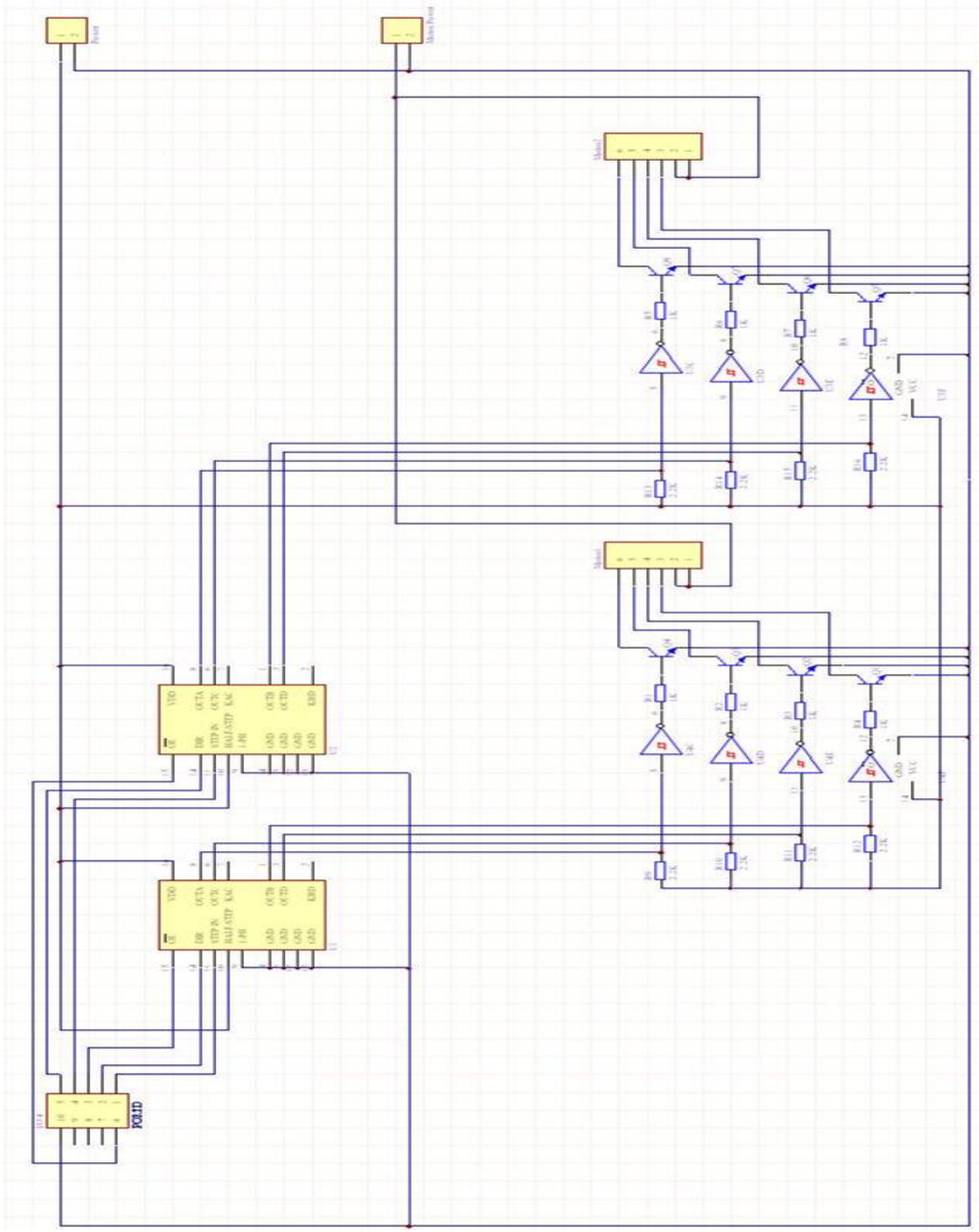
//0.2 x 48 x 256 = 2.46ms - adim motorlar için açık/kapali

```
void interrupt stepper()
{
drive_step = drive_step ^ on;
steer_step = steer_step ^ on;
TMR0=0xD0;
T0IF=0;
}
```

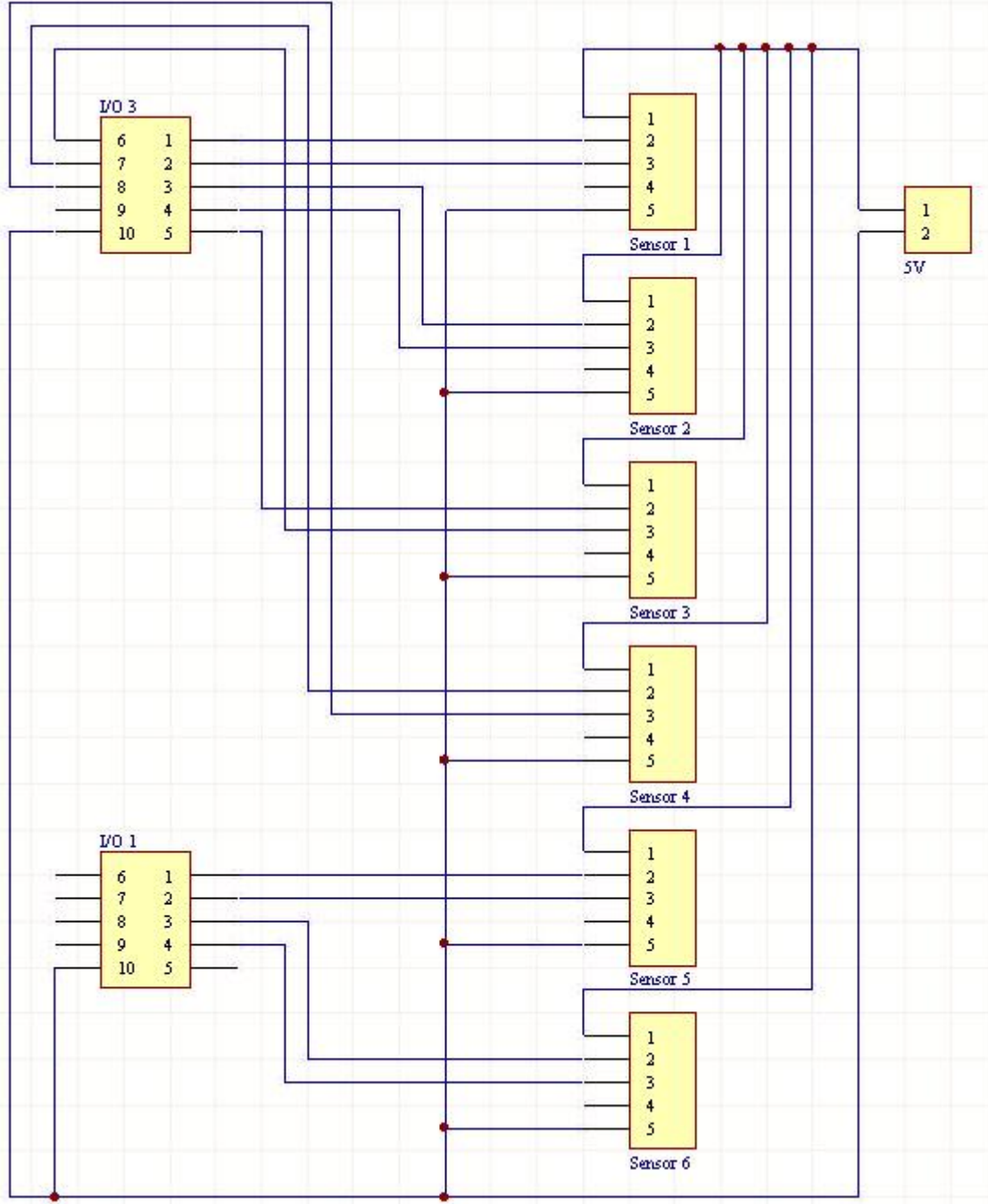
## EK-2 TASARLANAN DEVRELER



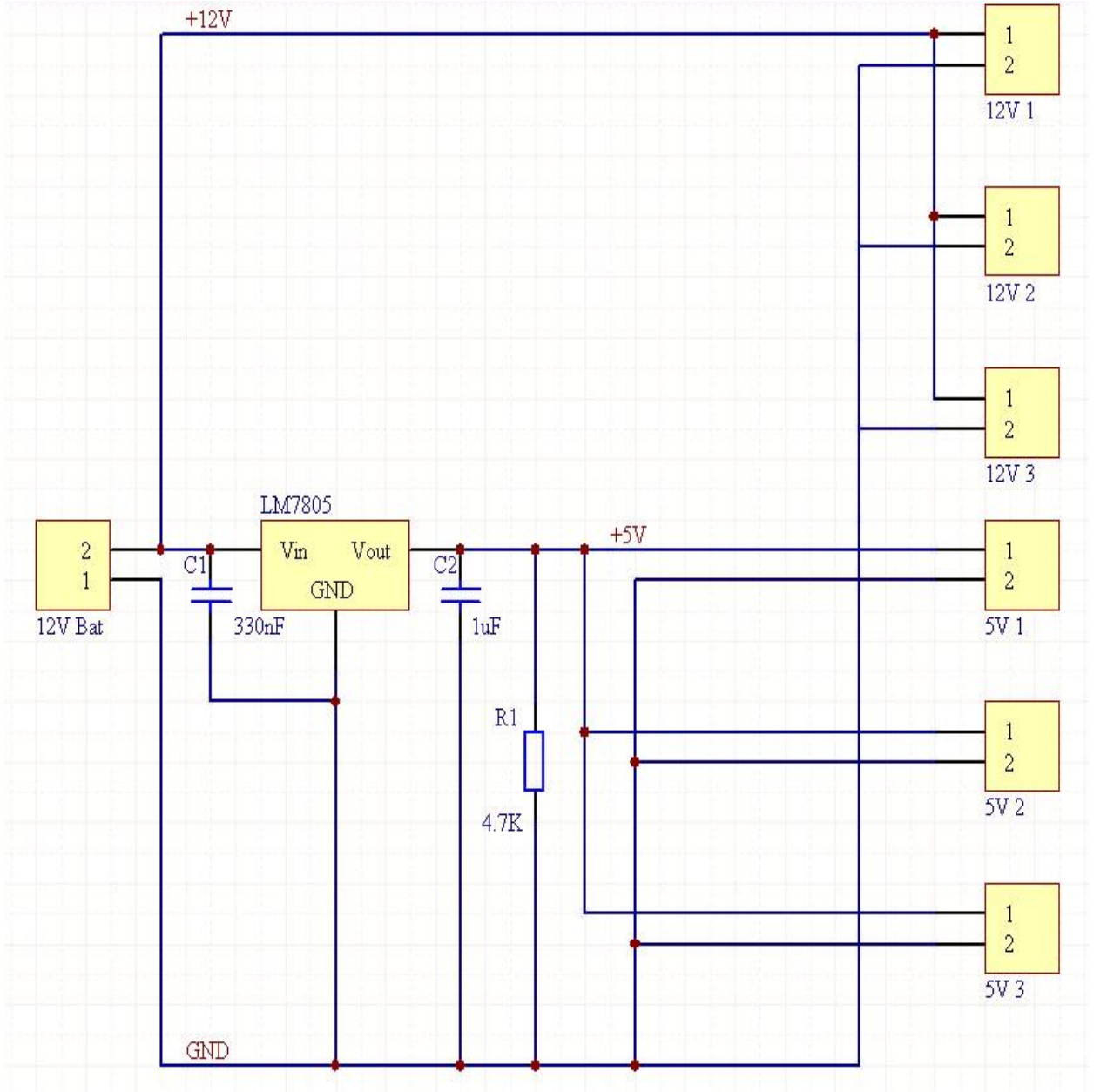
Şekil.EK-2 Anakart Devresi



Şekil EK-2 Motor Sürücü Devresi



Şekil EK-2 Sensör Devresi



Şekil EK-2 Güç Devresi

## EK-3 MOBİL ROBOTA AİT FOTOĞRAFLAR

