Robot Kol Tasarımı

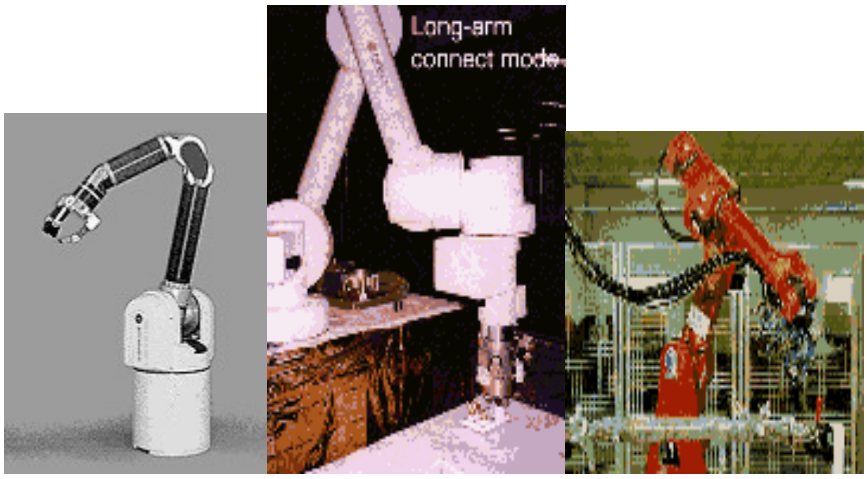
[1.1] Proje Kavrami

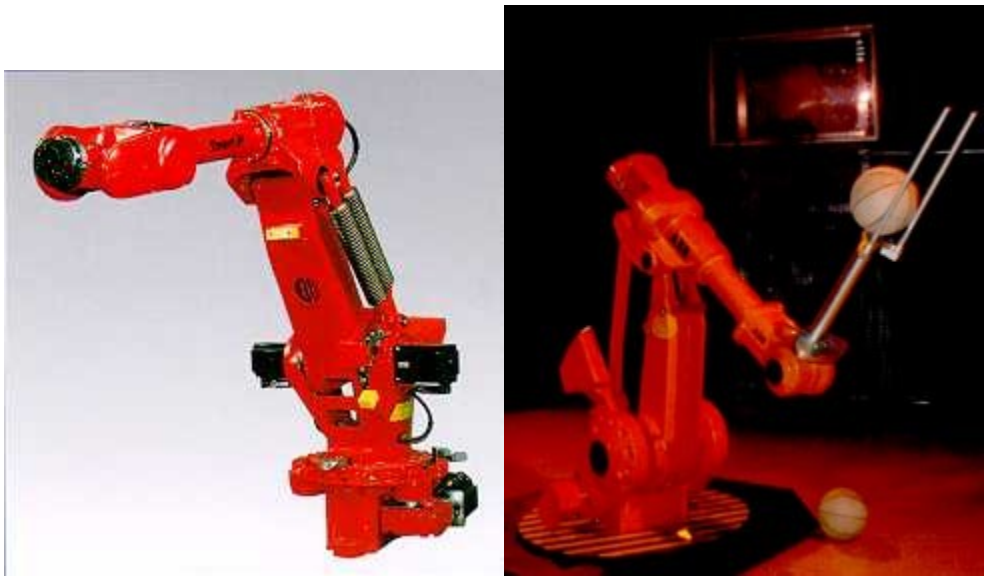
Proje bir amaç için gelistirme, arastirma için yapilan ve yapilacak olan kavramlar olarak yer alsada üiversite ve dengi okullar için bu kavram yillarca alinan egitimin uygulamaya dökülerek amaca uygun çalismalardaki son sinav olmaktadir. Iste bu yüzden bir ögrenci için proje kapsami bence hayatinin diger yarisi için çok uygun temel ve çok güzel bir baslangiçtir. Eger ögrenci projesi ile yenilik ve avantaj getirebiliyorsa yada sistemleri yürütebiliyorsa egitimini tamamlamis ve üretim moduna geçmeye baslamis demektir. Ama bu asla var olan ve daha iyisi için yapilmayan projeler için çok büyük bir gerçegi ortaya koymaktadir.

Sonuç olarak mutlak olarak alinan proje mutlaka bir yenilik yada özellik içermeli ve yapitaslari içermelidir. Aksi halde yapilan bir is arastirma mantiginin disina çikip kopyalama durumuna girer!. Iste bu yüzden gelismekte olan ve ihtiyaç duyulan bilgisayar destekli Robot Kol (eng: Robot ARM) tasarimini proje olarak seçtim.

[1.2] Projemin Hedef ve Kapsamlari

Projede hedef robot kol mantigini daha esneklestirip bilgisayar destekli insan kol yapisinin yerini alabilecek bir yapiyi olusturmaktir. Kol piyasada üretilen assagida görülen modellerden farkli olarak daha ekonomik ve genis acili kullanima hitap etmesi için yüksek hassaiyetli bir sistemi ayni zamanda düsük ama hizli hassasiyeti düsük olarkta çalisabilecek bir yapidada çalisabilmesi vede asil olarak bilgisayar yazilim yapisinin open-source mantigiyla kullanim alanlarinin artirilip sistem için sistem degilde herhangi bir olay için tasarim mantiginin getirilmesidir. Bu yüzden mekanik açidan sorunsuz olabilmesi için eklem serbestligi getirmeye çalisacagim.





Robot Kol projesi benim için temel olarak assagidaki yapilari içermektedir: (Sistem sirasiyla)

1. Mekanik bakis

Temel madde seçimleri - Eklem yapi ve yerleri - Baglanti ve türleri

2. Hareket sistemleri

Motor yapilari - Disli kutusu - Encoder (Sayici) - Motor Kontrölcüsü - Mikroislemci

3. Temel yazilim ve bilgisayara baglanti.

Mikroislemci yazilimi - Port baglantilari ve güvenli iletisim

4. Yazilim ve Uygulama teorileri

[1.3] Projemden beklenen sonuçlar.

Projemden beklenen sonuçlarin basinda elbetteki tam sonuç isteniyor ama asil olarak bu tür uygulama gelistirme yapilarininda okul programlarinda gerçeklestirilebileçegi kanisinin okulumuza yerlesmesi ve bizim gibi gelecege bakan ögrencilerin tesvik edilmesi. Unutulmamasi gereken basari sadece bir kisiye veya kuruma ait degildir basari geçmise katkisi olan ve su anda destek olan her sahsa aittir. Basari basarilari tesvikler ise tesvikleri getirir. Ama asil önemli olan uygulamadaki basarinin mutlak olmasi, projenin tamamlanarak bir sonraki arkadasin gelistirilmesine uygunlastirilmasidir...

[2.1] Genel Olarak Bir Robot Kavrami:

Ilk olarak genel robot tanimlarini incelersek;

" Robot, mekanik sistemleri ve bunlarla iliskili kontrol ve algilama sistemleriyle bilgisayar algoritmalarina bagli olarak akilli davranan makinelerdir. Genel bir enstitü tarafindan ise robot su sekilde tanimlanmaktadir: Robot Yeniden programlanabilen; maddeleri, parçalari, aletleri, programlanmis hareketlerle yapilacak ise göre tasiyan veya isleyen çok fonksiyonlu makinelerdir". (Robot Institute of America, 1979)

" Robot nedir? Robot, bir kaide üzerinde en az bir kol, tutma organlari(genelikle pensler, vantuzlar veya elektromiklatislar), pnömatik, hidrolik veya elektriksel sensorler ile konumu ve basinç algilayicilariyla, bilgi islem organlariyla donatilmis kontrollü-mekanik maniplelerdir."

Yukaridaki tanimlarda görüldügü gibi robot temel bir islemi yerine getirebilen, yetileri olan yeniden programlanbilen aygitlardir.Temel olarak bir robotun assagidaki özellikleriini olmasi gerekir:

1. Islem yapma yetisi : Bir islemi fiziksel veya farazi olarak yerine getirebilmelidir yoksa robot olmaz sadece bir madde olur.
2. Islemin sonucunu belirleme yetisi : Islemi yaptiktan sonra mutlak olarak islemin sonucunu belirlemelidir ki islem tam olarak yapilmis olsun.
3. Karar verme yetisi. : Islemin sonucuna göre yada dis etmenlere göre mutlaka bir yargi kurabilmelidir.

Iste bu yapilari içeren her sisteme genel olarak robot diyebiliriz. Örnegin sinav robotu, isik izleyen robot, vana açip kapayan robot veya

robot kol gibi. Fakat asil robot kavrami bu yapilarin çok daha ilerisine giderek dogada en karmasik yapi olan insan oglunun yetilerini taklit etmek için yapilan makinalardir. Robot kavramida onlar üzerine kurulmus olmasina ragmen tanim genel olarak farazi yapilarida içermektedir.

Çesitli ileri gelen dünya ülkeleri tarafindan robot çesitli sekilde tanimlanmis olmakla beraber hepsinin bulustugu nokta Robotlar yeniden programlanabilen mekanik aksamlardir. Bu kurumlarin ayrildiklari nokta algilama ve sonuca varma yetisidir. Japon konseyi bu konuda algilama ve karar verme mekanizmasi olmayan robot kavramini kabul ederken Ingiliz ve Amerikan Enstitüsü buna karsi çikmaktadirlar.....

[2.2] Robotlarin Tarihçesi:

[2.2.1] Robot Kavraminin Dogusu:

Medeniyet, kölelik düzeni sayesinde bugunku olaganüstü seviyeye ulasmistir. Bir köle tükettiginden fazlasini üretmek zorundadir. Amerikada seker kamisi toplayan zenci köleler sahiplerini milyarlarca dolar kazandirirken kendileri açliktan sagliksiz ortamlardan ölmüslerdir. Bu çirkin olaylar amerika iç savasi ve ardindan dünya çapinda kölelik düzeninin sona ermesiyle çalisan ihtiyaci duyan isverenler eskisi gibi ucuza isçi bulamamakta bulduklariylada istedigi kadar verimi alamamaktaydi çünkü artik zorlama yoktu çalisma haklari ve sendikalari vardi. Iste bu kötü durumdan insanlari isten çikarip yerlerine birkaç insanin isini yaptirabilecekleri mekanizmalar aramaya baslandi.Ve o günden günümüze dek teknolojiyle baglantili olarak robot kavrami degisti ve gelisti. Ama asil asamayi bilgisayar kavramiya birlikte gelen karar verme yetisi (algilama-karar verme-uygulama) sayesinde saglandi. Ve günümüzde 1Ghz islem gücündeki bilgisayarlar ile robotlarin gücü ve kapasite siniri insanin ötesine geçti. Artik robotlar bir saç kilini iki esit parçaya ayirabiliyor ve bunu saniyenin onda biri kadar bir sürede yapabiliyorlar.

Genel açidan robot kavraminin tarihdeki gelisimi:

Ortaçagda Selçuklu Türklerinden Sükman boyundan Cizreli Ebul-Iz, yalniz suyun potansiyel ve kinetik enerjilerinden faydalanarak makineler ve robotlar yapti. Ilkel otomatlar 17. ve 18. yüzyillarda Avrupa'da bulundu. Bunlar birer mekanik harikasiydilar. O zamanlarda mucitler ilginç makineler icat edip üretime geçtiler. Kilise ve katedrallerin tepesinde bulunan devasa saatlerde gerçek boyutlarda insan, melek,

seytan gibi figürler vardi. Bunlar ellerindeki tokmagi çana dogru giderek vuruyorlardi, vurus sayisi saatide belirliyordu.

Ötücü Kus: Kurulu düzenek tarafindan miller ve kaldiraçlar yardimiyla kusun kanatlari, kafasi ve gagasi kontrol edilebiliyordu. Ayrica vana ve pistonlar sayesinde kus sesi çikartiliyordu. Çalisirken kafasi ve kanatlari hareket ettirip öterken de gagasini oynatailiyordu. Bunlardan çok sayida üretildi ve ev dekarasyonunda kullanildi. Kusun hareketleri belirli bir sirayla yapiliyordu bu sira takibi için miller kulaniliyordu, iste bu miller ilk bilinen ROM yapisidir.



Millerle yapilan rom yapisina iyi bir örnek: (*Meillardet'in Otomati dan alinmadir*)

Otomatik Flütçü: Müzisyen kiyafet giydirilmis otomatik flütçü dudaklarina yapisik flüte hava pompalarken parmaklariylada flütün deliklerini açip kapatarak müzik yapabilmekteydi. Yine millerle saglanan bir takim hareketlerle islem gerçeklestiriliyordu. Bu flütçü 1738 yilinda Jacques de Vaucanson tarafindan paris'de yapilmistir. Otomat tamamen el isiydi. yaplian bu otomat bir binada sergilendi ve insanlar bunu görebilmek için çok uzaklardan bile geldiler. Hatta krallar ve Imparatorlar bile özellikle memnuniyetlerini belirtiler. Iste bu ilgi ve piyasa kavrami pek çok mucidi otomatlar üzerinde çalismaya tesvik etti.



Yaptigi diger önemli otomatlar:

Meillardet'in Otomati: Belki de kurulu düzenekli otomatlar içinde en karmasik olani 1805 yilinda Londra'da Henri Meillardet tarafindan yapilan yazi yazabilen ve resim yapabilen inanilmaz otomattir. Genis bellegi ve titiz hareketleri vardir. Örnegin bir gemi resmini aslina uygun, bütün detaylariyla bes dakikada çizebilmekteydi. Yapabildikleri arasinda Fransiz stilinde yazilmis bes dizelik bir siir bile vardir. Bu otomat halen Franklin Enstitüsünde sergilenmektedir. DAHA FAZLA BILGI!


Meillardet'in siir yazan ve resim çizen otomati.


R.U.R. Kullanilan Otomat.



Dünya litaratüründe ilk defa "Robot" kelimesi 1917 yilinda Karel Capek'in kisa hikayesi olan Opilec de geçmistir. Fakat asil kavram olarak robot anlayisini 1921 yilinda yine ayni yazarin Rossum's Universal Robots (R.U.R.) adli tiyatro eserinde ortaya atililmistir. Eserde robotlar Rossum ve oglunun topluma hizmet için olusturdugu insan görüslü yaratiklardi.
Robot kelime olarak ise çek dilinden gelmektedir. Agir, sikici, angarya is manasindadir.

Dünyada ilk olarak robotlarla ilgilene bilim dalina "Robotic" ifadesini kullanan kisi Issac Asimov'dur. (Dogum Temmuz 2, 1920, Ölüm Nisan. 6, 1992). Kelimenin kullanildigi eser Runaround(1942) adli hikayesidir, bu eser *1,Robot* adli kitabinda yer almistir(1950).Ünlü bilimkurgu yazari hikayelerinde henüz olamayan fakat ileride olmasi muhtemel sorunlarla ilgili durumlari anlatmaktadir.Dünyaca ünlü bazi eserleri bazilari *I Robot*

(1950), *The Foundation Trilogy* (1951-52), *Foundation's Edge* (1982), *The Gods Themselves* (1972) bu kitabiyla Hugo vede Nebula ödüllerinin ayni anda kazanmistir. DAHA FAZLA BILGI!



Issac Asimov göre robot kavraminda insanligin gelecegi için üç önemli kuram vardir: (Daha sonradan 0. kurami eklemistir.)

- 0.kuram: Robotlar asla insan olgusuna zarar vermemelidir.
- 1.kuram: Robotlar asla insanliga zarar vermemelidir. Diger assagidaki kuramlar tarafindan aksi idda edilemez.
- 2.kuram: Robotlar insanoglundan aldigi emirleri yerine getirmelidir. Diger assagidaki kuramlar tarafindan aksi idda edilemez.
- 3.kuram: Robotlar keni varliklarini diger kuramlari bozmadan ellerinden geldikçe korumalidirlar.

1918-1938 2 dünya savasi nedeniyle bir çok arastirma durdu ve bazilari bir daha asla devam edemedi...!!!

[2.3] Robot Kavraminin Gelisimi: (Asil Robot kavramin olusmasi ve gelisimi)

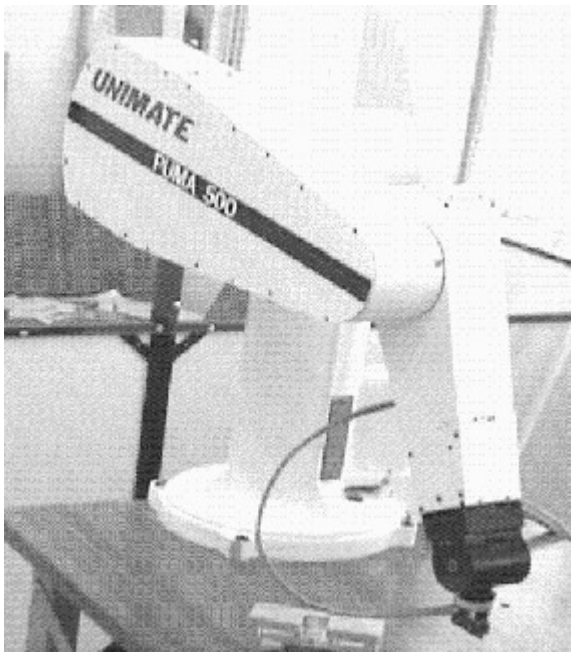
Sayisal kontrol ve Uzaktan kumanda kavramlarin gelismesiyle robotic çalismalarinda önemli gelismeler ortaya çikmistir. John Parson tarafindan uzaktan kumandali olarak yapilan makine 1940 yillarinda Amerika Birlesik Devletleri Hava Kuvvetleri tarafindan ardindanda Atom Enerjisi Komisyonu tarafindan kullanilmaya baslandi. Radyoaktif maddeler üzerine yapilan çalismalarda önemli islemlerde kullanildi. Bu sistem endüstri alaninda kullanilmaya basladi. Cyril Walter Kenward birlikte 1954 ün Mart ayinda patenti alindilar. Böylece ilk endüstriyel robot denilebilecek bir sistem tasarlandi.

Belli basli gelismelerden birinide Joseph F. Engelberger (Fizik Müh.) ile George C.Devol tarafindan gerçeklestirildi.Robotlarin Babasi olarak adlandirilan parça aktarim robotu yaptilar. Bilgileri magnetik ortamlarda saklamakti önemli gelisme çünkü artik daha fazla veri daha küçük alanlarda rahatlikla istenlidigi zaman degistirilebilecek sekilde saklanabildi. Ve çalismalari sonucunda "Unimate " adli firmayi kurdular 1949, bu firma ilk robotic üzerine kurulan firmadir.

Bu tarihten sonra dünya üzerinde özellikle Amerika, Avrupa ve

Japonyada pek çok firma robotic üzerine çalismaya basladilar. Ve bu ilgi gelismeyide beraberinde artirdi. Bu gelismeler arasinda göze çarpan ilk robot uygulama dili olan " WAVE" Stanford Akedemisi tarafindan gelistirildi ve robotic bilimine kazandirildi. 1974 de "AL " ve ticari amaçli olan " VAL " yapildi. Val Ultimate tarafindan gelistirlmisti ve buna bagli olarak "PUMA (Programmable Universal Machine for Assembly)" gelistirilerek üzerinde uygulandi. Nispetten kisa eklemli bir robottu fakat temel olarak General Motors firmasinin montaj hatti baz alinmisti. 1979 yilinda Yamanashi Üniversitesi tarafindan montaj amaçli olan "SCARA (Selective Compliance Arm for Robotic Assembly)" gelistirldi.Bu sistem ticari olarak 1981 piyasaya sürüldü.

Scara Hakkinda kaynak bilgiler ve örnekler için >



unimate'in puma robotu.

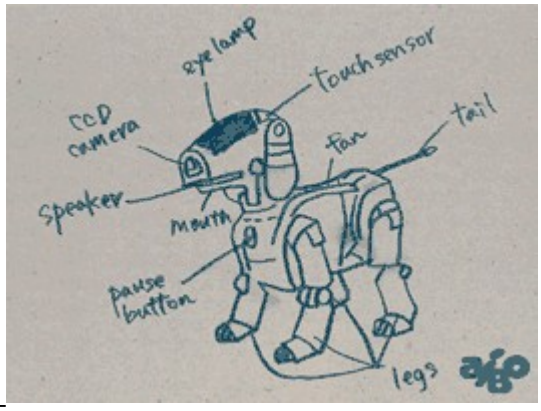
90 yillara gelindiginde robotlar artik çok çesitli alanlarda ve özellikle insanlarin rahatlikla yapamiyacagi isleri kusursuz yaparak insanoglu'nun yasam sürecinde yerlerini aldilar. Assagidaki örneklerde insan oglunun inemedigi derin sularda arastirma yapabilen, Hiçbir mola vermeden yillarca çalisabilen montaj robotlari, Cerrahlarin hata yapmasini engelliyen hatta cerrahlik meslegini oratdan kaldiracak kadar iddali olan ameliyat robotlari, insan kolunun yerine takilabilen yapay kol sanirim bu gelismelere en iyi örneklerdir.





Fakat asil gelisme Sony firmasi tarafindan evlerimize kadar sokulan robot köpek "Aibo" olmalidir. Yapilan robotta algilama karar verme vede komut dogrultusunda uygulama yetileri muvcut olmakla beraber genis açili bir hafiza ile sahibini tanimlama kadar birçok özellik yer almaktadir: (1998)





Sonuc olarak Robotic teknolojisindeki çalismalar artik büyük oranda bilgisayar teknolojisindeki gelismelere dayaniyor. Robotik endüstrisi dogdugunda bilgisayarlar mevcut olmasina ragmen 1970'lerin sonuna kadar boyutlari nedeniyle robot kontrolünde kullanmiya elverisli degildi. Bugün pazardaki tüm robotlar bilgisayar kontrolü kullanilmaktadir. Fakat

halen Robotic biliminin alaninin *makine* vede *bilgisayar* bilimlerini kapsadigi kesindir.

[2.4] Otomasyon Kavrami:

Otomasyon kavrami çesitli sekillerde ifade edilebilir ama kavram mutlak olarak mekanizasyonu içermektedir. Peki nedir bu mekanizasyon ve otomasyon.

- Mekanizasyon: Insan enerjisi disinda saglanan hareket olgusuna Mekanizasyon denir. Mekanizasyon kismen veya tam olabilir, bazi mekanik hareketler insan gücü katilarak tamamlanabilir buda mekanizasyonu kesinlestirmez.
- Otomasyon: Iste bu mekanizasyonlar sonucunda olusan toplu islem bloguna ototmasyon denir. Kontrol yani denetimden gelmektedir.

Otomasyonda önemli olan üretim ortamlari olan tezgahlar ilk olarak klasik tezgahlardi daha sonra Kam kontrollü, Pim kontrollü tezgahlarin kullanilmaya baslanmasi, üretim hizinin ve kalitanin artirilmasini sagladi. Daha sonraki gelisme Sayisal Kontrollü **(NC)** tezgahlarin uygulanmasi biçiminde olmustur. Biligisayar sisteminin tezgaha uygulanmasi sonucunda Bigisayar Kontrollü Tezgahlar **(CNC)** ve Bilgisayar kontrollü sistemler **(CIM)** ortaya çikmistir. Sadece üretimde degil Bilgi islem, Dökümantasyon, Arastirma ve Gelistirme çalismalarinda özel hazirlanmis olan bilgisayar programlari kullanilmaktadir. Iste bu sayede 20. yüzyil basinda insan oglu daha az çalisarak daha kisa sürede en mükemmelini üretmektedir.

Otomasyon kavrami için robotlar çok önemlidir çünkü ne kadar çok insana bagli olmadan çalisabilen robot sistemde yer alirsa sistem o kadar çok tam mekanizasyona vede hatasiz üretime geçmektedir. bugün gelismis ülkeler düzeyi denen toplulukta yer alan ülkelerin tamami üretimlerini otomasyon zincirleriyle tamamlamaktadir. Buna en iyi örnek robot için kullandigim motorun Isviçre'de tamamen insan eli degmeden 0.1 mikron çözünürlügünde yapilip test edilmesi sayesinde olusan mükemmel motor yapilaridir.

Ilk olarak tarihte otomasyon kavrami Henry Ford la birlikte baslamistir. Henry Ford 1800 yillarda Ford araba fabrikasini kurup isletirken aldigi araba siparislerini karsilayamiyordu. O da en kisa sürede montaji nasil tamalayacagini düsünürken iki seçenegi göz önünde bulundurmus.

- 1. Fabrikayi büyüterek daha çok isçi ve daha çok tezgah almak.
- 2. Tezgahlari bir sekilde hizladirarark bir tezgahi iki veya daha çok tezgah gibi çalistirabilmek.

Iste bu iki kavramdan birincisinde üretim kesinlikle artacakti ama kurulmasi vede çalismasi çok daha fazla is getirecekti fabrikaya bunun için Ford tezgahlari fabrikda üretim sirasina dizdi ve bir hareketli band

ile tezgahlar arasinda baglantiyi kurdu. Iste o günden sonra Ford firmasinin üretimi katlanarak büyüdü. O günler için yeterli görülen bu kavram birkaç sene sonra yetmemeye basladi çünkü insanlar yürüyen bandlara yetisemiyor hata yapiyor ürünlerin yüzde ellilere varan hatali üretimlere dönüsmeye basliyordu. Isçiler daha çok para daha az çalisma istiyorlardi. Sendika kurup güçlendiler. Vardiyali çalisma dönemi baslamisti ama halen bir sorun vardi üretim kazancinin nerdeyse yarisi fabrika giderlerine yani maaslar vede hatali üretim giderlerine gitmeye baslamisti, ürünlerin fiyati artiriliyor ve maliyetin 3 katini asan fiyatlara satiliyordu. Ama bir gün, bir Japon bilmadami mikrochip denen küçültülmüs elektronik komplex parçayi yapinca isler biraz degisti artik üretimde 24 saat çalisabilecek maas istemeyen, hata yapmayan, hasta olmayan, greve gitmeyen mükemmel isçi gelmisti, robotlar gelmisti. bu olgunun yararlarini assagida maddelemeye çalistim:

Otomasyonda robotlarin Yararlari:

- Üretim artisi.
- Üretim maliyetinin düsmesi.
- Kalite artisi
- Tehlikeli ortamlarda çalisabilme.
- Yönetim ve denetim kolayligi.
- Tümlesik sistem organizasyonu.
- Is esnekligi.
- Uzun ömür.

[2.5] Robotlarin yapisal çesitleri:

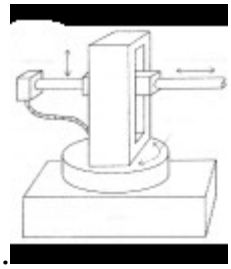
Temel olarak robotlar kapsam ve yetenkleriyle siniflandirilabilirler ama bence asil olarak yapisal yeteneklerine göre siniflandirilabilir:

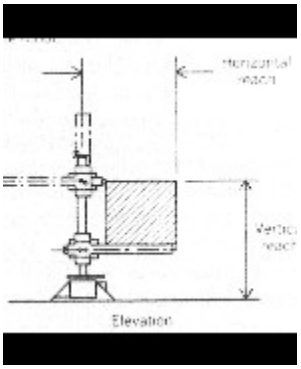
- **Kartezyen Robotlar:** X,Y,Z koordinat düzleminde her kol bir öncekine göre dik açiyla kayar. Dikdörtgen sekline bir çalisma alanlari vardir.

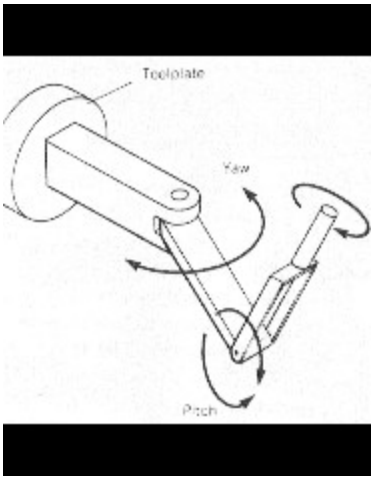
- **Silindirik Robotlar:** Çalisma alanlari silindiriktir.Kolun bir bölümü dikey, diger bölümü ise yatay hareket eder.

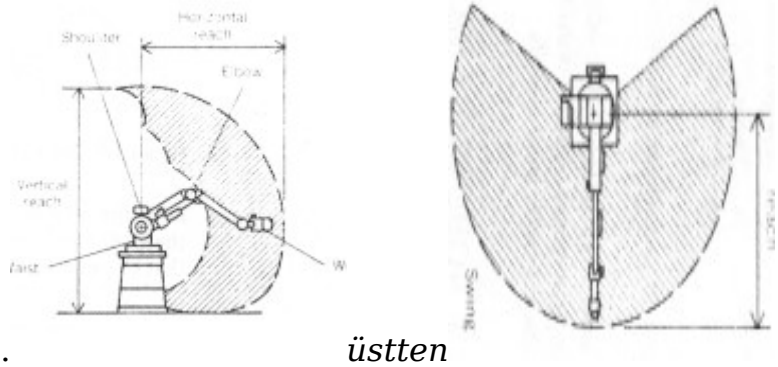
- **Polar Robotlar:** Kol taban etrafinda dönebilir. Kolun bir parçasi içeri disari öteleme hareketi yapabilir. Bir bölümüde assagi yukari dönebilir.

- **Revolüt Robotlar:** Dönel veya küresel eklemler sahiptir. Taban eklemine bagli olan kol taban etrafinda döner ve diger iki kismi tasir. Dönel eklemler yatay ve dikey olarak birlestirlmistir.Yarim küre seklinde bir çalisma alanlari vardir.

1. 

2. 

3. 

4. 

üstten

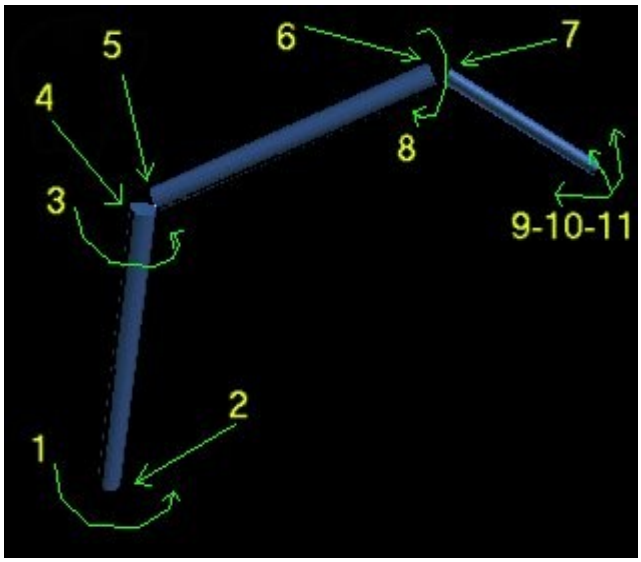
[2.2.5] Günümüzde kullanilan belli basli robot örnekleri:

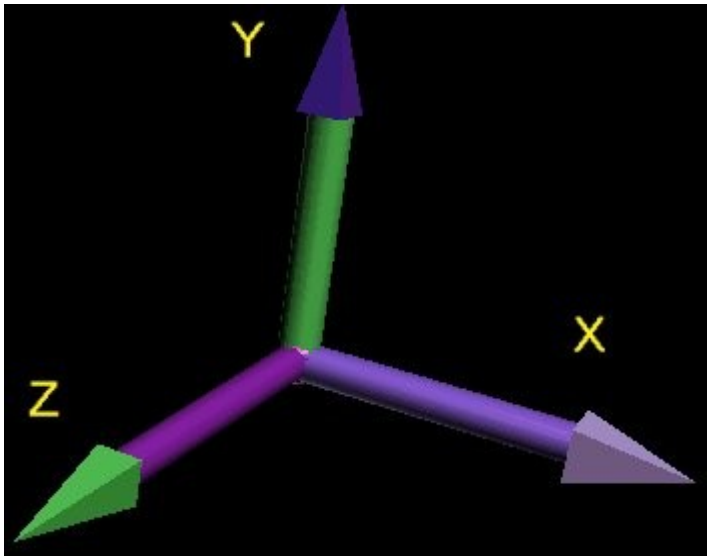
Piyasaya bakildiginda satilan robot türlerinde bir genellesme dikkat çekici. Robotlar genel olarak bir amaç için degilde bir amaç robota uyarlaniyor, yani ek parça tanimlayici vede özellikle program olgulariyla yapi kusursuz olarak tamalaniyor.

Piyasa robotlarini su sekide siralayabiliriz:

- Robot Kollar (özellikle tip alanindaki kullanimda çok hizli bir ilerleme var) örnek.
- Su alti robotlari (birkaç firmanin su alti madencilik ve arastirma üzerine araçlari var ama bir yavas ilerleyen çalismalar sanirim) örnek.
- Uzay arastirma robotlari (nasa bu konuda devreye giriyor.) örnek.
- Mobile robotlar (Bataryali olduklarindan gelisme kisitli oluyor ama mikro islemcili tasarimda bayagi ilerleme saglanmis) örnek.

[3] Fiziksel Olarak Robot Kol





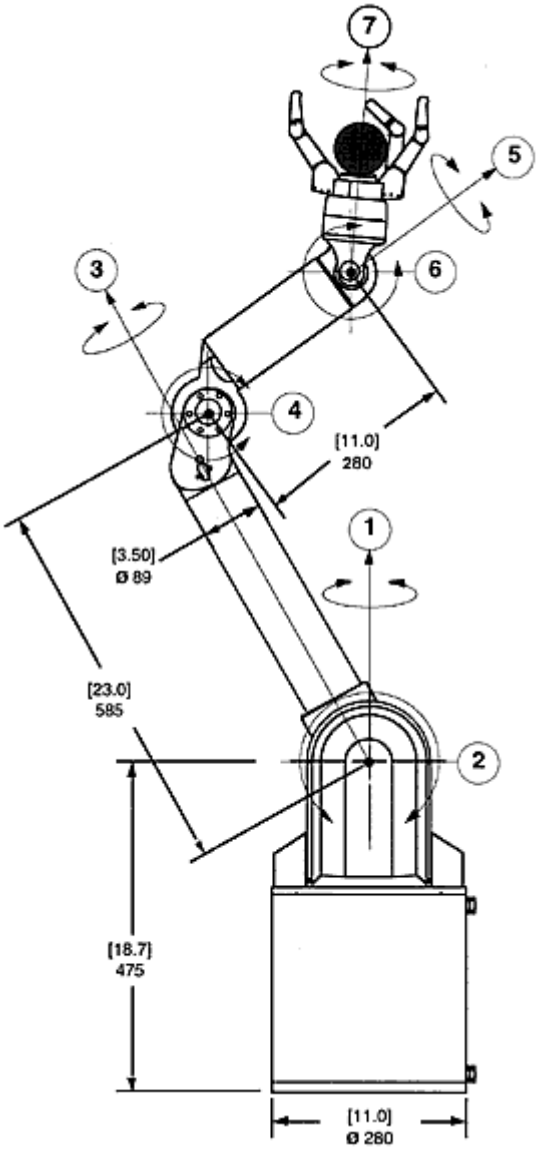
Fiziksel olarak bir robot kol ucu uca eklenmiş cubukların 3 boyutlu uzaydaki salınımına benzer. Fakat dünya şartlarında bu ivmelere yer çekimi, sürtünme ve aşınma gibi ufak tefek görünen ama sonucu etkileyen faktörler yani vektörler vardır. İlk olarak 3 boyutlu uzay kavramıyla üç eklemli bir kolu yapısal olarak incelersek;

Eklemler birbirlerini izlerken bu izlemeden doğan hareket bağımlığı ilede birbirlerini etkilemektedirler. Kısacası en altan en üste doğru doğrusal bir zincir yapısı vardır demek doğru olabilir.

Eklemler biribrlerini farklı yönler ve şekillerde hareketlendriebilrler buda bize üç boyutlu uzayda erişim alanını artırmamızı sağlar. Daha doğrusu hareket yeteneği kazandırır.

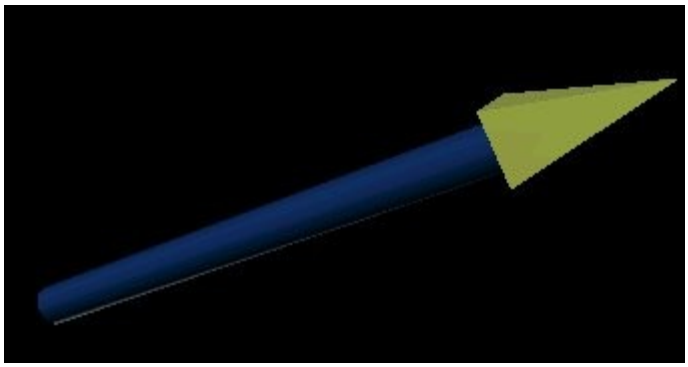
Ayrıca şekilde her eklemin kendine ait 3 eklemi vardır bu eklemeler silindirik eksenli olup kendi etraflarında, ilerisindekini yukarı aşşağı oynatan vede altındakini sağa sola çekebilen eklemlerden oluşur vektöriyel merkezleri motor başları olup 2 si üst kısmında diğer 1 ise alt kısmı ucundadır. Sadece tutucu kısmı aynı anda iki eksenli bir ekleme sahiptir.

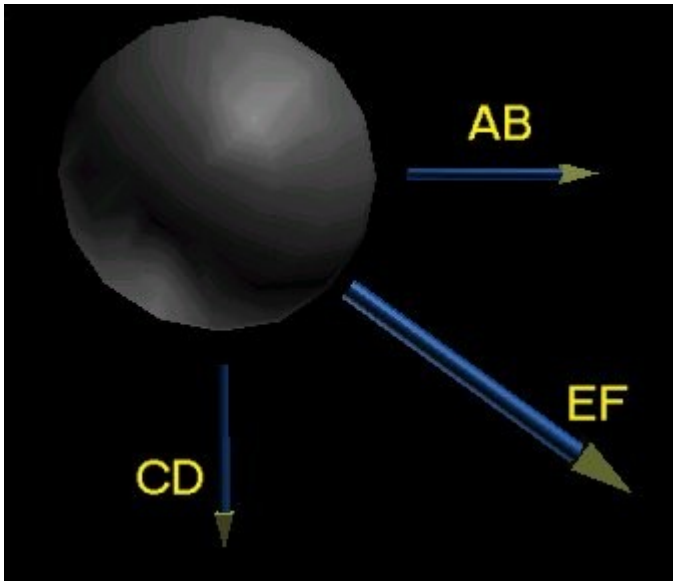
Aşşağıda ise tasarlanmış bir robotun ana hatlarını görmektesiniz:



[3.1] Temel Fizik Kavramları ve Hareket Teoremi

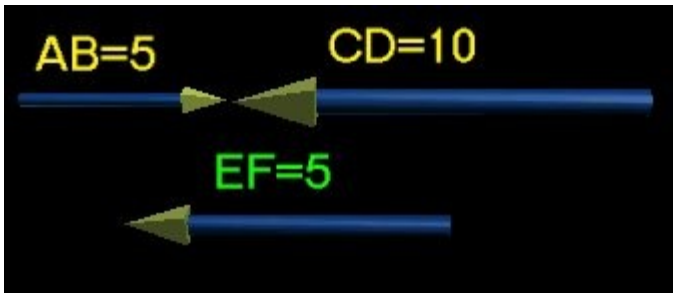
Vektör bir hareket göstergecidir. Hareketin yönünü kuvvetini gösterir. Temel olarak bir hareket konumsal vektörlerle tanımlanabilr.

vektör: 



Şekilde görüldüğü gibi küresel objenin hareket yönü AB vektörleri ile gösterilmiştir fakat objeye uygulanan yerçekim ivmesi CD vektörüyle aşşağıya doğru bir çekme uyguladığı için ortalama(bileşke) vektör olan EF vektörü yönünde obje hareket edecektir bu EF vektörü AB büyüdükçe AB'ye doğru yakınlaşacaktır. tam tersi durumdada yani CD vektörü büyüdükçe CD vektörüne doğru yaklaşacaktır.

Sonuç olarak fizikte temel hareket birimi vektörlerle ifade edilebilir çünkü her hareketin bir şiddeti vede bir yönü vardır. Vektörler üç boyutlu uzayda ifade edilebilirken her hareket bileşeni vektörlere ayrılması zorunludur. Aynı zamanda ters etkime yapan karşı vektörlerde bu kurala dahildirler.



Vektörler temel olarak sade ve tek düze olmalarına rağmen fatöriyel şeklindeki katlamalı artımlarda sonuç vektörünü tespit etmek için ihmaller hesabı ortaya atılmış ve sonuca öylece varılabilmiştir. Fakat robotumuzun sonuca mikron düzeyinde ulaşmasını istiyorsak mutlak olarak bu ihmal hesaplarınıda mikronun altında tutmalıyız.

[3.2] Kinematiksel İnceleme

Kinematik kavramı robotların temelini oluşturmaktadır, çünkü robotlara yaptıracak hareket ve sonuçta ulaşılacak konum bu dal tarafından incelenmektedir. Eklem kinematik için çok büyük bir yere sahiptir ve genelde robotlar açısından eklemler bir zinciri andıracak şekilde ard arda bağlanarak kullanılmaktadır işte bu yüzden kinematik temel eklemler açısından ikiye ayrılmaktadır. Zincirin açık olduğu sistemler ve kapalı zincir yapısındaki sistemler.

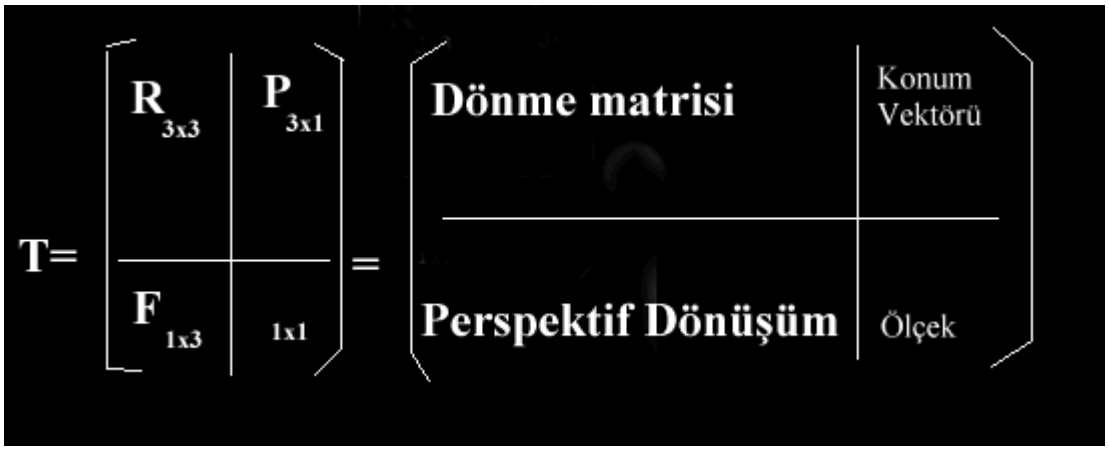
- Açık uçlu kinematik zincirde bir ucu serbest iken diğer ucu sabit bir aksama bağlı olan eklemdir.
- Kapalı uçlu kinematik zincir ise her iki ucuda sabitlenmiş olan eklem türüdür.

Bir robot kolu kinematiğin en önemli başlangıç ilkesi kinematik hesabına nereden ve ne biçimde başlanacağıdır, bunun için iki ana başlangıç noktası vardır. Bunlardan birincisi sabit ağırlıktan (en altta yer alan sabitleme kütlesi) başlayarak en uca ulaşmak buna ***düz kinematik*** denir diğeri ise en uç noktadan başlıyarak sabite ulaşmaktırki bunada ***ters kinematik*** denir. Unutulmaması gereken kinematik hesapları kayıpsız konum algılaması yapar, robot koluna uygulanan momentleri dikkate almadan hesap yapıldığından gerçek sonuç bulunamamıştır. Yapılan tüm işlemler kolun eklem sayısına bağlı olarak değişen katlara sahiptir bu katlar mutlak olarak eklem serbestlik ve tepkime değerlerinide içermektedir.

Geçmişten günümüze geçen sürede birçok sistem ve uygulama gerçekleştirilmiştir ve oluşan sorunlar karşısında yeni çözümler oluşturularak sonuça varılmıştır. Bu sistemlerin çoğunda yapılan hesap programları mutlak olarak ters kinematiği kullanmaktadır. Aşşağıda temel olarak iki çözümün ayrıntıları yer almaktadır;

Robot Kolunun Düz Kinematiği: Kolun yönlendirlmesinde her eklem hareket noktası bir merkezi XYZ eklemsel düzlemleri için başlangıç kabul edilip hareket sonucunda oluşacak yeni konum en alttan en üsteki ekleme

kadar takip edilmektedir. Üç eksenli bu yapıda her eklem için bir 3x3'lük matris tanımlanır eksenlere uygulanan dönme hareketi sonucu bu matrisin 4x4 olmasını sağlar ve 4-4 elemanı daima 1 olacak şekilde 4 satır ve 4 sütun bilgileri sonuç konumudur. İlk olarak Denavit Hartenberg tarafından kullanılmıştır.



Robot Kolunun Ters Kinematiği: Düz kinematiğe göre daha kapsamlı bir hesap gerektiren ters kinematik işlemleri daha çok hedefe geometrik ve matematiksel ifadelerle yaklaşarak sonuca ulaşmaya çalışmaktadır. Hedef konuma yaklaşımlarda integral gibi tersleyici fonksiyonlar kullanıldığından sonuç kesin bir yolla çıkmaz değişik yönerge ve denemlerle sonuca ulaşılabilir, geometrik açıdan çözümselliği daha elverişli olmasına karşın her robotla aynı şekilde kullanılamaz. robotların kısıtlayıcı etmenleri olan eklem ve dönüş serbestlikleri devreye girmektedir. Altı eksenli bir robotun her hareketi yerine getirebilineceği idda edilmektedir fakat sonuçta kolun yapısı ve hedef konum bilinmeden eklem sayısı olayaı netleştirmez.

Sonuçta kinematik hersaplar bize hedef konuma ulaşmak için enkısa yolu değil, en uzun tasarımı sağladığı için gereksizdir.

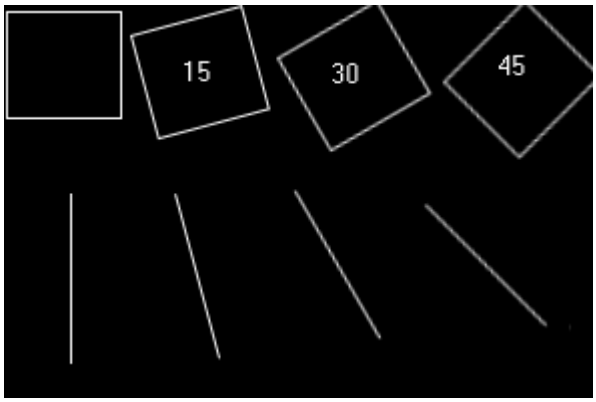
[3.3] Üç Boyutlu Tasarım ve Matematik



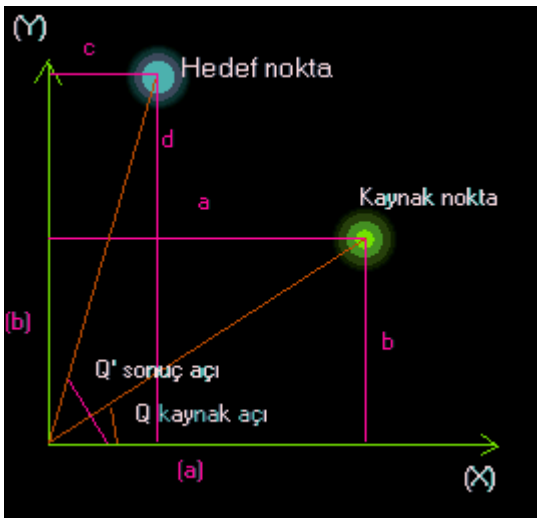
X,Y,Z eksenlerinden oluşan üç boyut kavramı beraberinde döndürme ve iki boyuttan üçüncü boyuta geçme gibi işlevleride getirmektedir.

Eksensel döndürme:

Cisimler bir koordinat düzleminde noktalardan oluşmakatdır bu cismi bellirli bir yönde döndürme işlevinin yapılması demek işte bu noktalarının açısal cevrim sonrası yerlerini belirlemektir. Aşşağıdaki örnekte bir karenin 45 derece sola döndürülmesi incelenmektedir;



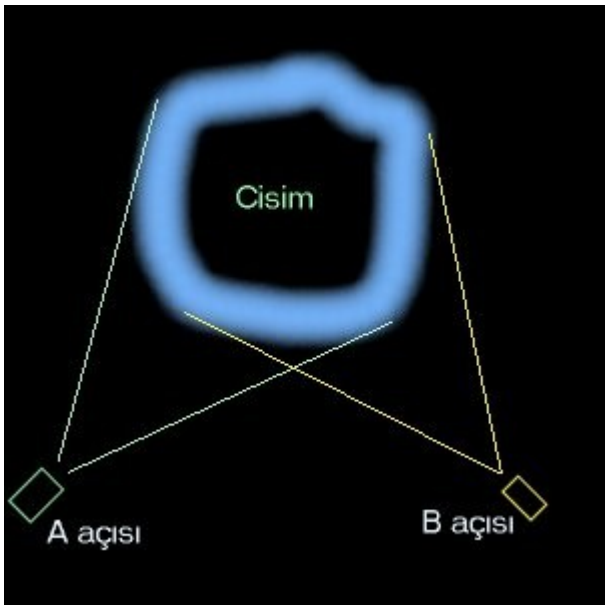
Matematiksel açıdan bir doğrunun cevrilmesi iki boyutlu düzlemlerde tan, arctan değerleriyle mümkündür. Yani bilinen iki noktadan aktif düzlemsel açı arctan ile bulunur daha sonra çevrim açısı eklenerek sonuç açıya ulaşılır vede sonuç açınısının tan değeriylede hedef noktaya ulaşılır. Aşşağıdaki örnekte bu olay görülmektedir;

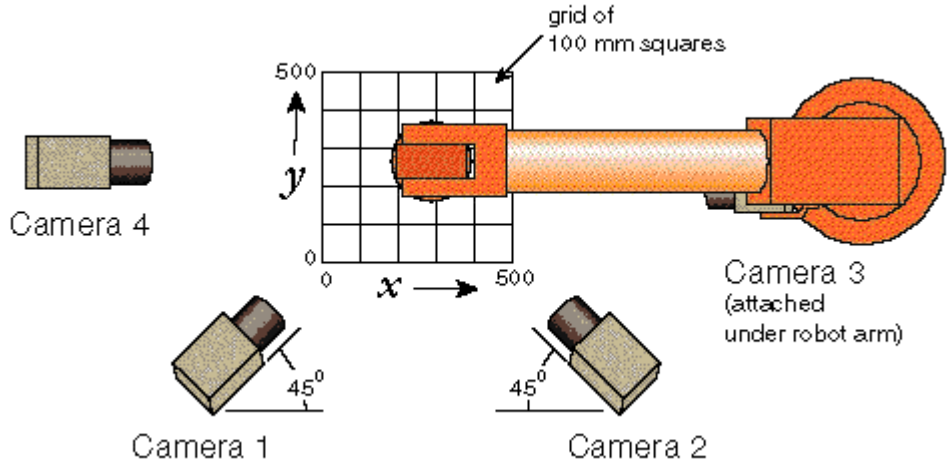


Örnekte Kaynak nokta a,b değerlerinde iki boyutlu düzlemdedir. Noktanın Q açısı arctan ile (**Q=arctan(b/a)**) bulunur ve açıya eklenecek değer eklendikten sonra Q' açısal değeri elde edilir vede yeni noktasal değerler d,c sinüs ve cosinüs işlemleriyle bulunur (**d=sin(Q')*f , c=cos(Q')*f, hipotenüs f=^(a²+b²)**). Noktasal hedefe varılmış olur. Üç boyutlubir düzlemde ise bu noktasal iki boyutlu çevrim iki şer eksenlere ayrılarak yapılır; X-Y; X-Z; Y-Z gibi.

İki adet iki boyuttan üçüncü boyut kavramı:

Cisimler genelde tek bir açıdan bakıldığından dolayı iki boyutlu görüntüleri mevcuttur buyüzden cismin tam olarak algılanabilmesi vede işlemlerin doğru bir şekilde yapılabilmesi için iki boyutlu görüntüden açısal dödürmeyle bir adet daha alınır bu sayede iki boyutlu 2 farklı görüntüden içsel üçüncü bir görüntü çıkarılabilir. çekilen görüntü sayısıyla doğruluk ve çekim açıları arasındaki oran sonucu doğrusal olarak etkilemektedir.





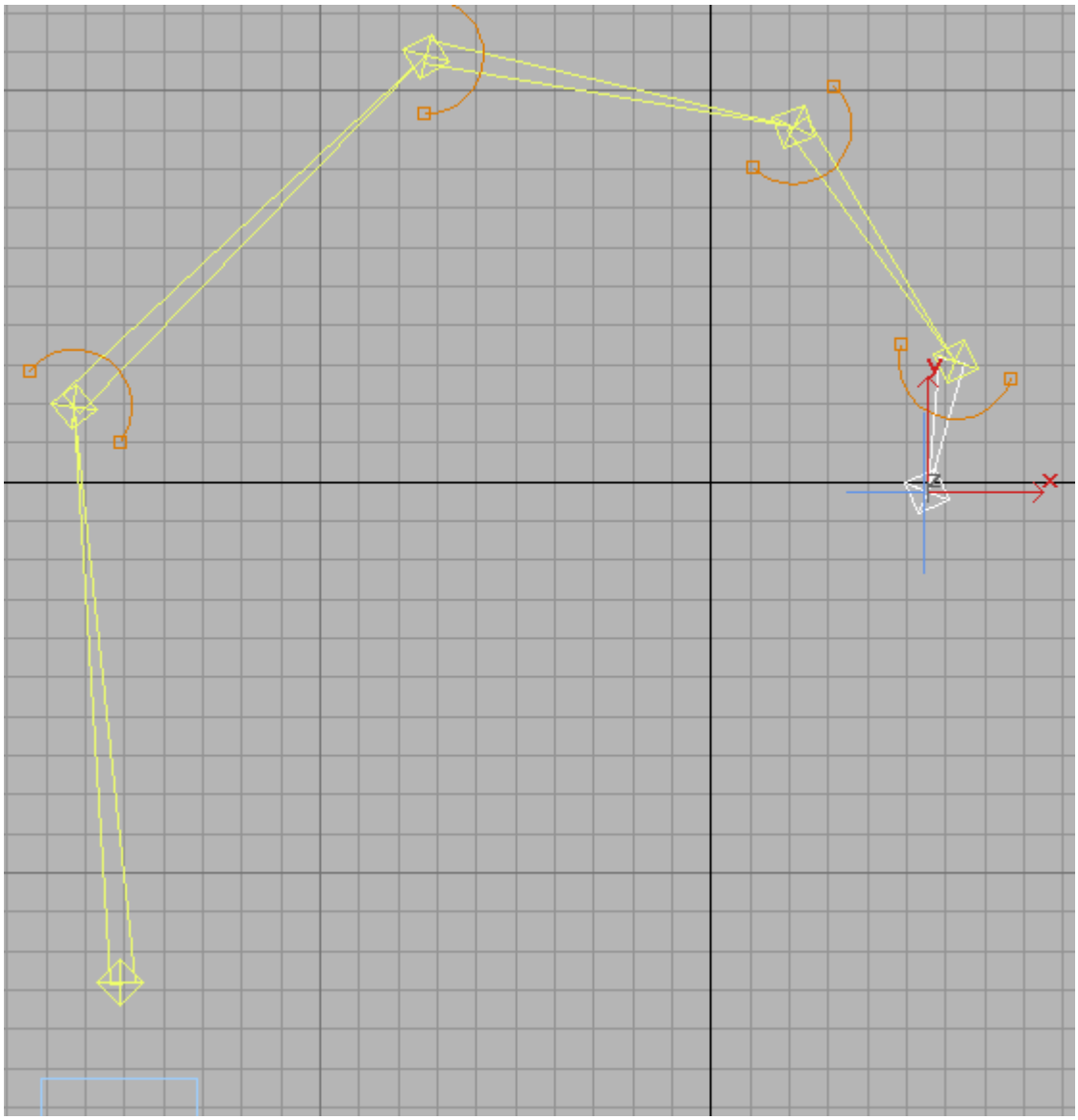
Sonuç olarak temel tirigonometri ve temel cebir robotic açısından konum bazlı büyük önem taşımaktadır. Robotun bir sonraki hamlesini hesaplamak doğacak hataları ve yanlş tanımlamaları engelliyecektir. Bu yüksen sabit noktasal hesap için bilgisayar işlemcisi çok uygun bir tabandır.

[3.3.1] Gelişmiş Konum Formül Tasarımı

İşlemler artıkça ve sonuca bağımlı yeni ortamlar real-time(gerçek zaman) bazlı çalışması gerektiğinde bu hesaplamaların gecikme yaratmaması vede sonucu işlem bittikten sonra yalanlamaması gerekmektedir. Bunun için yapılan hesaplar artıkça işlemler birleştirilmeli ve kalıpsal tanımlara kullanılarak bir tampon oluşturulmalıdır. İşlem sonuçlarının bitirilmesi kısaldıkça sistem optimize edilerek gereksiz işlemler taranmalı ve ayıklanmalıdır. bu işlemler hem bilgisayarın hemde mekanizmanın hata yapmasını ve yavaşlamasını engelliyecektir.

[3.3.2] Bilgisayar kavramı ve dinamiksel çözümler(Acad14-3DS max 2.5)

Bilgisayar kavramı işte asıl bu kavramların hesaplanması ve analizinde önemli bir yer almaktadır. Üç bouytlu olarak bir noktayı ilerletmek için yapılan işlem mikro saniyeler aldığından robotun hareketinde bir gecikme faktörü olmaktan çıkıp geri besleme vede hata aramada yer almaya başlamaktadır. Olmazsa olmaz kavramında yer alan bilgisayar aynı zamanda bize dizayn aşamasında şimdi olduğu gibi özgün yardımcı olarak yerini almaktadır. Ama bilindiği gibi bilgisayar programlanmadan bir hiç olduğundan bizim işimizi görebilecek temel bir üç boyutlu tasarım ve geliştirme programına ihtiyacımız bulunmaktadır işte burada devreye alman yapımı olan mühendislik harikası olan AutoCad programı girmektedir. Program temel kalıp çizimleri ve yapısal analiz yapabilmektedir ama yinede daha ortada parça yokken o parçayı test edebileceğimiz ve sonuçlar çıkarabileceğimiz bir program olan 3DS MAX adlı programa ihtiyaç duymaktayız. Max programıyla Olmayan kolu cizip eklemleri kısıtlayarak yapabileceği ve yapamıyacağı hareketleri görebiliriz tabiki bu bize çok önemli bir avantaj sağlamaktadır. Aşşağıda 3DSmax 2.5 cizilmiş temel bir örnek yer almaktadır bu örnek üzerinden kemik(bone) yapısı ve eklem kısıtlamayı inceleyebilirsiniz.

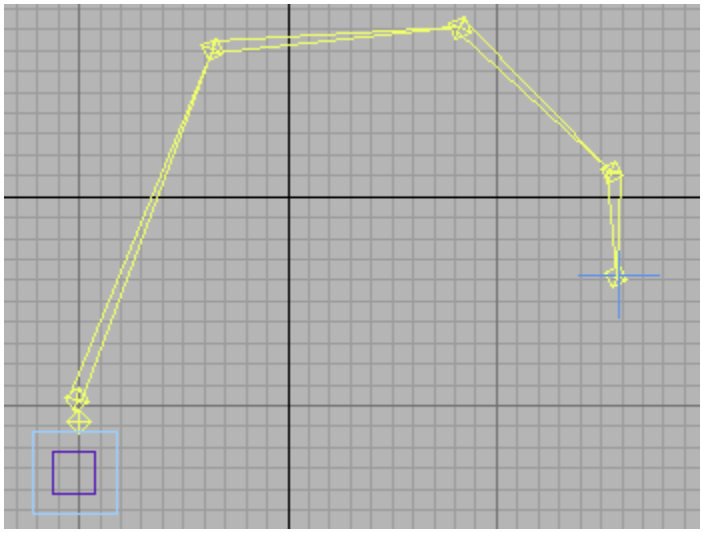

Y
X

3D studio max programında çizim kısmından çarklar şeklindeki ikon seçilerek bones(kemik) seçilerek ekranda temel bir çizgi çizercesine robotun temel eksenleri çizilerek ikonu altındaki IK bölümünden *Rotational Joints* kısmından kemik başlarının herket kısıtlaması eksensel olarak limitlenebilir. Daha sonra yapılmak istenen hareket ANIMATE butonuna basılarak **1/100** yazısı üzerinde bulunan bar çekilip oradaki hareket konumlandırılarak animate kapatılıp play butonuna basılır, yapdığımız hareketi bize tekrarlar tabiki eğer mümkünse?

Sonuç olarak bilgisayar ve doğru program her dalda olduğu gibi robot dizaynındada biz yol göstermekle kalmayıp hatlarımızı minumuma indirebilmektedir.

[4] Mekanik Olarak Robot Kol:

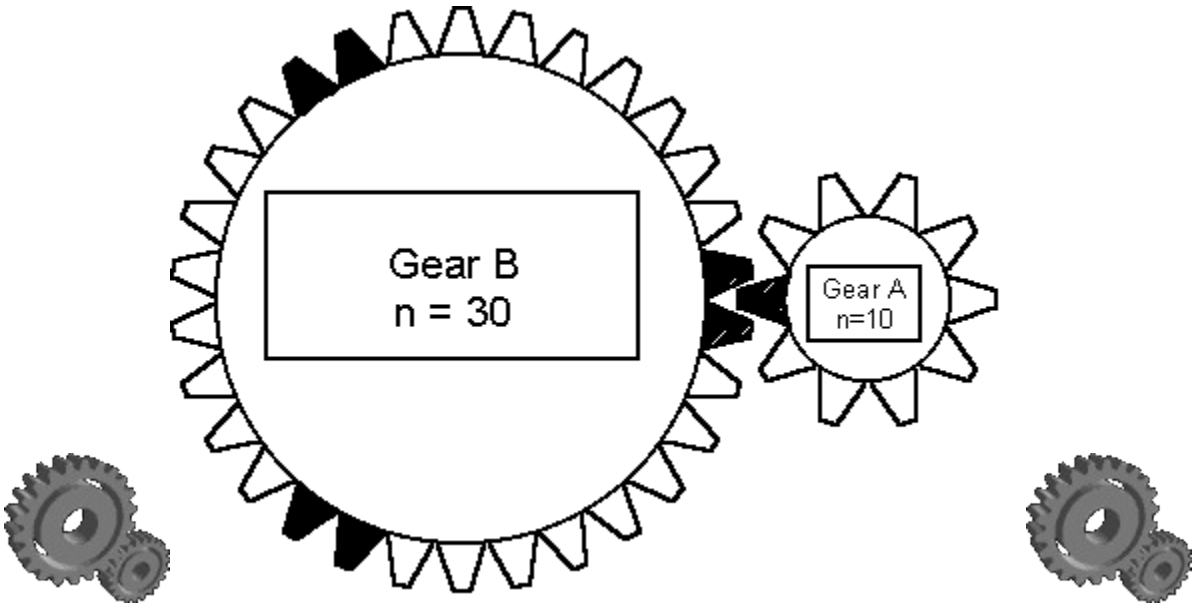
Mekanik olarak robot kavramı fiziksel olarak doğan hareket kavramlarının gerçekleştirilmesi için gerek duyulan mekanik aksamlar yani eklemlerin oluşmasıya otaya çıkmıştır. Mekanik açıdan robot kavramı EKLEMSEL olarak inceleme ve madde üzerindeki hareket, tepki ve sürtünme gibi kavramları içerir. Kısacası yapılmış olan bir robotta hesaba katılması gereken vektörlerden biride mekanik açıdan tepki ve etki vektörleridir. Ayrıca robotun yapabileceği hareket ve yapamıyacağı hareket kıstasları da mekanik olarak yapılan dizayn sonucunda ortaya çıkmaktadır. Aşşağıdaki örnekte kemiksel olarak dört eksenli bir robot kol görülmektedir;



Robot dizaynında ilk olarak yapılamsı gerekenlerden biri robotun yapması gereken harekeler doğrultusunda yapılacak eklemlerin konum ve serbesliklerine karar vermektir. Bu karar robotun kapasite, serbeslik, çözünürlük, erişebilrilik vede hız gibi ana temellerini doğrudan etkilemektedir, eklemleri uzunluk ve hacimleri ise yine robotun hareket ögesinde yer edinmekle beraber robotun eklemlerinin kırılması veya eklemlerin titremesiyle sonuçlanabilir.

[4.1] Mekaniğin Temelleri:

Mekaniğin temeli yine hareket ve madde ögeleri üzerinde yoğunlaşmaktadır. Mekanikte amaç bir maddesel ögeyi belli bir şekilde hareketlendirme ve durdurma işleminin en az güc ve en fazla hareket ivmesiyle gerçekleştirmektir. Tabi bunun yanında mekanik aksamların aşınma, dayanma ve süreç gibi kavramlarınıda hatapaylı hesaplayarak önlem belirleme işide mekanik dizayna düşmektedir. Mekanik açıdan bir olayın analizini yapmakla işin elle tutulur bir kavram olmasında önemli bir yol katedilmiş olur fakat her zaman mutlak olarak yapılma istenen hareket karşısındaki ters kuvvet veya engelleyici etmenler olayı gerçekleştirmemizde en önemli karşı kuvvetler olur, hareketin yapılması için mutlak olarak sadece bir tek yol yoktur bunun için yapılacak hareketin başka bir etmene yıkılarak kaybın kabul edilecek miktarın altında olması sağlanabilir. Bu konuda en iyi örnek dişlilerdir, dişliler kuvveti kendinden bir sonraki etmene aynı şekilde artırıken hızından kaybettirerek güç kazandırmaktadır buda bize motor gücünün kaldıramıyacağı yükleri daha yavaş ama kaldırılıp hareketlendirilebilecek düzeye getirilmesini sağlar. Aşağıdaki şekilde bir dişli yapısı görülmektedir;

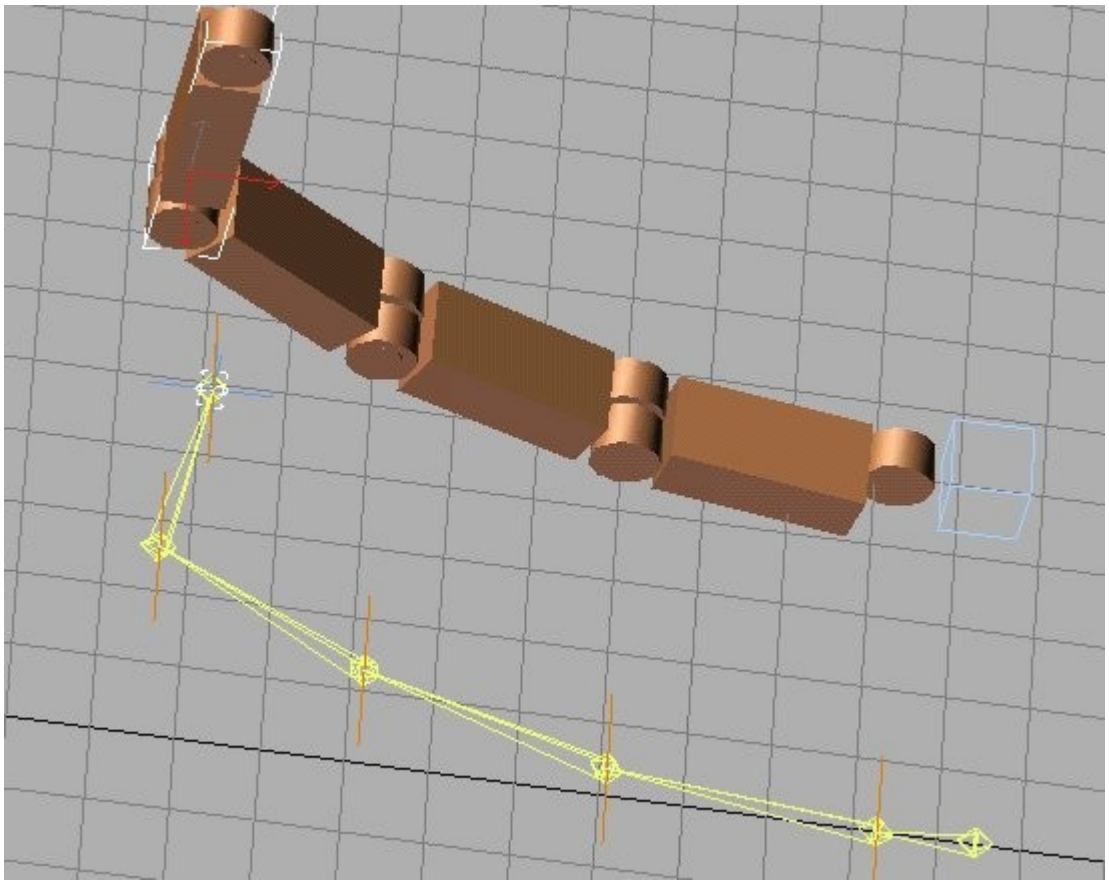


Yukarıdaki örnekte A dişlisinden verilen kuvvet B dişlisinden 3 kat daha büyük olarak ama 3 kat azalmış bir hızla bize geri dönmektedir, buda bize

robotumuzun eksen hareklerini motorları büyütmeden ve ağırlığı arttırmadan istenilen şekilde yapılmasını sağlar.

[4.2] Eklem Analizi:

Her ne kadar istenilen kadar eklem tasarlanıp ard arda eklenir gibi gözüksede hedef en az eklemle istenilen hareketi yapabilmektir. Elimizdeki motor ve katı tutucular zamanla aşınma ve esneme gösterdiğinden eklemlerin uç uca eklenmesiyyle oluşan ağırlığa ve hareket sırasında oluşan ters kuvvetlerin etkilerine dayanamayıp istenilen hassasiyetin dışına çıkabilmektedir. Aşağıda görülen örnek mükkemmel derecede insan kolunu taklit edebilen bir yapı göstermekle birlikte gerçek ortamda yetersiz kalmaktadır.



İlk olarak yapılması gereken hareketler çıkarılmalıdır bunun için istenilen yön ve şekildeki harekler sınırlandırılmış bir otamda tasarlanmalıdır. Eklem bağlantı noktaları ve eklem uzunluklarını hedef olayda en etkili şekilde etkileyen etmendir. Eğer eklem bağlantıları hatalı ise eklemler istenilen hareketleri yapamaz veya motorlar hesaplananın dışında yüklendiğininden istenmiyen durumlarla karşılaşılılabilinir.

[4.3] Gücün Ağırlığa Oranı:

Bazı sistemlerde büyük motorlarının kullanılması yüzünden ağılığın artmasıyla eklem bloklarıda güçlendirilmiş ve motorların kuvveti tam

olarak yükü kaldırmaya ve hareketleri yapmaya yeterli olamamıştır, bu durumu düzeltmek için piston ve hidrolik sistemler tasarlanmıştır. Günümüzde hem motorlar hemde sürücü katları ilerlemesi çok iyi bir konumdadır, cm^2 ye 80 Newton uygulayabilen 8000 devirlik dc servo motolar ile birlikte kullanılan spur yapıdaki 415:1 oranındaki dişli kutularıyla kuvvet gerekenin 100 kat üzerine bile çıkabilmektedir.

Dikkat edilmesi gereken nonta kuvvet/ağırlık oranının yaklaşın 1 olması gerektiğidir. Tersi durumlarda hareket ivmeleri veya ağırlık çekmeleri sonucunda kısa zamanda robotumuz devre dışı kalmaktadır. Buna en iyi örnek yüksek kuvvet gerektiren sistemlerde gücü artırmak için yapılan işlemlerde ağırlığında artmasıya yeniden gereken kuvveti kazanmanın imkansız olduğu görülünce hedefe ulaşma yönteminin değiştirilerek eklem ve eksen azaltılmasıyla olay çözümlenmiştir. Bu tip durumlarda ters ağırlık veya zıplama ivem hareketleri sadece çözümü imkansız hale getirmemekte çalışma zamanınıda kaybetmemizi sağlamaktadır.

[4.4] Mekanik Açıdan Robot Kol:

Robot kolun en önemli görülen ve ilk olarak tasarlanması gereken konularından biridir mekanik analiz, çünkü robotun yapabileceği ve yapamıyacağı hareketler bilinecek elektronik dizaynıyla bilgisayar programıda ona göre ayarlanacaktır. Oysa ortada çalışacağı ümit edilen mekanik üzerine yazılmış programı ve muhteşem elektronik dizaynı gerçekleşemiyen mekanik eklemleri çözememektedir.

Şimdi sırasıyla dikkat edilmesi gereken mekanik konuları ve çözümlerine bir bakalım;

[4.4.1] Mekanik Kısıtlamalar:

Mekanik bir paçanın eğer oynuyan bir parçası varsa mutlak olarak bu parçanın bir şekilde korunması yani ana kütleye sabitlenerek serbest bırakılması yani kısıtlanması gerekmektedir. Bu kısıtlamalar yapılacak harekette gerek duyulmayan şekilde yapılmalıdır, eğer yapılması gereken bir hareketi eklem kısıtlaması yüzünden yapmıyorsak işin ta başında başarısız oluruz. Eklemin serbestlik açıları çıkarılmalı eriye kalan bütün yönler kısıtlanacak şekilde eklem kısıtlanmalıdır, bu sayede mümkün olan en yüksek güçte sağlamlık ve esneklik sağlanmış olur.

[4.4.2] Madde Aşınması ve Ömür:

İki metal birbirine hareketli bir şekilde monte edildiğinde hareketler sonucunda zamanla metal aşınması yapmakta ve iki oynayan yüzey arasında boşluk oluşturmaktadır bu boşluk belirli bir düzeyden sonra istenmiyen ters haretklerin yapılmasını sağlamakta ve başarısızlıkla sonuçlanmaktadır. Yapılması gereken en az yüzey sürtünmesinin kurulması ve gerekli yüzeylerdede çelik halka veya teflon yüzey kullanılmalıdır.

[4.4.3] Mekanik İhmaller ve Sonuçları:

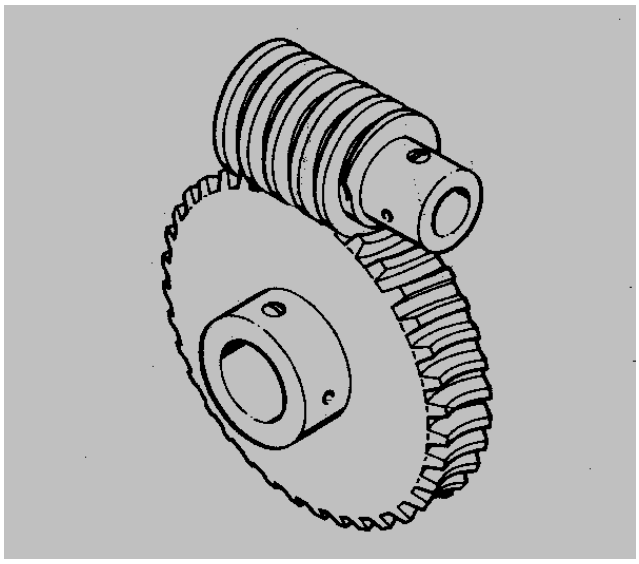
Oluşacak sürtünme ters kuvvetlerinin ve yer yüzü çekiminin ihmal edilmesi ilk denemelerde her ne kadar başarılı gözüksede zaman içinde çalışan yüzeyler arasında sürtünme yüzünden boşluk ve yer yüzü Bilgisayar kavramı işte asıl bu kavramların hesaplanması ve analizinde önemli bir yer almaktadır. Üç bouytlu olarak bir noktayı ilerletmek için yapılan işlem mikro saniyeler aldığından robotun hareketinde bir gecikme faktörü olmaktan çıkıp geri besleme vede hata aramada yer almaya başlamaktadır.

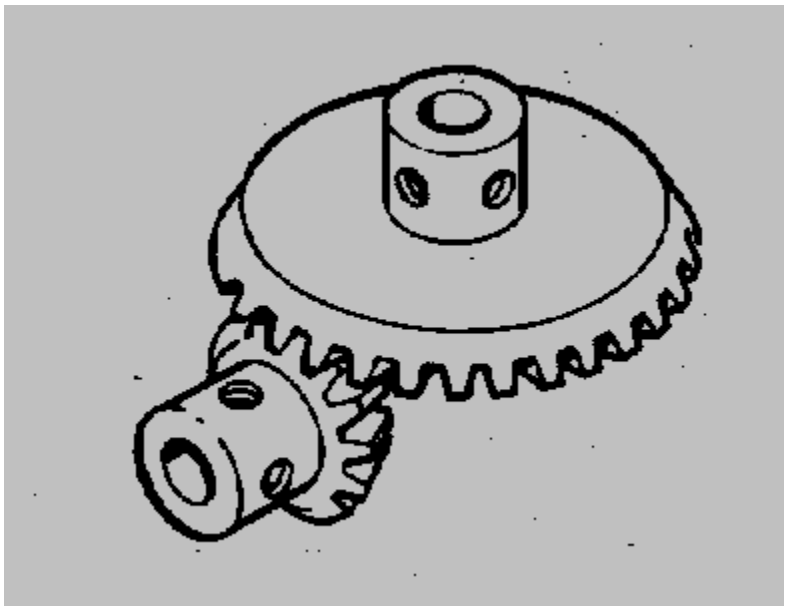
[4.5] Mekanik Hareket:

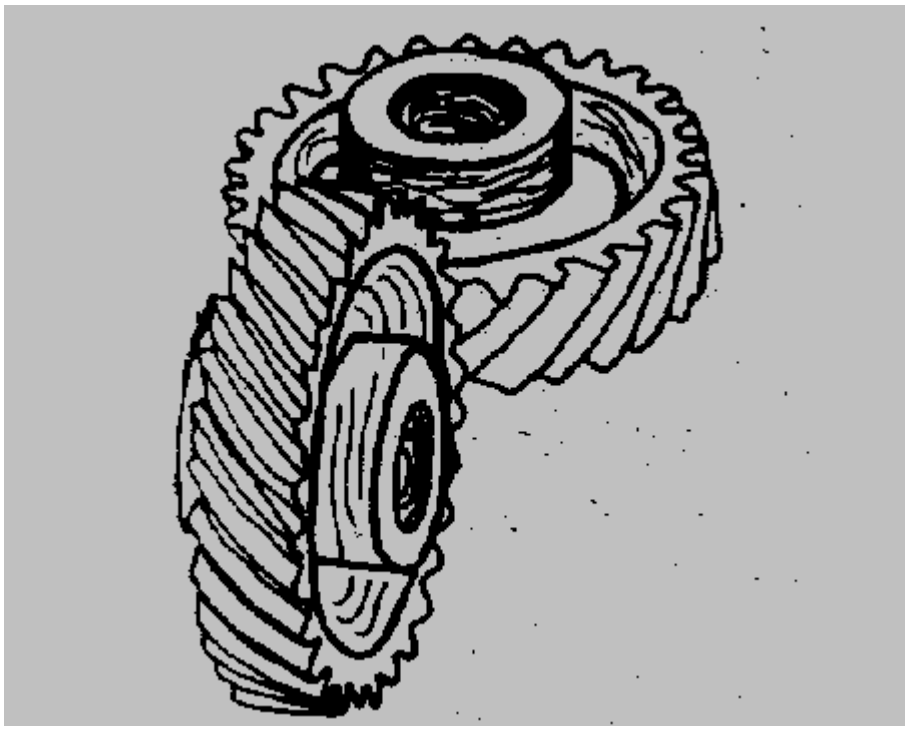
Yapılacak hareketin mekanik olarak tanımlanması kavramı ilk olarak eklemlerin hareketlenmesiyle gözlenebilir. Eklemler hareket ettiren kuvvet olan motorlarla birlikte bir hareket şekli göstersede dış etmenler sonucunda istenmiyen veya zorlayıp hiçbir hareket yapamadan olduğu yerde kalabilir. bu yüzden mekanik harekette normal hareket hesaplamalarının dışında hareket ettirilen metal aksamın ve sabitlerin davraışlarıda hesaplanıp ona göre yeni çözüm ve teknikler uygulanmalıdır. İlk olarak hareketin yönü ve şekli incelenmelidir;

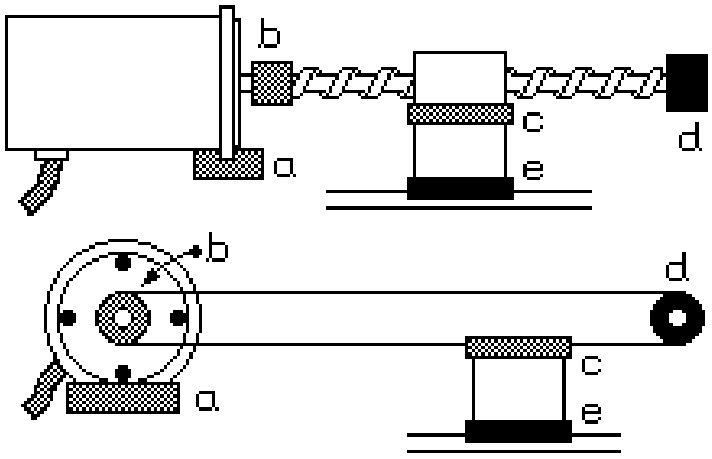
[4.5.1] Hareketin Yönü ve Şekli:

Yapılması istenen hareket için bir yönelge ve tepki hesaplanırken mutlaka hedef sonuç için başka türlü sonuca varabilecek çözümlerde aranmalıdır. Aşağıda çeşitli hareket çevrimleri saysesinde sonuca ulaşilmıştır;



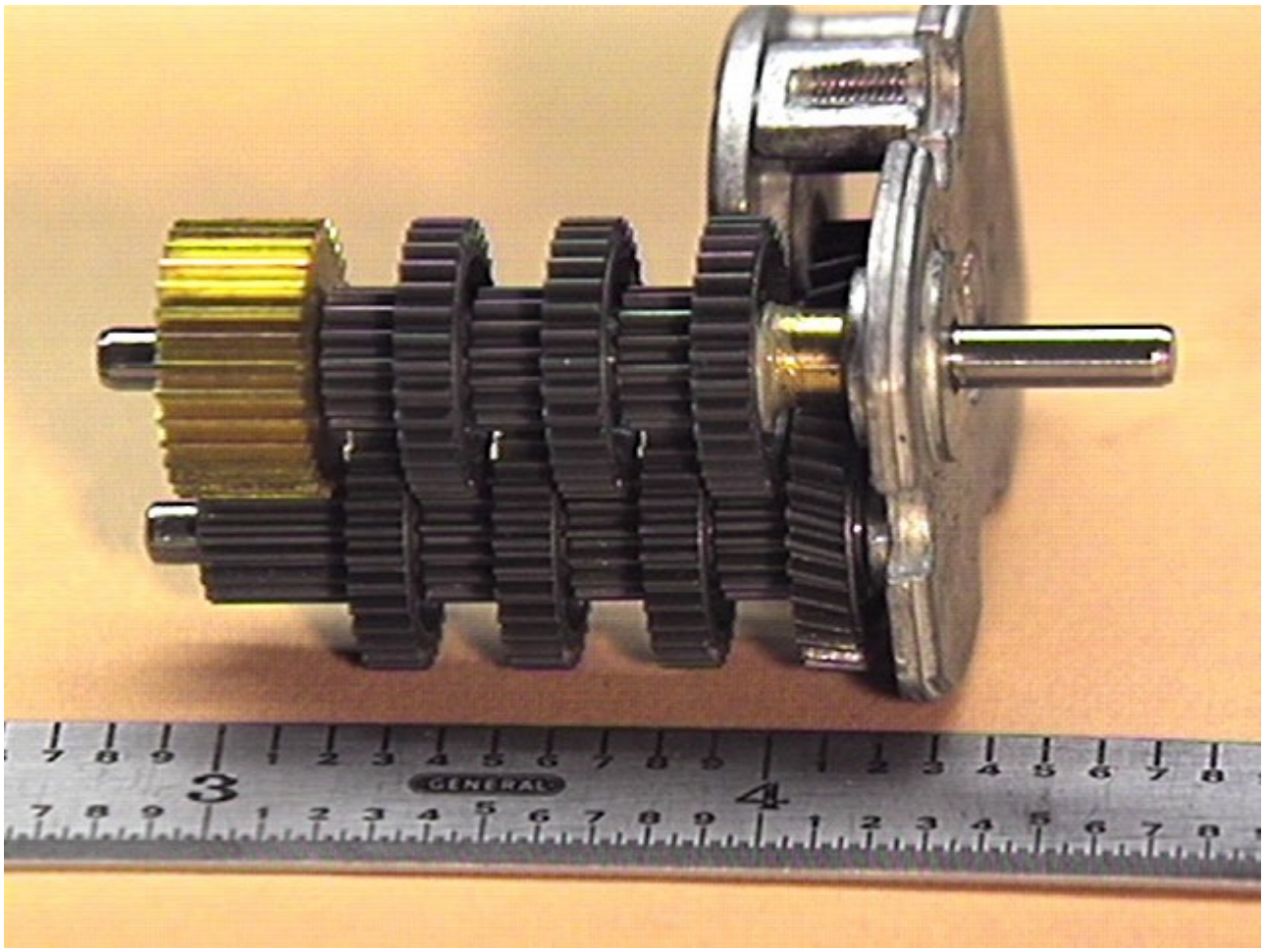




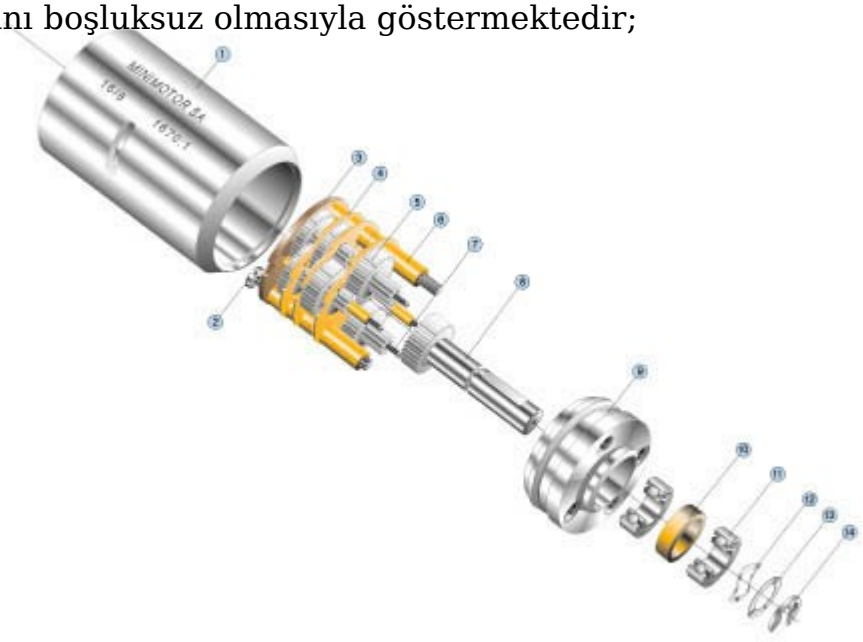


[4.5.1] Hareketin Çevrimleri:

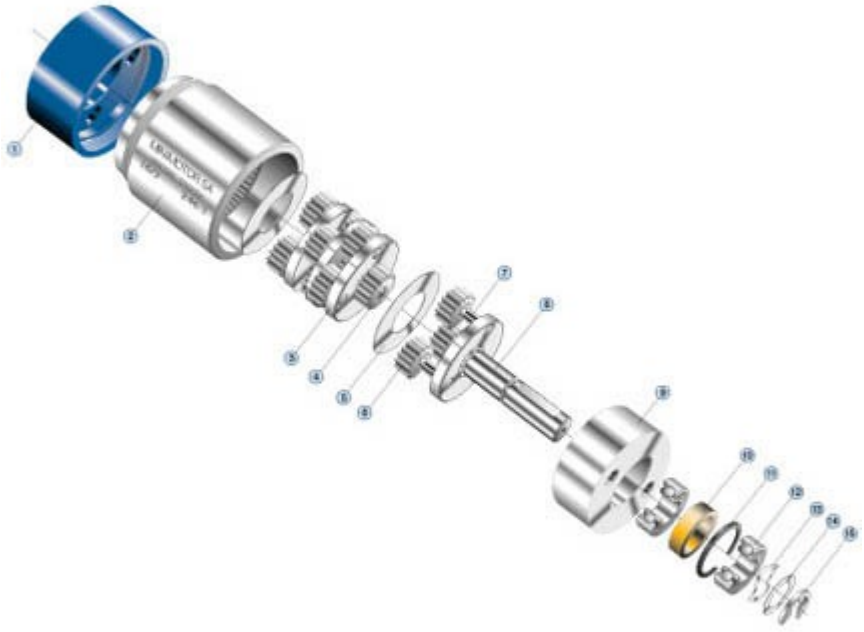
Yaplıma istenen hareket yön ve şekil değiştirerek yani modellenerek yeni bir kalıba konur bu olaylar sayesinde gereken güç veya ivme yönü kazanılabilir. Aşağıda buna en iyi örnek olan dişli kutuları ve onların yapıları görülmektedir;



Dişli kutuları projemiz açısından çok büyük bir öneme sahiptir. Bu kutular kendi aralarında iki gruba ayrılırlar. Spur ve Pleantary. Pleantary yukarıda görülen örnekteki gibi her dişli bir sonraki tek dişliye değerek döngüsünü tamamlamaktadır. Spu ise her dişli kendinden sonra üç dişliye ve üç dişlide bir dişliye değecek şekilde dizilmiştir bunun en önemli yararını boşluksuz olmasıyla göstermektedir;



Copyright © 1998
by Minimotor SA

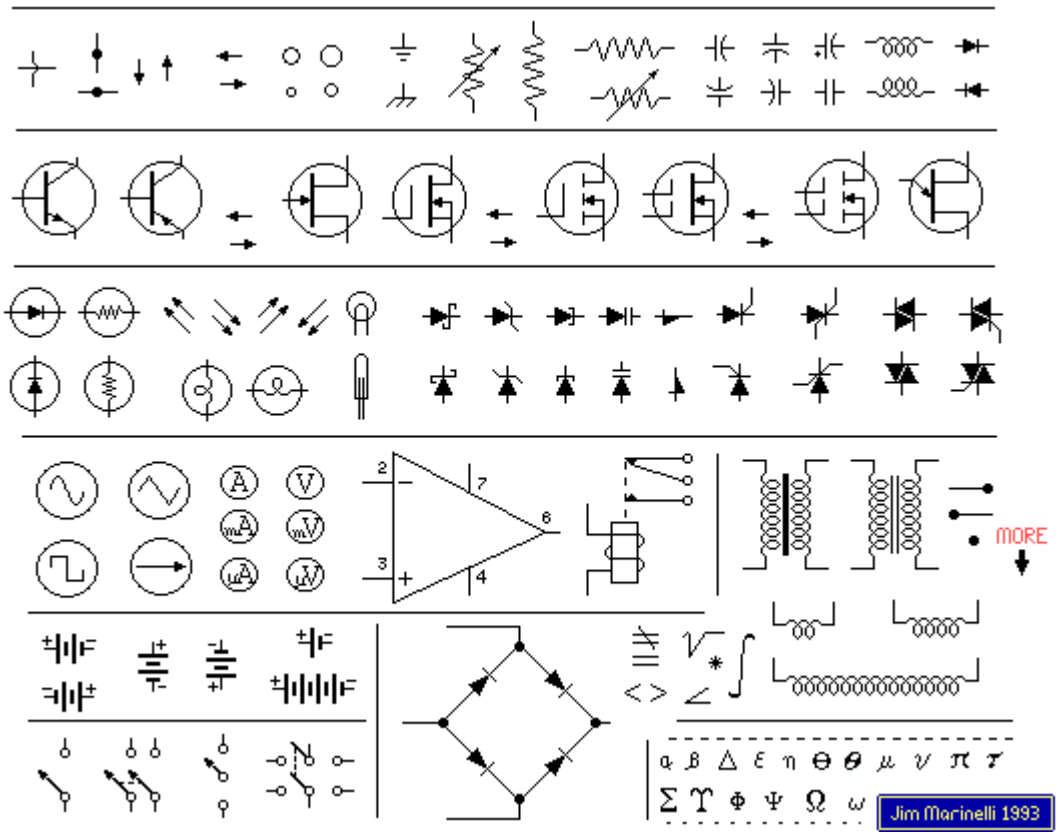


[5] Eletronik Olarak Robot Kol:

Temelde robot tasarımda Pc kullanımından sonra Elektronik kontrol ve tasarım ön plana çıkmıştır. Sistem elektronik karkarlar verebilmek için elektroniksel ölçme ve hareket olgularını edinmeye başlamıştır. İlk olarak elektronik robot tasarımında bilgisayarlar ile haberleşme konusunda karşımıza çıkmaktadır, bilindiği üzere bilgisayarlar lojiksel ifadeler ile haberleşirler (0-0,5V :0 manasına / 4,5-5V ise 1 manasına gelir) işte bu haberleşmeyi yapabilmek için bir veya birden fazla hat üzerinden zamansal olarak veri aktarabilecek sistemler tasarlanmaktadır. Asıl işi

mikro işlemciler yapsada yapı elektronik tasarımla tanışmaktadır. Daha sonra ise elektronik olarak veri taşınarak yapay sinir ağları kurulmakta ve eklemler hem birbirleriyle hemde bilgisayarla haberleşmektedirler vede işte bu olay yapının tamamen esneklik ve kararlılık kazanması yolunda atılan en büyük adım olmaktadır. Enson basamak olarakta hareket ve geri beslem sistemlerinin elektroniğe taşınarak (motorlar ve encoderler sayesinde) yapıya eklem ucundan bilgisayara kadar elektronik bir sistem kurulmaktadır. Gelin bu ana yapılara temelleriye birlikte genel bir göz atalım...

Örnek devre elemanları:



[5.1] Temel Elektronik Bilgileri:

Eğer insan elektronik kavramı tam olarak kavrayabilirse işlem basamaklarını ancak o zaman istediği gibi geçebilir. Yapılması gereken her işte olduğu gibi elektronik devreler hayatımızda önemli bir yere gelmiştir; kumandalar, adaptörler ve daha niceleri. Herkez bir şekilde bu devreleri sökmüş ve incelemişizdir. Aynı başka bir dili öğrenmek gibi

elektroniğide bilen bir şahıs zorlanmadan istediği gibi bu devreleri onarmakla kalmaz yenisini yani daha iyisini bile yapabilir.

Elektronik tanımını kısaca yapmak istersek ilk olarak eletrik akımı yani temel olaya inmemiz gerekir bu olayıda anlamak için mutlak olarak atomlarla ilgili bilmemiz gereken noktalar vardır;

Atomlar doğada aktif olarak üç farklı şekilde yer alırlar;

- Katı
- Sıvı
- Gaz

Modlar arasındaki önemli farklar ağırlık ve yoğunluktur. Bu atomların hareketsel bir yapısı olduğundan enerji bakımından ise ikiye ayrılırlar;

1.Potansiyel Enerjili : Pozistonunu koruyan enerjidir.

2.Kinetik Enerjili : Hareket enerjisidir.

Atomlar birbirlerine çok benzeselerede yapısal olarak çok küçük parçalardan oluşurlar. Bu parçacıklar sayesinde atomlar 92 farklı cins oluştururlar. Bu ayrımı gösteren periodik tabloda atomların numaraları ve ağırlıkları yer almaktadır. 92 olan sayı son yıllarda yapılan nükleer çalışma sonuçlarıyla 14 adet eklenerek 106'ya çıkmıştır. Yandaki linkde ayrıntılı bir tablo yeralmaktadır. http://www.webelements.com/

1845 yılından sonra atomun içyapısı tam olarak açıklanabilmiştir. Yapıda ELEKTRON, PROTON ve NÖTRON bulunmaktadır. Elektronlar serbest yörüngede gezen negatif(-) yüklü parçacıklardır. Proton ise merkezde yer alan pozitif(+) yüklü parçacıklardır, sabit konumludur. Nötronlar ise yüksüz ve merkezde yer alan hareketsiz parçacıklardır. Yapıda hareketli yapıda bulunan elektronlar atomun tipini belirler. Eğer elektronlar protonlara göre fazla ise atom NEGATİF, Elektronlar Portonlardan az ise POZİTİF, Elektronlar Protonlara eşit ise atom NÖTR yapısındadır.

İşte burada iki atom yakınlaştırıldığında birinin yükü diğerine göre zıt yönlü ise Negatifden Pozitife geçiş olacaktır, bu olaya elektrik akımı denilebilir. Elektrik akımı daha önceki yıllarda Pozitifden negatife olarak bilinmekteydi fakat prensip olarak Negatiden Pozitifedir.

İşte bu olguları matematiksel olarak ifadelendirsek genel kavramları belirlememiz gerekir;

Amper(A): Bir iletken üzerinden gecen elektrik akım miktarı. (Birimleri: Amper,MiliAmper, MikroAmper)

1 Amper(A)= 1 kulomb (Q) eşittir.

Volt (V): İki farklı yükle yüklenmiş ortamlar arasındaki yük farkı yani kısaca gerilim farkı. (Birimleri: KiloVolt,Volt,miliVolt,mikroVolt)

V=W/Q W joule ile gösterilen güçsel fark, Q kulomb olarak bilinen bir sabite.(1 kulomb 6,240,000,000,000,000 elektron yada protondur)

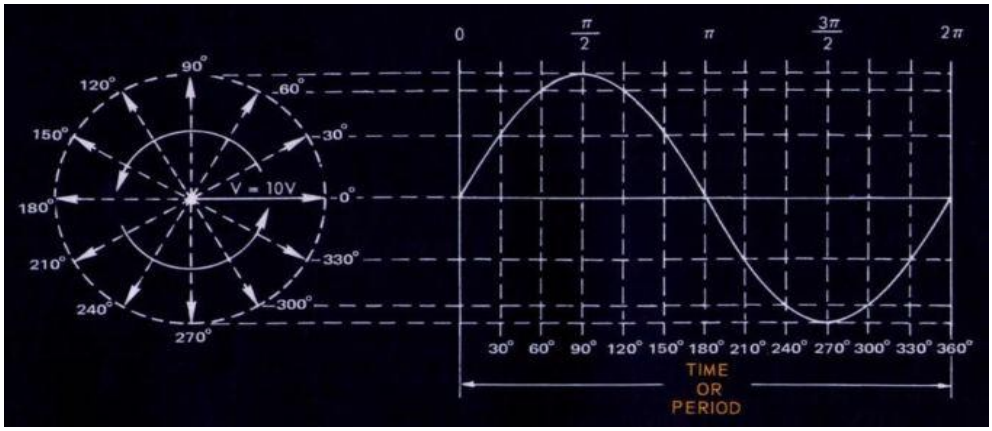
Yukarıda görülen kavramlar anlık kavramlarıdır. Elektriksel güç yani zamana yayılmış olan enerji ise P ile gösterilir birimi watt dır.

P=W/t dir. W Ayak_ağırlığı (food-pounds) , t ise zamanı gösterir.

1 HP = 768 watt dır. HP beygir gücüdür.

P= I x E yede eşittir. E gerilim farkını I ise akım gücünü temsil eder.

Devreye uygulanan akım ve gerilim birbirleriyle doğru orantılı olarak değişirler fakat bu değişim zamansal bazlı olarak süregelirse işte o zaman akım ve gerilim AC diye tabir edilen dalgalı yani sabit olmayan gerilim ve akım olacaktır. Örneğin sehir şebeklerinde gerilim -320V ile +320V arasında değişmesine rağmen biz o gerilimi 220V olarak bilmekteyiz çünkü zaman yayılan bu gerilim efektif olarak yani ortalama olarak 220V gözükmektedir. Bunun tersi olan yani sabit bir gerilimleri olan piller ise DC diye tabir edilen bir gerilim ve akıma sahiptirler. Aşağıda açısal olarak bir sinüs dalga tasviri yer almaktadır.



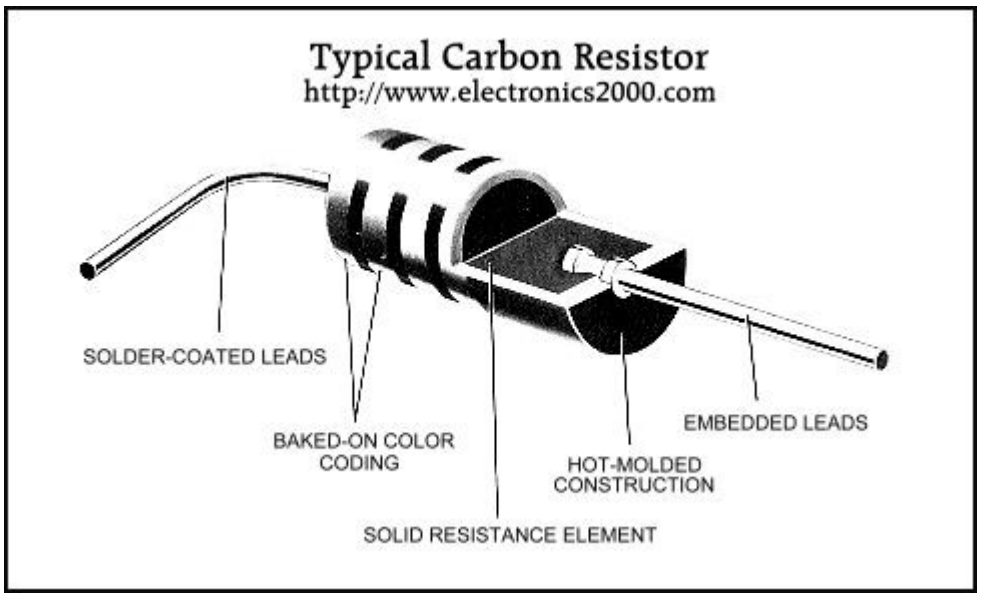
DİRENÇ:

Devreye uygulanan gerilim ve akım bir uçtan diğer uca ulaşıncaya kadar izlediği yolda birtakım zorluklarla karşılaşır. Bu zorluklar elektronların geçişin etkileyen veya geçiktiren kuvvetlerdir. İşte bu kuvvetlere DİRENÇ

denebilir. kısaca Ωohm ile gösterilir. Aşşağıda örnek bir direnç ile maddelerin iletkenlik oranlarını görebilirsiniz.



Madde	*p* (rho)	
ALUMINUM	17.	
CARBON	2500-7500	x copper
CANSTANTAN	295.	
COPPER	10.4	
GOLD	14.	
IRON	58.	
NICHROME	676.	
SILVER	9.8	
STEEL	100.	
TUNGSTEN	33.8	



Bir direncin iç yapısı

Dirençler şekildeki gibi tasarlandıkları gibi farklı maddelerden farklı şekil ve bağlantılarlada tasarlanabilirler;

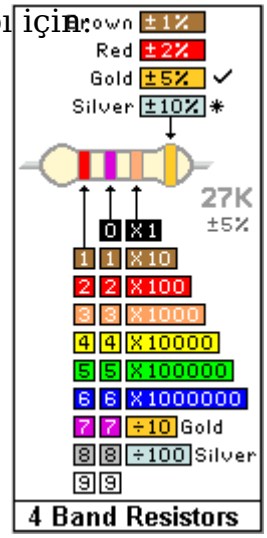
Carbon Dirençler: Şekilde görülen basit devre direncidir.

Güç Dirençleri : Yüksek güçlü akımlar altındada rahatlıkla çalışabilen dirençlerdir.

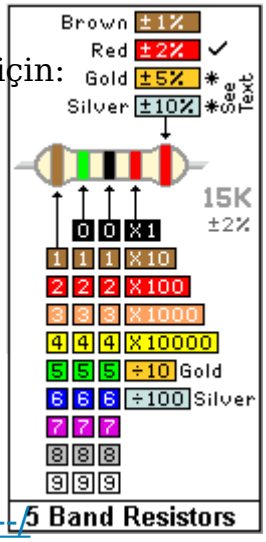
Potansiyometre : Üç uçlu ayarlanbilir bir dirençtir.

Bu dirençlerin hacimlerinin ufak olması ve sabitsel olarak kodlanabilmesi için renksel direnç kodları oluşturulmuştur, aşşağıda bu kodları inceleyip hesap yapan siteler mevcuttur;

4 bandlı direnç hesabı için:



5 bandlı direnç hesabı için:



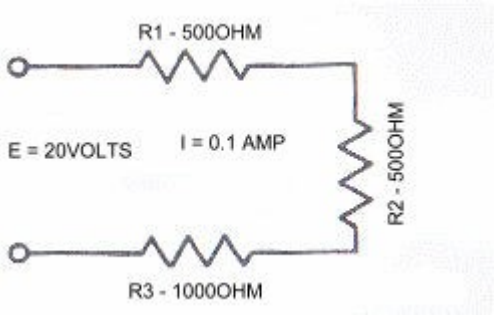
HESAP YAPAN BİR SAYFA-1

Ohm Kanunlarıyla Hesap!!!

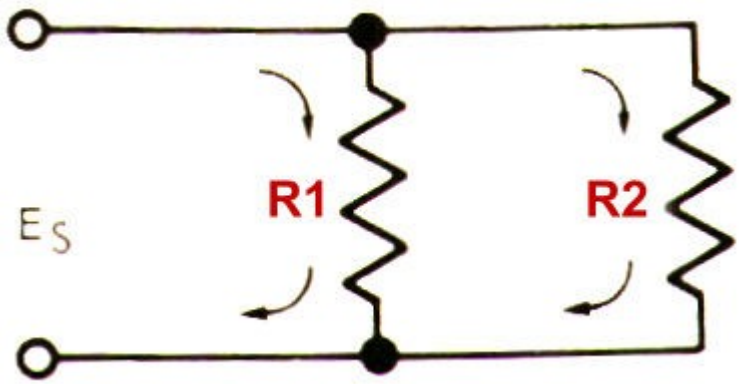
http://www.xtronics.com/Kits/rcode.htm, http://aloha-mall.com/a-1/pcresist.htm, http://rlan.net/NostalgiaAir/RMACodes/reschart.htm, http://www.testeq.com/charts/resclr.html, http://www.drbob.net/rcode.html , genel görünümlü direnç tablosu:

Devrede bulunan elemanlar üzerinden geçen akım ve oluşan gerilim elemanların bağlantı şekillerine göre ikiye ayrılabilir;

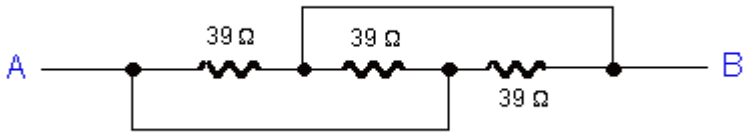
Seri Bağlama: Elemanlar üzerinden akım geçerken bir sırayı takip ediyprmuş gibi önce birinden sonrada diğerinden geçerek gider. Akımlar sabit Gerilimler farklıdır. Örnek şekil aşağıdadır.



Paralel Bağlama: Elemanlar ard arda değilde yan yan bağlanmıştır, akım aynı anda ikisinden birden geçebilir. Gerilimler aynı Akımlar farklıdır. Örnek şema aşağıdadır.



Örnek Soru: Sekilde gorulen devrenin toplam direnc degeri nedir?



Cevap:>>>>

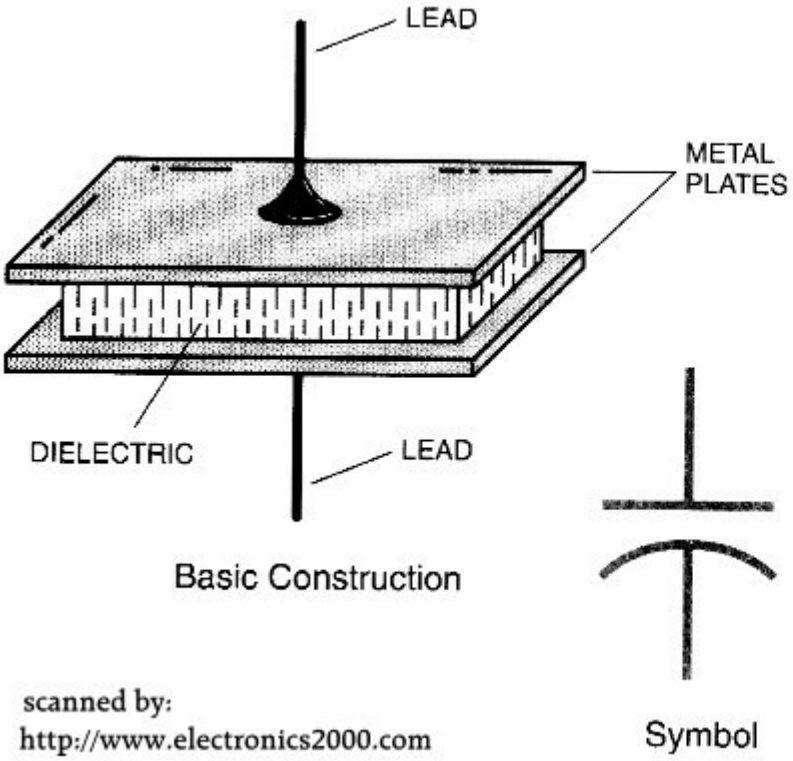
Devrelerde yer alan temel diğer devre elemanları ise;

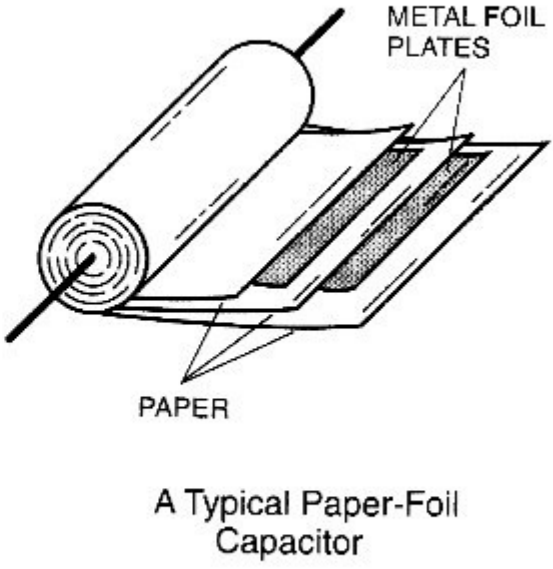
- Direnç (Yukarıda değinmiştik)
- Kapsitans
- Bobin
- Transformatör
- Diyot
- Transistör
- IC² ler yada bilinen adıyla Entegreler

Şimdi kısaca bu elemanları tanıyalım;

KAPASİTANS(KONDANSATÖR): 

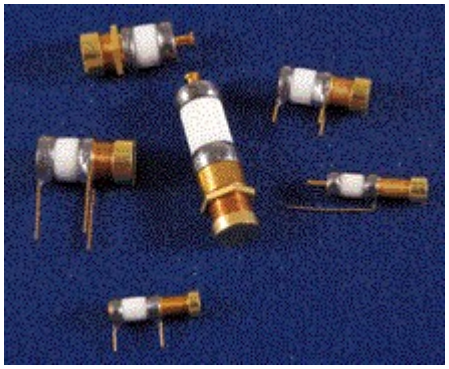
Kapasitans kavramını açıklamak istersek; kısaca küçük bir pile benzetmek yanlış olmaz çünkü iki farklı ucun arasındaki dielektrik(yalıtkan) madde sayesinde iki taraftada birikim oluşur. Oluşan bu birikim uçlar kısadevre edilince kendini tamamlayarak bir akım oluşturur bu olayı gerçekleştiren elemanlara kısaca kondansatör denebilir. Aşağıda solda yer alan şekilde temel yapısını gösteren iyi bir örnek ve sağda bulunan şekilde ise bir kondansatörün şarjı görülmektedir.





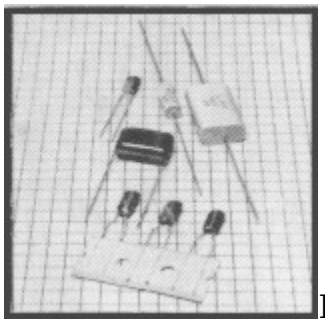


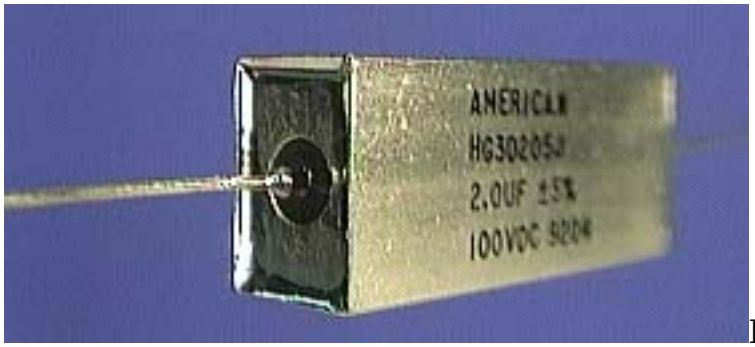
Fiziksel olarak temel yapı bir maddenin üzerinde hapsedebildiği statik elektrik denen bir temel elektrik depolama işlevinin iki metalin etkileşimi ile artırılarak yön ve miktar verilebilmesidir. Miktar kondansatörün iletken ve yalıtkan kısımlarının özelliklerine bağlı olarak hapsedebileceği maximum gücü belirler, Yön bilgisi ise seçilen maddelerden dolayı ortaya çıkan bir kutuplaşmadır. Aşağıda örnek kondansatör çeşitlerini görmektesiniz;





Ayarlanabilir Kondansatörler

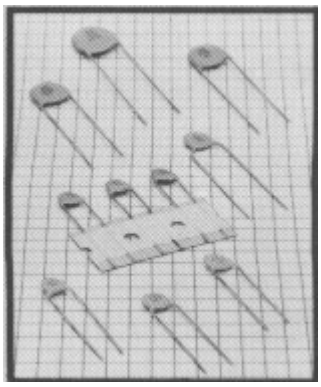

Film Kondansatörleri



Havasız ortamlı Kondansatör.



Kağıt Kondansatörler



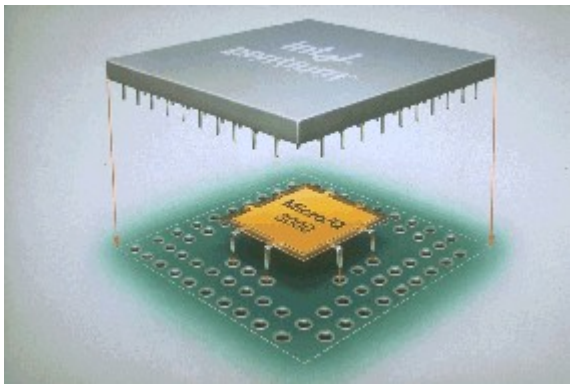
Seramik Kondansatörler



Elektrolitik Kondansatörler



Mica Kondansatörler


Minyatür kondansatörler.


Mikrodalga Kondansatörleri

Kondansatörler görüldüğü gibi çok çeşitli yapılara sahiptirler bunun nedeni ise devrede ihtiyaç duyulaçak özelliklere göre ihtiyacın karşılanmasıdır. Örneğin mica ve seramik kondansatörler ufak kapasitelerde, elektrolitik ise daha büyük kapasitelerde kullanılmaktadır. Tabiki herşey iki metalin bir yalıtkanla birbirinden ayrılması olayı değildir, o yüzden çeşitlilik çok fazladır.

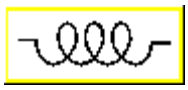
Kondansatörlerin birimi FARAT dır, C ile gösterilir.

i = çekilen akım,**v** = gerilim düşümü,**t** = uygulanan süre ise

$$C = i / (v/t)$$ dir.

(1 milifarad 1/1000 yani bin farada eşittir.1 microfarad 1/1,000,000 yani kısaca 1 milyon farada eşittir.1 picofarad (1/1,000,000 da 1/1,000,000 farada eşittir))

Dikkat edilmesi gereken noktalardan en önemlisi devrede oluşacak kapasitans etkisinin AC açıdan incelenmesi gerektiğidir.

BOBİN: 

Devrelerde bulunan akım yollarının hepsi genel açıdan birer bobin görevi yapmaktadır. Bobinin bu yollardan farklı olan yanı uzunluk ve kendi üzerine olan etkisidir. İlk elektrik akımı bulan insanoğlu uzun bir teli metal parça üzerine sararak akım geçirdiğinde metalin mıklatıs görevi yaptığını ve akımın yönüne göre metalin uçlarında NS kutuplarının oluştuğunu bulmuştur. Bulan kişi yine Farday(1791-1867)'dır. Kısacası bobin bir iletkenin üzerinden geçen akımı mağnetik alan çizgilerine çevirerek yapısal olarak enerji dönüşümünü gerçekleştirmiştir.

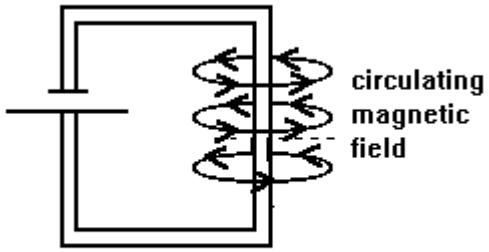
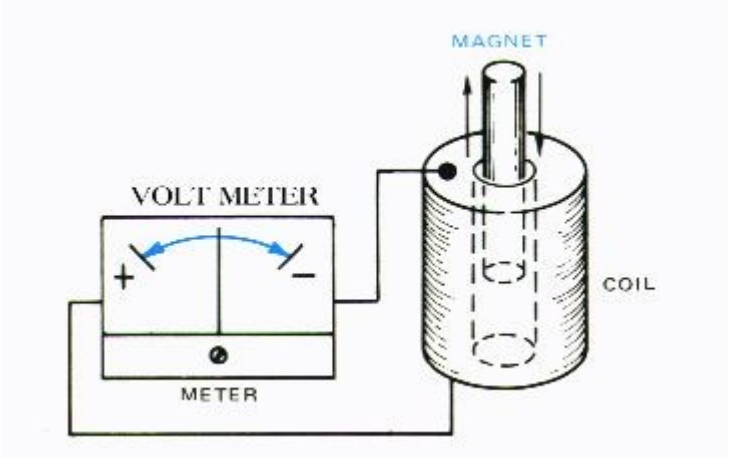


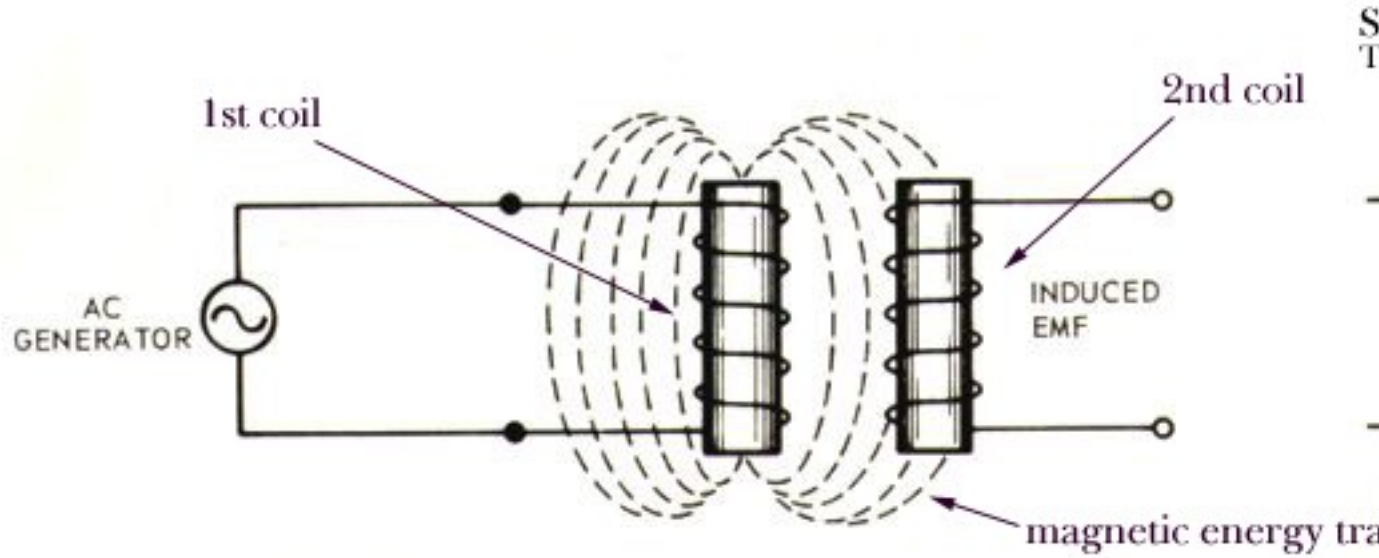
Figure 5



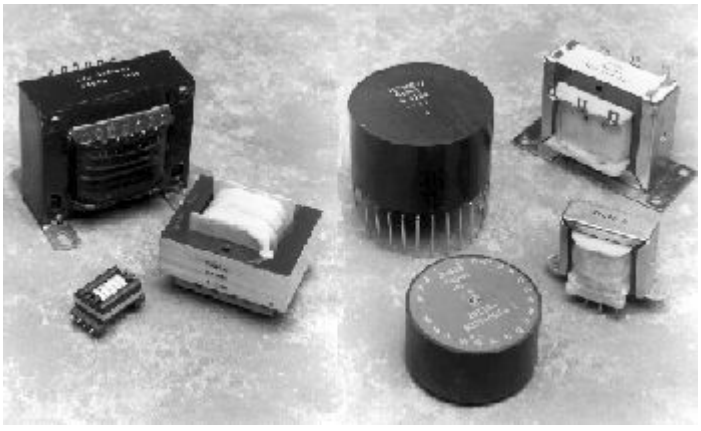
Tersi durumundada yani bir mağnetik cisim mağnetik alan çizgileri bobini kesecek şekilde hareketlendirilirse bobin üzerinde bir akım oluşur işte bu temel akımdır. Normal devre içi kullanımında oluşan bu mağnetik çizgiler farklı şekillerde sarıldığında kendisi üzerine ters mağnetik alan kuvveti uygulayarak üzerinden geçen akımı yavaşlatmıştır. Bu sayede gecikmiş bir akım çıkıştan alınabilir. Bu durum yine kondansatörede olduğu gibi AC devrelerde kullanılır. Etrafında bulunan dielektrik madde ve kullanılan iletkenin özelliği bobinin özelliklerini belirler.

TRANSFORMATÖR (Trafolar):

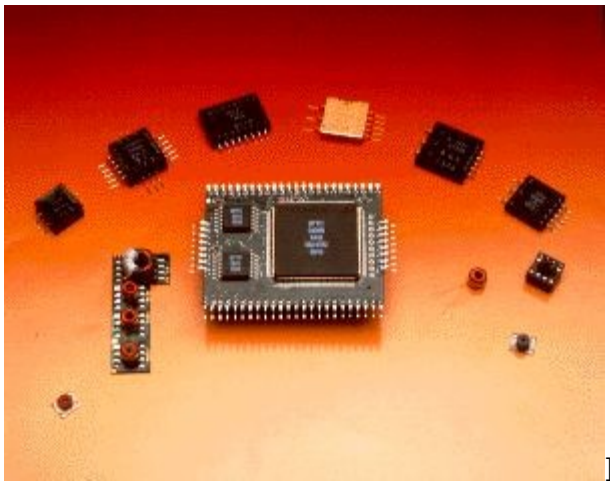
Elektrik akımının herzaman üretildiği büyüklükte kullanılamadığı gerektiğinden voltaj değerini azaltan veya çoğaltan bir devre elemanına ihtiyaç duyulmuştur. Bir direnç gösterilerek akım düşürülmeli ve böylece gerilim azaltılmalıydı ama aktif dirençler bu işi göremezlerdi çünkü istenen değerler tam olarak sabit olmamakla birlikte güç tüketimi düşüktü. Kondansatörler ise ihtiyaca uyacak özellikli değildi sadece depolar ve geri verirlerdi ama Bobin işimizi görebilecekgibiyidi.

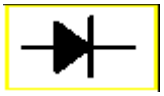


Yukarıda görüldüğü gibi birbirlerine elektriksel teması bulunmayan iki demir kütle üzerine sarılmış olan bobinler arasında sarım sayısı vede demir kütlerin özelliklerine bağlı olarak akım transferi oluşmaktadır. Aşağıda temel iki tip transformatörler görülmektedir;

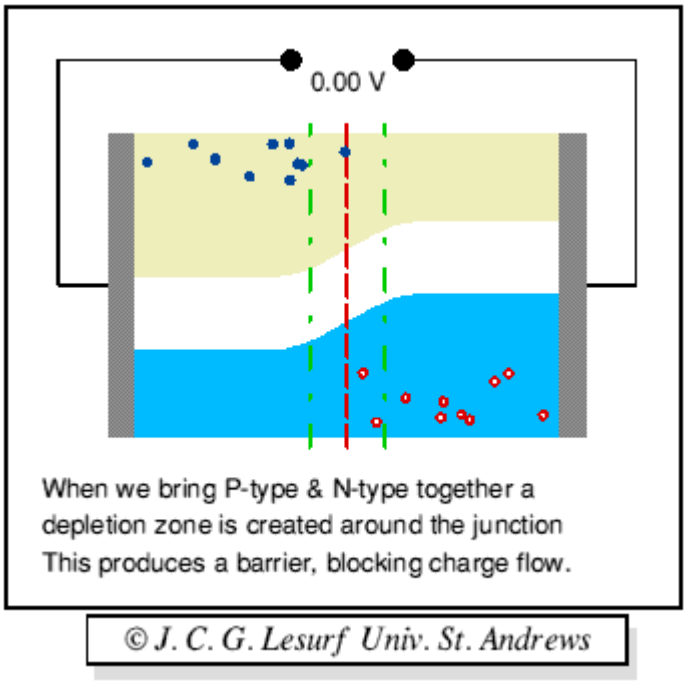


Güç Transformatörleri


Küçük çaplı transformatörler

diyot: 

Eketrik akımı normal iletkernler kullanılarak birçok elaman geliştirilmiştir fakat asıl gerekli olan karar verme yetisine sahip olan bir elektriksel elemandır. İşte buna ihtiyaç duyan insanoğlu dah önce keşfettiği temel atom bilgilerinin üzerine iki farklı simge kurdu, N ve P diye adlandırıldı. N tipi madde atomlarında bulunan elektronlarının sayısı 5 olan atomlardan oluşuyordu bunla genel vericiydiler, P tipi madde ise en dış yörüngesinde 3 elektron bulunan atomlardan oluşuyordu gelen alıcıydılar. Ardarda konan bu iki madde üzerinden geçen akım elektronlar açısından N den P'ye olacak şekilde geçebiliyorlardı. Tersinde ise yani P den N'e geçiş belirli bir sınır içerisinde imkansızdır. Bu elamana diyot olarak tanımlanmıştır. İki ucu olan elemanın artı olan ucuna Anot, eksi olan diğer uca ise Katot denilmektedir. Akım Anottan Katota doğru yol alır.(+ dan - ye tabirinde)

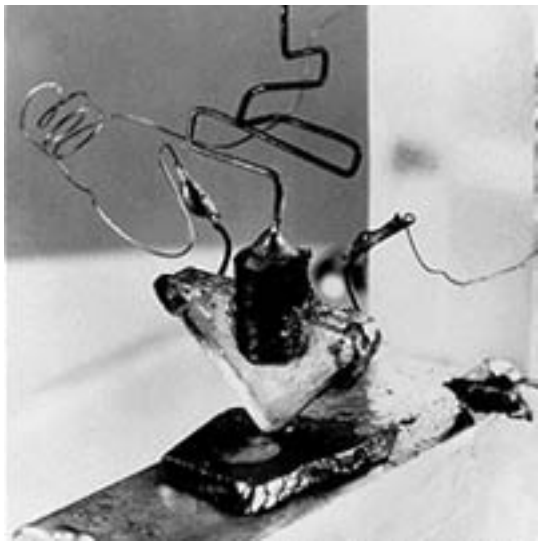




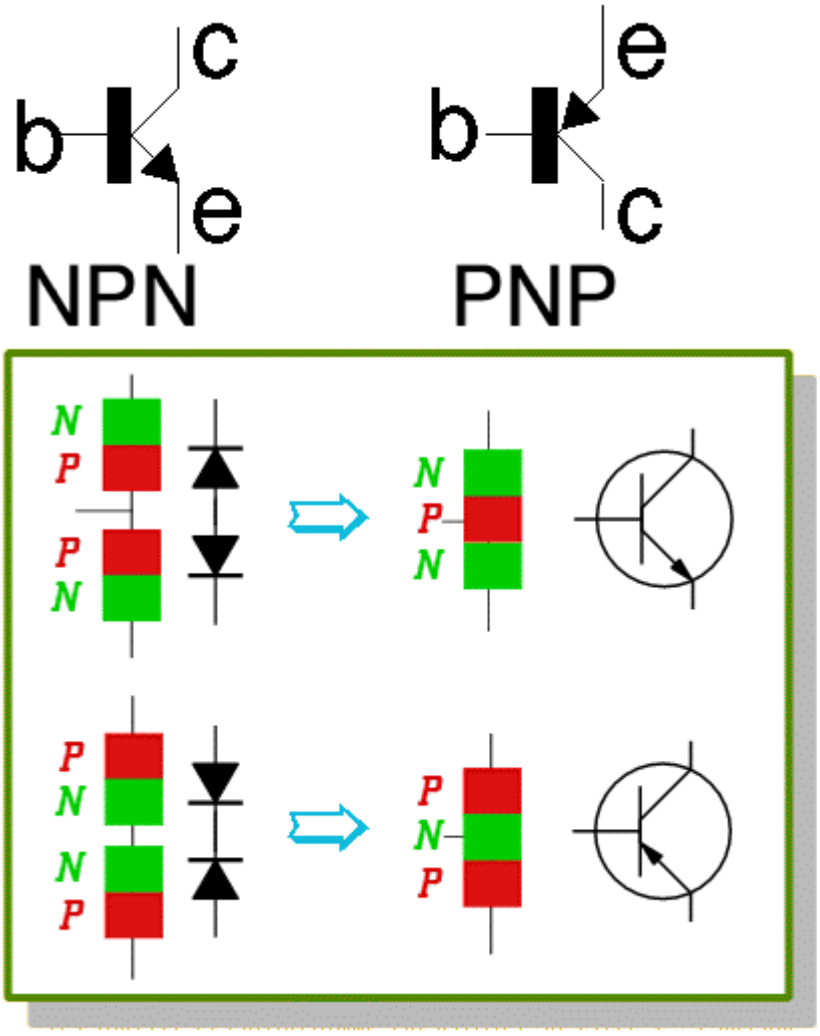
Peki bu sınırlar nelerdir diye bir tabiri açıklarsak, diyodun yapıldığı maddelerin ileri ve geri kutuplanma diye tabir edilen kutuplanması esnasında içindeki yalıtkan zar açılmak için yani ileri kutuplanmada iletime geçmek için belirlibir sabit gerilime ihtiyaç duyar bu gerilim kullanılan maddeye göre değişmektedir. Geri kutuplamada ise zar aktif durumunu koruyarak belirli bir voltaja kadar iletimi sağlamaz yine bu voltaj kullanılam maddelere göre değişmektedir. Encok kullanılan maddeler silisyum ve Germanyumdur.

TRANSİSTÖR:

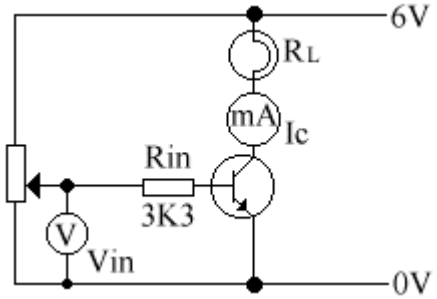
Transistör ilk olarak 1947 AT&T's Bell Labaratuarında John Bardeen, Walter Brattain ve William Shockley tarafından geliştirilidi. Amaç Yarıiletken yani istenildiğinde akımı geçirecek istenildiğinde ise yapı akım geçişini engelliyerek yapısal bir kapı görevi görecekti. Silisyum ve Germanyum yarıiletlenleri kullanılarak üç kutuplu bir yapı oluşturuldu(iki yana Emitör ve Kollektör orta uca ise Beyz denildi). Yapı orta ucundan aldığı gerilimle diğer iki karşılıklı uç arasındaki bağlantıyı kuruyordu. Yapılan bu elektronik kompanend sayesinde günümüz yarı iletken teknolojinin temeli atılmıştır. Şu anda intel firmasının yapdığı mikroişlemciler içerisinde yaklaşık olarak 5 milyar civarında transistör bulunmaktadır. Aşşağıda ilk protipi görmektesiniz:

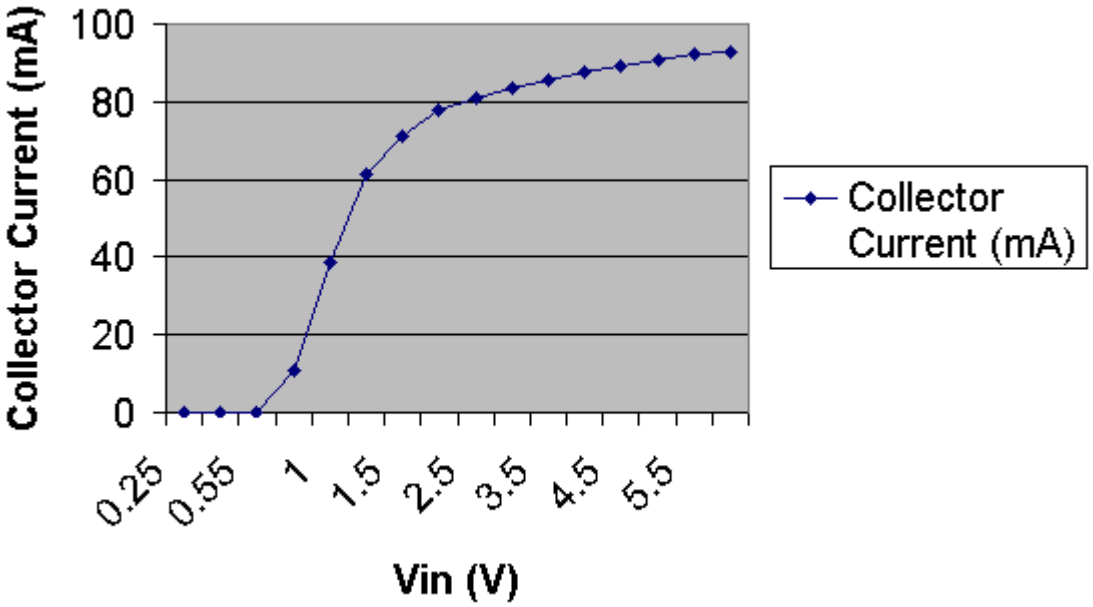


Transistör yapısal olarak Silisyum ve Germanyum dizilişine göre NPN ve PNP olarak ikiye ayrılmıştır, yapısal bu ayrım transistörlerin akım geçirme yönünü etkilemiştir. Aşşağıda genel şemaları ile diyotsal tabirlerini görmektesiniz....

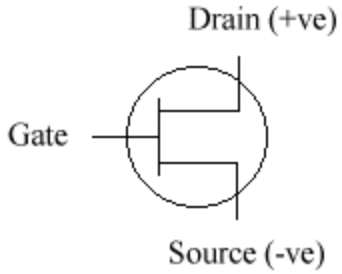


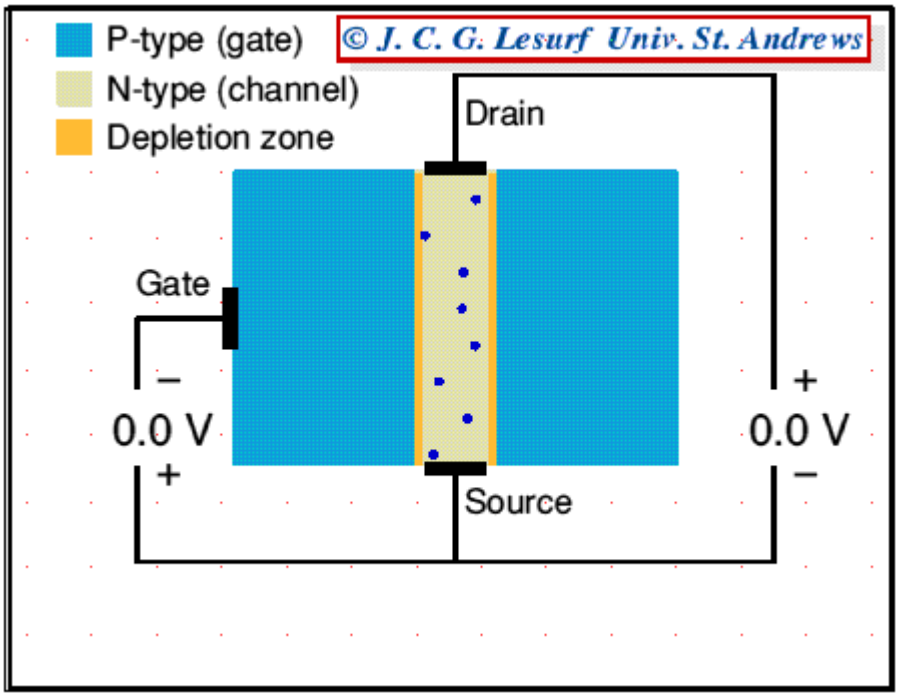
Kullanımı oldukça kolay görülen bu anahtarlama elemanının yapısından kaynaklanan bazı kısıtlama ve zararlarıda yer almaktadır fakat bu şartlar genel kullanım ve özel kullanım örnekleriyle eşitlenen transistörler ile aşılmıştır. Aşşağıda görülen örnek devrede beyze verilen gerilimin oranıyla kollektörden emitöre olan geçişi ve bu geçişin grafiğini görebilirsiniz. Kullanılan transistör NPN dir.

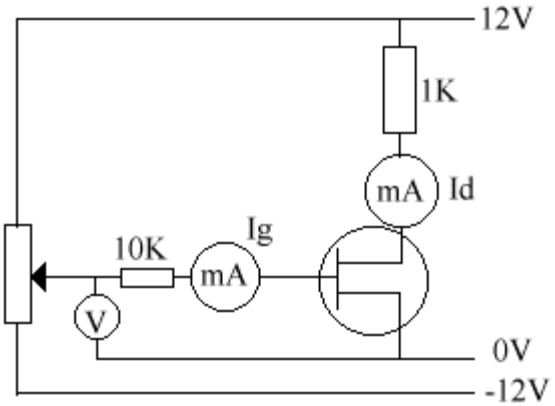


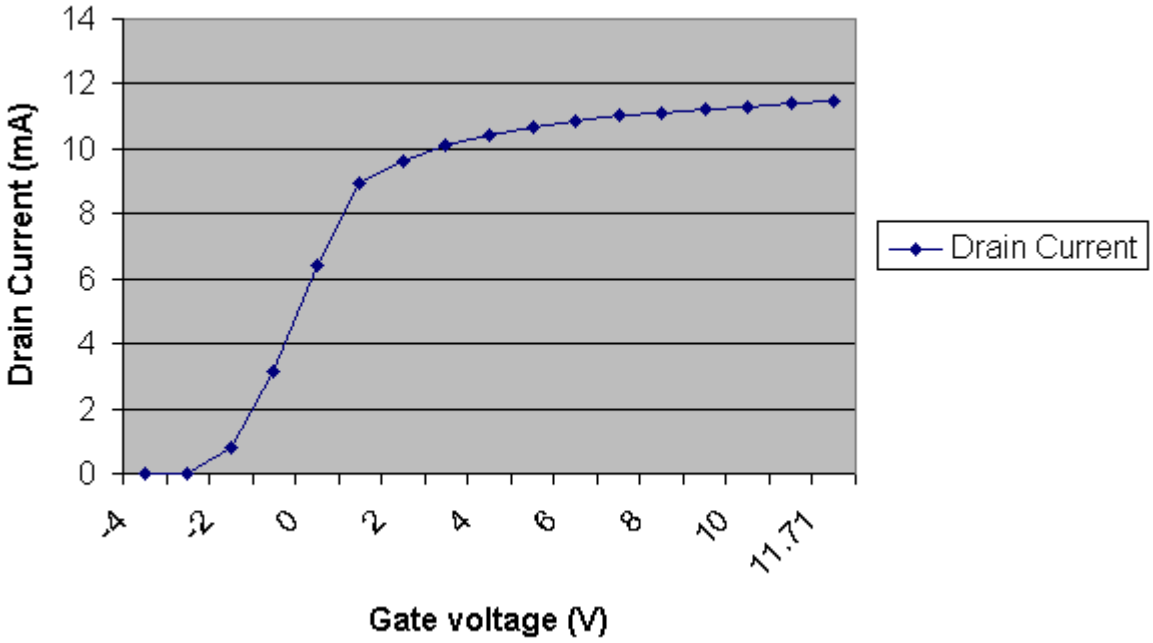


Yapsıal olarak mükemmelleştirilmek istenen transistör araştırmalarıyla Fet ve Mosfet diye tabir edilen yeni yarı ileten anahtarlama elemanları yapılmıştır. Bunlar transistörün eksik olarak görülen yönlerini kapatmak için geliştirilmiştir. Örneğin Fet yüksek hızlı ve kararlıklı bir eleman olmasına rağmen düşük akımlarda çalışarak bir yapı teşkil etmiştir.Fet'de transistörün üç ucu olan beyz,emitör ve kollektör yerine Gate,Source ve Drain uçları yer almıştır. Aşşağıda genel olarak FET'in yapısı örnek bir devre ilegrafiksel olarak tutumu görülmektedir.


Drain (+ve)
Gate
Source (-ve)


P-type (gate)
N-type (channel)
Depletion zone
© J. C. G. Lesurf Univ. St. Andrews
Drain
Gate
–
0.0 V
+
+
0.0 V
–
Source





Mosfet ise yine yüksek hızlı ve yüksek akımlarda çalışabilen bir transistör türüdür.

Transistör seçmek için kullanılabilecek bir sayfa: http://www.tanker.se/lidstrom/trans.htm

---

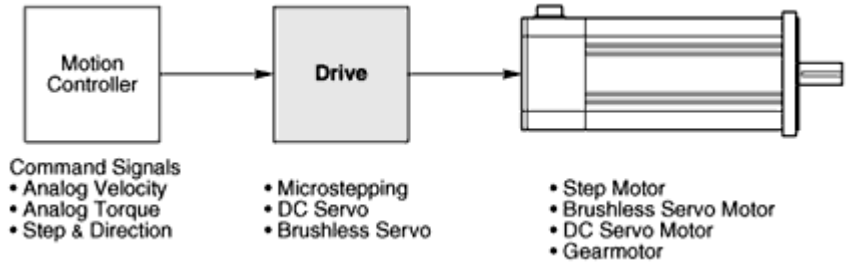
[5.2] Güç Devreleri:

Elektronikte mekanikte olduğu gibi küçük olan bir olguyla büyük olan olguyu kontrol etmek gerekmektedir. Örneğin ATX bilgisayar kasalarında bilgisayar üzerideki her alet maximum 24V çalışmasına karşın bilgisayar mikro işlemcisi 220V güç kaynağını bir yarıiletken ile kontrol edebilmektedir. Bu yapıda bizim için önemli olan minimum güçle maximum güçü kayıpsız olarak hızlı ve kesin bir şekilde kontrol etmektir. Robotic alanında ise güç devreleri özellikle lojik ibarelerden oluşan hareket ifadelerini motora anlatmak için kullanılmaktadır. Bu işlemler

için özel güç katları vede entegreler tasarlanmaktadır bunlardan bazıları L293, L297, L297D, UMB 2003 gibi 5-30 volt arasında çalışabilen güç katlarıdır.

[5.3] Ana Hatlarıya Sürücü Devreleri:

Güç devreleriyle kazanılan lojiksel kontrolü doğru ve manalı bir hareket çevirmek için devreye sürücü devreler girmektedir. Sürem asıl anlam olarak yön verme mansaına gelmekle birlikte yapısal olarak atılacak bir sonraki adımı belirleme işlemidir. Roboti'de motor sürücü devreleri motorun yerine ve bir sonraki pozisyonuna geçişini belirlemektedir. Aşşağıda genel olarak bir model görülmektedir:



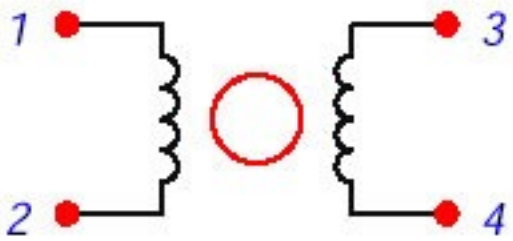
Sürücü devrelerinde genel olarak motor kontrolü yapacağımızdan gelin şimdi elektriksel motor nedir onu inceliyelim:

[5.4] Elektriksel Motorlar:

Elektriksel Motor kavramı elektriksel alanın yani magnetik alan kuvvetinin elektrik akımıyla oluşturulmasıyla ortaya atılmıştır. Bir demir nüve üzerine sarılan çok spirli bobin telin üzerinden akım geçirildiğinde demir nüve kutup başları olan N S kutuplarını oluşturarak ceçresindeki zıt kutupları veya metal cisimleri çekmektedir. Bu teoriden çıkan bilimadamları çekme itme kuvvetini ardarda koyarak ve bu hareketi dairesel harekete çevirerek ilk elektriksel motoru yapmışlardır. Bu motorlar kendi aralarında ana hatlarıyla üç guruba ayrılırlar:

[5.4.1] DC Motorlar:

Mantık olarak bobin üzerinden geçen akımın sonucunda oluşturduğu mağnetik kaçaklar sayesinde oluşturduğu kutuplaşmayı ileri ve geri yönlü olarak kullanarak yani zıt kutupların çekmesi vede aynı kutupların birbirin itmesi prensibinin dairesel harekete dönüştürlmesini baz alınan en basit yapıdır. Diğer motorların tamamı bu mantık üzerine kurulmuştur.



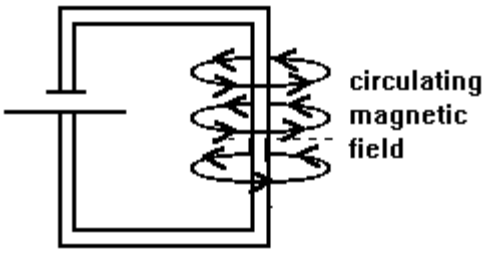
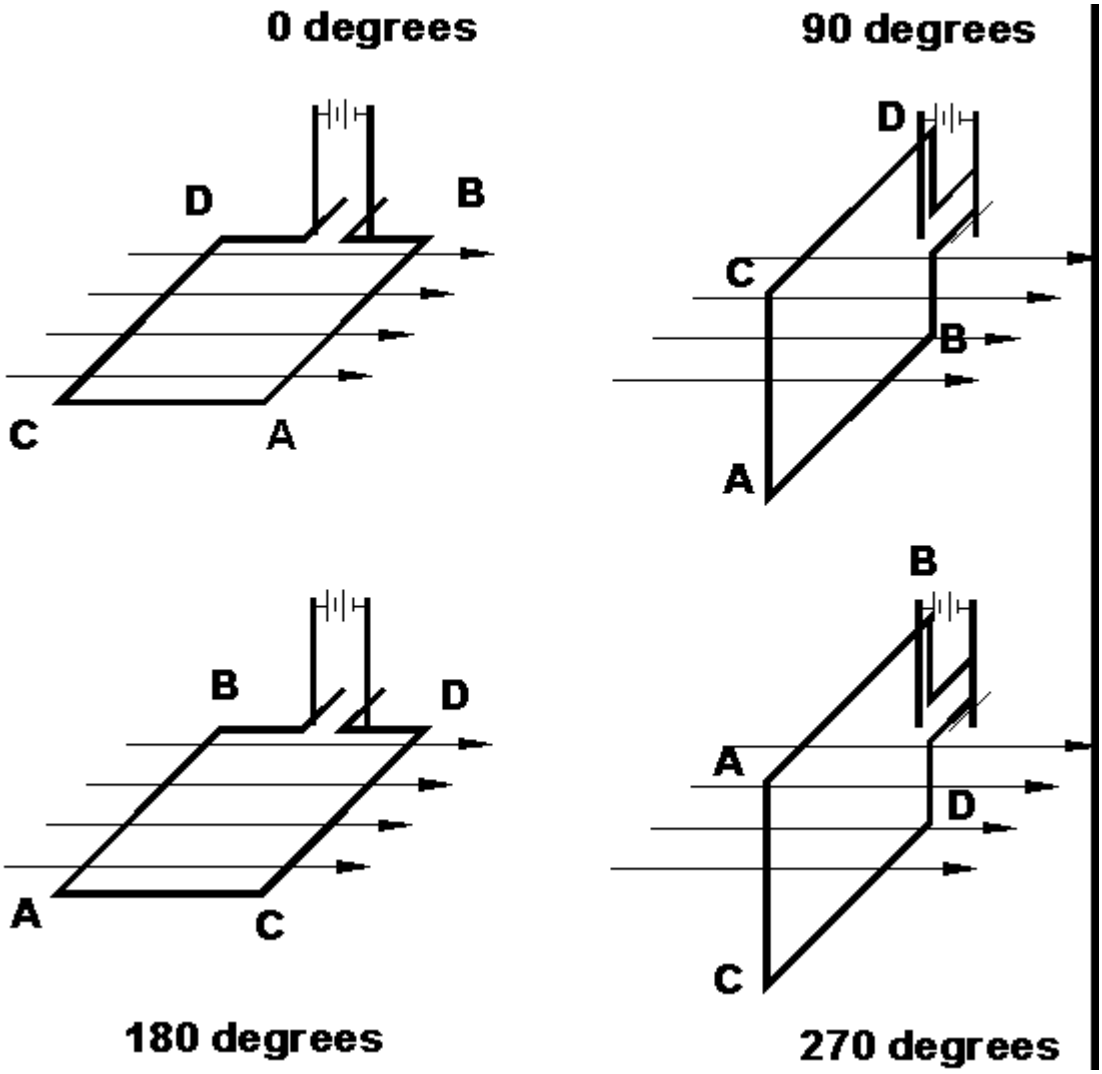
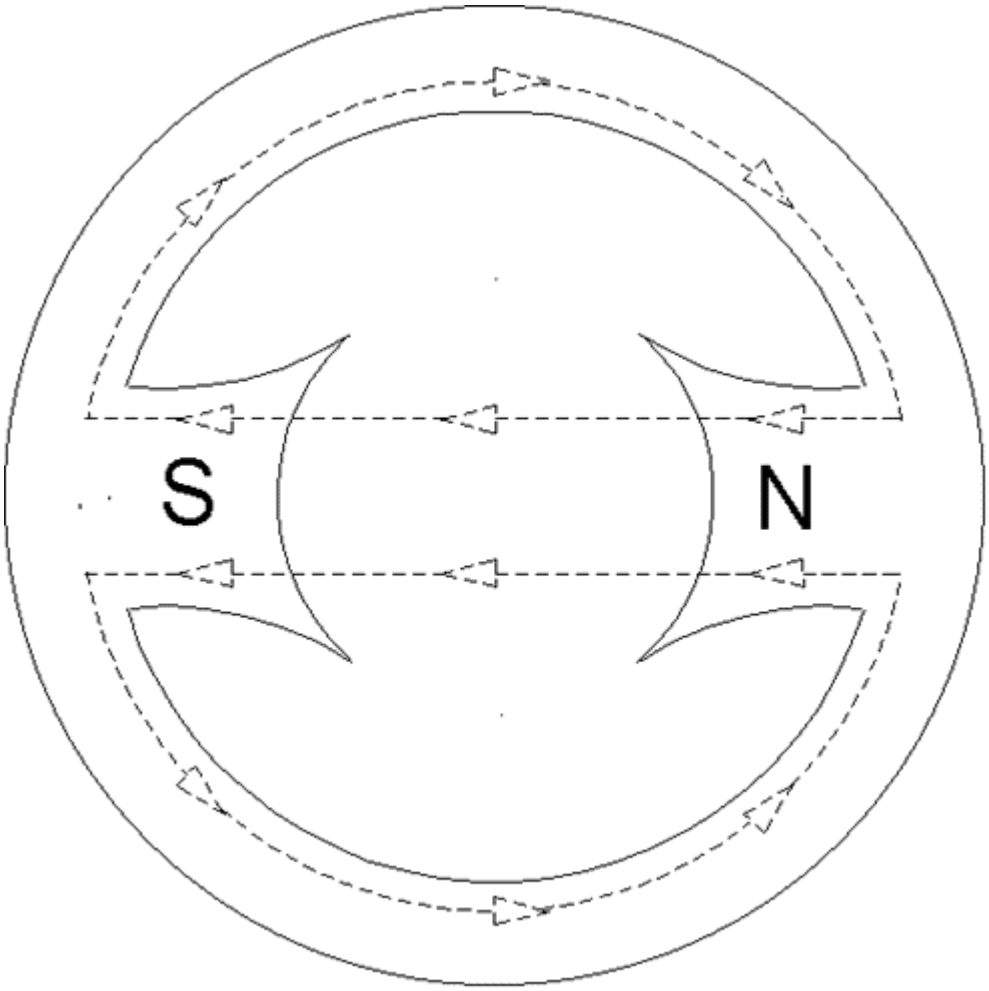
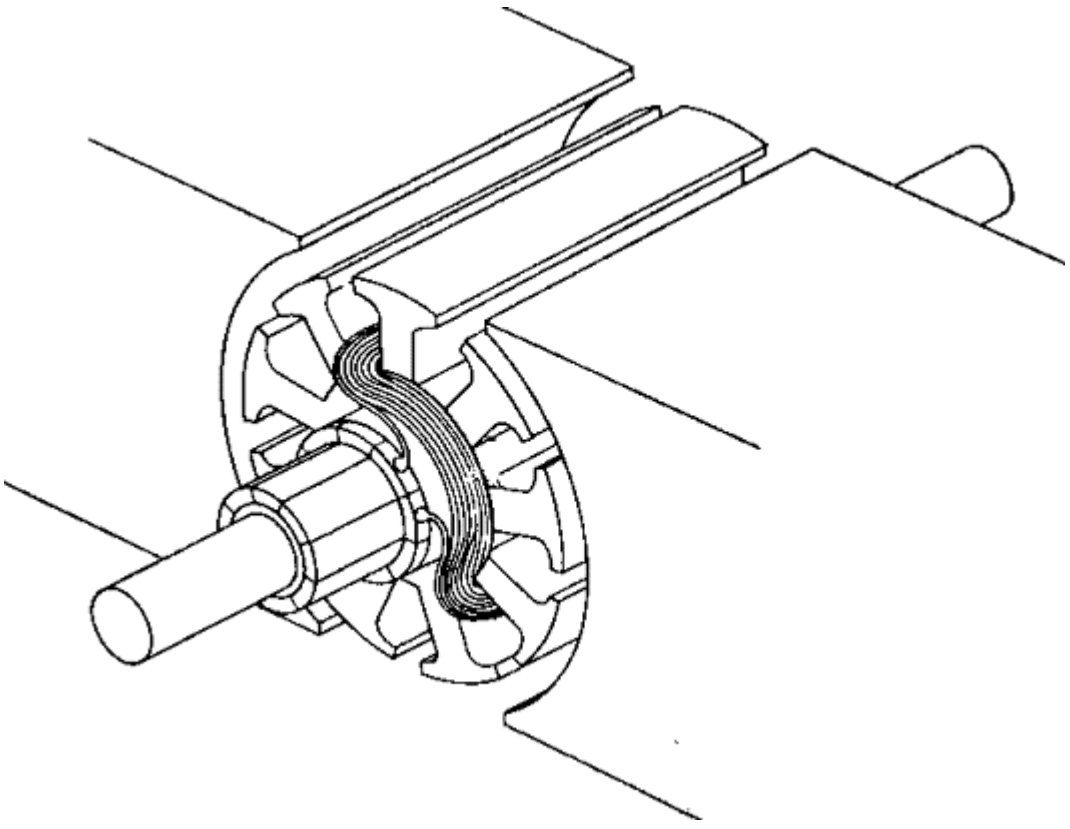


Figure 5

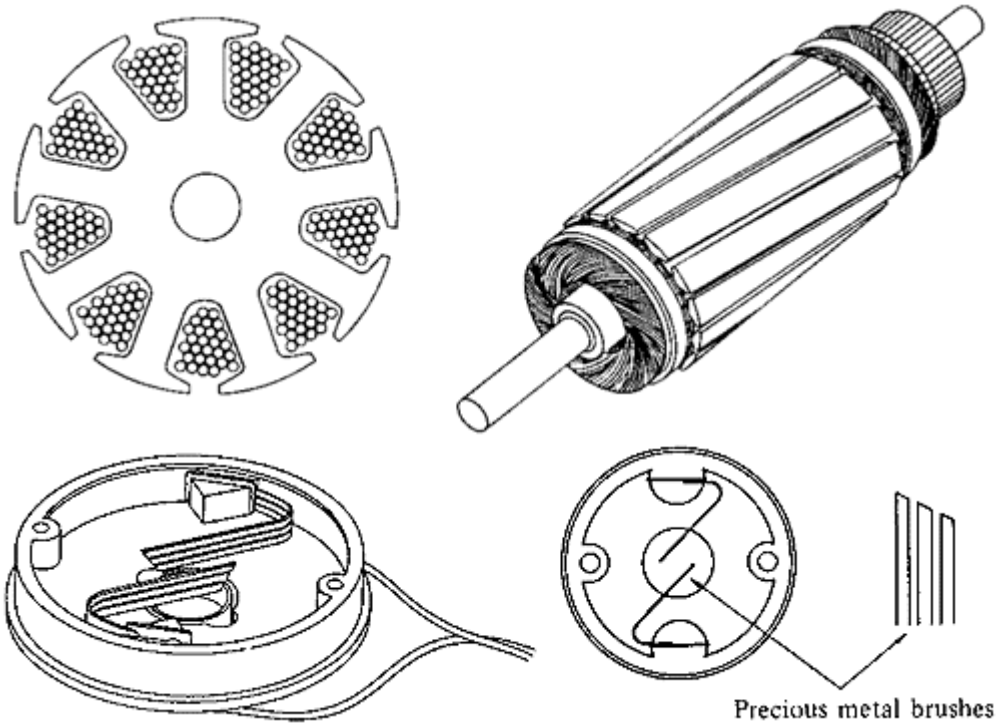


Şekillerde ve akımın yönünün iletkenin şeklinin Magnetik alanı ve de kuvvet yönünü nasıl etkilediğni görmektesiniz... Aşağıda ise bu hareketin dairesel harekete dönüştürülmesini görebilmekteyiz:

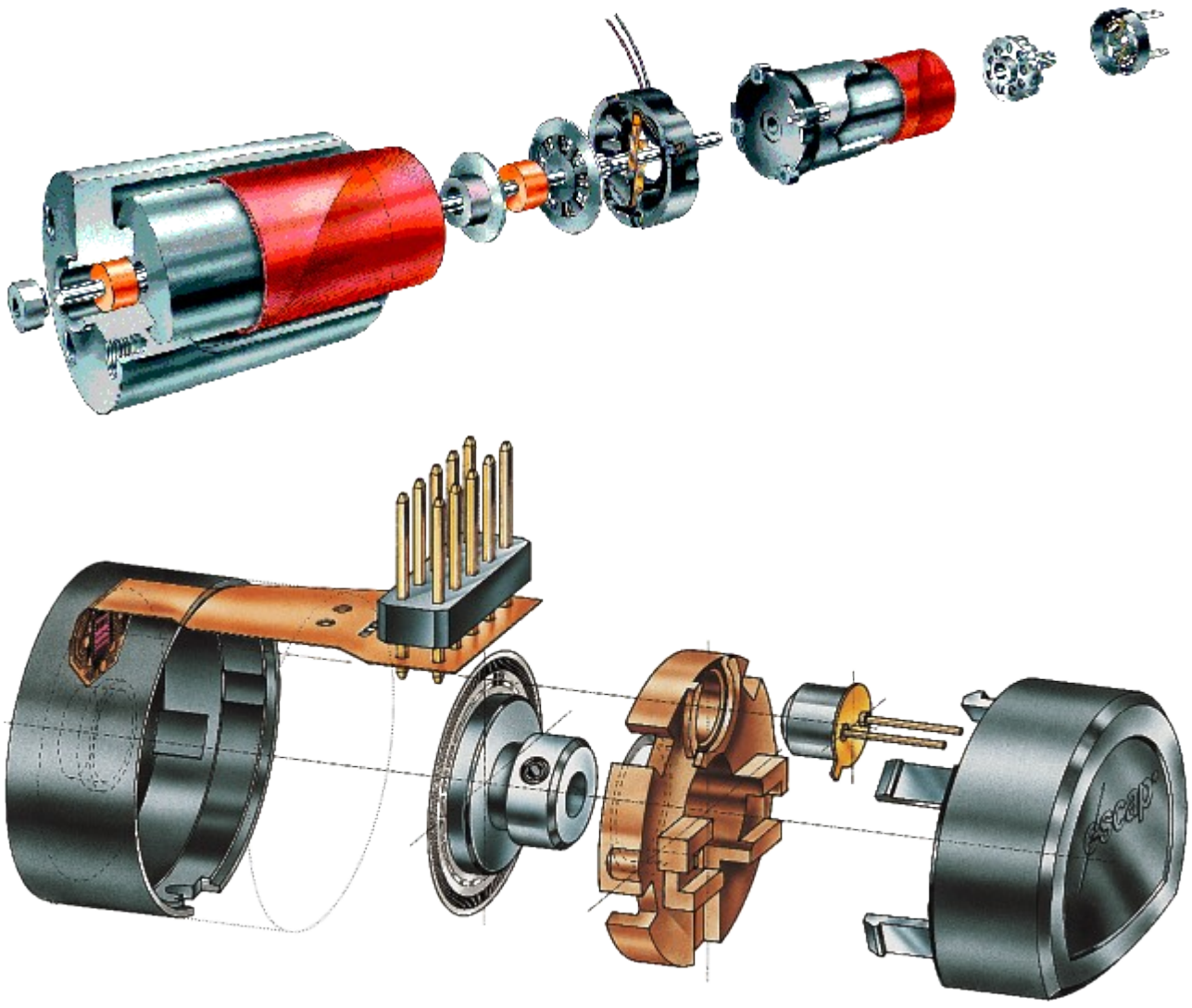

S
N



Mıklatıs mantığında oluşan N S Kutuplaşması kullanılarak N kutbu tarafına S kutbu gelecek vede S kutbu tarafınada N kutbu gelecek şekilde akımın yönü değiştirilerek hareket ivmesi kazandırılmaktadır...

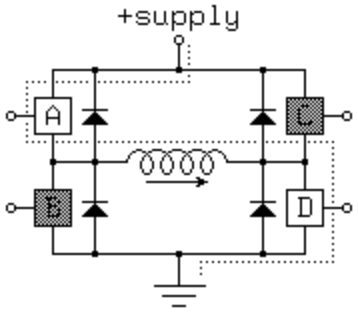


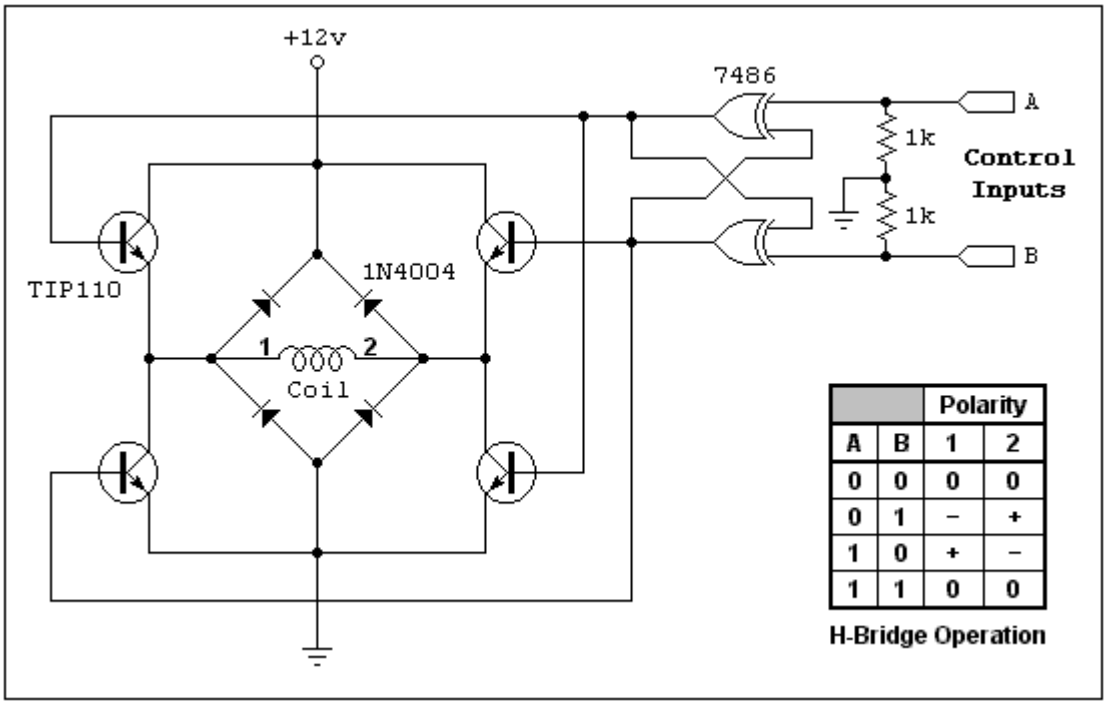
Ayrıca yapıda bobin sarımı vede kutup başlarının akımın yönünün değişim için kullanımı görülmektedir. Aşşağıda ise piyasada satılan temel dc motor iç yapısının blok şekli görülmektedir. İlk şekilde ana şaft üzerine sarılmış çapraz sargılı ince yüksek duyarlılıklı bobin ve çok yakınlaştırılmış dairesel mıklatıs görülmektedir diğer şekilde ise temel bir encoder yapısı görülmektedir.



[5.4.1.1] Sürücü Devresi:

DC motorlarda yapısal olarak akım geldiği sürece motor akımın yönüne bağlı olarak döner fakat akımın siddeti ve sekli motoru doğrudan etkiler. Aşşağıda ilk olarak Wien köprüsüyle akımın yönünü değiştrimeyi vede akımın zamansal bölünmesini ayarlayarakta hızını etkileyebilriz.

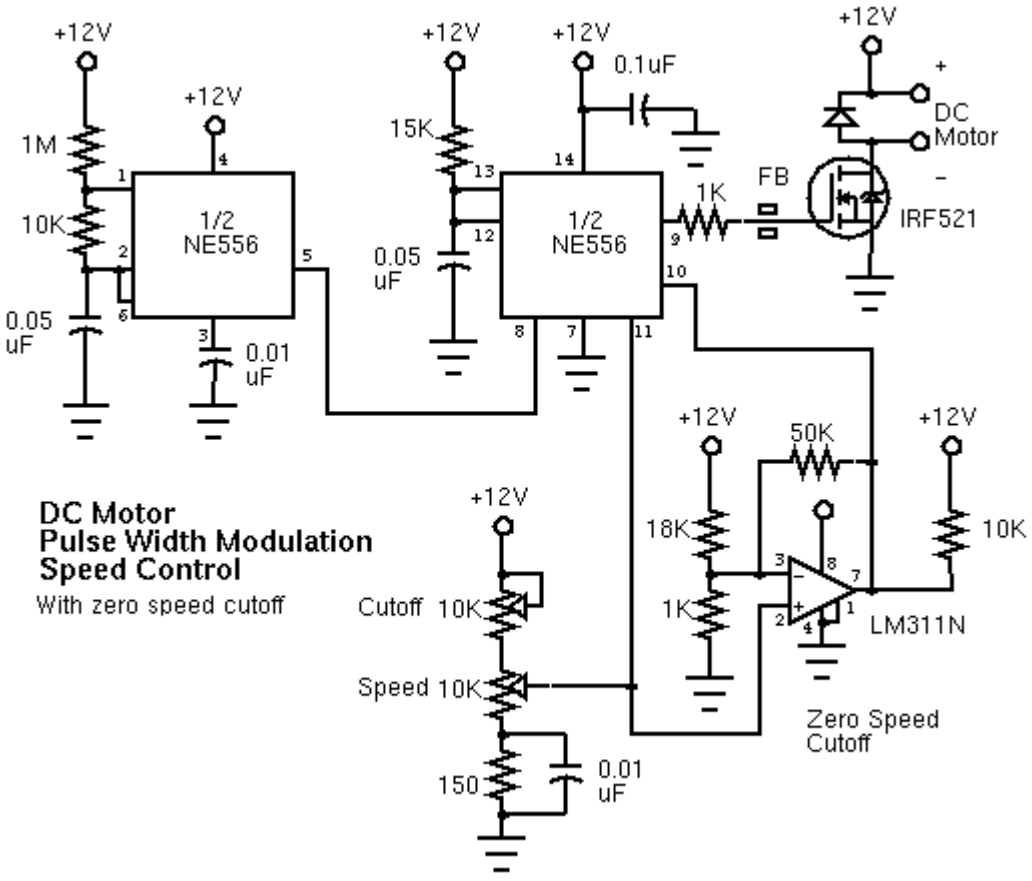




		Polarity	
A	B	1	2
0	0	0	0
0	1	–	+
1	0	+	–
1	1	0	0

H-Bridge Operation

Devrede görülen ABCD uçları aynı tipteki transistör yani anahtarı temsil etmektedir. Anahtarların çapraz açılışına göre motor ileri yada geri tetiklenerek dönüş yönünü değiştrimektedir. Diyotlar ise ters akımlardan transistörleri kısaca besleme devrelerini korumak için konulmuştur.



Yukarıdaki DC hız ayarlayıcısı devre pulse mode yani darbe modlu çalışan bir hız devresidir devrede ana gerilim transistörü tetiklenerek DC motora gerilim verilerek motro hareketlendirilip transistör kesime götürülerek DC motor beslemesi kesilmektedir bu sayede motor tetiklenme aralığına göre hız kazanmaktadır. Tork beklenen yapılar için sakıncalı bir devredir tercih edilmesi gereken akıma dayalı gerilim bölme yada kısıtlama yapılarıdır.

[5.4.1.2] Olumlu Olumsuz Yapıları:

Olumsuz yanları ilk üretilen motorlardan biri olması nedeniyle bayağı çok fakat halen çoğu alanda kullanıldığını düşünürsek yapısal olarak bir imkansızı başarmıştır fakat robotic acısından sıfırn altında bir kontrol alanına sahiptir çift kutup değişimi yüzünden açısal veya konumsal kontrol yapılamamakta emule edilen (micro wave) motor sürücü devreleri ise ağır şartlar altında iflas ettmekte ve uzun ömürlü olmamaktadır.

[5.4.1.3] Piyasa Durumu:

Özellikle uzak doğuda üretim potansiyeli bayağı yüksek olan bir motor türü almanyada bir iki firma DC motorlar ve Dişlikutularını birleştirerek amatör veya hassasiyet ve güç gerektirmeyen işlemlerde kullanabilinecek motorlar üretmektedirler. Ulaşabileceğiniz internet adresleri ve yuvarlak fiyatlarına gelince:

www.asiansources.com yaklaşık 20 kadar firma var motorlar voltaj ve yapılarına göre 5-30$ arasına fiyatlara sahipler.

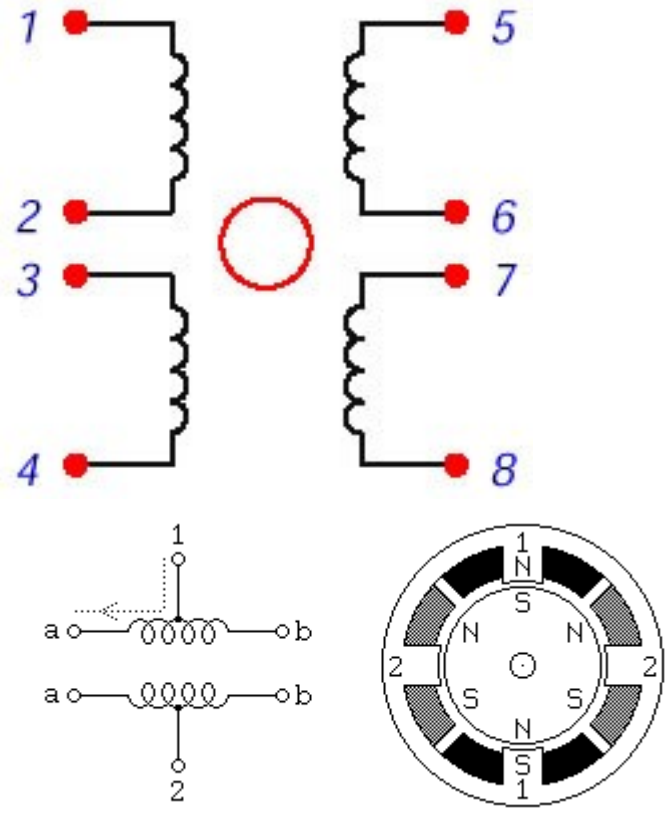
www.minimotor.ch Çok kaliteli motorlar fakat fiyatları çok yüksek 50$ a çok basit bir motoru satıyorlar ama sagı kaliteleri ve torksal güçleri mükemmel

www.allcorp.com Fiyatlar ve mallar bayağı iyi fakat kalite düşük.

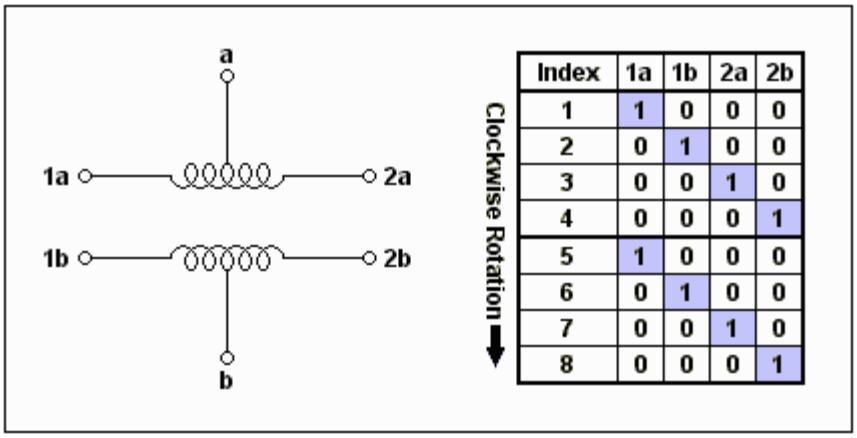
www.alltronics.com En iyi yerlerden biri.

[5.4.2] Adım Motorları:

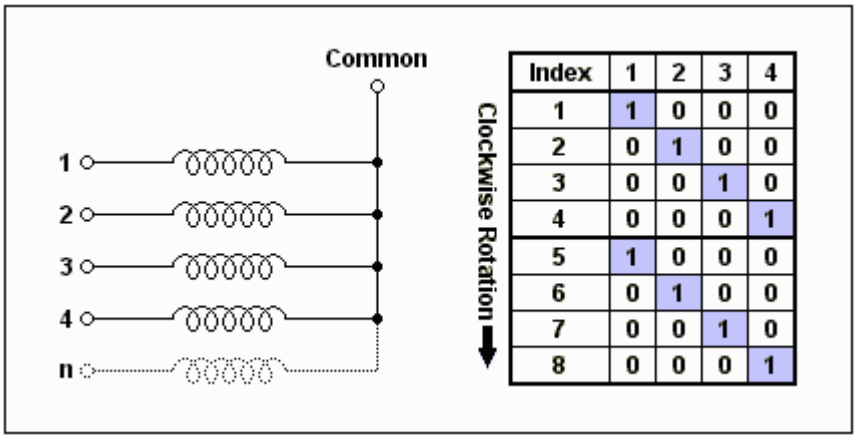
Yapısal açıdan DC motorla çok benzemekle beraber bobin sargı sayısıyla sarım şekli değişmektedir. Bu sayede motor adım adım ilerletilebilmekte ve herhangi bir adımda durdurulabilmektedir. Aşşağıda temel olarak bir Adım Motorunun yapısını görülmektedir:



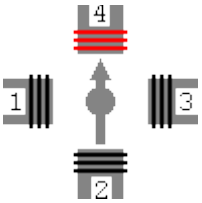
Görüldüğü gibi akımın geçebileceği birden fazla boboinsargıları vardır bu sargıların sayısı artırıldıkça adım atabilme yetisi yani açısal dönüşü o kadar küçülür. örneğin 4 ayrı sargılı fakat 100 parçalı bir adım motoru bir devirede 100*4 = 400 adım atabilir açısal olarak ise 360/400 = 0,9 derecedir. Aşşağıda bir adım motorunun çalışması canlandırılmıştır.

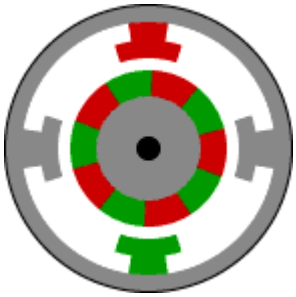


Index	1a	1b	2a	2b
1	1	0	0	0
2	0	1	0	0
3	0	0	1	0
4	0	0	0	1
5	1	0	0	0
6	0	1	0	0
7	0	0	1	0
8	0	0	0	1



Index	1	2	3	4
1	1	0	0	0
2	0	1	0	0
3	0	0	1	0
4	0	0	0	1
5	1	0	0	0
6	0	1	0	0
7	0	0	1	0
8	0	0	0	1





Adım motorunda aynı anda iki kutbun aktif olamsıyla dönen kısım olan rotor bu iki kutbun ortasında durabilmekdedir.Fakat Rotor üzerinde uygulanan güç iki farklı noktaya zıt yönlerde çekildiğinden tutuş gücü olan tork düşmektedir. Aşşağıda tam, yarım adım şemaları karşılaştırılmalı olarak görülmektedir;

Clockwise Rotation

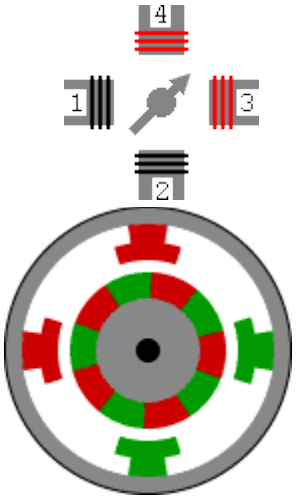
Index	1a	1b	2a	2b
1	1	0	0	1
2	1	1	0	0
3	0	1	1	0
4	0	0	1	1
5	1	0	0	1
6	1	1	0	0
7	0	1	1	0
8	0	0	1	1

Alternate Full Step Sequence
(Provides more torque)

Clockwise Rotation

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

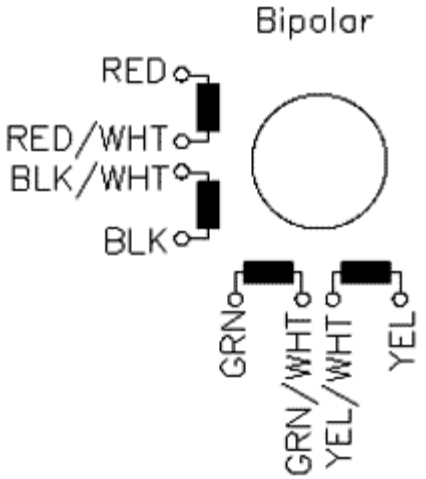
Half Step Sequence

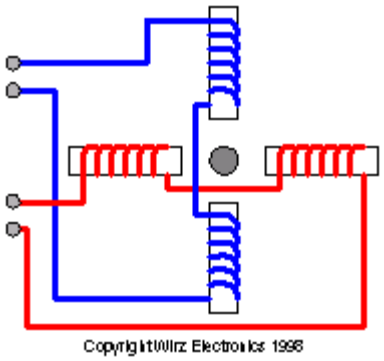


Bu noktadan hareketle adım motrunun atacağı 400 adımı half step (yarım adım) sayesinde ikiye katlıyarak 800 adıma çıkarıyoruz. ayrıca adım açısı 0,9 dereceden dan 0,45 dereceye düşmektedir. Bu bize daha hassas işlem yapabilme gücü getirdiği düşünülsede kayıp olarak bizden step motorun en büyük avantajı olan torksal tutunabilme kabiliyetini çalmaktadır.

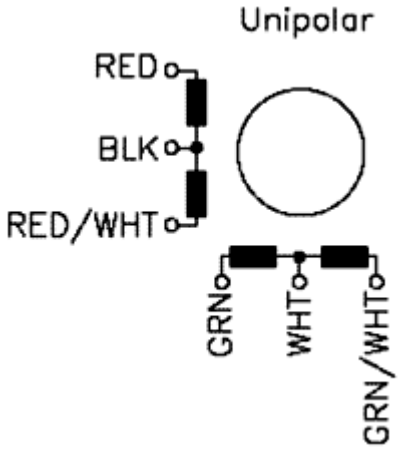
Ayrıca half step den sonra birde mikro step adı altında çalışmalar yapılmaktadır burada ise amaç 2 kutubu ayrı ayrı farklı voltajlarda yükleyerk rotoru konumsal olarak arada tutarak daha hassas adım attırmaktır ama yine güç kaybı herşeyin sonu olmaktadır.

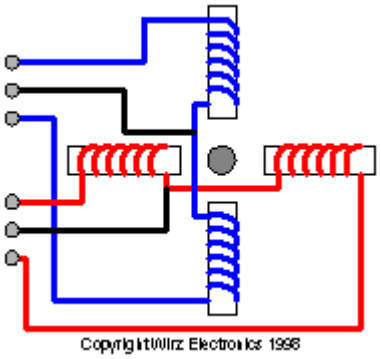
Aşşağıda iki tür olan Bipolar ve Unipolar step motorların yapısal özellikleriyle konumsal kontrol avantajları yer almkatadır:





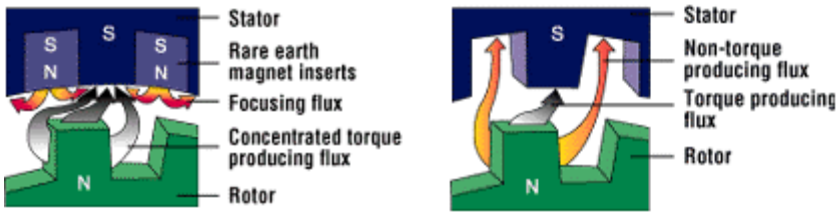
ve





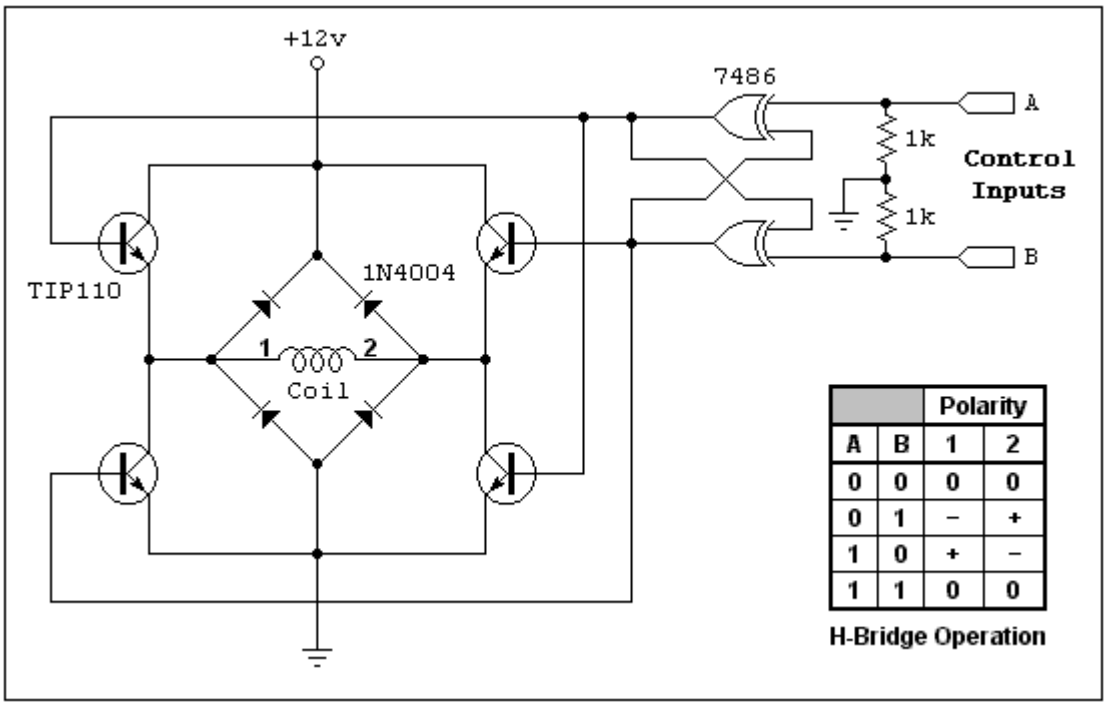
STEP		FULL STEP MODES								HALF STEP			
DIRECTION		1 PHASE EXC				2 PHASE EXC							
CW	CCW	P1	P2	P3	P4	P1	P2	P3	P4	P1	P2	P3	P4
S	8	■				■	■			■			
1	7		■				■	■		■	■		
2	6			■				■	■		■		
3	5				■	■			■		■	■	
4	4	■				■	■					■	
5	3		■				■	■				■	■
6	2			■				■	■				■
7	1				■	■			■	■			■
8	S	■				■	■			■			

Bipolar ve Unipolar adım motorları arasındaki en büyük farklılık bobin srımlarıyla bu boboinlerin baglantı şekilleridir. Aşşağıda ise NEMA standartına uygun bir adım motorunun olması gereken adım aralığı gösteilmekte ve full step desteklenmektedir.

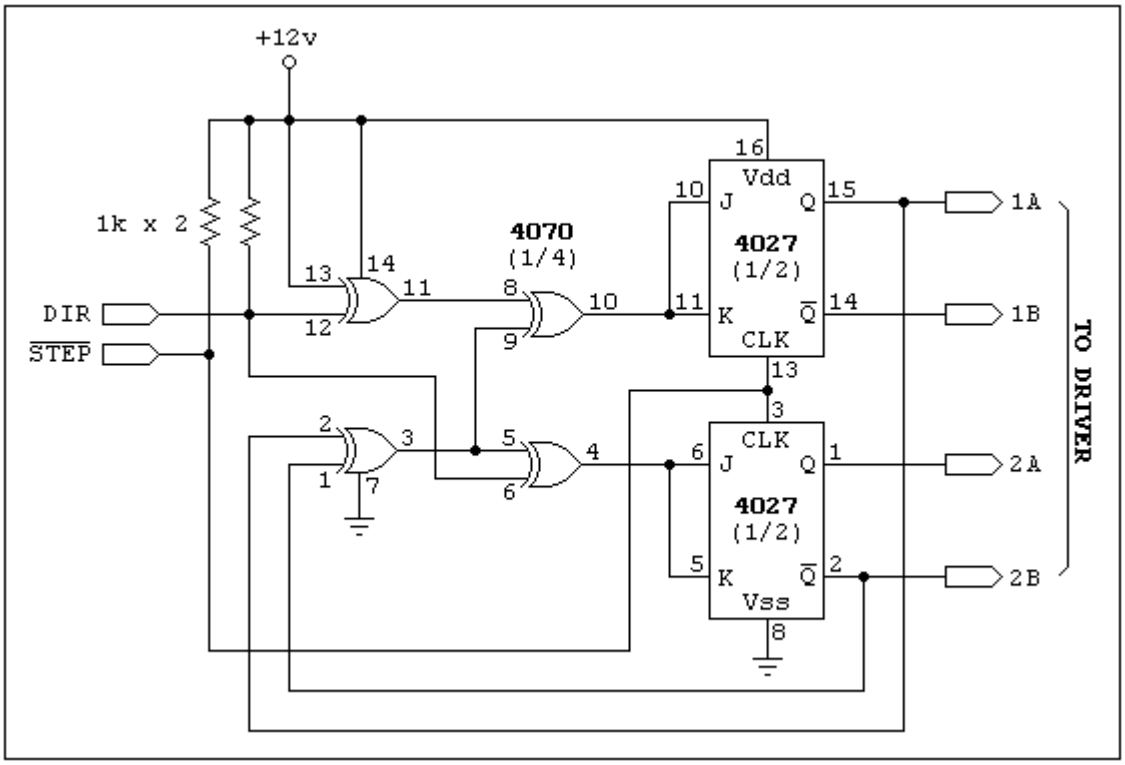


[5.4.2.1] Sürücü Devresi:

Yapılarındaki kolaylıktan dolayı adım motorlarının sürülmesi oldukça kolaydır. Amaç adım motorunun bobin uçlarına adım şemasına göre sırayla tatbik edilmesidir. Uygulanan gerilim aracı bir tanpon katla güç devresi olarak kullanılmaktadır. Aşşağıda örnek devre yer almaktadır;



Yukarıdaki devre ana şema olarak bir step motorun tetiklenmesini içermektedir. Devre DC motor yapısında olduğu gibi 4 ana noktadan farklı kutuplarla tetiklenerek akım yüklenmektedir fakat DC motorda olduğu gibi bu farklılık dönüş yönünü değil bir sonraki adımın yerine karar vermektedir. Lojiksel bir kontrol sağlanmakla birlikte aşşağıda çok daha kolay bir devre göze çarpmaktadır.



Ana adım yapısı XOR kapılarıyla sağlanmaktadır bu yapı iki adet JK flip flopa uygulandığında bobinler için gerekli olan adım atma yapısı 2 lojik uç kontrolüyle sağlanmaktadır. DIR direction yönden gelmektedir bir sonraki adımın yönüne karar verir. STEP ise adıma karşılık gelir bir adım atılmasını söyler sıfırda aktiftir.

[5.4.2.2] Olumlu Olumsuz Yapıları:

Olumsuz yanları hız problemleri var belli bir hızın üstünde adımı kaçırma ve gerileme hatta tepki verme durumuna giriyor. Kararlılığı ısısal olarak çok çabuk değişiyor. Full stepte genel olarak başarılı fakat half step veya mikro stepte etkisini yitiriyor. Hata anında eski pozisyonuna dönmede problem yaşanıyor. Dc motorlardan avantajlı fakat servolara göre biraz HACI MURAT kalıyor....

[5.4.2.3] Piyasa Durumu:

Yine uzak doğuda üretim potansiyeli bayağı yüksek. Dişli kutusu eklenmiş adım motorları bayağı makul ama kaçırma açıları neredeyse 3 dereceyi buluyor. Ulaşabileceğiniz internet adresleri ve yuvarlak fiyatlarına gelince:

www.asiansources.com yaklaşık 10 kadar firma var motorlar kalite ve yapılarına göre 25-80$ arasına fiyatlara sahipler.

www.crouzet.com veya www.yorum-automation.com Çok kaliteli motorlar fakat fiyatları çok yüksek.

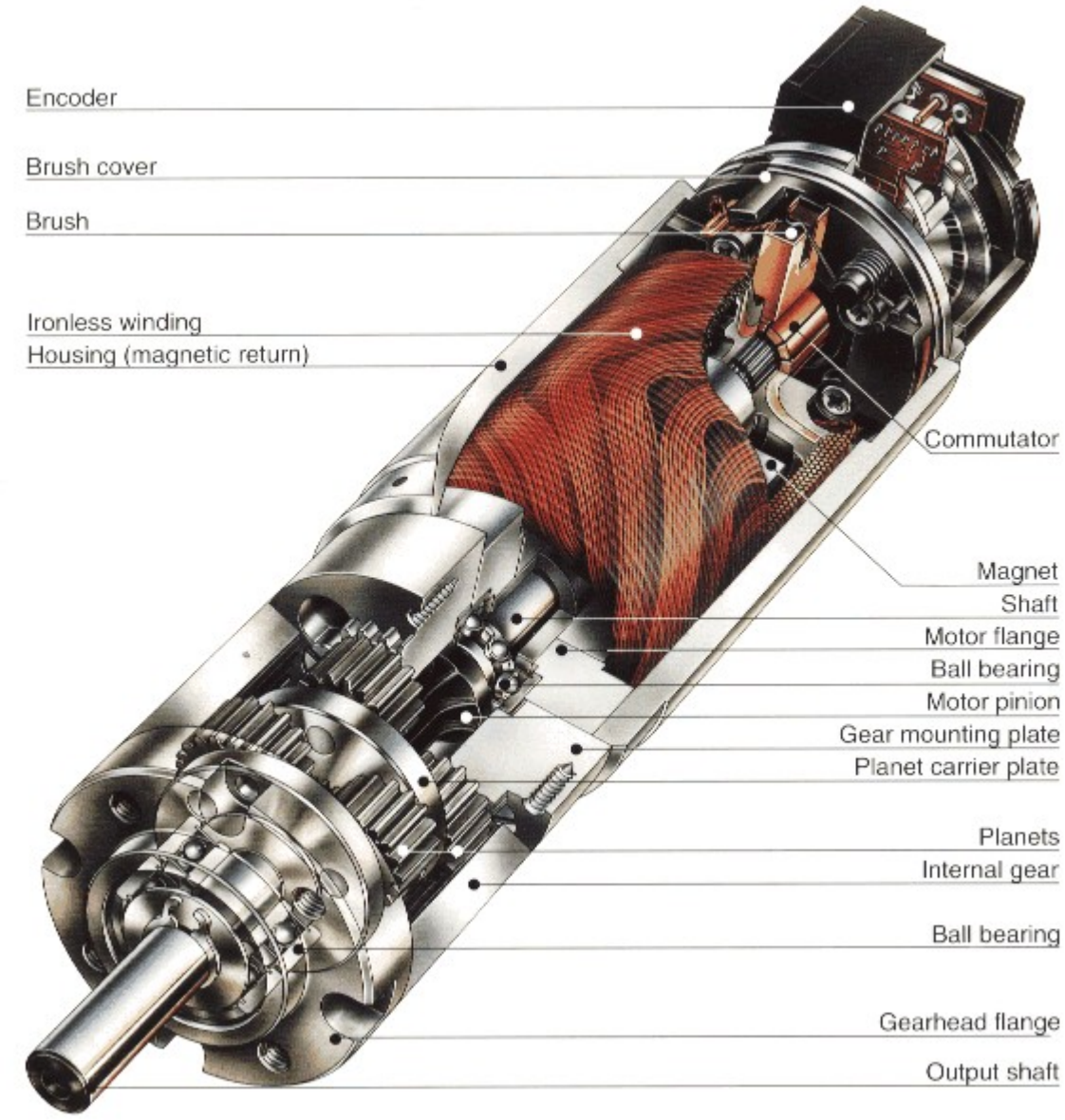
www.allcorp.com Fiyatlar ve mallar bayağı iyi fakat kalite düşük.

www.pulsemotor.com sıradan ama komplex aletler var.

www.alltronics.com Güzel ve kalite mallar var ama yeterli değil?

[5.4.3] Servo Motorlar:

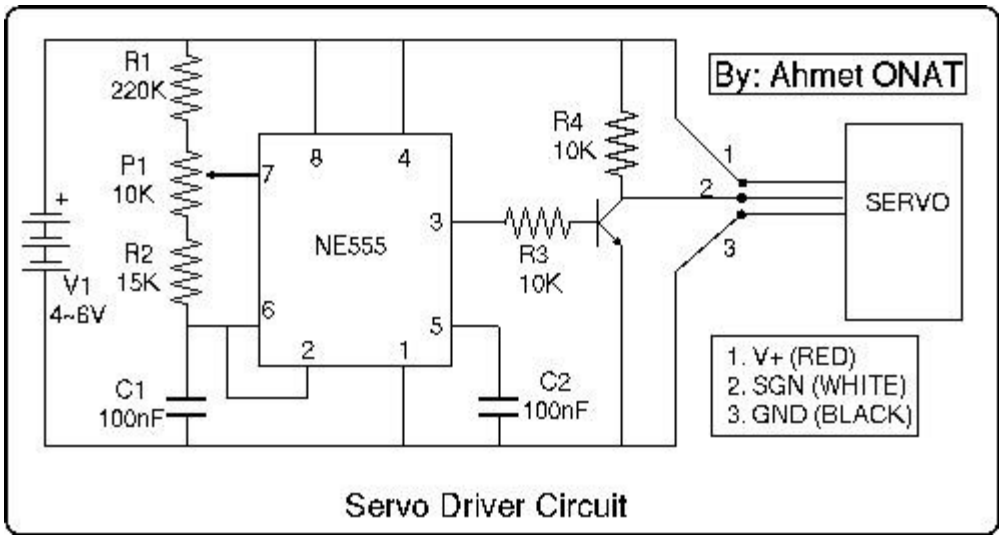
Adım motorunun yapısal frenleme ve çözünürlük sağlamasından dolayı amaça uygun olarak gözüksede sorun olarak karşımıza yetersiz çözünürlük vede hız problemleriyle çıkan adım kaçırma sorunları çıkınca cazibesini yitirmektedir. Servo motor kavramı burada ortaya çıkmaktadır. Yapı Step motorla DC motorun birleştirilmesiyle oluşmuştur. Üç ana dış bobin yapısıyla step motorun parçalama teporisine döndürme prensipiylede DC motoru çağrıştırmaktadır. Extra olarak konum algılama sensörleri ve gelişmiş sargı teknikleri yer almaktadır. Aşşağıda gelişmiş bir SERVO-GEAR(DİŞLİ)-ENCODER(SAYICI) üçlemesinin yer aldığı bir örnek görülmektedir.



Bu Motor Nasa'nın Pathfinder adlı marsa inen ilk aracında kullanılmıştır.

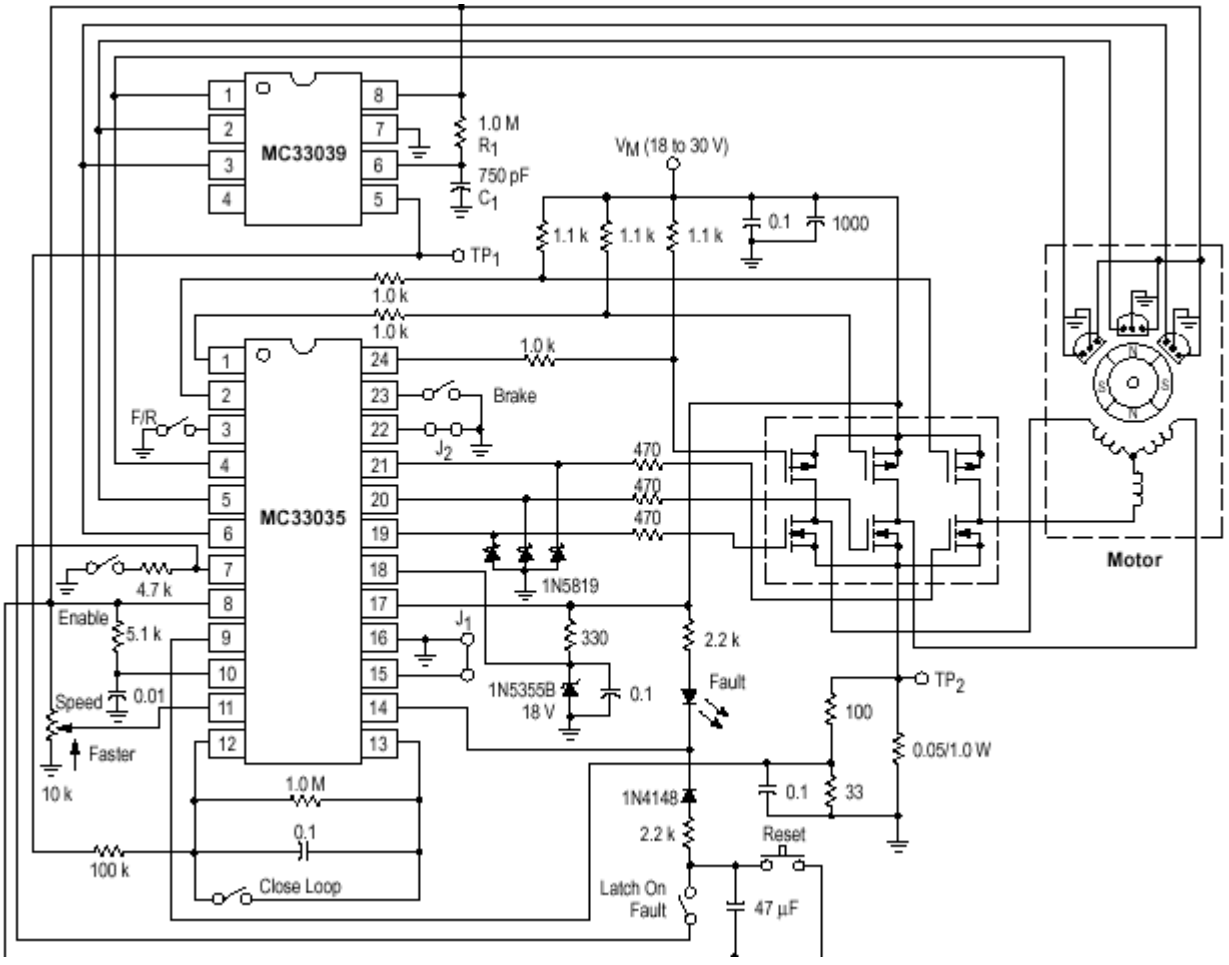
[5.4.3.1] Sürücü Devresi:

Yapısal olarak dc benzemesi ve kullanım açısandan steb'i çağrıştırmasına rağmen step gibi kullanım kolaylığı olan bir motor türü değildir servo motorlar. Motor zamansal olarak verilen palselere yeniden ateşlenerek döndürülmesi devam ettirilmektedir palseler ne kadar sık olursa o kadar hızlı, palseler ne kadar güçlü olursada o kadar güçlü hareketler sağlamaktadır. DC motorlardaki kalkış problemi yaşanmamakla birlikte yüksek gerilimde kullanılmak zorundadırlar.....



Devrede NE555 entegresi tipik osilatör kaynağını oluşturarak temel üç uçlu bir servodc motoru tetikleyerek döndürmektedir.

Bu örnekte ise motorola ciplerinin kullanıldığı bir dc servo lojik sürücüsü görülmektedir.



[5.4.3.2] Olumlu Olumsuz Yapıları:

Olumsuz yanı pek yok gibi teknolojini en iyisi durumunda yüksek hızlarda hibir sorunu yok neredeyse ayrıca kaliteli bir kontrolcüyle çok daha iyi sonuç veriyor gearboxlı yapıları önemle tavsiye edilir. Kontrolcüsünü yapmak biraz zorlar ama hazır satılanlar bayağı ucuz. Yüksek devirli kullanacaksanız mutlak olarak ball bearingli alın. Ben kol hareket motivasyonu için SERVO-GEAR(DİŞLİ)-ENCODER(SAYICI) birleşimini içeren minimotor firmasından bir ürün grubunu seçtim.

[5.4.3.3] Piyasa Durumu:

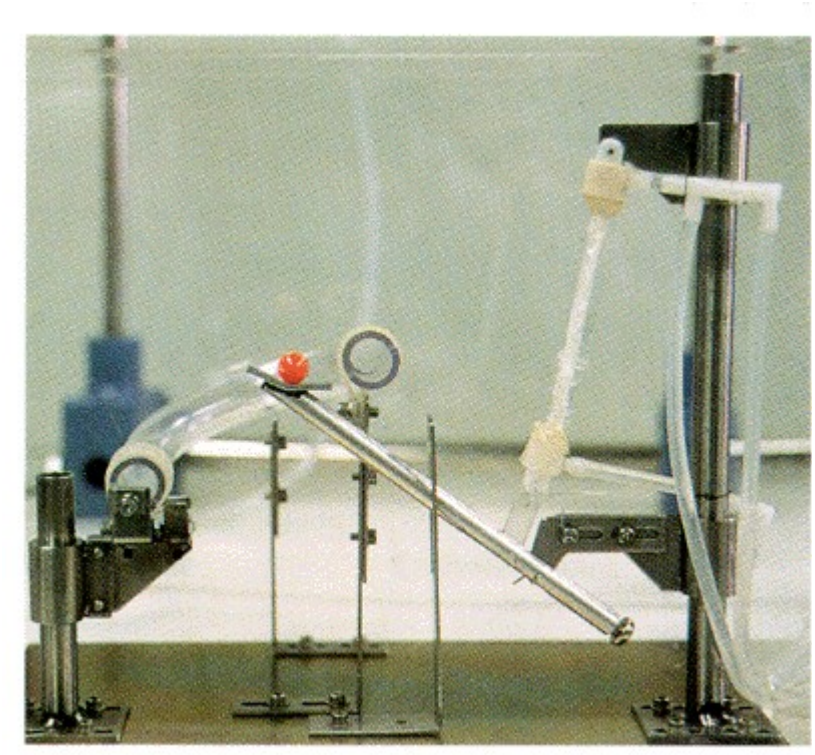
Tek problem motor sürücü devresini çok komplex olması bu yüzden devreyi mümkünse hazır alın daha ucuza gelmesi yanında hatada yapmıyor. Motorlar büyüklük ve hitap ettikleri alan göre fiyatlandırılıyor amaca uygun motor seçilirse kazançlı çıkılmış olunur.

www.minimotor.ch Çok harika motorlar var ama fiyat problem.

www.alltronics.com Güzel ve kalite mallar var ama yeterli değil?

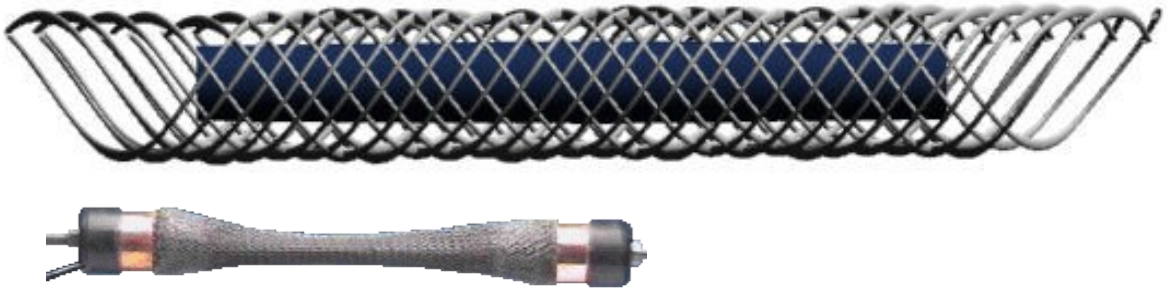
[5.4.4] Yapay Kas Sistemleri:

Yapay kas sisteminde keşvedilen bir silikon birelşimli bir maddeye elektrik verince kasılamaktadır fakat madde karasız olmakla beraber uzun ömürlü bir kasılma hareketinide gerçekleştirememektedir. Sonuç olarak henüz test aşamasındadır.



Aşağıdaki örnekte ise hava basıncı ile çelik tellerin dairesel hareketini kullanarak kasılan ve bırakan bir yapay kas sistemi görülmektedir. Bu sistem insan kasını taklit eden en başarılı çalışmalardan biridir, piyasa satışıda başlamıştır. Tam isim olarak Air Muscle olarak adlandırılmaktadır.

Ayrıntılı bilgi için: http://www.shadow.org.uk/products/airmuscles.shtml



[5.5] Geri Besleme(Feed Back):

Bilgisayar kavramı işte asıl bu kavramların hesaplanması ve analizinde önemli bir yer almaktadır. Üç bouytlu olarak bir noktayı ilerletmek için yapılan işlem mikro saniyeler aldığından robotun hareketinde bir gecikme faktörü olmaktan çıkıp geri besleme vede hata aramada yer almaya başlamaktadır. Motorların dönme hareketini açısal olarak yani adımsal olarak belirli bir çözünürlükte aktarılabilmesi için gerekli sistem sonucunda encoder yapısı ortaya atılmıştır. Temel olarak iki ana yapıda yer almaktadırlar;

1. Mağnetik
2. Optik

[5.5.1] Mağnetik Ortamlar:

Mağnetik kavramındanda anlaşılabileceği gibi mağnetik ortamlar olan endüvisel bobin ve mıklatıs yapılarından oluşan bir geribesleme türüdür. Hareket bir mıklatış çembere aktarılır ve çembere en az iki kutup başı olmak üzere birden fazla kutup yerleştirilerek imal edilir çemberin bir noktasında yeralan magnetik alan algılayıcı olan transistör yapılı entegre ile kutup değişimleri algılanabilir. Ne kadar fazla kutup başı olursa sistem o kadar çok adım gösterir, veya çemberin çevresel uzunluğuda kutup sayısıyla birlikte encode sistemine etki eder. Sistemde kutup başları arasındaki geçişlerdeki boşluk ve hassasiyet yüksek çözünürlüklü

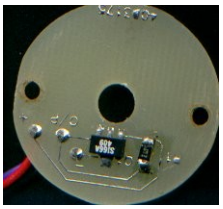
sistemler için kabul edilebilecek sınırların dışındadır. İki farklı kullanım türü vardır;

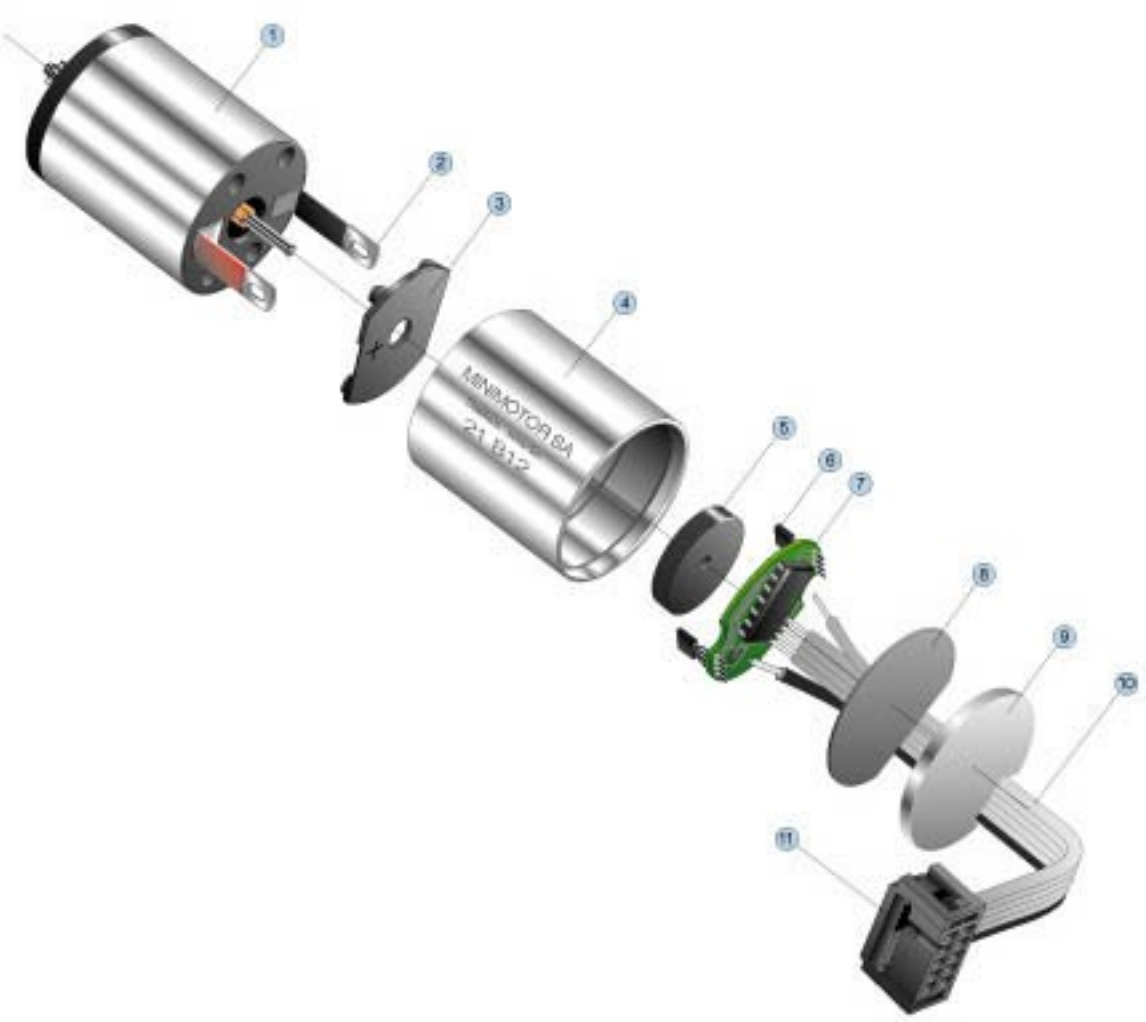
[5.5.1.1] Potansiyometre:

Her ne kadar mağnetik alan içermesede yapısal olarak mağnetik encoderleri çağrıştırmaktadır. Bilinen Pot dirençler bu işlem için kulanılır, potun sonsuz turlu olması gerekmektedir. Bu sistemin encode işlemlerindeki en büyük avantajı anlık konum bilgisini verebilmesidir yani adımları sayarak değilde sıfırıncı dirençten itibaren olan direnç değişim farkından açısal dönüş maximum dirençe göre çıkarılabilir. Dez ajantajı ise sonsuz turlusunun fiyatının yüksek olması ve 0(sıfır) noktasındaki bir miktar olan fakat sistem genelini etkileyen boşluktur.

[5.5.1.2] Mağnetik Encoder:

Mağnetik ortamlarda anlatıldığı gibi çembersel bir mıklatısın kutup değişimini algılayan bir mağnetik alan değişim entegresi ile sistem encode yapmaktadır. Bu sistem anlık pozisyon yerine adım bilgisi vermektedir. Aşşağıda örnek resimleri görülmektedir;





Copyright © 1998
by Minimotor SA

5.Mağnetik Çember 6-7.Kutup Algılayıcı

[5.5.2] Optik Encoder:

Yukarıda işlenen mağnetik encoderlardan farklı olarak optik yapı önplana çıkmaktadır. Mağnetik ortamdaki mıklatıs çember yerine belirli adetlerde var olan delikler devreye girmektedir bunlar kutup başları gibi adım atmaları belirtmek için kullanılır. Mağnetik kutup değişimi entegresi yerine optic bir led ile tam karşısına optic algılayıcı konarak önünde yer

alan delikli çemberin dönüşünü algılamaktadır. İleri ve geri hareket ayrımını yapabilmesi içinde optic çember delikleri ikili olarak dizilmişlerdir vede cift optic led ve algılayıcıyla tek-çift mantığı sayesinde hareket çözülmektedir. Aşşağıda buna ilişkin tablo ve örnek yapılar yer almaktadır.



1.Motor

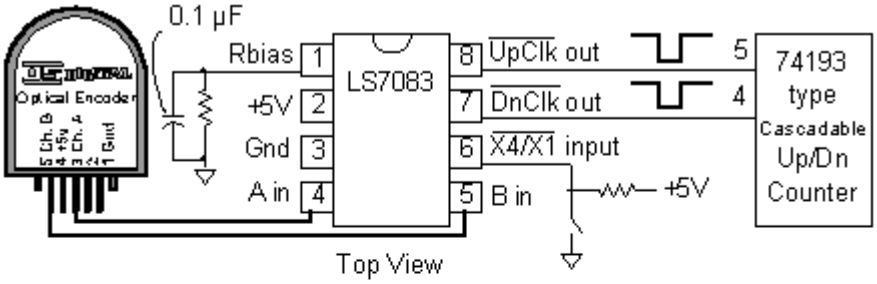
2.Koruyucu Kesit

3.Optic Verici ve yansıyanı alıcı

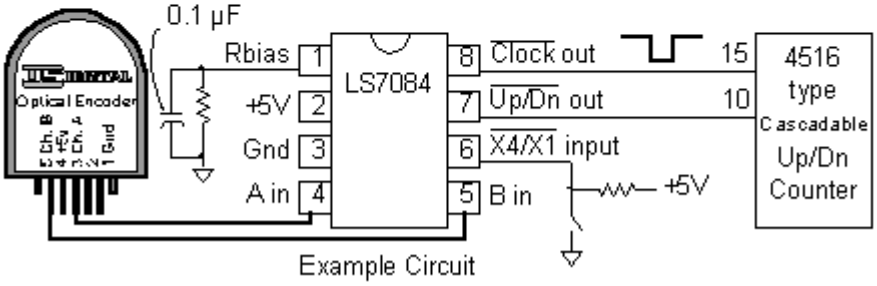
4. Ana bağlantı kablosu

5. Ters ışıkların girmesini önleyen koruyucu kap.

© 1999 by Minimotor SA

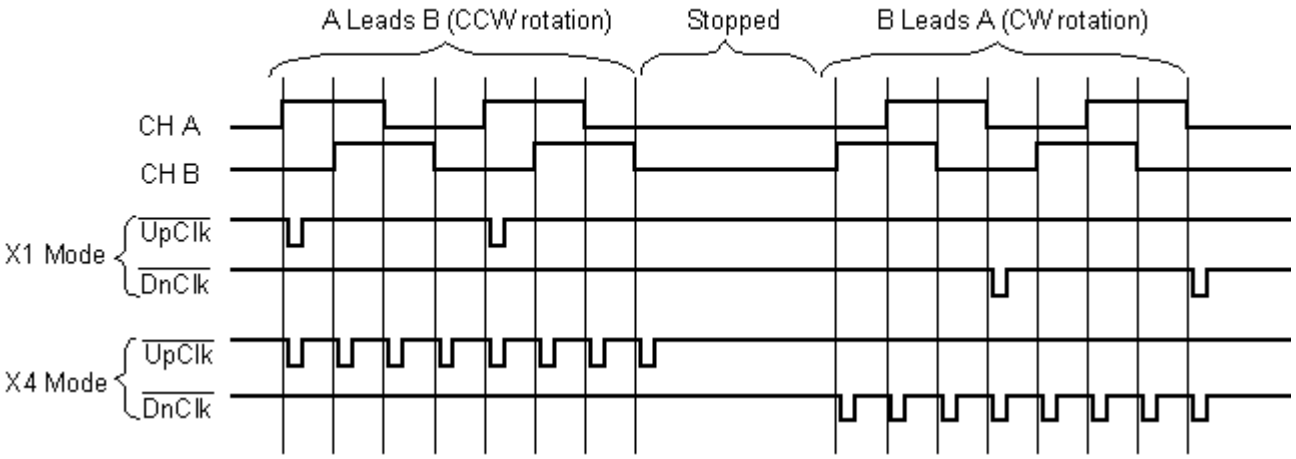


Top View

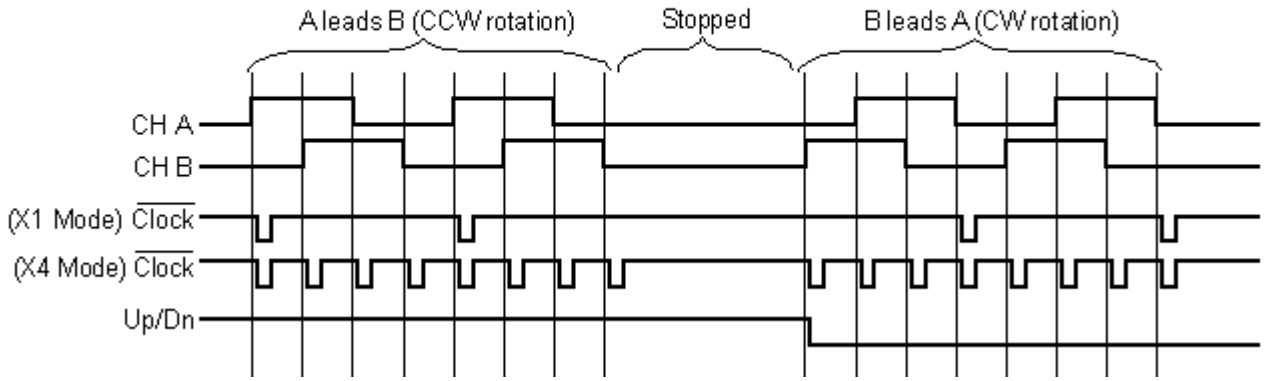


Example Circuit

LS7083



LS7084

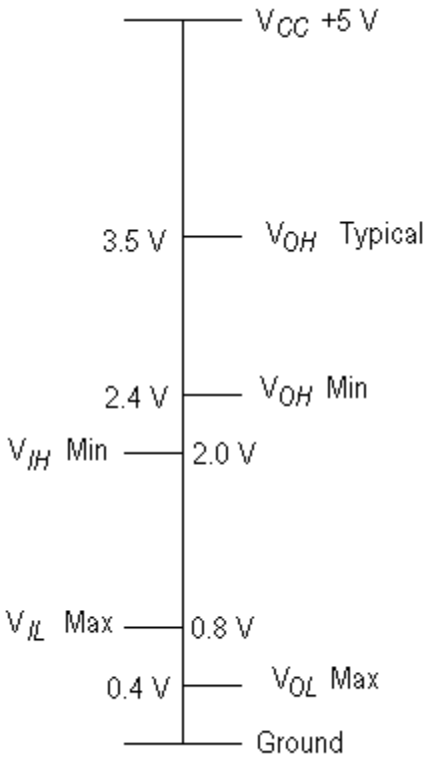


[5.6] Lojiksel Elektronik:

Elektronik kavramında 1970 yıllarda oluşan aşamalardan sonra ortaya sayısal elektronik yani birimsel sistemler ortaya çıkmıştır ve bu sistemler elektroniğin temeli olan voltaj kaynağına 1 yada 0 atayarak sonuca ulaşıp veri taşımak istenmiştir bu yüzten TTL olarak alandırılan temel lojik sistem oluşturulmuştur. Cmos ve Hcmos bu standardın alternatifi ve kapsam farklılıkları gösteren alt ürünleridir ama yinede en fazla ve esnek ortamlarda kullanılan TTL'ler piyasayı ele geçirmiştir. Şu anda en ufak bir lojik işlem içeren bir aleti incelesin mutlak olarak temel lojik elemanlar yani TTL entegreler ile karşılaşacaksınızdır.

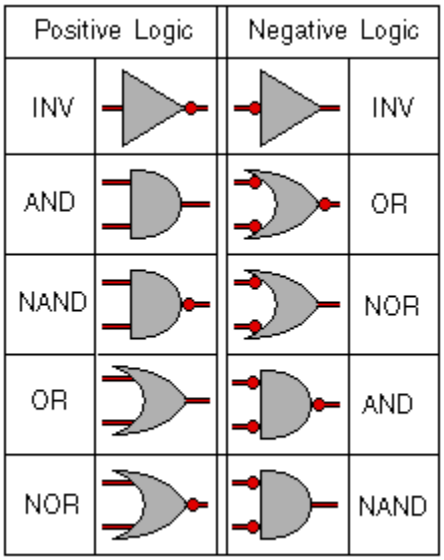
[5.6.1] TTL standardı:

Genel olarak TTL standadı denince akla 74xxx serisi entegreler akla gelmektedir fakat asıl kapsamı oluşturan entegre içerisindeki lojik voltaj alıcı ve yollayıcı devre karakteristikleri 74 serisi entegreleri TTL kapsamına almaktadır. Aşağıda lojiksel olarak voltaj değerleri yer almaktadır;



Lojik elektronikte voltaj aralığı 5v ile GND yani 0 volt arasında alınmaktadır. 5V ile 2.4V arasındaki bölge lojiksel olarak 1 yani High(Yüksek) uç olarak benimsenmiştir, 2V ile 0.4V arası ise lojik olarak 0 yani Low(Düşük) uç olarak alınmıştır. 2.4V ile 2V arası kara bölge olarak bilinir ve devreuçlarının lojik işlemlerde bu aralıkta değer vermesi veya alması sonucu hataya uğratmaktadır.

Lojik olarak tanımlanan voltaj değerleri ile işlem yapmak içim temel mantık işlemleri devreler haline getirilip bunlara isimleri ve nitelikleri verilmiştir. Aşağıda lojiksel simgeleri ve uçları görülmektedir;



Lojik kapı olarak tanımlanan bu mantıksal işlem birimlerini kısaca inceleyelim;

INV: Tersleyicidir girişinden aldığı lojik ifadenin tersini alıp çıkışından verir.

AND: Ve mantık işlemini gerçekleştirir. Girişlerinden aldığı değerlerden herikisi birden lojik olarak 1 ise çıkışı lojik olarak 1 olur.

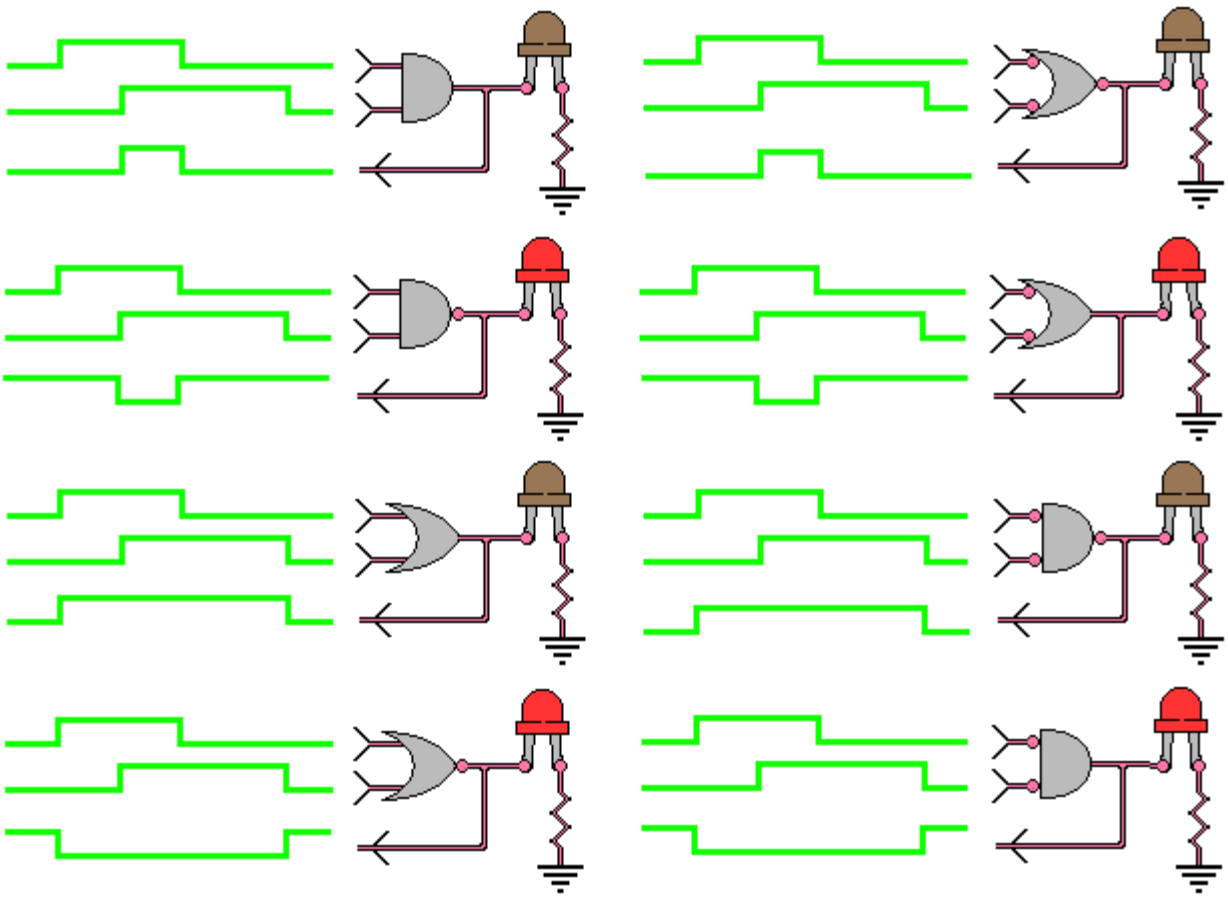
NAND: Ters Ve işlemidir, bir AND kapısının çıkışına INV takılmış gibi çıkışı Ve kapısının tersidir yani girişlerinin her ikiside 1 ise sonuç sıfır aksi durumda 1 dir.

OR: Veya mantık işlemi gerçekleştirir. Girişlerden herhangibiri lojik olarak 1 ise çıkış 1 olur aksi halde logik 0 dır.

NOR: Veya aişleminin terslenmiş halidir. Girişlerden herikisi birden lojik 0 ise sonuç lojik 1 dir aksi halde çıkış lojik 0 dır.

XOR: Veya işleminin çapraz kullanımıdır yani giriş uçlarından sadece bir lojik 1 ise çıkış lojik 1 dir aksi durumlarda lojik 0 olacaktır.

Aşağıda bu kapıların çalışma şekli görülmektedir, ledin kırmızı olduğu yerlerde lojik 1 sonucu elde edilmiştir;



74 serisi entegrelerde kendi aralarında hız ve kapasite açısından ayrımlar göstermektedir aşağıdaki tabloada bu ayrımların sonuçları görülmektedir:

SERIES	V_{CC} (V)	$V_{IL(max)}$ (V)	$V_{IH(min)}$ (V)	V_T (V)	dt/dv (ns/V)
SN74	4.75–5.25	0.8	2	1.4	15
SN74LS	4.75–5.25	0.8	2	1.4	15
SN74S	4.75–5.25	0.8	2	1.4	15
SN74ALS	4.5–5.5	0.8	2	1.4	15
SN74AS	4.5–5.5	0.8	2	1.4	15
SN74F	4.5–5.5	0.8	2	1.4	15
SN74HC	2	0.3	1.5	1.4	625
	4.6	0.9	3.15	2.25	110
	6	1.2	4.2	3	80
SN74HCT	4.5–5.5	0.8	2	1.4	125
74AC	3	0.9	2.1	1.5	10
	4.5	1.35	3.15	2.25	10
	5.5	1.65	3.85	2.75	10
74ACT	4.5–5.5	0.8	2	1.4	10
SN74BCT	4.5–5.5	0.8	2	1.4	10
SN74ABT	4.5–5.5	0.8	2	1.4	5/10
SN74LV	2.7–3.6	0.8	2	≈1.5	100
SN74LVC	2.7–3.6	0.8	2	≈1.5	5/10
SN74LVT	3.0–3.6	0.8	2	1.4	10

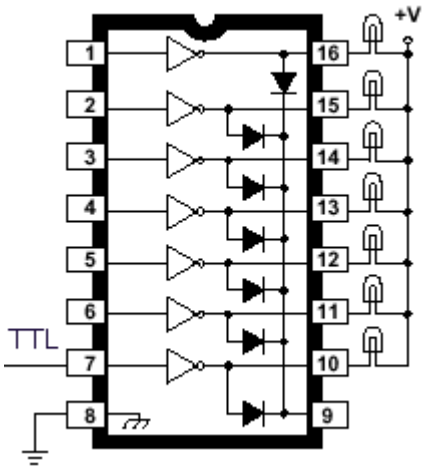
FAMILY	INPUT CAPACITANCE (pF)	OUTPUT CAPACITANCE (pF)	
		OPEN-COLLECTOR OUTPUT†	BUS DRIVER
SN74	3	5	–
SN74LS	3.5	3.5	5
SN74S	3.5	3.5	9
SN74ALS	2	4	5
SN74AS	4	–	10
SN74F	5	5	9
SN74HC	3	3	9
74AC/SN74AC	4	–	10
SN74BCT	6	–	12
SN74ABT	4	–	8
SN74LV	3	–	8
SN74LVC	4	–	8
SN74LVT	4	–	8

74 serisi entegrelerde HC ve HCT tercih edilen tiptir çünkü gerek hız gerekse parazitsel etkiler açısından yararlıdırlar fakat F serisi vede özellikle 75F serisi entegreler neredeyse hatasız bir ortam

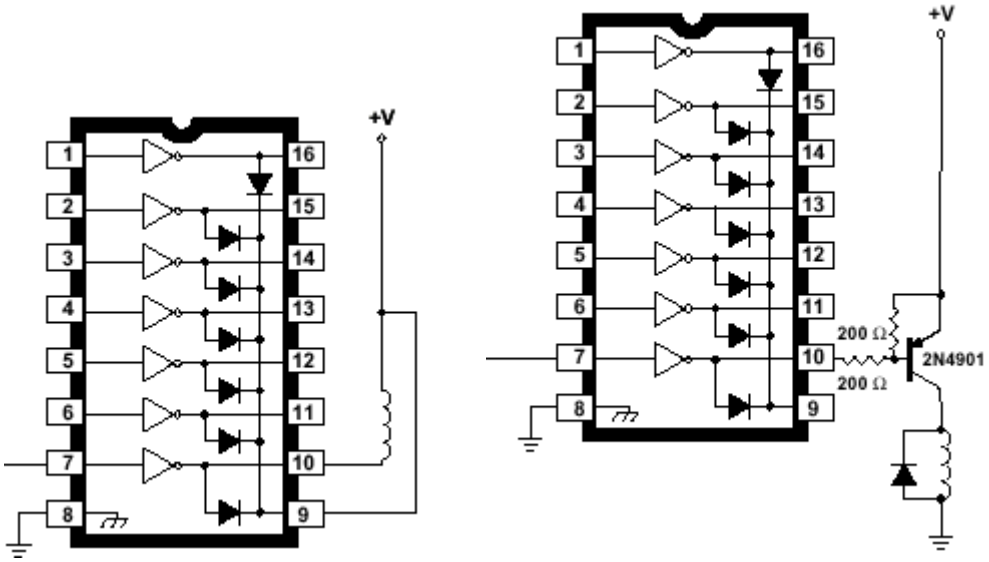
oluşturmaktadır. Eğer hıza ihtiyaç yoksa ynai Khz'de işlem yapılacaksa 74LS seriside iş görecektir.

[5.4.1.1] Sürücü Devresi:

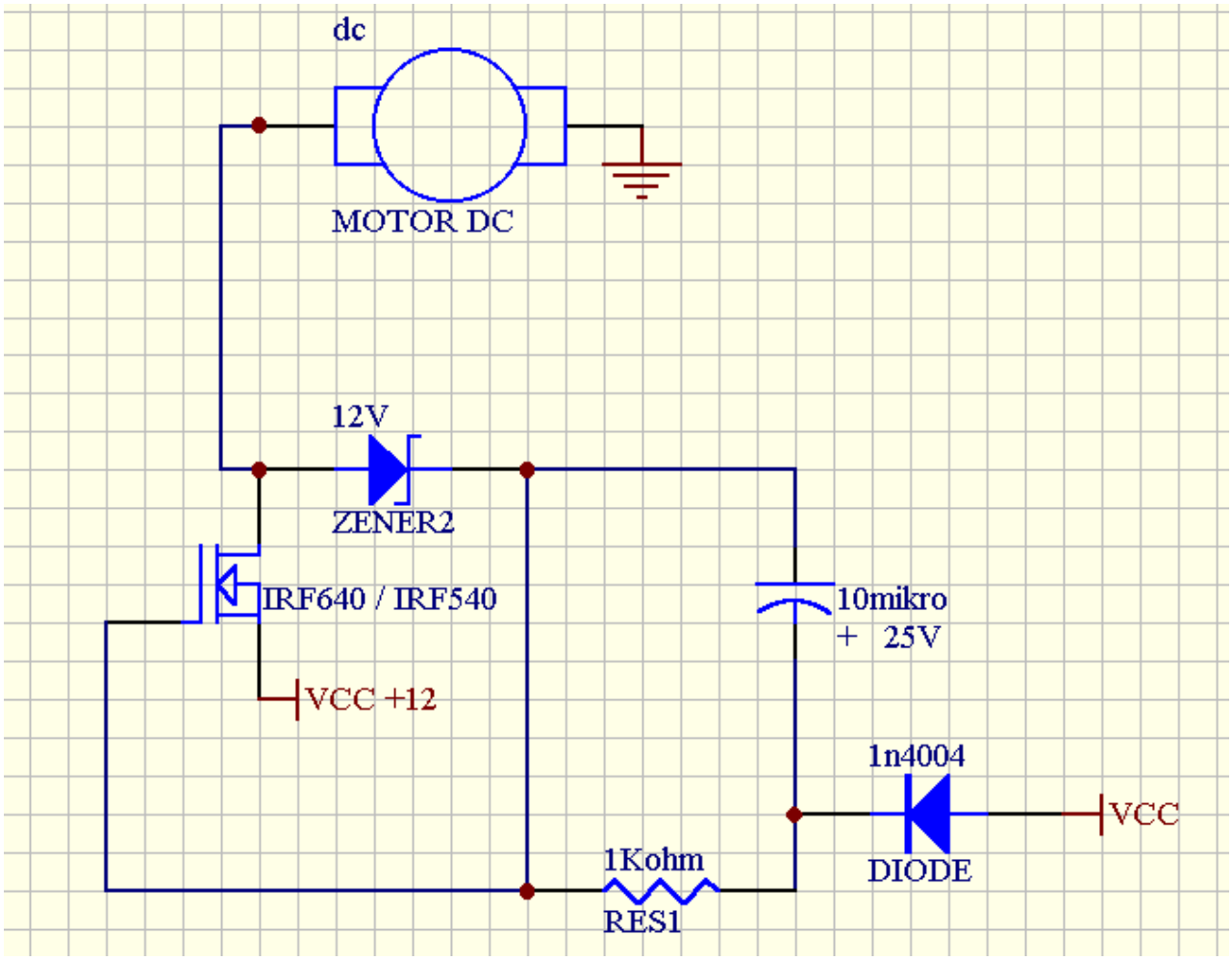
Lojiksel yapılan işlemler sonucunda oluşan sonuca göre bir veya birden çok elektronik anahtar veya güç elemanı çalıştırılması sayesinde sonuca varılmak istenir fakat lojiksel birimler için bu elemanları çalıştırmak, sürmek imkansızdır çünkü gerekli akım veya gerilim yani güç yeterlilik derecesi lojiksel birimler için yetersizdir. Lojiksel elemanlardan güç elemanlarına geçişi sağlayan elemanlara Power Drivers (güç sürücüleri) denir. Örnek olarak aşağıdaki ULN2003 entegresi verilebilir;



Şekilde görüldüğü gibi 16 uçlu 2003 entegresinin GND ve Vcc çıkarılınca geriye kalan uçlarla 7 adet güç çıkışı elde edilmektedir. 7 nolu uçtan verilen lojik 1 ile 10 ucuna bağlı olan lamba sönecek ve lojik 0 verilincede lamba yanacaktır, lamba 30V'luk olmasına karşın devremizin güç katı ile terslenerek ve diyot korumasıyla sonuç dışarıya yalıtılarak çıkarılmaktadır. Aşağıdaki örnektede bir motor bobinin 2003 ile lojik olarak sürülmesi görülmektedir;



Bir diğer kullanımda olan yöntem ise transistörlerle yükseltmek daha doğrusu akımı korumaktır ama bu yöntem hem yavaş hemde standardı, montajı zor olan bir yöntemdir



[6.1] Mikrokontolcülerin Tarihçesi:

Ilk olarak temel olarak bir mikroislemci nedir bunun tanimini yapacak olursak temel manada mikroislemci mantiksal ve aritmetik islemleri gerçeklestirebilen komplex devre topluguna verilen addir dersek sanirim dogru olur.Belirli bir aygita mikroislemci olarak tanimlamak alanin ana hedefi olan askeri çalismalardan dolayi kesinlik kazanamamaktadir. Fakat ticari piyasa açisindan ilk mikroislemci *INTEL* firmasina ait olan 8008 mikro islemcisidir.

Intel 8008'den önce San Antonio/Texsas/USA kurulu bulunan Datapoint firmasi akilli bilgisayar çevre elemanlari üretiyordu.1968-69 yillarinda çok basit bir bilgisayar tasarlayan Datapoint mühendisleri, kendi tasarimlarinin tek bir yonga üzerinde imali için Intel ve Texas Instruments firmalariyla ile antlasma yaptilar. Datapoint mühendislerinin amaci, degisik çevre elemanlarinin ihtiyaçlarinin degisik kontrol komutlariyla kontrol edecek olan genel amaçli bir kontrol devresi üretmekti. Bu yaklasim her yeni çevre elemani için yeni bir kontrol devresi yapmaktansa tek bir yonga ile islevsellik kazandirmakti.

Sonuçta üretilen entegre çok iyi çalisti. Fakat Datapoint devreden istenilen hizda islem yaptiramadigi için projeyi iptal etti ve eski sistemine döndü, Intel firmasi ise bu yonga düzenine 8008 adini vererek piyasaya programlanabilir mantiksal kontrolör olarak sürdü. Yavas çalisan bu entegre piyasada büyük ilgi gördü ve birçok alanda bu mikrokontroler üzerinde projeler gelistirildi. Bu yogun ilgiden dolayi intel ürününü gelistirmek için çalismalara basladi.

Asil gerilerden gelen bu temel Intel Firmasiyla uygulama alani buldu. 1947'de bulunan transistör kavrami ve 1950 baslarinda ayni yonga üzerinde "transistör ,diyot, direnç, kapasitans" birlestirilmesi bile hedefi bu kadar belirginlestirmemisti. Intelin chipini alanlar artik hazir bir alt yapinin üstünde dans ediyorlardi.

Bu tarihten sonra piyasa isteklerine ve teknolojinin ilerliyisine göre mikroislemciler çesitlilik ve hiz kazandilar. Kisaca Ana hatlariyla mikroislemci ailesi:

i4004 4 bitlik programlanabilen ilk chipti. 1946 üretildi ve o sene 30 tonluk ENIAc adli bilgisayar üretildi. Saniyede 100,000 islem yapabiliyordu.

i4040 Yine 4 bitlik bir islemciydi fakat hizi artirilmisti.

i8008 1971 yilinda üretilen 8 bitlik bir 4004 islemciydi.

i8080 Yine 4040 gibi 8008'in hizi artirilarak 1973 yilinda üretildi.

MC6800 Motorola Firmasi tarafindan üretilen 8 bitlik bir islemcidir.

i8085 1977 yilinda 8080 güncellenmis bir versiyonu üretildi.Hizi dahada artirildi.

Z80 AMD (Advandel Micro Devices) firmasi tarafindan üretildi 8085 ayni komutsetine sahipti.

i8086 1978 yilinda modern tabirini kullandigimiz ilk microkontroler üretildi. 16 bitlik ilk microchipdir.

i8088 1979 yilinda üretilen 8088 yine 8086'in hizlandirilmis versiyonudur. IBM firmasi 1981'de Intelin 8088'den ilk PC lerini üretti.

i80286 1983 yilinda üretildi. 8086'in güncellenmis versiyonudur.

i80386 1986 yili yapimi olan 386 yine bir ilke imza atti, 32 bitlik bir mikro islemci.

i80486 1989 yili yapimi olan 486 islemcisi 387 adli matematik islemciyle bütünlestirildi ve hizi 50 MHZ çikarilidi.

Pentium 1993 yilinda ilk olarak üretilen Pentium isim hakkindan dolayi isim degisikligine ugradi. 486 farkli olarak veriyolu hizi vede cache miktarlari artirildi.

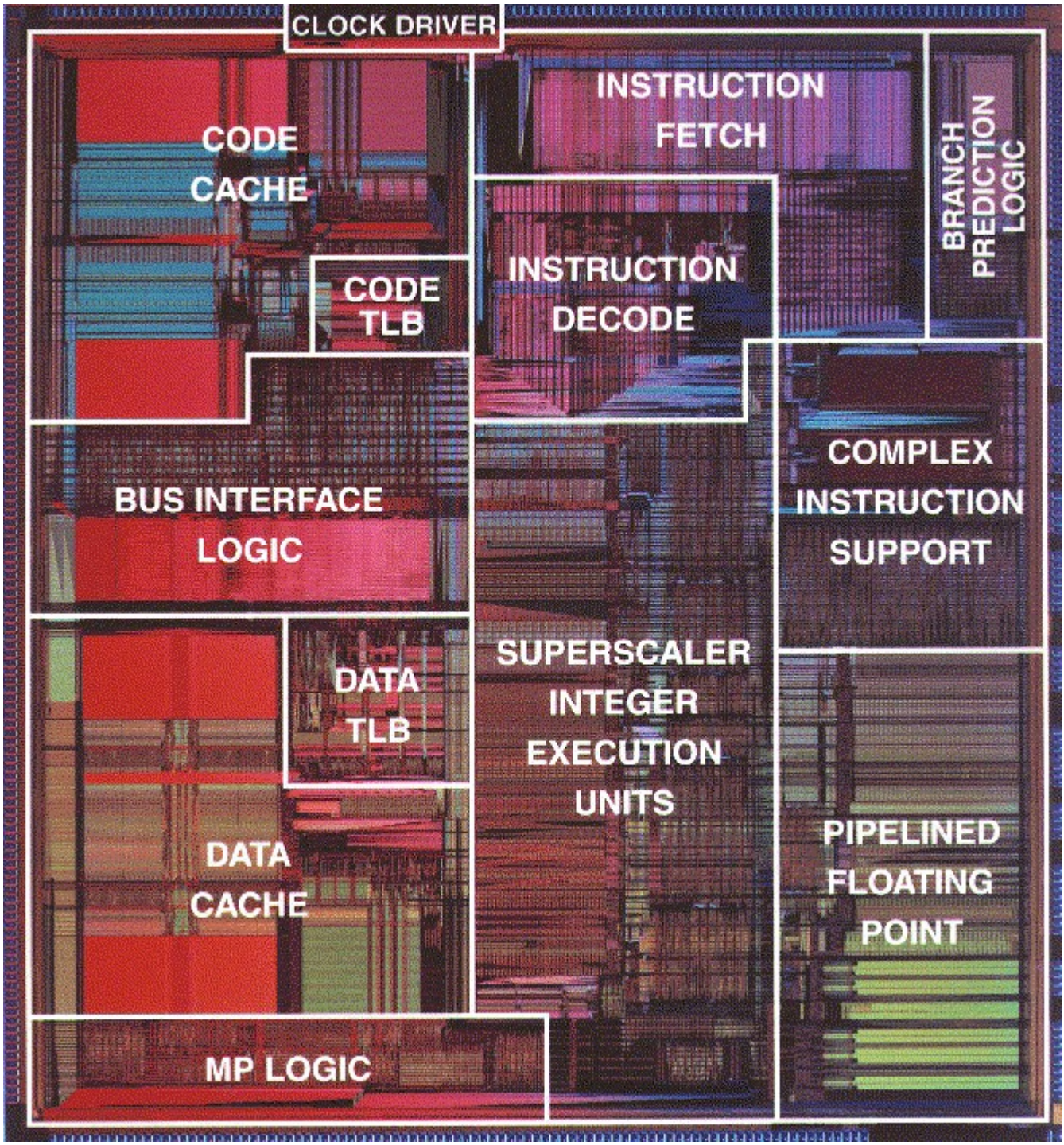
PentiumPRO 1995 yilinda üretildi ve iki level cache ile birlikte tamamiyle 64 bitlik bir islemci oldu.

Asagida özet olarak bir tablo görmektesiniz;

TABLE 1–2 Many modern Intel and Motorola microprocessors

Manufacturer	*Part*	*Data Bus Width*	*Memory Size*
Intel	8048	8	2K internal
	8051	8	8K internal
	8085A	8	64K
	8086	16	1M
	8088	8	1M
	8096	16	8K internal
	80186	16	1M
	80188	8	1M
	80251	8	16K internal
	80286	16	16M
	80386EX	16	64M
	80386DX	32	4G
	80386SL	16	32M
	80386SLC	16	32M + 1K cache
	80386SX	16	16M
	80486DX/DX2	32	4G + 8K cache
	80486SX	32	4G + 8K cache
	80486DX4	32	4G + 16K cache
	Pentium	64	4G + 16K cache
	Pentium Overdrive (P24T)	32	4G + 16K cache
	Pentium Pro processor	64	64G + 16K L1 cache + 256K L2 cache
Motorola	6800	8	64K
	6805	8	2K
	6809	8	64K
	68000	16	16M
	68008Q	8	1M
	68008D	8	4M
	68010	16	16M
	68020	32	4G
	68030	32	4G + 256 cache
	68040	32	4G + 8K cache
	68050	32	Proposed, but never released
	68060	64	4G + 16K cache
	PowerPC	64	4G + 32K cache

Daha Ayrıntılı bilgi için: http://bwrc.eecs.berkeley.edu/CIC/archive/cpu_history.html . Aşağıda Intel firmasına ait Pentium tipi bir mikrişlemcinin devre şema PCB si görülmektedir.



[6.2] Mikrokontrolün Yapisal Analizi:

Ana hatlariyla bir mikroislemci kendini var eden temel etmenlerden olusur. Bunlar kisaca MPU, RAM, ROM, IO dur. Simdi kisaca bu yapilara deginelim;

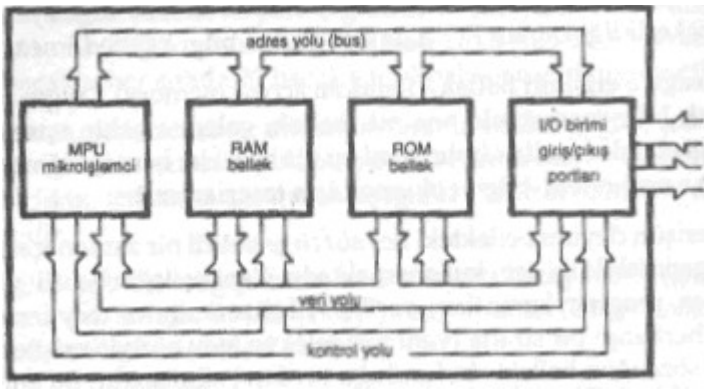
MPU: Merkezi Islem Birimi olarakda anilan birim mikroislemcimizin ana yapisal islemleri olan mantiksal islemler, aritmetik islemler ve yürütme islemlerini gerçeklestirir yani kisaca diger birimlerden aldigi verileri isleyerek yeniden baska bir birime aktaran kisimdir. Belirli bir saat

frekansinda çalisir yani saniyede belli sayida islem gerçeklestirir. Kendi içerisinde ROM,RAM ve IO'suda vardir.

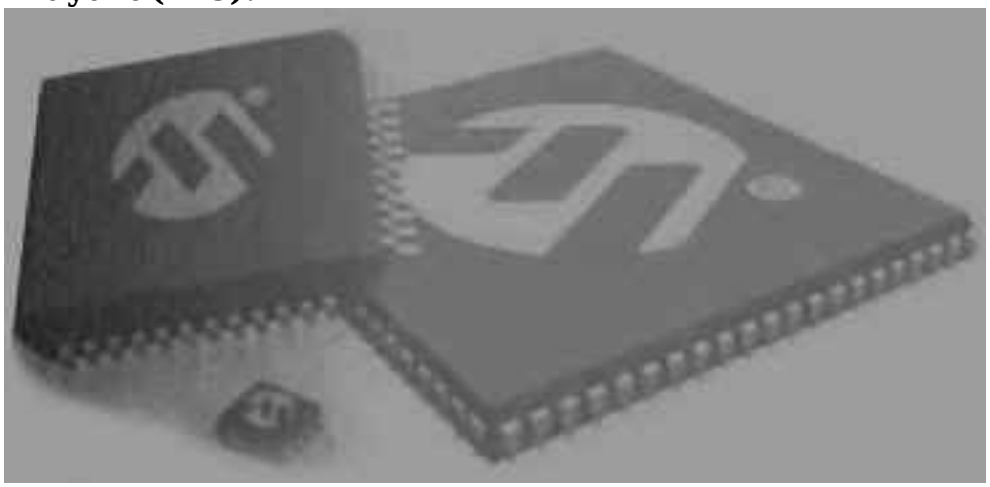
ROM: Yanlizca okunabilen bellek olarak adlandirilan ROM üretim esnasinda bir kerelik olarak doldurulan hafiza parçasidir.

RAM: Asil bellek olarak bilinen kisimdir aktif olarak merkezi islem birimi tarafindan kullanilir azligi veya çoklugu sistem kapasitesini etkiler. Elektrik kesilince silinir.

IO: Mikrokontrolcünün dissal bagidir. Disaridan alinan bilgiyi MPU ve yine MPU 'dan aldigi veriyi dissal bir porta aktaran korunmuş kisimdir. Sistemdeki en yavas aygitlardır.



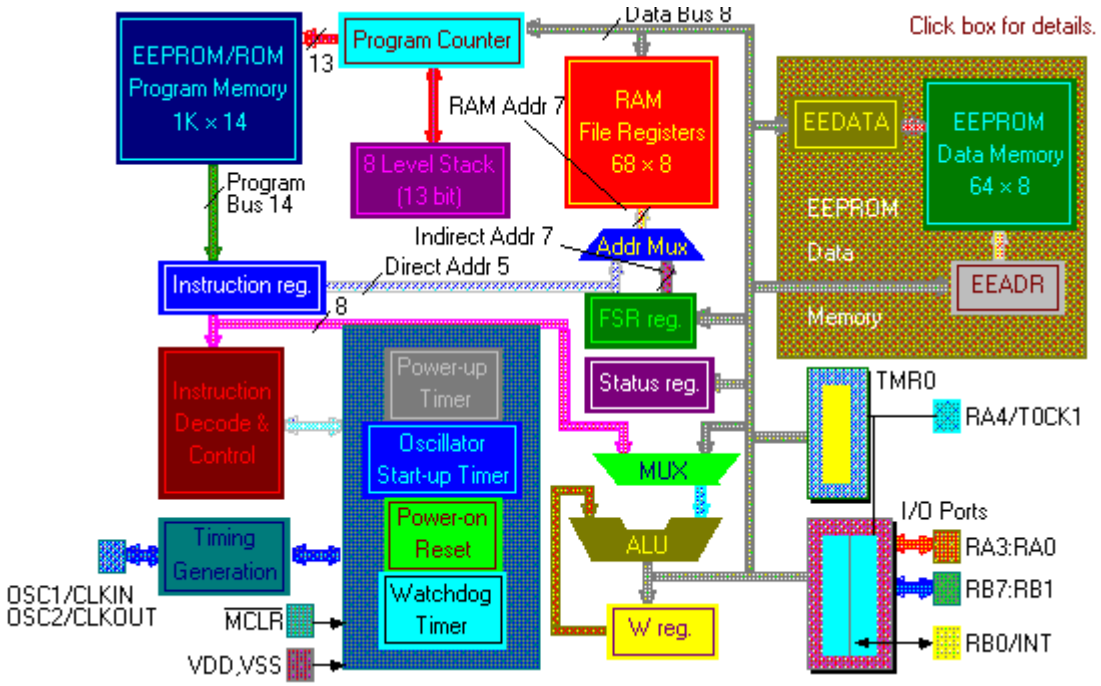
[6.3] Programlanabilir Kontrol Arayüzü(PIC):



Pic üretilmis bulunan mikroislemcilerden farkli olarak basite indirgenmis fakat hizi ve etkisi düsürülmemis olan bir mikrochiptir. MICROCHIP firmasi tarafindan uygulama formlari için üretilmistir. Basit yonga seti, uygun portlari vede opsiyonel islevleri bakimindan hedef kontrolü için bulunmaz bir mikroislemcidir PIC. Yapisal olarak risc

islemcileri çagristirmakla beraber gelistirilmis komut yapisi sayesinde emsallerinden çok daha çabuk ve hizli programlanabilir. 8-16 bitlik islemcilerdir. 20 MHZ kadar çikabilirler. Bazi tiplerinde üzerindeki portlar ADC ve DAC görevini tek word üzerinden görebilmektedirler. Rakiplerine göre hizlari ve çalisma ortamlari dah esnek ve kararlidir. 2.5V-6V arasinda çalisabilirler ve son olarak cazip fiyatlari bu tip ihtiyaclari tam olarak karsilamaktadir. (Ayrica enson teknoloji her zaman için bir öncekinden dahada avantaj ve yenilik içerir.)

Benim seçtigim model PIC 16C serisidir. Bu serideki islemcilerde EEprom chip içerisinde mevcut oldugundan ekstradan eprom baglamak ve isletmek için port ayirmak gerekmemektedir. Asagida temel olarak bir pic 16c serisi mikrochip'in blok semasini görmektesiniz;



Semada görüldügü gibi dissal ve içsel interruptlari(kesmeleri) mevcuttur. Bende devremde motorun encoderi real time çalismasinin saglanmasi için A portunun 5 numarali bacagi ile uart sinyalizasyonuna uydurmaya çalistim.

[6.3.1] Temel Konut Seti:

Toplam 37 komutu bulunan PIC mikroislemcisi temel bazli tüm komut kümesine sahiptir. i8086 biraz farkli olmakla beraber 8086 temel komut setinden çok daha fazla kullanisli bir komut kümesine sahiptir. Dikkat edilmesi gereken nokta komutlarin sonundaki f ve w ibareleridir. Asagida temel komut setini görmektesiniz;

PIC16F8X

TABLE 9-2 PIC16FXX INSTRUCTION SET

Mnemonic, Operands		Description	Cycles	14-Bit Opcode MSb			LSb	Status Affected	Notes
BYTE-ORIENTED FILE REGISTER OPERATIONS									
ADDWF	f, d	Add W and f	1	00	0111	dfff	ffff	C,DC,Z	1,2
ANDWF	f, d	AND W with f	1	00	0101	dfff	ffff	Z	1,2
CLRF	f	Clear f	1	00	0001	lfff	ffff	Z	2
CLRW	-	Clear W	1	00	0001	0xxx	xxxx	Z	
COMF	f, d	Complement f	1	00	1001	dfff	ffff	Z	1,2
DECF	f, d	Decrement f	1	00	0011	dfff	ffff	Z	1,2
DECFSZ	f, d	Decrement f, Skip if 0	1(2)	00	1011	dfff	ffff		1,2,3
INCF	f, d	Increment f	1	00	1010	dfff	ffff	Z	1,2
INCFSZ	f, d	Increment f, Skip if 0	1(2)	00	1111	dfff	ffff		1,2,3
IORWF	f, d	Inclusive OR W with f	1	00	0100	dfff	ffff	Z	1,2
MOVF	f, d	Move f	1	00	1000	dfff	ffff	Z	1,2
MOVWF	f	Move W to f	1	00	0000	lfff	ffff		
NOP	-	No Operation	1	00	0000	0xx0	0000		
RLF	f, d	Rotate Left f through Carry	1	00	1101	dfff	ffff	C	1,2
RRF	f, d	Rotate Right f through Carry	1	00	1100	dfff	ffff	C	1,2
SUBWF	f, d	Subtract W from f	1	00	0010	dfff	ffff	C,DC,Z	1,2
SWAPF	f, d	Swap nibbles in f	1	00	1110	dfff	ffff		1,2
XORWF	f, d	Exclusive OR W with f	1	00	0110	dfff	ffff	Z	1,2
BIT-ORIENTED FILE REGISTER OPERATIONS									
BCF	f, b	Bit Clear f	1	01	00bb	bfff	ffff		1,2
BSF	f, b	Bit Set f	1	01	01bb	bfff	ffff		1,2
BTFSC	f, b	Bit Test f, Skip if Clear	1 (2)	01	10bb	bfff	ffff		3
BTFSS	f, b	Bit Test f, Skip if Set	1 (2)	01	11bb	bfff	ffff		3
LITERAL AND CONTROL OPERATIONS									
ADDLW	k	Add literal and W	1	11	111x	kkkk	kkkk	C,DC,Z	
ANDLW	k	AND literal with W	1	11	1001	kkkk	kkkk	Z	
CALL	k	Call subroutine	2	10	0kkk	kkkk	kkkk		
CLRWDT	-	Clear Watchdog Timer	1	00	0000	0110	0100	$\overline{TO}$,$\overline{PD}$	
GOTO	k	Go to address	2	10	1kkk	kkkk	kkkk		
IORLW	k	Inclusive OR literal with W	1	11	1000	kkkk	kkkk	Z	
MOVLW	k	Move literal to W	1	11	00xx	kkkk	kkkk		
RETFIE	-	Return from interrupt	2	00	0000	0000	1001		
RETLW	k	Return with literal in W	2	11	01xx	kkkk	kkkk		
RETURN	-	Return from Subroutine	2	00	0000	0000	1000		
SLEEP	-	Go into standby mode	1	00	0000	0110	0011	$\overline{TO}$,$\overline{PD}$	
SUBLW	k	Subtract W from literal	1	11	110x	kkkk	kkkk	C,DC,Z	
XORLW	k	Exclusive OR literal with W	1	11	1010	kkkk	kkkk	Z	

Note 1: When an I/O register is modified as a function of itself (e.g., `MOVF PORTB, 1`), the value used will be that value present on the pins themselves. For example, if the data latch is '1' for a pin configured as input and is driven low by an external device, the data will be written back with a '0'.

2: If this instruction is executed on the TMR0 register (and, where applicable, d = 1), the prescaler will be cleared if assigned to the Timer0 Module.

3: If Program Counter (PC) is modified or a conditional test is true, the instruction requires two cycles. The second cycle is executed as a NOP.

Aşağıda ise komut setindeki komutların kullanım teori ve pratikleri yer almaktadır. Yazılan kaynak kod Matlab 6.xx ile derlenebilir;

ATAMA KOMUTLARI:

```
MOVLW     Belirtilen operant W yazmacina konulur. Operant 0-7F arasinda
olmalidir
        Örn:    movlw   H'22'
                movlw   B'01010101'

MOVWF   Belirtilen hedefe W yazmacı içerisindeki bilgiyi aktarır.
           Örn:  Movwf Sayi1
MOVF    Bir operantın içerisindeki bilgiyi W yazmacına aktarır.
           Örn:  Movf Sayi1,0
```

CLRF   Temel olarak bir ifadeyi temizler.
          Örn: Clrf Sayi1
CLRW   W yazmacını temizler.

BSF   Operantın belirtilen bitini set yani 1 yapar.
    Örn: BSF Sayi1,0
          BSF Sayi1,7
          BSF porta,0
BCF   Operantın belirtilen bitini reset yani 0 yapar.
    Örn: BCF Sayi1,0
          BCF Sayi1,6
          BCF Porta,0

ARITMETIK ISLEMLER:

INCF Kaynak gösterilen ifadeyi 1 arttırarak sonucu seçilen hedefe aktarır.
      Örn: Incf   Sayi1,0
            Incf   Sayi1,1
DECF Kaynak gösterilen ifadeyi 1 azaltarak sonucu seçilen hedefe aktarır.
      Örn: Decf   Sayi1,0
            Decf   Sayi1,1
ADDWF W yazmacıyla Kaynak Bilgiyi toplar sonucu belirtilen konuma göre aktarır.
    Örn: movlw   H'10'
          movwf   sayi1
          movlw   H'10'
          Addwf   sayi1,1   {20 hex sonucu sayi1'e aktarilir.}
SUBWF Verilen Operantdan W registerini çıkarıp sonucu istenilen hedefe yükler.
    Örn: Movlw H'10'
          Movwf Sayi1
          Movlw H'08'
          Subwf Sayi1,1   {08 sonucu sayi1'e aktarilir.}
XORWF W yazmacının değeriyle verilen değişken içeriğini Özel Veya işlemini uygulayarak sonucu istenen hedefe aktarır.
    Örn: Movlw H'01'
          Movwf Sayi1
          Xorwf Sayi1,0

KARSILASTIRMA KOMUTLARI:

BTFSC   Pic serisinin temel karsılastırma komutudur. Belirtilen ifadeinin bir bitini kaşılaştırır, eğer bit sıfırsa kendinden iki sonrakini değilse sonrakini işleme koyar.
   Örn: Btfsc Sayi1,0
         Clrf Sayi1
         Nop   {Eger sayinin birinci biti bir ise sayiyi sifirlar}
BTFSS   Pic serisinin temel karsılastırma komutudur. Belirtilen ifadeinin bir bitini
            kaşılaştırır, eğer bit sıfırsa kendinden iki sonrakini değilse sonrakini işleme koyar.
   Örn: Btfss Sayi1,0
         Goto Atla
          Clrf Sayi1   {Eger sayinin birinci biti bir ise sayiyi sifirlar}
        Atla
         Nop

Esitmi karsilastirmasi

            Temel olarak pic programlama dilinde esitmi ve farklimi diye tanimlanabilecek bir komut
           olmamasina ragmen temel bayraklar ayni PC makina dilindeki gibi kullanilabilir ozellikle

```
        carry(elde) ve zero(sifir) bayraklari bu is icin uygundur.
   Örn:  Movlw H'11'
         Xorwf Sayi1,0
         Btfsc Status,Z
         Goto Esit
         Nop
        Esit
```

BITSEL ISLEMLER:

```
RLF    Ifadeyi sol tarafa dogru bir bit cevirir.

RRF    Ifadeyi sag tarafa dogru bir bit cevirir.
```

GENEL ISLEVLI KOMUTLAR:

```
NOP    Hicbirsey yapmaz bir cycle komut ifadesidir.

GOTO   Kendinden sonra gelen satıra kosulsuz atlama yapar.
CALL    Kendinden sonra çağırılan alt program parçacığını işletip RETURN
görünce CALL satırının bir alt satırını işleme koyar.
```

Örnek Bir program:
Program ds1820 ısı entegresini kullanarak yapılmış seri porttan ayarlanabilen 7 haneli seven segmentli temel bir saat örneğidir.

```
 LIST    P=16F84
 ERRORLEVEL    -302
     __CONFIG      3FF9H


PCL    EQU    2
STATUS  EQU    3
PORTA   EQU    5
PORTB   EQU    6
INTCON  EQU    0BH
TRISA   EQU    85H
TRISB   EQU    86H
OPTREG  EQU    81H


C      EQU    0          ; bits in STATUS
Z      EQU    2
RP0    EQU    5

T0IF   EQU    2           ; bit in INTCON


     CBLOCK 22H
  ZZ               ;1 decremented every tick (9.15 ms or 8.19 ms)
       YY              ;2
       OXX              ;4
       IXX             ;5
       XX              ;6
       VALUE             ;7
       SING             ;8
       TICKS            ;10
```

```
        TICKS2              ;9
        ISIX               ;11
        SNY                ;12
        DAK                ;13
        SAAT                ;14
        GUN                 ;15
        AY                 ;16
        YIL                ;17
        BEKTARIH            ;18
        BEKSAAT             ;19
        BEKSICAK            ;20
        BEKNEM              ;30
        SIRA               ;21
        SYC1                ;22
        DECOD               ;23
        BOLUM               ;24
        KALAN              ;25
        SAATSIRA            ;26
        TARIHSIRA           ;27
        SICAKSIRA           ;28
        NEMSIRA             ;29
        BEKLEME             ;31
        SSAYI              ;32

         XXXXXXXX
         YYYYYYYY

 ENDC

 ORG     0
       GOTO    INIT

BDELAY    MOVLW   H'18'
        MOVWF   TICKS
BDELAY_   NOP
       DECFSZ  TICKS,F
        GOTO    BDELAY_
        RETURN

BDELAY2    MOVLW   H'FF'
        MOVWF   TICKS
BDELAY2_   NOP
       DECFSZ  TICKS,F
        GOTO    BDELAY2_
        RETURN

_BDELAY   MOVLW   H'0C'       ;HALF WAIT!
        MOVWF   TICKS
_BDELAY_  NOP
       DECFSZ  TICKS,F
        GOTO    _BDELAY_
        RETURN

DELAY   MOVWF   TICKS

DEL1    BTFSS   INTCON,T0IF     ; wait for TMR0 to roll over
 GOTO    DEL1
```

```
 CLRF    INTCON
 DECFSZ  TICKS,F
      GOTO    DEL1
      RETURN

      ;{COM PORTUNDAN VERI YOLLAMA}

OUTCOM    MOVLW   8
       MOVWF   XX
       BSF     PORTA,0    ;START BIT
       CALL    BDELAY

CDONGU    RRF     OXX,F
       BTFSC   STATUS,0
       GOTO    CLEAR
       BSF     PORTA,0
       GOTO    DEVAM1
CLEAR     BCF     PORTA,0
DEVAM1    CALL    BDELAY

        DECFSZ  XX,F
        GOTO    CDONGU

        BCF     PORTA,0       ;STOP BIT
        CALL    BDELAY

        RETURN

      ;{COM PORTUNDAN VERI ALMA}
INCOM     MOVLW   8
       MOVWF   XX
       CLRF    IXX

BEKLE1    BTFSS   PORTA,1
       GOTO    BEKLE1
       CALL    _BDELAY
       BTFSS   PORTA,1
       GOTO    BEKLE1

CDONGU2   CALL    BDELAY
       BTFSC   PORTA,1
       GOTO    DEVAM2
       BSF     STATUS,0
       GOTO    DEVAM3
DEVAM2    BCF     STATUS,0
DEVAM3    RRF     IXX,F

        DECFSZ  XX,F
        GOTO    CDONGU2

        CALL    BDELAY

        RETURN
```

```
        ; {DS1820 ISI BILGISINI ALMA!}
DISABLEINT
 BSF     STATUS,RP0      ; select register bank 1
 BCF     OPTREG,7
 BCF     STATUS,RP0      ; reselect bank 0
       RETURN
ENABLEINT
 BSF     STATUS,RP0      ; select register bank 1
 BSF     OPTREG,7
 BCF     STATUS,RP0      ; reselect bank 0
       RETURN

DSDELAY     MOVWF TICKS2
DSDELAY_    NOP
    NOP
      NOP
    NOP
    NOP
    NOP
    NOP
         DECFSZ  TICKS2,F
         GOTO    DSDELAY_
         RETURN

M1HI       BSF STATUS,RP0           ;goto to page 1
         BSF TRISA,4              ;make RA2 an input
         BCF STATUS,RP0           ;back to page 0
         RETURN                   ;master 1 wire line hi

M1LO       BCF PORTA,4
         BSF STATUS,RP0           ;goto to page 1
         BCF TRISA,4              ;make RA2 an output
         BCF STATUS,RP0           ;back to page 0
         RETURN                   ;master 1 wire line low

TXBYTE1     MOVLW H'8'              ;eight bits of data
         MOVWF XX                 ;a counter for bits
TX1LOOP     RRF ISIX,f              ;roll rightmost bit into carry
         BTFSS STATUS,0           ;if carry 0 want low bit output
         GOTO TXHIBIT1            ;else send a 1
         CALL M1LO
         CALL M1HI
         MOVLW H'8'
         CALL DSDELAY
         GOTO TXDONE1             ;are we finished?
TXHIBIT1    CALL M1LO
         MOVLW H'8'
         CALL DSDELAY
         CALL M1HI
TXDONE1     DECFSZ XX,f             ;1 less data bit, skip when 0
         GOTO TX1LOOP             ;more bits left
         RETURN
```

```
RXBYTE1     MOVLW H'8'              ;eight bits of data
        MOVWF XX                ;a counter for bits
        CLRF  ISIX

RX1LOOP     CALL M1LO               ;a quick low pulse
        NOP
        CALL M1HI               ;and back high
   NOP
   NOP
   NOP
   NOP
   NOP
        NOP
        BTFSC PORTA,4
        GOTO RXHIBIT
        BCF  STATUS,0
        GOTO RXDONE
RXHIBIT     BSF  STATUS,0
RXDONE      RRF  ISIX,f             ;roll carry into RXREG MSB
        MOVLW H'8'              ;delay 80 microseconds 0C
        CALL  DSDELAY
        DECFSZ XX,f             ;1 less data bit, skip when zero
        GOTO RX1LOOP            ;more bits left
        RETURN


RESETDS  CALL M1HI
       CALL M1LO
       MOVLW H'3C'
       CALL DSDELAY
       CALL M1HI
       MOVLW H'28'
       CALL DSDELAY
       RETURN

ISIAL    CALL   DISABLEINT
       CALL   RESETDS

       MOVLW  H'CC'
       MOVWF  ISIX
       CALL   TXBYTE1
       MOVLW  H'44'
       MOVWF  ISIX
       CALL   TXBYTE1

       MOVLW H'50'
       CALL  DSDELAY

       CALL   RESETDS

       MOVLW  H'CC'
       MOVWF  ISIX
       CALL   TXBYTE1
       MOVLW  H'BE'
       MOVWF  ISIX
       CALL   TXBYTE1
```

```
        CALL   RXBYTE1
        MOVF   ISIX,0
        MOVWF  VALUE

        CALL   RXBYTE1
        MOVF   ISIX,0
        MOVWF  SING

        CALL   RESETDS
        CALL   ENABLEINT

        BTFSC  SING,0
        GOTO   SETBT
        BTFSC  SING,7
        GOTO   ERRORDS
        GOTO   DONEDS
SETBT   BTFSS  SING,7
        GOTO   ERRORDS
        GOTO   DONEDS
ERRORDS CLRF   SING
        MOVLW  0
        MOVWF  VALUE
DONEDS  MOVLW  H'FF'
        XORWF  VALUE,W
        btfsS  STATUS,Z
        GOTO   BBISI
        CLRF   VALUE
        CLRF   SING
        RETURN
BBISI   MOVLW  H'7F'
        XORWF  VALUE,W
        btfsS  STATUS,Z
        RETURN
        CLRF   VALUE
        CLRF   SING
        RETURN



;dotmatrix display
AYLAR  ADDWF   PCL,F
     DT     0,20,1D,20,1F,20,1F,1F,20,1F,20,1F,20,0,0,0,0
     RETURN

SSEG   ADDWF   PCL,F
     ;DT     77,14,6E,3E,1D,3B,7B,16,7F,3F,63,0F,0A,8,5,1,0F,0,0
           ;0  1  2  3  4  5  6  7  8  9  C  o  / - : . %
     ;DT     77,24,3B,3E,6C,5E,5F,34,7F,7E,53,78,2,8,5,1,7,0,0
     ;normal  0  1  2  3  4  5  6  7  8  9  C  o  / - : . %

     DT     77,41,3B,6B,4D,6E,7E,43,7F,6F,36,0F,0A,8,5,1,0F,0,0
    ;ESKIKALIN0  1  2  3  4  5  6  7  8  9  C  o  / - : . %

     RETURN
```

```
BAKMA  MOVLW   H'77'
       MOVWF   OXX
       CALL    OUTCOM
       MOVLW   0
       MOVWF   DECOD
       MOVLW   H'0A'
       CALL    SEGMENT
BAKMA_ CALL    INCOM
       MOVLW   H'76'
       XORWF   IXX,W
       btfss   STATUS,Z
       GOTO    BAKMA_
       NOP

       MOVLW   1
       MOVWF   DECOD
       MOVLW   H'0A'
       CALL    SEGMENT

       CALL    INCOM
       MOVF    IXX,W
       MOVWF   SNY
       CALL    INCOM
       MOVF    IXX,W
       MOVWF   DAK
       CALL    INCOM
       MOVF    IXX,W
       MOVWF   SAAT
       CALL    INCOM
       MOVF    IXX,W
       MOVWF   GUN
       CALL    INCOM
       MOVF    IXX,W
       MOVWF   AY
       CALL    INCOM
       MOVF    IXX,W
       MOVWF   YIL

       MOVLW   3
       MOVWF   DECOD
       MOVLW   H'0A'
       CALL    SEGMENT

       CALL    INCOM
       MOVF    IXX,W
       MOVWF   BEKSAAT
       CALL    INCOM
       MOVF    IXX,W
       MOVWF   BEKTARIH
       CALL    INCOM
       MOVF    IXX,W
       MOVWF   BEKSICAK
       CALL    INCOM
       MOVF    IXX,W
       MOVWF   BEKNEM

       MOVLW   4
       MOVWF   DECOD
```

```
      MOVLW   H'0A'
      CALL    SEGMENT

      CALL    INCOM
      MOVF    IXX,W
      MOVWF   SAATSIRA

      CALL    INCOM
      MOVF    IXX,W
      MOVWF   TARIHSIRA

      CALL    INCOM
      MOVF    IXX,W
      MOVWF   SICAKSIRA

      CALL    INCOM
      MOVF    IXX,W
      MOVWF   NEMSIRA

      CALL    INCOM
      MOVF    IXX,W
      MOVWF   SSAYI

      MOVLW   5
      MOVWF   DECOD
      MOVLW   H'0A'
      CALL    SEGMENT

BAKMA2_
      CALL    INCOM
      MOVLW   H'75'
      XORWF   IXX,W
      btfss   STATUS,Z
      GOTO    BAKMA2_
      RETURN


ARTTIR INCF     SNY,F
      MOVLW    H'3C'
      XORWF    SNY,W
      btfss STATUS,Z
      RETURN
      CLRF     SNY
      INCF     DAK,F
      MOVLW    H'3C'
      XORWF    DAK,W
      btfss STATUS,Z
      RETURN
      MOVLW    H'01'
      MOVWF    DAK
      INCF     SAAT,F
      MOVLW    H'18'
      XORWF    SAAT,W
      btfss STATUS,Z
      RETURN
```

```
      CLRF     SAAT
      INCF     GUN,F

      MOVF     AY,W
      CALL     AYLAR
      MOVWF    ZZ

      MOVLW    H'02'
      XORWF    AY,W
      btfsS    STATUS,Z
      GOTO     AYTAM

      BTFSC    YIL,0
      GOTO     AYTAM
      BTFSC    YIL,1
      GOTO     AYTAM
      INCF     ZZ,F

AYTAM  MOVF     ZZ,W
      XORWF    GUN,W
      btfss STATUS,Z
      RETURN
      MOVLW    1
      MOVWF    GUN
      INCF     AY,F
      MOVLW    H'0D'
      XORWF    AY,W
      btfss STATUS,Z
      RETURN
      MOVLW    1
      MOVWF    AY
      INCF     YIL,F
      RETURN

YAZSAAT
       MOVLW   H'03'
       XORWF   AY,W
       btfsS   STATUS,Z
       GOTO    IKINCI
       MOVLW   H'01'
       XORWF   GUN,W
       btfsS   STATUS,Z
       GOTO    IKINCI
       MOVLW   H'01'
       XORWF   SAAT,W
       btfsS   STATUS,Z
       GOTO    IKINCI
       MOVLW   H'00'
       XORWF   DAK,W
       btfsS   STATUS,Z
       GOTO    IKINCI
       INCF    SAAT,F
IKINCI  MOVLW   H'10'
       XORWF   AY,W
```

```
        btfsS   STATUS,Z
        RETURN
        MOVLW   H'29'
        XORWF   GUN,W
        btfsS   STATUS,Z
        RETURN
        MOVLW   H'02'
        XORWF   SAAT,W
        btfsS   STATUS,Z
        RETURN
        MOVLW   H'00'
        XORWF   DAK,W
        btfsS   STATUS,Z
        RETURN
        DECF    SAAT,F
        RETURN


SEGMENT CALL    SSEG
        MOVWF   YY

        BSF     PORTA,3
        BSF     PORTA,2
        BSF     PORTB,7

        ;BTFSC   DECOD,2
        ;BSF     YY,7
        BTFSS   DECOD,2
        BSF     YY,7

        ;MOVF    YY,0
        COMF    YY,W
        MOVWF   PORTB

        BTFSS   DECOD,0
        BCF     PORTA,3
        BTFSS   DECOD,1
        BCF     PORTA,2


        CALL    BDELAY2
        CALL    BDELAY2
        CALL    BDELAY2

        RETURN

DIVO    MOVWF   YY
        CLRF    BOLUM
        CLRF    KALAN
DIVO_   INCF    BOLUM,F
        MOVF    YY,W
        MOVWF   KALAN
        MOVLW   H'0A'
        SUBWF   YY,F
```

```
      BTFSC   STATUS,C
      GOTO    DIVO_
      DECF    BOLUM,F
      RETURN

CARPMA  CLRF    YY
CARPMA_ MOVF    BOLUM,W
      ADDWF   YY,F
      DECFSZ  KALAN,F
      GOTO    CARPMA_
      MOVF    YY,W
      MOVWF   BOLUM
      RETURN

CLOCK   MOVF    SAAT,W
      CALL    DIVO
      MOVLW   0
      MOVWF   DECOD
      MOVF    BOLUM,W
      CALL    SEGMENT
      MOVLW   1
      MOVWF   DECOD
      MOVF    KALAN,W
      CALL    SEGMENT

      MOVLW   2
      MOVWF   DECOD
      BTFSC   SNY,0
      GOTO    SILIK
      MOVLW   H'0E'
      GOTO    TOKMM
SILIK   MOVLW   H'12'
TOKMM   CALL    SEGMENT
      MOVF    DAK,W
      CALL    DIVO
      MOVLW   3
      MOVWF   DECOD
      MOVF    BOLUM,W
      CALL    SEGMENT
      MOVLW   4
      MOVWF   DECOD
      MOVF    KALAN,W
      CALL    SEGMENT
      MOVF    SNY,W
      CALL    DIVO
      MOVLW   5
      MOVWF   DECOD
      MOVF    KALAN,W
      CALL    SEGMENT
      RETURN

DATE    MOVLW   5
      MOVWF   DECOD
      MOVLW   H'12'
      CALL    SEGMENT
      MOVF    AY,W
      CALL    DIVO
      MOVLW   4
```

```
        MOVWF   DECOD
        MOVF    KALAN,W
        CALL    SEGMENT
        MOVLW   3
        MOVWF   DECOD
        MOVF    BOLUM,W
        CALL    SEGMENT
        MOVF    GUN,W
        CALL    DIVO
        MOVLW   2
        MOVWF   DECOD
        MOVLW   H'0C'
        CALL    SEGMENT
        MOVLW   1
        MOVWF   DECOD
        MOVF    KALAN,W
        CALL    SEGMENT
        MOVLW   0
        MOVWF   DECOD
        MOVF    BOLUM,W
        CALL    SEGMENT
        RETURN



TERMO   MOVLW   0
        XORWF   SING,W
        btfsS STATUS,Z
        GOTO    EKSI

        MOVLW   0
        MOVWF   DECOD
        MOVF    VALUE,W
        MOVWF   YY
        BCF     STATUS,C
        RRF     YY,W
        CALL    DIVO
        MOVF    BOLUM,W
        CALL    SEGMENT
        MOVLW   1
        MOVWF   DECOD
        MOVF    KALAN,W
        CALL    SEGMENT
        MOVLW   2
        MOVWF   DECOD
        MOVLW   H'12'
        CALL    SEGMENT
        MOVLW   3
        MOVWF   DECOD
        MOVLW   H'0B'
        CALL    SEGMENT
        MOVLW   4
        MOVWF   DECOD
        MOVLW   H'0A'
        CALL    SEGMENT
        MOVLW   5
        MOVWF   DECOD
        MOVLW   H'12'
```

```
      CALL    SEGMENT
      RETURN

EKSI    MOVLW   5
      MOVWF   DECOD
      MOVLW   H'0B'
      CALL    SEGMENT
      MOVLW   4
      MOVWF   DECOD
      RRF     VALUE,W
      CALL    DIVO
      MOVF    BOLUM,W
      CALL    SEGMENT
      MOVLW   3
      MOVWF   DECOD
      MOVF    KALAN,W
      CALL    SEGMENT
      MOVLW   2
      MOVWF   DECOD
      MOVLW   H'12'
      CALL    SEGMENT
      MOVLW   1
      MOVWF   DECOD
      MOVLW   H'0D'
      CALL    SEGMENT
      MOVLW   0
      MOVWF   DECOD
      MOVLW   H'12'
      CALL    SEGMENT
      RETURN


NEMDSP  MOVF    VALUE,W
      MOVWF   ZZ
      BCF     STATUS,C
      RRF     ZZ,F


      MOVLW   1
      XORWF   AY,W
      btfsC   STATUS,Z
      GOTO    NTAMAM
      MOVLW   2
      XORWF   AY,W
      btfsC   STATUS,Z
      GOTO    NTAMAM

      MOVLW   H'0A'
      XORWF   AY,W
      btfsC   STATUS,Z
      GOTO    NTAMAM
      MOVLW   H'0B'
      XORWF   AY,W
      btfsC   STATUS,Z
      GOTO    NTAMAM
      MOVLW   H'0C'
      XORWF   AY,W
```

```
      btfsC   STATUS,Z
      GOTO    NTAMAM

      MOVLW   2
      GOTO    NTAMAM2

NTAMAM  MOVLW   1
NTAMAM2 MOVWF   KALAN

      MOVF    ZZ,W
      MOVWF   BOLUM
      CALL    CARPMA


      MOVLW   H'19'
      ADDWF   BOLUM,W
      CALL    DIVO


      MOVLW   H'0A'
      XORWF   BOLUM,W
      btfsC   STATUS,Z
      GOTO    NYUZ
      MOVLW   H'0B'
      XORWF   BOLUM,W
      btfsC   STATUS,Z
      GOTO    NYUZ
      MOVLW   H'0C'
      XORWF   BOLUM,W
      btfsC   STATUS,Z
      GOTO    NYUZ
      MOVLW   H'0D'
      XORWF   BOLUM,W
      btfsC   STATUS,Z
      GOTO    NYUZ
      MOVLW   H'0E'
      XORWF   BOLUM,W
      btfsC   STATUS,Z
      GOTO    NYUZ
      MOVLW   H'0F'
      XORWF   BOLUM,W
      btfsC   STATUS,Z
      GOTO    NYUZ

      MOVLW   5
      MOVWF   DECOD
      MOVLW   H'12'
      CALL    SEGMENT
      MOVLW   4
      MOVWF   DECOD
      MOVF    KALAN,W
      CALL    SEGMENT
      MOVLW   3
      MOVWF   DECOD
      MOVF    BOLUM,W
      CALL    SEGMENT
      MOVLW   2
```

```
      MOVWF   DECOD
      MOVLW   H'10'
      CALL    SEGMENT
      MOVLW   1
      MOVWF   DECOD
      MOVLW   H'12'
      CALL    SEGMENT
      MOVLW   0
      MOVWF   DECOD
      MOVLW   H'12'
      CALL    SEGMENT

      RETURN
NYUZ    MOVLW   H'0A'
      SUBWF   BOLUM,F
      MOVLW   5
      MOVWF   DECOD
      MOVF    KALAN,W
      CALL    SEGMENT
      MOVLW   4
      MOVWF   DECOD
      MOVF    BOLUM,W
      CALL    SEGMENT
      MOVLW   3
      MOVWF   DECOD
      MOVLW   H'01'
      CALL    SEGMENT
      MOVLW   2
      MOVWF   DECOD
      MOVLW   H'10'
      CALL    SEGMENT
      MOVLW   1
      MOVWF   DECOD
      MOVLW   H'12'
      CALL    SEGMENT
      MOVLW   0
      MOVWF   DECOD
      MOVLW   H'12'
      CALL    SEGMENT
      RETURN



ATA_SAAT  MOVF    BEKSAAT,W
       MOVWF   BEKLEME
      RETURN
ATA_TARIH  MOVF    BEKTARIH,W
        MOVWF   BEKLEME
      RETURN
ATA_SICAK  MOVF    BEKSICAK,W
        MOVWF   BEKLEME
      RETURN
ATA_NEM    MOVF    BEKNEM,W
        MOVWF   BEKLEME
      RETURN

SART    DECFSZ  BEKLEME,F
      RETURN
```

```
TEKAR_  BCF     STATUS,C
        RLF     SIRA,F
        BTFSS   SIRA,4
        GOTO    DEVAM_
        MOVLW   H'01'
        MOVWF   SIRA
        CALL    ISIAL
DEVAM_
        MOVF    TARIHSIRA,W
        XORWF   SIRA,W
        btfsC   STATUS,Z
        CALL    ATA_TARIH

        MOVF    SAATSIRA,W
        XORWF   SIRA,W
        btfsC   STATUS,Z
        CALL    ATA_SAAT

        MOVF    SICAKSIRA,W
        XORWF   SIRA,W
        btfsC   STATUS,Z
        CALL    ATA_SICAK

        MOVF    NEMSIRA,W
        XORWF   SIRA,W
        btfsC   STATUS,Z
        CALL    ATA_NEM
        NOP

        MOVLW   H'0'
        XORWF   BEKLEME,W
        btfsC   STATUS,Z
        GOTO    TEKAR_

        BTFSC PORTA,1
        CALL    BAKMA
        NOP
        RETURN


INIT    CLRF    PORTA
        CLRF    PORTB
 BSF    STATUS,RP0      ; select register bank 1
        MOVLW   B'00010010'
        MOVWF   TRISA
        CLRF    TRISB
 MOVLW  4
 MOVWF  OPTREG
 BCF    STATUS,RP0      ; reselect bank 0

        CLRF    IXX
        CLRF    OXX

        CLRF    SNY
        CLRF    DAK
        CLRF    SAAT
```

```
        MOVLW   H'08'
        MOVWF   GUN
        MOVLW   H'03'
        MOVWF   AY
        CLRF    YIL

        MOVLW   H'50'
        MOVWF   SYC1
        CALL    ISIAL
                        ;SABIT ATAMALAR
        MOVLW   1
        MOVWF   SAATSIRA
        MOVLW   2
        MOVWF   TARIHSIRA
        MOVLW   4
        MOVWF   SICAKSIRA
        MOVLW   8
        MOVWF   NEMSIRA
        MOVLW   5
        MOVWF   BEKLEME
        MOVWF   BEKSAAT
        ;MOVLW   0
        MOVWF   BEKTARIH
        MOVWF   BEKSICAK
        MOVWF   BEKNEM

        MOVLW   H'1'
        MOVWF   SIRA


ANA     MOVF    TARIHSIRA,W
        XORWF   SIRA,W
        btfsC   STATUS,Z
        CALL    DATE

        MOVF    SAATSIRA,W
        XORWF   SIRA,W
        btfsC   STATUS,Z
        CALL    CLOCK

        MOVF    SICAKSIRA,W
        XORWF   SIRA,W
        btfsC   STATUS,Z
        CALL    TERMO

        MOVF    NEMSIRA,W
        XORWF   SIRA,W
        btfsC   STATUS,Z
        CALL    NEMDSP

        NOP

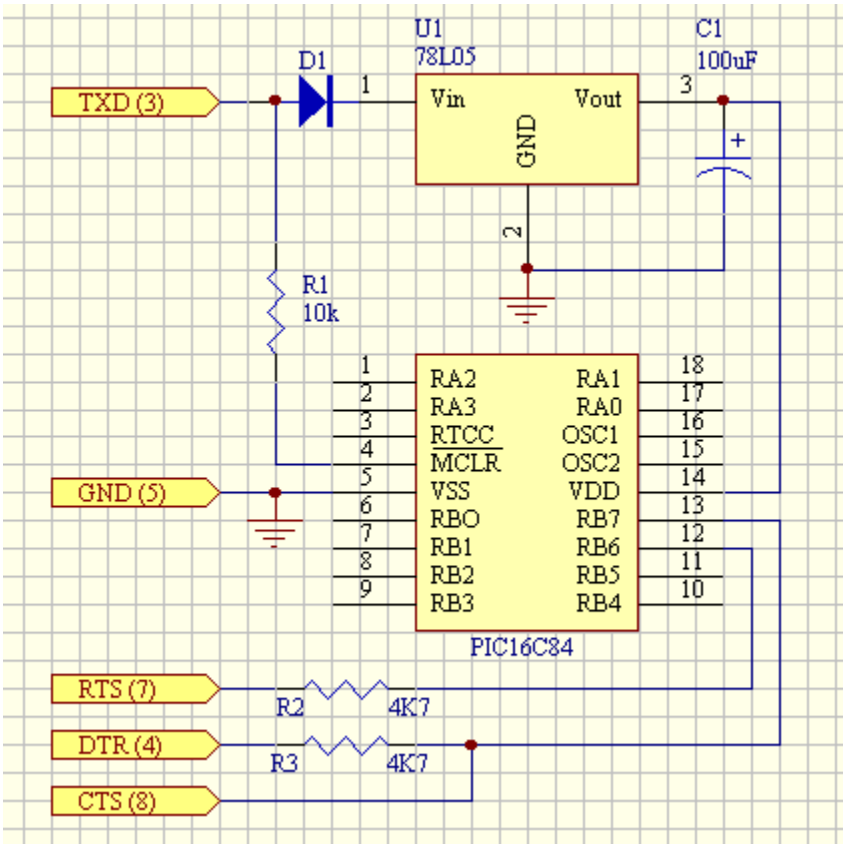
        BTFSS   INTCON,T0IF
 GOTO    ANA
        CLRF    INTCON
        DECFSZ  SYC1,F
        GOTO    ANA
```

```
        MOVLW   H'50'
        MOVWF   SYC1
        CALL    ARTTIR
        CALL    YAZSAAT
        CALL    SART
        GOTO    ANA

 END
```

[6.3.2] Kullanim Teorileri

Pic mikroislemcileri içlerinde bulundurduklari eepromlari doldurulup standart devre sartlari verildiginde sirasal olarak yükelen programi islerler. Bu yükleme islemi için genel olarak yetenekli programlayicilar kullanilabilecegi gibi asagida yer alan temel bir programlayicida isimizi görecektir;

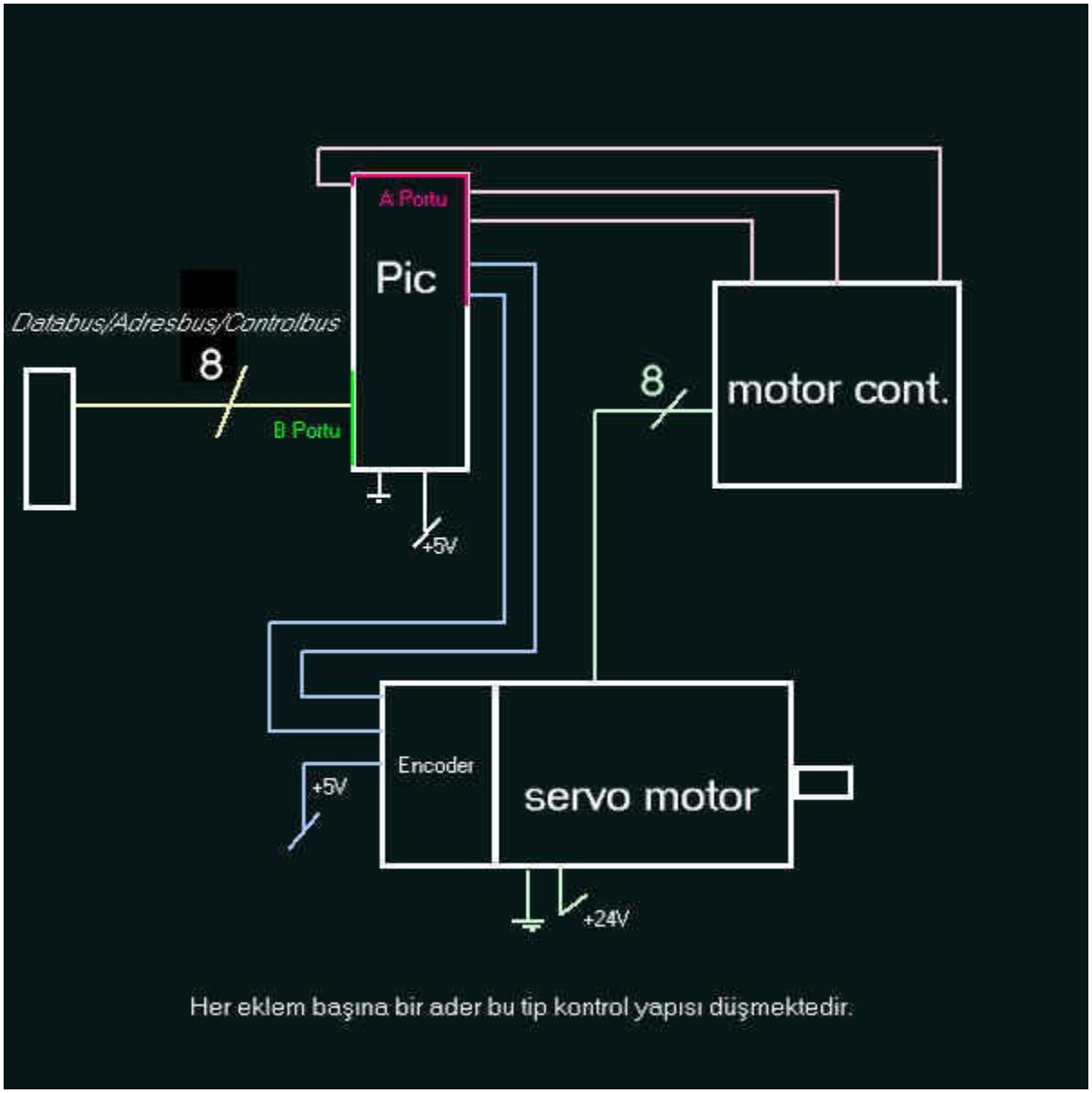


Daha Fazla Pic Programcısı için:

BENIM 16f84 programlayıcım için:

Görüldügü gibi programlayici com portundan Pc baglanarak çalismaktadir. Pic seris mikrochipler MCLR ucundan islevsel yapilarina karar verirler. Eger MCLR ucundan 11-14V arasi bir voltaj verirseniz pic programlama moduna gececektir, geçisten sonra RB7 data için RB6'da clock için kullanilarak sirasal olarak program pic içerisine atilabilmektedir. MCLR ucu 2-5V arasina getirildiginde ise pic standart çalisma moduna geçerek hafizasinda yer alan porgrami isletmektedir. Vdd +5 volt, Vss 0V ve OSC1,OSC2 arasina seramik bir 4Mhz'lik 30pf bir kristal-kondansatör bagi atilmasi mecburidir.

Iki portu bulunan pic 16f84 mikroislemcisi temel olarak iki aygitin arasinda islev görecegi için portlarinin birini kaynak aygita digerini ise hedef aygita verebiliriz. Asagida encoderli bir motor kontrolü için ayrilmis pic'i görmektesiniz;



Görüldügü gibi dar kapsamli olan 5 bitlik A portu Seri baglanti ve encoder çözümü için daha genis olan B portu ise motor kontrölcüsüne komut vermek için kullanililmaktadir.

[6.3.3] Çesitleri

Picler genel olarak yapilarina göre kodlamayla ayrilmislardir, 12-15-16-17-18 serileri olmak üzere. kullandigimiz 16 serisi bize en uygun oldugunu yukarida açiklamistik fakat ilerki asamalarda ihtiyaç duyulacak ana bir kontrolcü mikrochip için pic serisinin 16F877 kodlu chipini kullanacagiz. Bu chip 16F84 farkli olarak yüksek hiz ve 4 adet portla kendini göstermektedir. Komut yapilari ve hafiza düzeni 16F84 aynidir. Dikkat edilmesi gereken sadece portlarinindan birisinin ADC özeliginin olduğudur.

Daha Ayrıntılı bilgi için: **Microchip**

[6.4] Piyasa Analizi

Piyasada genel olarak 80C51 veya 80C35 gibi hantal mikrokontrolerler kulanilmaktadir. Alternatifleri olan MC68 serisi ise hem pahali hemde çok çesitli devre elemanlarina ihtiyaç duymaktadir. Dikkat edilirse piyasa uygulamalari yavas yavas pic serisine kaymaktadir. Aktif kullanilan devrelerde göze çarpan en büyük etki genis kartlar ve yüksek güç tüketimidir. Ihtiyac duyulan islevler için yeni çikan çogu chip çiktigi firma disinda kabul görmemektedir çünkü kullanima açilan yanlari kisitli vede dökümantasyon saklanmaktadir. Seçim için fazla zorlanmadan pic seçilebilir, bulmak ve çalistirmak kolaydir.

7. Pc Mimarisi ve Robot Kontrolü

Önceki konularda bahsedildiği gibi robot kontrolünün oluşması ve gelişmesinde PC'lerin yeri çok büyüktür. Robot az çok yapabileceklerine karar verip bir esneklik kazdırabilir ama asıl işlevleri sağlayan akıllı yazılımlar ancak bir Pc kaynakları sayesinde robotu işlevsel ve tamamlayıcı bir etki verebilir. Bu konu aynı adres bilmeyen bir şahsın adres aramasına bnzetilebilinir, eğer adama adres hakkında temel ipuçları verilip karar vermesi konusunda yardımcı olunursa adresi kolayca ve hatasız bulabilir. Ve yine en önemli olan değişkenlik, algılama öğeleri ancak yapısal bir PC ile sağlanabilir.

Görüldüğü gibi Pc-Robotlar bir bütündür eğer bu iki parçayı birbirinden ayırısak olayımız amacımız dışına çıkar. Amaça uygun işlevsel program yapısını bir sonraki konuda inceleyeceğiz fakat bir önemli nokta olan Pc ile Robot kontrol kitinin haberleşmesi için Pc tabanlı kullanılabilecek PORT(Dışsal giriş/çıkış birimi) yapılarını genel olarak inceleyelim.

7.1 Mikroişlemciyle çok yönlü iletişim:

Genel olarak micro işlemcilerde üç ana veri yolu bulunmaktadır, bunlar DATA BUS,CONTROL BUS vede ADRES BUS dır. Adres bus yani adres veri yolu ile mikroişlemci istenilen aygıtı seçmede kullanır örneğinin ram'ın ilk bloğu gibi, Data veri yolu ise seçilen bölgeden alınan ve verilen (Read/Write) bilgileri mikroişlemci ile dışsal aygıt arasında taşıyan veri yoludur. Kontrol veri yolu'da verilerin yönünü yani yazmamı okumamı olduğuna yada hatalımı sorularına yanıt arayan bir veriyoludur. Buradan çıkarılacak en önemli sonuç bütün veri yolları aygıtlara paralel olarak dağılmaktadır seçimsel veya yapısal bir arızada iki aygıtın birbirine karışması çok kolay olacaktır. Bütün bu işlemlerin sırasına mikroişlemci içerisindeki kristal yani mikrochipin hızı karar vermekle birlikte yazılım ve donanımsal yazılımlarda bu hızı etkileyebilirler, bu etileşim dışında bire bir yani eş zamanlı çalışan aygıtlarada real time(gerçel zamanlı) devices(aygıtlar) denir.

Örneğin temel bir PC de LPT1 portundan 1 bilgisini göndermek için aşşağıdaki programı kullanırsak;

```
1. mov   dx,378h
2. mov    al,1
3. out    dx,al
4. ret
```

Burada asıl işlevi yapan OUT komutudur hedef DX registerı(yazmacı) içerisindeki porttan AL yazmacı içerisindeki bir bytelık bilgiyi gönderir. Şimdi bunu işlemci açısından işlersek birinci basamakta DX yazmacı içerisine 378 hex bilgisi koy, AL yazmacı içerisine 1 bilgisini koy, Porttan çıkış yap hedef port DX yazmacı bilgi AL yazmacı yani adres veri yolunda DX yazmacını data veri yolunda AL yazmacını ve kontrol veri yolunda ise çıkış işlevini yerleştirerek bir birim zaman bekle.

Ana işlemler bu kadar basit olsada dikkat edilmesi gereken zamanlama ve doğacak hata kontrolünün yapılmasıdır.

7.2 Dışsal Aygıtlarla iletişim portları:

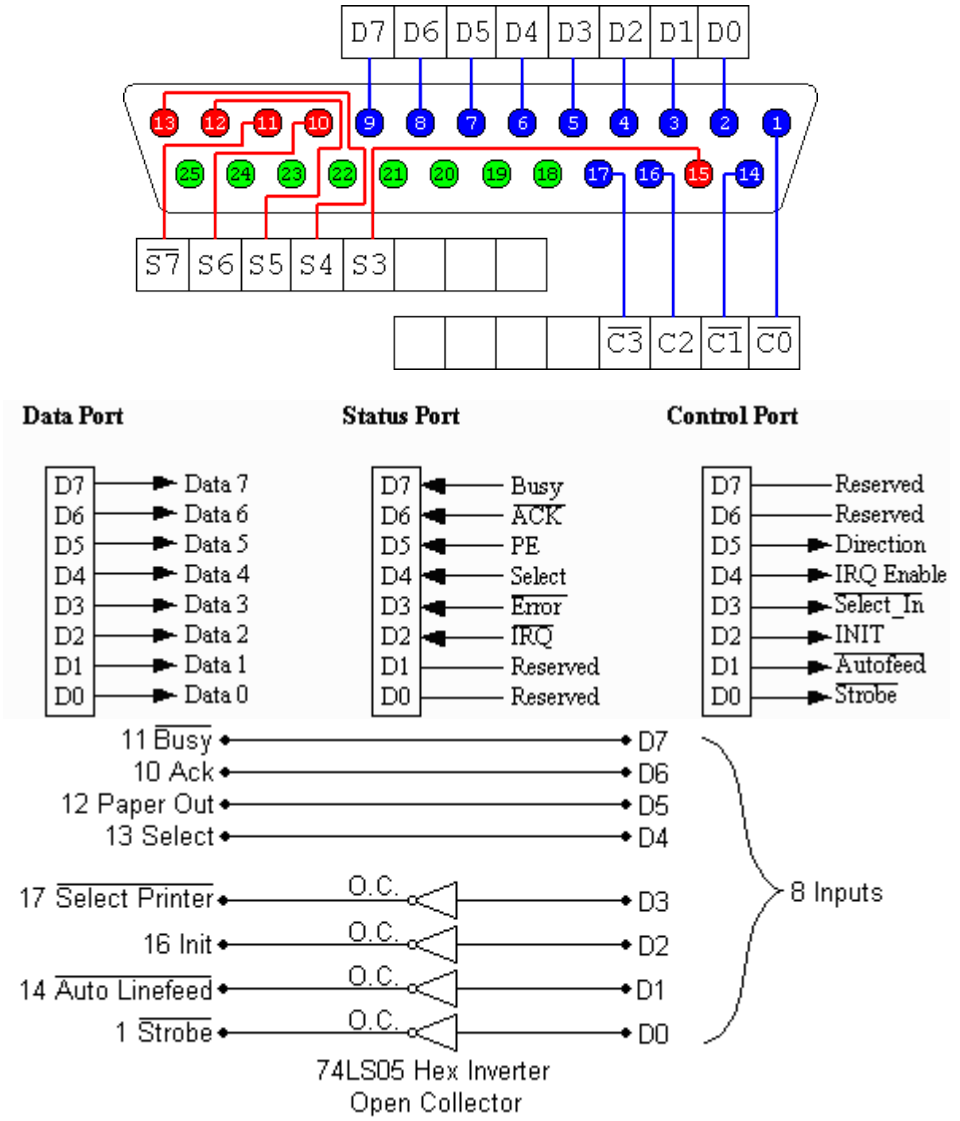
Genel kullanımda yeralan intel uyumlu chipsetler sahip bilgisayarlarda üç ana port yani dışsal IO yer almaktadır bunlar:

- Paralel Port
- Seri Port
- USB portu

Şimdi kısaca bu portları inceleyerek kullanım seçim yapmaya çalışalım;

[7.2.1] Paralel Port:

Pc'de yer alan en eski portlardan biridir paralel port. Adındanda anlaşılabileceği gibi bilgiler paralel olarak yani birden fazla kablo ile taşınmaktadır. Sekiz bitlik bir veri yoluna sahiptir amacına uygun olarak sadece çıkış için tasarlanmış olmasına rağmen içerisinde yer alan dışsal hata kodları için ayrılmış giriş uçları kullanılarak sekizli olarak giriş çıkış birimi olrak kulanılabilmektedir. Populer olarak yazıcılar tarafından kullanılan LPT portları yazıcı dışında tarayıcı, ide çıkışı, zip drive ve kamera içinde kullanılabilmektedir. Portun yapısı ve işleyişi çok kolay anlaşılması ve kolay uygulamaları sayesinde günümüze kadar popülaritesini korumaktadır. Aşşağıdaki resimde D25 yani 25 pinlik bir paralel portun pinlerinin görevleri görülmektedir. 1 ucu strobe yani darbesel frekanslama için kullanılan bir uçtur.2-8 Arasındaki pinler sadece çıkış için kullanılan sekiz bitlik(1 byte) bir bölümümdür. 14-16-17 nolu pinler ise yine çıkış işlevi için kullanılsada amaç kontroldür, bu pinler çift yönlü çalışabilir. 10-11-12-13-15 ise sadece giriş olarak kullanılan hata kontrol uçlarıdır.18-25 uçları kullanılmamakla birlikte GND(ground yani toprak) olarak kullanılabilir.



Burada ise genel olarak sekiz girişin LPT portuna emüle edilmesi görülmektedir. 11-10-12-13 uçları genel olarak giriş için kullanılabilinsede 17-16-14-1 uçları ters çevrilmiş olarak karşımıza çıktığından yani open kollektör (yükseltilmiş uç) olarak konumlandığından bir ttl tersleyici ile sonuca ulaşılabilmektedir.

Pc'ningenel olarak portları sekiz bitlik olduğundan 24 bitlik bir bilgi ancak üç port ile çözümlenebilmektedir.

Bir Pc'de maximum dört adet paralel port bulunacak şekilde adresleme yapılmıştır. Temel hedef port adresleri hafızada yer almaktadır, adres 0x40:0x8 adresinden başlayarak 4 word uzunluğundadır (ilk olarak düşük seviyeli byte'ın verildiği unutulmamalıdır). (Bu adresleri ekranda yazan program). Geçtiğimiz yıllarda eski teknolojiye yeni gelişmeler eklenerek iki tip paralel iletişim standardı ortaya çıkmıştır bunlar ECP ve EPP dir. Farklı olarak uygulanan yüksek hız ve kontrol için donanımsal bir programlamadır ama eski standartlar tamamiyle korunmuştur bu yüzdende belirgin bir ilerleme görülmemektedir.

Sonuc olarak paralel port her ne kadar kullanışlı ve kolay görünsede bu çerçevede yapılmış olan tüm dışsal aygıtların sorunlar yaşadığı açıktır özellikle hız ve kayıp veri konusunda ayrıca 8 çıkış+8 gitiş =16 kablo+2 besleme=toplam 18 adet kablonun bilgisayarla yapılan alet arasında taşınmasıda oldukça zor olmaktadır.

[7.2.2] Seri Port:

Yine ismindende anlaşılabileceği gibi seri olarak bilgilerin aktarıldığı bir portur. Veriler bir hat üzerinden zamansal olarak sıra ile yollanarak istenilen veriler sıra ile karşıya bit bit ulaştırılır. Burada zamanlama yani frekans uyumu çok önemli bir yer almaktadır bunun için protokol kavramı oluşmuştur. Aynı zamanda çift yönlü iletişim ile seri iletişim paralel iletişimin önüne geçerek daha az kablo ile maximum mesafeye veri ulaştırır hale gelmiştir. Veriler TTL boyutundan çıkarak lojik seviyelerini +10V ile -10V arasında düzenlenmiştir. Aşşağıda bir byte'lık bir verinin seri olarak aktarılması görülmektedir;

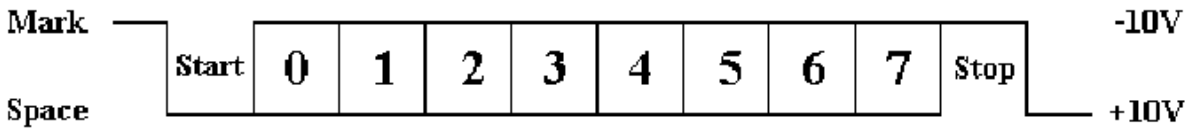


Figure 5 : RS-232 Logic Waveform

Görüldüğü gibi bir bytelık verinin sekiz ayrı biti lojik 0 -10V ve lojik 1'ide +10V olarak belirli periodlarla karşıya aktarılırken aktif low olarak başlangıç manasına gelen START biti ve sonunda da aktif high olarak

STOP biti bulunmaktadır buda protokolün bittiğini belirtir. Seri port paremetrelerine dikkat edilecek olunursa gönderilen verinin bit uzunluğu 5 ile 8 bit arasında değişebildiği gibi verinin hızı vede veri kontrolü için çift yada tek party biti kullanılabilmektedir. Fakat kullanım kolaylığı ve gereksinimler sonucunda partysiz, 9600 baud (baud= sn gönderilen bit sayısı 9600/8=1200 byte sn yollanabiliyor.) hızında ve 8 bitlik veri protokolü kabul görmüştür.



1.Kontrol Biti(CD) 2.Gelen Bilgi(RD) 3.Gönderilen Bilgi(TX) 4.Hazır Sinyali (DTR) 5. Şase (GND)

6. Bilgi Kaydı(DSR) 7. Gönderim İsteği(RTS) 8.Gönderim Hazır (CTS) 9. Dış Seslenme (RI)

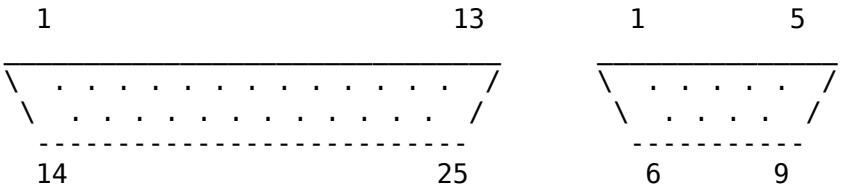
Yukarıdaki sekilde görülen uçlardan bizler sadece aktif olarak tx,rx ve gnd uçlarını kullanmaktayız. Diğer uçlar özel ve geliştirme amaçları için tasarlanmıştır.

Seri olarak giden ve gelen verilerin zamanlaması protokol nezlinde olduğundan kısıtlama sadece bilgisayarın maximum desteğine ve kablonun kapasitesi ile sınırlıdır. Genel olarak kullanılan hızlar aşşağıdaki tabloda yer almaktadır;

Speed (BPS)	Divisor (Dec)	Divisor Latch High Byte	Divisor Latch Low Byte
50	2304	09h	00h
300	384	01h	80h
600	192	00h	C0h
2400	48	00h	30h
4800	24	00h	18h
9600	12	00h	0Ch
19200	6	00h	06h
38400	3	00h	03h
57600	2	00h	02h
115200	1	00h	01h

Table 6 : Table of Commonly Used Baudrate Divisors

Port çıkışı olarak D25 veya D9 luk çıkış şeklinde kullanılmaktadırlar. Temel pin çıkışları aşşağıdaki gibidir;

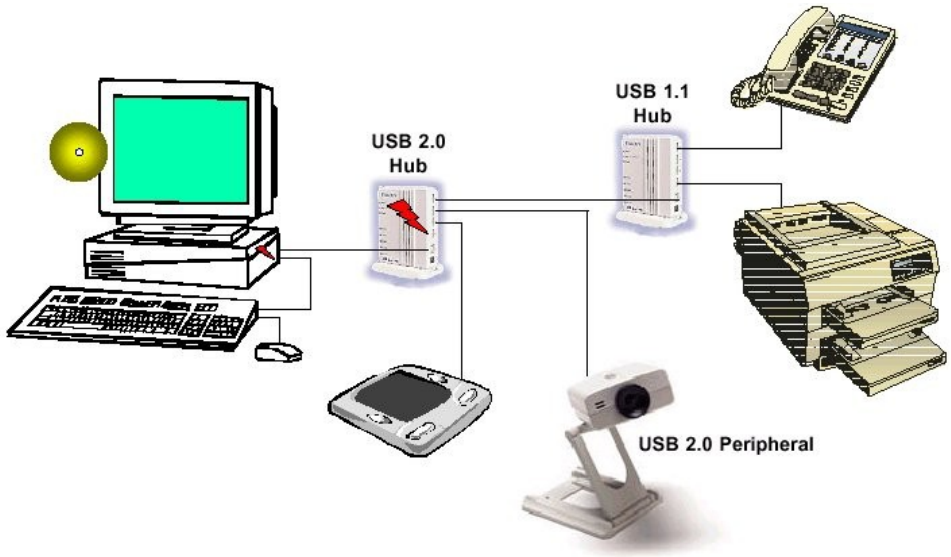


Name (V24)	25pin	9pin	Dir	Genel Tanımı	Açıklama
TxD	2	3	o	Transmit Data	Data
RxD	3	2	i	Receive Data	Data
RTS	4	7	o	Request To Send	Elsıkışma
CTS	5	8	i	Clear To Send	Elsıkışma
DTR	20	4	o	Data Terminal Ready	Durum
DSR	6	6	i	Data Set Ready	Durum
RI	22	9	i	Ring Indicator	Durum
DCD	8	1	i	Data Carrier Detect	Durum
GND	7	5	-	Signal ground	Katılım seviyesi
-	1	-	-	Protective ground	GND

Pc de maximum dört adet seri port bulunabilmektedir. Yine LPT olduğu gibi bu portların adresleri hafızada yer almaktadır yeri 0x40:0x0 dan başlayarak dört word dur. (Bu adresleri ekranda yazan program).

[7.2.3] USB Portu:

Universal Serial Bus yni Genel Seri Veriyolu olarak tanımlanan yeni bir standart olan usb aslında seri porta çok benzemekle beraber ondan farklı olarak genel bir protokol yapısı içerisinde paket teoreminin kullanılmasıdır. Ilk olarak 1995 yılında içerisinde bazı temel ögeleri barındırarak ortaya çıkmış ve zamanla günümüze kadar gelişme göstermiştir. Çalışma sistemindeki paketler aslında bellirli bir uzunlukta olan veri bloklarıdır. Bu Portta farklı olan asıl etmen ise direkt olarak işlemci ile muhatap olunduğundan veri yolu işlemci ile aynı hızda yani veri yolu hızından bağımsız olarak çalışabildiğinden dolayı bilgisayara yani işlemciye göre değişmesidir. Pentium 3 işlemciye sahip bir makinaya takılan bir usb kamera ile Pentium 2 işlemciye takılan aynı kameranın çektiği görüntününü karesel hızı bize bu olayı açıkça göstermektedir. Fakat geliştirilen bu yeni standart ta asıl yapılam istenen tak kullan mantığıyla alakalıdır, yeni alınan veya var olan bir device(araç) makina açıkken dahi kolayca takılıp sökülebilmektedir usb portları üzerinden ayrıca veri yolunun ulaştığı bu korkunç hız bilgisayar ortamlı ağ yapısınıda bu çerçeveye taşınmasını sağlamıştır. Kısmen veri bloklarını incelediğimizde veri bloklarının başında kimden geldiği ve kime gittiği gibi birtakım bilgiler yer almaktadır bu bilgilerden gelen, giden device bir byte ile tanımlandığından bir usb portu üzerinden aynı anda 255 yani maximum bir byte'lık aygıt işletilebilir.



Var olan bu yeni teknolojiyi seçmememin nedeni ise asıl olan güvenlik ve uyumluluk problemleridir.

7.3 Veriyolları ve kart uygulamaları:

Genel kullanımda yeralan intel uyumlu chipsetler sahip bilgisayarlarda üç ana port yani dışsal IO yer almaktadır bunlar:

- ISA

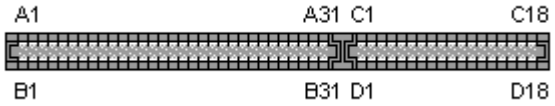
- PCI
- AGP

Şimdi kısaca bu portları inceleyerek kullanım seçim yapmaya çalışalım;

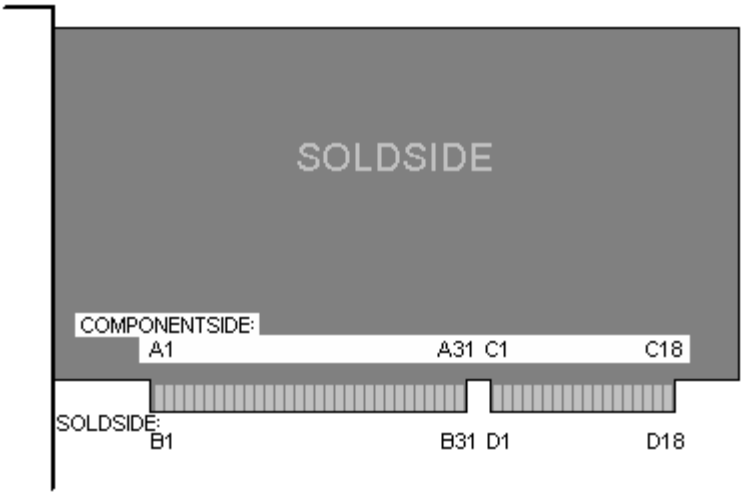
[7.3.1] ISA veriyolu:

Isa veriyolu ilk temel pc olan i8086 bu yana kullanılan en yaygın ve başarılı tabir edilebilecek kolay kullanımlı geniş amaçlı bir porttur. Kelime açılılmı ISA industry standard (Endüstri standardı) dır, 8 mhz'lik bir clock hızına sahiptir fakat düşük hızına ve yapısındaki yavaşlığa rağmen asıl önemini anakart üzerinde olmasından ve direkt olarak PC'nin kalbi olan CPU'ya bağlı olmasıyla kazandığı hız ve esnekliktir. Port temelde 64 bitlik bir pin bağlantı şeması yer almasına karşın sadece 8 adet data ve 16 bitlik bir adres bus'a sahiptir, isa portunun bir sonraki versiyonunda ise Eisa(Extended ISA) adıyla bilinen 16 bit data ve 24 bit adres taşıyabilen portdur. Şu anki P3 ve P2 bilgisayarlarının anakartları temelde isa portu kullanabilmesine karşın artık gelişen diğer portlar yüzünden ISA portu devreden çıkarılmış ve tarihin kalın yaparaklarına gömülmüştür. Yinede bilgisayarın merkezi işlembirimine en çok yaklaşabildiğimiz, kolay ve esnek kullanımlı bu portu incelemek hem PCI portunu hemde temel adres, data bus mantıklarını kavramamaızda etkili olmaktadır.

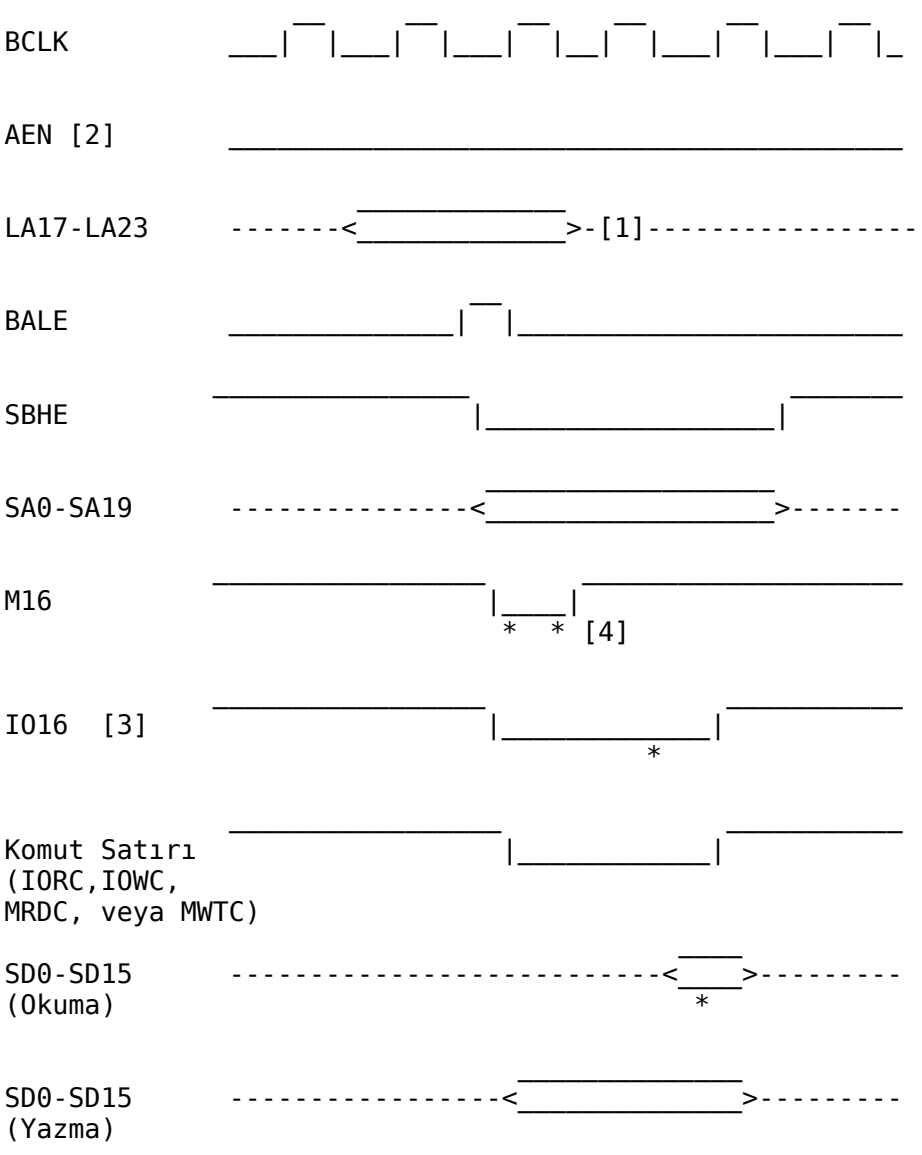
Aşşağıda temel olarak bir ISA veri yolunun pinout(bacak bağlantı) şemasını görmekteyiz:



Bağlantıda görülen A,B,C ve D bölgeleri zamanla geliştirilen ek hizmetleri ve ayrıntıları temsil etmektedir. A bölgesi genel 8 bitlik datayı ve 16 bit adresi barındırıan kaynak bölgedir. B bölgesi ise ilk isa versiyonunda kullanılan temel işaretçilerini taşır. D ve C bölgeleri ise B-A bölgelrinin geliştirilmek istenince eklenen ek bağlantı bacaklarıdır. Aşağıda bu bağlantı yapısına uyan bir ISA kart görülmektedir:



Aşağıda Isa portunun 16 bitlik zamansal aktarım lojik diyagramını görmektesiniz;

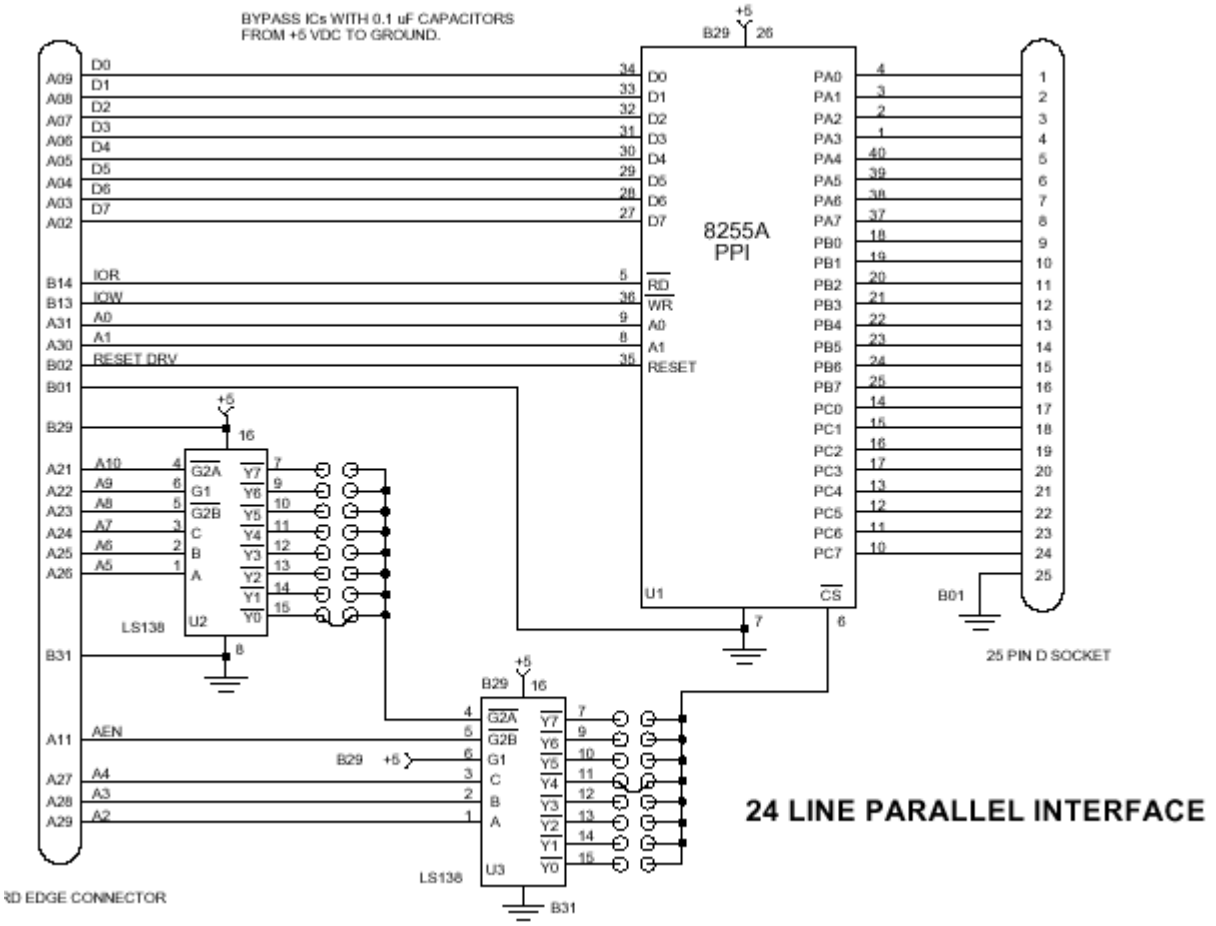


24 bitlik bir paralel giriş çıkış kartı tasarlamak istersek bu uygulamayı Isa veriyolu nasıl yapabiliriz sorusuna yanıt ararken sanırım veri yolunu daha iyi tanıyabiliriz. İlk olarak 8 bitlik bir data kaynağımız olduğunun farkındayız bu yüzden üç adet sekiz bitlik veriyi porta yazmalı veya almalıyız, bu işlemler için en uygun olan Intel firmasına ait 82c55 entegresini kullanabiliriz. Bu entegre yapsında üç adet sekiz bitlik veri yolu ve bir adet control byte ile bu portların in-out yordamını kontrol edebilmekteyiz. O zaman sekiz bitlik data yolunu direkt olarak 82c55 entegresinin Data ucuna bağlayabiliriz ve Adres Enable(Adres seçici) uçlarınada bizim son iki bitlik adres veriyolunu bağlayabilir ve dört farklı adresi 82c55 içerisinden seçerek istenilen porta istenilen bilgiyi yazabiliriz. IoWR(yazma) ve IoRD(okuma) uçlarınıda Isa veriyolundan alarak 82c55 iord ve iowr uçlarına bağlayarak yazma ve silme işlemlerinide gerçekleştirmiş oluruz, geriye sadece Isa veriyolu üzerinde tanımlanacak adres numarasını belirlemek kalmıştır bunun için aşağıdaki tablodan kullanılmayan bir alan seçerek AEN(adres seçildi) ucunu ve Adres uçlarını lojik işlemlere tabi tutarak 82c55'in Aen ucuna veririz. İşte bu sayede temel olarak bir 82c55 ile 24 bitlik 3 ayrı port ve yapıdan oluşan bir giriş çıkış kartı tasarlamış olduk. Dikkat edilmesi gereken nokta adres ve data veriyollarının doğru şekillendirilmesi ve IO uçlarının kullanım mantığıdır.

```
Port Adreleri için karşılık gelen aygıtlar:
Port (hex)
000-00F   DMA Controller
010-01F   DMA Controller (PS/2)
020-02F   Master Programmable Interrupt Controller (PIC)
030-03F   Slave PIC
040-05F   Programmable Interval Timer (PIT)
060-06F   Keyboard Controller
070-071   Real Time Clock
080-083   DMA Page Registers
090-097   Programmable Option Select (PS/2)
0A0-0AF   PIC #2
0C0-0CF   DMAC #2
0E0-0EF   reserved
0F0-0FF   Math coprocessor, PCJr Disk Controller
100-10F   Programmable Option Select (PS/2)
110-16F   Müsait
170-17F   Hard Drive 1 (AT)
180-1EF   Müsait
1F0-1FF   Hard Drive 0 (AT)
200-20F   Game Adapter
210-217   Müsait
220-26F   Müsait
270-27F   Parallel Port 3
280-2A1   Müsait
2A2-2A3   clock
2B0-2DF   EGA/Video
2E2-2E3   Data Acquisition Adapter (AT)
2E8-2EF   Serial Port COM4
2F0-2F7   Reserved
2F8-2FF   Serial Port COM2
300-31F   Müsait
320-32F   Müsait
```

```
330-33F   Reserved for XT/370
340-35F   Müsait
360-36F   Network
370-377   Floppy Disk Controller
378-37F   Parallel Port 2
380-38F   SDLC Adapter
390-39F   Cluster Adapter
3A0-3AF   reserved
3B0-3BB   Monochome Adapter
3BC-3BF   Parallel Port 1
3C0-3CF   EGA/VGA
3D0-3DF   Color Graphics Adapter
3E0-3EF   Serial Port COM3
3F0-3F7   Floppy Disk Controller
3F8-3FF   Serial Port COM1
```

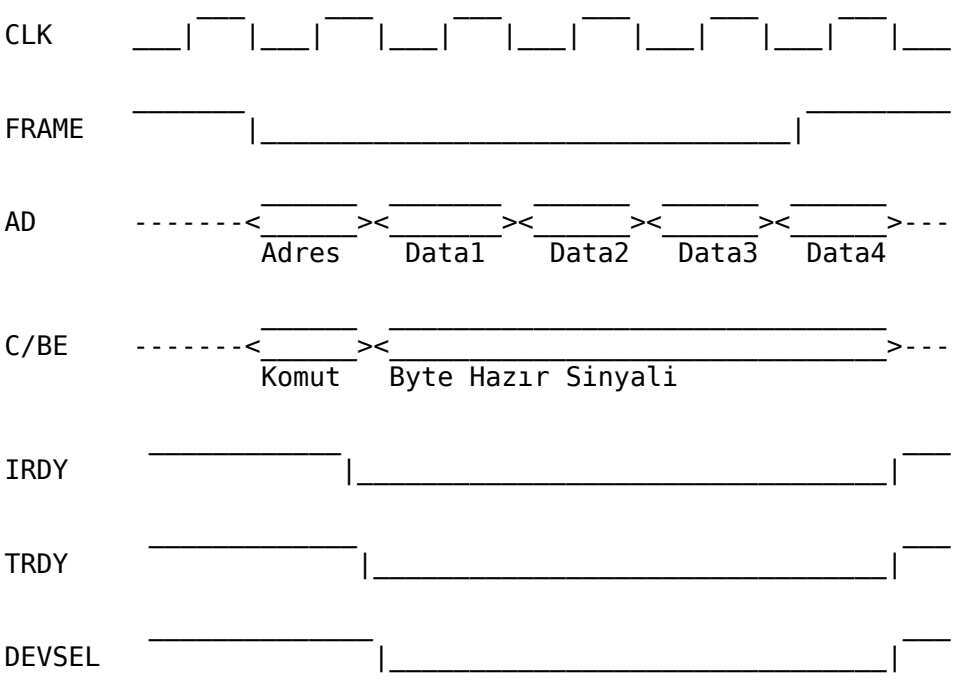
Aşağıda 24 bitlik giriş/çıkış kartının elektronik devre şeması görülmektedir;



Çok rahat ve sorunsuz bir kullanımı olan Isa veriyolu yapısı ne yazıkki günümüz uygulamaları için artık yavaş ve hantal kalmaktadır. Bu yüzden geliştirilen her yeni uygulamada veri yolu minimum PCI olarak tasarlanmıştır fakat USB basit yapılar için tercih sebebidir.

[7.2.2] PCI veriyolu:

PCI veriyolu isa veriyoluna alternatif olarak çıkarılmış olan MCA(Micro Chanel Architecture) ve VLB(Vesa Local Bus) sonra gelen ve neredeyse işlemcinin clock hızına erişebilen seri modlu, değişebilen configürasyon ve adres yapılı bir veri yolu tipidir. Günümüzde ekran kartı dışında çok tercih edilen ve geçerliliğini koruyan bir veriyoludur. 33 Mhz ve PCI2 de 66 Mhz veri transfer hızına sahip olan PCI veriyolu 3.3-5V luk voltaj standardına ve saniyede 264 Mb'lık veri transfer hızına sahiptir. Adres ve data veriyolları PCI veri yolu yapısında ardarda kullanıldığından adres ve data veri yolları aynı bitsel genişliğe sahiptirler, 32 bitlik yapıda olan PCI ve 64 bitlik yapıda olan PCI2 bulunmaktadır. Aşağıda temel olarak bir PCI portuna veri basımı zamansal lojik şeması vede pin çıkışları görülmektedir.



İçerik.	Pin		İçerik.
-12 V	B01	A01	TRST#
TCK	B02	A02	+12 V
GND	B03	A03	TMS
TDO	B04	A04	TDI
+ 5 V	B05	A05	+5 V
+5 V	B06	A06	INTA#
INTB#	B07	A07	INTC#
INTD#	B08	A08	+5 V
PRSNT1#	B09	A09	Reserved
Reserved	B10	A10	+5 V (I/O)
PRSNT2#	B11	A11	Reserved
GND/3,3 V Key	B12	A12	GND/3,3 V Key
GND/3,3 V Key	B13	A13	GND/3,3 V Key
Reserved	B14	A14	Reserved
GND	B15	A15	RST#
CLK	B16	A16	+5 V (I/O)
GND	B17	A17	GNT#
REQ#	B18	A18	GND
+5 V	B19	A19	Reserved
AD[31]	B20	A20	AD[30]
AD[29]	B21	A21	+3,3 V
GND	B22	A22	AD[28]
AD[27]	B23	A23	AD[26]
AD[25]	B24	A24	GND
+3,3 V	B25	A25	AD[24]
C/BE[3]#	B26	A26	IDSEL
AD[23]	B27	A27	+3,3 V
GND	B28	A28	AD[22]
AD[21]	B29	A29	AD[20]
AD[19]	B30	A30	GND
+3,3 V	B31	A31	AD[18]
AD[17]	B32	A32	AD[16]
C/BE[2]#	B33	A33	3,3 V
GND	B34	A34	FRAME#
IRDY#	B35	A35	GND
3,3 V	B36	A36	TRDY#
DEVSEL#	B37	A37	GND
GND	B38	A38	STOP#
LOCK#	B39	A39	+3,3 V
PERR#	B40	A40	SDONE
+3,3 V	B41	A41	SBO#

SERR#	B42	A42	GND
+3,3 V	B43	A43	PAR
C/BE[1]#	B44	A44	AD[15]
AD[14]	B45	A45	+3,3 V
GND	B46	A46	AD[13]
AD[12]	B47	A47	AD[11]
AD[10]	B48	A48	GND
GND	B49	A49	AD[09]
GND 5 V Key	B50	A50	GND 5 V Key
GND 5 V Key	B51	A51	GND 5 V Key
AD[08]	B52	A52	C/BE[0]#
AD[07]	B53	A53	+3,3 V
+3,3 V	B54	A54	AD[06]
AD[05]	B55	A55	AD[04]
AD[03]	B56	A56	GND
GND	B57	A57	AD[02]
AD[01]	B58	A58	AD[00]
+5 V (I/O)	B59	A59	+5 V (I/O)
ACK64#	B60	A60	REQ64#
+5 V	B61	A61	+5 V
+5 V	B62	A62	+5 V
64 Bit Space			64 Bit Space
Reserved	B63	A63	GND
GND	B64	A64	C/BE[7]#
C/BE[6]#	B65	A65	C/BE[5]#
C/BE[4]#	B66	A66	+5 V (I/O)
GND	B67	A67	PAR64
AD[63]	B68	A68	AD[62]
AD[61]	B69	A69	GND
+5 V (I/O)	B70	A70	AD[60]
AD[59]	B71	A71	AD[58]
AD[57]	B72	A72	GND
GND	B73	A73	AD[56]
AD[55]	B74	A74	AD[54]
AD[53]	B75	A75	+5 V (I/O)
GND	B76	A76	AD[52]
AD[51]	B77	A77	AD[50]
AD[49]	B78	A78	GND
GND	B79	A79	AD[48]
AD[47]	B80	A80	AD[46]
AD[45]	B81	A81	GND
GND	B82	A82	AD[44]
AD[43]	B83	A83	AD[42]
AD[41]	B84	A84	+5 V (I/O)
GND	B85	A85	AD[40]
AD[39]	B86	A86	AD[38]
AD[37]	B87	A87	GND
+5 V (I/O)	B88	A88	AD[36]
AD[35]	B89	A89	AD[34]
AD[33]	B90	A90	GND
GND	B91	A91	AD[32]
Reserved	B92	A92	Reserved

Reserved	B93	A93	GND
GND	B94	A94	Reserved

[7.2.3] AGP veriyolu:

Dahada hızlandırılan ve veri yolu 128 bite çıkarılan bu ayrıcalıklı port hakkında yayınlanmış fazla ve kesin bir döküman olmadığından ayrıntılı veri verememekteyiz ama bu veriyolu ile şu anda kullanılan PC veriyollarının sonuna ulaşmış bulunuyoruz.

8.Programlama Dili

Programlama Dilini tanimlayacak olursak ilk önce dip kökün anlamina bakmamiz gerekir. Dil insan oglunun gelistirdigi ilk iletisim sistemlerinden biridir, amaç iki veya daha faza insan arasindaki iletisimi saglamaktir. Program ise bilgisayar ortaminda yer alan bir amaç dogrultusunda yazilan islevsel dökümandir, yani bilgisayarin anlayacagi dilden tasfir ve emir gurubu dökümanidir. Bu ikisini birlestirirsek ortaya bilgisayar hedefli yeni bir lisan kurami çikar. Diger diller gibi kendisine has kurallari ve gereksinimleri vardir. Bu dili olusturanlarin belirleyip korudugu bir ifade toplulugudur, diger normal dillerden en büyük farki dil yordamininda bilgisayarla olusturulmus olmasidir. Genel olarak proje hedefli dillere bakacak olursak ilk olarak bilgisayar ortamindan bagimsiz görünen Robot Programlama Dilleri görebiliriz...

8.1 Robot Programlama Dilleri:

Geçmiste olusturulan bilgisayar disi aygitlarin temel bir standart ile kendi aralarinda uyumlu çalisabilmesi için genel kuram ve zorumluluklar konulmak istenmistir fakat günümüzde oldugu gibi firmalar yüksek basari ve kalite için kendi standartlarini koymuslardir örnegin su anda yazici kurulumunda en önemli etmen HP olup olmamasidir, Hp kendi ansi standartini olusturmus ve kalitesi ile bunu dünyaya kabul ettirmistir. Gelisen isletim sisitem fonksiyonlari ile program yönetimi sanal ortamlara tasinarak yapilan ana islevler çok basitlestirilmistir. Iste bu yüzden yapilan yeni bir donanim için yeni bir standart ve yazilim gelistirmek kaliteyi etkiliyorsa kaçinmak hatali bir karar olacaktir. Fakat yinede genel olarak Robot Programa Dillerine deyinmek eski hatalari görmemize yardimci olacaktir;

Ilk olarak "Teach by Lead" yani göster ve uygulat olarak tanimlayabilecegimiz bir yapiyi görmekteyiz bu yapida hedef olan programlam dili olmasina karsin robot kendine yaptirilan bir hareketi defalarca tekrarlayabilmektedir. Daha çok boyama robotlarinda kullanilan bu teknik sayesinde yapilacak boyama islevi kolu çekistirerek yapilir bu yapilan islemler sirasinda "Teach(Ögrenme)" kademsinde olan robot algilayicilarindan gelen verileri Epromuna kaydederek islevi

sonladirilinca tekrarlayabilir, enbüyük dez avantaji kolun hareketinin insan ögesi ile tekrarlamaktir.

Bundan sonraki asamalarda devreye uygulayici olarak bilgisayar yani yapisal bir program devreye girmektedir.

VAL ve VAL II(tm): Bu programlama dili ile daha çok PUMA robotlari programlanmaktadir. Bu Basic tipi benzeri hareketlerin tanimlanmasini saglayan bir özellik tasimaktadir. VAL ve gelistirilmis versiyonu olan VAL II (TM) yüksek seviyeli dil olarak tanimlanmaktadir. Bu programlama dilinde kullanilan komutlar ve özellikleri asagidadir.

SPEED 100 ALWAYS : Robot yeni bir komut gelene kadar 100 ünite degerindeki hizla hareket edecektir. Robot hizlari ünite olarak tanimlanir ve 0,01 ila 327,67 sinirlari arasindadir

MOVE POINT : Robot kol halihazirda bulunulan konumdan POINT olarak tanimlanmis yeni noktaya hareket edecektir.

MOVE 2POINTS : Robot kol dogrusal olarak tanimlanmis iki noktaya hareket edecektir. Önce birinci hedefe daha sonra ikinci hedefe ulasacaktir.

DRIVE 1 , 30 , 100 : Bu komut ile 1 nolu birlesim robot tabani 30 derecelik dönmeyi ve 100 ünite hizinda gerçeklestirecektir.

DRAW 100 , 200 , 50 : Draw komutu takim merkez noktasi için kullanilmaktadir. Robot tutucusu 100 cm x ekseninde 200 cm y ekseninde ve 50 cm z ekseninde kartezyen koordinat sisteminde hareket edecektir.

APPRO POINT , 100 : Bir önceki pozisyon noktasindan 100 cm ileriye hareket komutudur. Burada 100 cm takimin robot ucun bulundugu noktadan z ekseni boyunca ilerlemesidir.

DEPART 100 : Ayni Appro Point komutu gibidir. Yalniz burada hareket Z ekseninin negatif yönünde olacaktir.

VAL dilinde yazilan her bir program bütün diger programlarin alt programi olabilir. Unimation VAL program dilini gelistirerek yeni bir versiyonu olan VAL II (TM) çikartmistir. Burada robot hareket komutlari trigonometrik fonksiyonlar, gerçek zaman iletisimi için ileri fonksiyonlarve çevre ekipmanlari ile kontrol sisteminin baglantisi için gerekli olan komutlarla desteklenmistir.

AML PROGRAMLAMA DILI:

Yüksek seviyeli dil olarak kabul edilen AML A Manufacturing Language kelimelerinin bas harflerinden olusmustur. Bu dil Endüstriyel robot sistemlerinin islemleri için özellikle dizayn edilmistir. Bu programlama

dili ile robot sistemlerinin dört temel fonksiyonu kontrol edilebilmektedir. Bunlari su sekilde siralayabiliriz.

- 1 - Manipulation (Kullanma Isletme) fiziksel objelerin ismerkezi etrafinda dödürülmesi.
- 2 - Sensing (Algilama)
- 3 - Intelligence (Haberlesme)
- 4 - Data processing (Bilgi isleme ve kaydetme)

Yukarida belirtilen özellikler kontrol edilmekte rapor olarak üretilebilmektedir. AML dili robotlarin programlanmasinda özel islemler için kullanilan dillerden biri degildir. Bu dilde döngüler kullanilmakta olup test, matematiksel islemler , sensörler ile bilgi alis verisi ve bilgi depolama ile ilgili uygulamalar saglanmaktadir. genellikle PUMA türü robotlarin programciliginda kullanilir. Bu dilde kullanilan bazi komutlar ve uygulama özellikleri asagidadir.

MOVE : En önemli komuttur ve bunun DPMOVE isimli komut çesidi vardir. DPMOVE robot kolun vektörel olarak hareketini saglamaktadir. [PT (x , y , z)] : Robot kolun çalisma bölgesi içerisinde tanimlanan bir noktaya hareketini saglayacaktir. PMOVE komutu istenilen noktaya ulasimi saglayacaktir. Tavsiye edilen AML program yapisinda bütün bölgeler ve hareketler alt program (Sub-Progran) olarak hazirlanmalidir. Endüstriyel robot uygulamalarinda kullanilan robot türlerinden biriside DEA robotlardir. (Digital Elektronic Automation)Bu robotlarda kullanilan programlama dili HELP programlama dilidir. Bu dil yüksek seviyeli dil olaraktanimlanmaktadir. HELP programlama dilinde üç temel komut vardir. Bunlar :

- 1 - Atama durumlari, Aritmetik deyimler, Sayma
- 2 - Kontrol komutlari GOTO, GOSUB; IF-THEN
- 3 - Sub programlar içerisinde kullanilan GRASP, UNGRASP,SIGNAL ve WAIT komutlari

Yukarida temel özelliklerin kullanilmasi ile robotlar programlanmaktadir. Ayrica MOVE (X , Y , Z) komutu ile robotun kartezyen koordinat sisteminde X , Y , Z eksenlerinde hareketleri saglanmaktadir.

MITSUBISHI PROGRAMLAMA DILI : En güçlü dillerden birisidir. komutlar çift karekterler ile tanimlanmaktadir. tanimlanan komutun parametrik degerleri ile birlikte kullanilmaktadir. Bu program dili 6 temel kisimdan meydana gelmistir. Bunlar:

- 1 - Position / Motion Control Instruction (Pozisyon ve hareket kontrol komutlari.)
- 2 - Program Control Instruction (Program Kontrol Komutlari.)
- 3 - Hand Control Instruction (El Kontrol Komutlari)
- 4 - I/O Control Instruction (Giris / Çikis Kontrol Komutlari)
- 5 - RS232C Read Instruction (RS 232C Kontrol Komutlari)
- 6 - Miscellaneous (Yardimci komutlar)

Pozisyon ve Hareket kontrol komutlari :

DP (Decrement Position) : Robot kolun bulundugu konumdan pozisyon numarasi daha küçük olan bir önceki pozisyon noktasina hareketi saglanmaktadir.

10 LPRINT " MO 3 "

20 LPRINT " MO 4 " 4 Nolu pozisyona hareket

30 LPRINT " MO 5 " 5 Nolu pozisyona hareket

40 LPRINT " **DP** " 4 Nolu pozisyona hareket.

DW (Draw) : Robot kolun x , y , z eksenlerinde hareketini saglayan komuttur. Verilecek arti degerler pozitif yönü eksi degerler ise negatif yönü tanimlamaktadir. DW 10 , 5 , 0 burada robot kol x ekseninde 10 mm y ekseninde 5 mm z ekseninde ise 0 mm hareket edecektir.

HE (Here) : Robot kolun bulundugu konumu pozisyon no olaraktanimlayan komuttu r. Pozisyon numaralari 1 ila 629 arasinda bir sayi olabilir.

10 LPRINT " MO 10 " 10 Nolu pozisyona hareket

20 LPRINT " DW 10,0,0 " X ekseninide 10 mm ilerle

30 LPRINT " **HE** 11 " Bu nokta pozisyon no 11 dir.

HO (Home) : Karetezyen koordinat sisteminde referans nokta tanimlama islemini gerçeklestirir. LPRINT " **HO** " **IP (Increment Point)** : Bulunulan konumdan pozisyon numarasi daha büyük olan bir önceki pozisyona gitme komutu. (Bu komutu DP komutunun tersi olarak düsünebiliriz.)

10 LPRINT " MO 7 " 7 Nolu pozisyona hareket

20 LPRINT " MO 6 " 6 Nolu pozisyona hareket

30 LPRINT " MO 5 " 5 Nolu pozisyona hareket

40 LPRINT " **IP** " 6 Nolu pozisyona hareket

MA (Move Approach) : MC (Move Contunious) : MJ (Move Joint) : MO (Move) :

MP (Move Position) : Robot elim uç noktasinin tanimlanan x,y,z eksenlerdeki tanimlanmis olan koordinat noktasina Pitch, Roll ve açi

degerleri verilerek hareket etmesini saglar MP [X eksen koordinati][Y eksen koordinati][Z eksen koordinati][Pitch açisi][Roll açisi]

örnek: MP 0,380,300,-70,-40 “ PD 1,0,380,300,-70,-40 ” “ MP 0,380,280,-70,-40 “

MS (Move Straight) : Robot kolun bulundugu noktadan, pozisyon no belirlenen noktaya belirlenen sayida basamaklarla ilerlemesini saglayan komuttur. MS [Pozisyon no, Basamak sayisi < O , C >] Burada 1<= Pozisyon no <= 629 Burada 1<= Basamak sayisi <= 99 “ HE “ “ MO 3 ” “ MS 2,3 C “ “ MO 3 “ “ MS 2,6,C “

MT (Move Tool) : Aktif aleti hareketlendirir.

RS 232-C Kontrol Komutlari :

CR (CXounter Read) : Kaç komut birikimi oldugunu belirtir.

DR (Data Read) : Bilgi okuma.

ER (Error Read) : Hatakodu bilgisini verir.

LR (Line Read) : Çogulsal bir okuma yapar.

PR (Position Read) : Aktif pozisyon bilgisini verir.

WH (Where) : Aktif konum bilgilerini verir.

Yardimci Komutlar :

RS (Reset) : Robot Pozisyonlarini resetler.

TR (Transfer) : Aktif durumu trasfer eder.

WR (Write) : Robota bilgileri yazar.

(Comment) : Komut uygulamasi gerçeklestirilmeden açiklayici bilgiler yazilmaktadir.

8.2 Yapisal Program:

Yapisal açidan bir program tabirini açiklamadan önce ilk olarak bir program nedir ve ne olmalidiri anlamak gerekir. Program bir görevi yerine getirmek için tasarlanmis asamalar bütünüdür. Bir program mutlaka hatasiz vede en kisa yoldan maximum çözüme ulasmalidir. Yani

program kisaca islerimizi kolaylastirmak için kurulan bir sebep - sonuç iliskisidir.

Yapisal açidan program tabiri ise olusturulacak programin temel taslarini yerlestirmede uygulanan yöntem ve belirgelerdir. Temel olrak ilk bilgisayar porgrama dillerinden bu yana çok yaygin olarak bilinen sekilsel yaklasimi inceleyelim. Assagida temel olrak hangi kutunun ne anlama geldigi görülmektedir;

Programi olusturacak olan sahis ilk olarak elindeki baslangiç verilerini ve ulasmasi gereken sonucu irdeleyerek isleme baslar. Elindeki bilgilerle sonuç bilgilerini karsilastirildiktan sonra izlenecek en kisa yol bulunmaya çalisilir, bu asamada çözüm bireye bagli oldugundan hiçbir zaman çözüm yolu tek bir iz degildir, birçok farkli yoldan sonuca ulasilabilir önemli olan minimum zamanda maximuma erismektir. Basit bir örnekte bunu görelim;

Ör: 30 ayri kümesten gelen yumurtalarin ayri ayri adetleri bilinmektedir. Ortalama kümes basina kaç yumurta düsmektedir???

1.asama: Baslangiç verisi 30 adet ayri ayri veri ve ulasilmasi gereken sonuç ortalama deger.

2.asama: 30 ayri çiftlikten gelen yumurta toplanir ve 30 bölünerek ortalam adet bulunur.

2.asama (çözüm 2): Her 2 çiftlikte bir ortalama bulunarak diger bir çiftlik ile yeniden 2 çiftlik olarak ortalama bulunur. Son çiftlikten sonra ortalama bulunmus olur.

3.asama: sonuç raporlanir.

Dikkat edilmesi gereken basitlestirilmis hizli ve pratik çözümdür.

[8.3] Pascalin Incelenmesi

Pascal programlama dili ilk olarak Fransa'da yer alan Borland firması tarafından ... yılında üretilmiş olan yapısal olarak üçüncü kuşak bir dil yapısıdır. Versiyon ayrımlarında 5. Versiyonda temel yapısını, 6. Versiyonunda da Unit ve Object-Oriented yapısını kazanmıştır en son versiyonu olan 7 ve 7.1 ise genel olarak windows ortamına hazırlık ve bazı hataların kapatılmasından oluşmaktadır. Dilde hedef en basit kullanımdan en karmaşığa kadar olan yapıda kullanıcılara temel olarak hitap ederek yol göstermektir. Yapısal olarak çözümlenen işlevler sayesinde complex yazılımlar hata kendi kendini oluşturan yazılımlar gerçekleştirilebilmektedir. Peki diğer dillerle karşılaştırmasını yapmak gerekirse Pascal temelde yapısal olarak C diline çok benzemekle birlikte yazım kolaylığı ve hedef ortam esnekliği ile C dilinden ayrılmaktadır fakat C dili gerek derleyicisin gelişmiş ve microsoft tarafından desteklenmesi sayesinde hız ve sağlamlık kazanmıştır bununla birlikte gelen yapısal

çekiştirmeler sonucunda da hedef sapması ve kullanım zorluğu ortaya çıkmıştır fakat asıl başarısını çeşitli ortamları desteklemesiyle kazanmıştır örneğin Unix ve Novell gibi temelsel diller bu dilde yazıldığından kullanım alanını bu yöne kaydırmıştır. Sonuç olarak pascal Ms-Dos ve Windows kullanmayı hedefleyen sistemler için eşi bulunmaz temel bir derleyicidir.

Temel Olarak Program Yazım Kuramları:

Temel olarak bilindiği gibi ingilizce dili üzerinde yazılan editörlerde türkçe harf yasağı vardır. Dikkat edilmesi gereken diğer bir ayrıntıda her satır sonuna konan noktalı virgüldür. Örneğin basit bir program yazmak istersek programımızı "Begin" ile başlatıp "End." İle bitirmemiz gerekir ve istenilen komut kümeleri bu iki uç noktası arasına yazılmalıdır;

```
Begin
  Write('http://get.to/Robotic');          {Ekrana http://get.to/Robotic yazar}
End.
```

Buradaki "Write" komutu ekran ortamına yazı yazmak için kullanılmaktadır. Komuttan sonra parametre olarak nitelendirilen ifadeler için parantez açılarak içerisine uygulanacak işlev yerleştirilir, parantez kapatılarak parametre tamamlanır ve noktalı virgül de komutun bittiğine işaret eder. Parametremiz tek tırnak içerisinde yazılmıştır çünkü parametre string yani alfasayısal(hem rakamları hemde harflerle özel işaretleri içerir) bir ifadedir. Write yerine "Writeln" kullanılırsa write komutundan tek farkı parametre ekranda gösterilirken ifade yazılıp alt satırın başına geçer. Ekran modu text (ekrandaki görüntü noktalardan değil de karakterlerden oluşan ekran modu tipi) moddur. Küme parantezi ise açıklama yapılması gerektiğinde yani program parçası olmayan ifadelerin yazılması için kullanılır parantez içini derleyici dikkate almaz. Diğer bir örnekte ise ekranın kullanımını görmekteyiz;

```
Begin
  Writeln('Yazıp assagı gecer');
  Write('Yazıp assagı gecmez');
  Writeln('devam eder ve assagı gecer');
End.
```

Dikkat edilmesi gereken önemli bir nokta temel işlevlerin aynı zamanda ASCII (American Standart Code Instruction information- Standart karakter seti) tablosundaki 13. kodun write sonuna konarak writeln elde edilmesidir, bundan sonra göreceğimiz her alt satırın başına geçme işlevi aslında bu kodla yapılmaktadır yani ekrana "Yazıp assagi gecer#13" yazılmaktadır. #13 ifadesi başındaki # karakteri ascii tablodan kodu işaret etmek için kullanılır, yani "Write(#65);" komutu ekrana "A" harfini yazacaktır.

Ekrandan gelen olarak yazı yazmayı öğrendik, bu yazılan yazıyı belirli bir konuma yazıldığı açıkça görülmekte işte bu konum "Cursor" olarak tabir edilen ekranda bulunan yanıp sönen bir işaretçidir ve bu işaretçi

ekranda kalınan son noktayı takip etmede kullanılmaktadır. Cursor'un yerini bir şey yazmadan değiştirmek için "Gotoxy" komutu kullanılabilir. Aşağıda temel bir örneği yer almaktadır;

```
Begin
  Writeln('Yazıp assagı gecer');
  Gotoxy(10,10);
  Write('Yer 10,10',' gecmistir.');
End.
```

Komutta parametreler X,Y konumlarıdır. Aşağıda ekranda görülen eksenler ve uzunlukları yer almaktadır;



Diğer bir temel komut ise ekrandan veri okuyan "Read" ve "Readln" komutlarıdır, komut parametresi değişkendir. Örnek olarak;

```
Var
   Adi:string;

Begin
  Gotoxy(10,10);
  Write('Adınızı Giriniz:');
  Readln(adi);
  Write('Merhaba ',adi);
End.
```

Örnekte begin başlangıç komutu üzerinde yer alan "Var" değişken tanımlama komutudur, ondan sonra begin komutuna kadar yazılan ifadeler değişken olarak tanımlanarak kullanılırlar peki nasıl bir değişken tanımlanır isterseniz ilk olarak temel değişken tiplerini görelim;

- Byte: sayısal bir ifadedir, 0 255 arasında bir değer içerebilir. 8 bitten oluşur, en küçük birimdir.
- Word: sayısal ifadedir, 0 65535 arasındadır.16 bitten oluşur. (1 word 2 byte eşittir).
- Integer: Sayısal ifadeler için kullanılır, sınırları -32768 ile +32767 arasındadır. (1 word uzunluğundadır.)
- Dword: sayısal ifadedir, 0 4294967295 arasındadır, 32 bitten oluşur yani 2 word,4 byte'a eşittir.

- Longint: sayısal ifadedir, -2147483648 2147483647 arasındadır popüler bir sayısal ifadedir. (Dword eşittir)
- String: Alfasayısal olarak adlandırılan 255 (1 byte maximum değeri) karakter uzunluğundaki değişkendir.
- Char: Karaktersel ifadedir, Ascii tablo içeriğini taşır. 1 byte uzunluğundadır kullanım farkları vardır.

Bu değişkenlerden oluşacak bir dizi tanımlamak için ise *ARRAY* (dizi) olarak tanımlanabilir. Arrayden sonra köşeli parantez içerisinde belirtilen rakam ararlığı kadar ([1..10] 10 adet demektir.1 den 10'a) değişken içerir demektir. Eğer dizide birden fazla aynı değişken bloğu kullanılacaksa dizinin boyutu artırılabilir.Örneğin;

```
Var
   Data:Array[1..20] of Integer; {Tek boyutlu bir dizi}
   Data2:Array[1..3,1..20] of Integer; {iki boyutlu bir dizi}
   Data3:Array[1..10,1..3,1..20] of Integer; {Üç boyutlu bir dizi}
Begin
   Writeln('1.Dizinin 15. elamanını giriniz: ');Readln(data[15]);
   Writeln('1.Dizinin  15. elamani: ',data[15]);
   Writeln('2.Dizinin 15. elamanını giriniz: ');Readln(data2[1,15]);
   Writeln('2.Dizinin  15. elamani: ',data2[1,15]);
   Writeln('3.Dizinin 15. elamanını giriniz: ');Readln(data3[1,1,15]);
   Writeln('3.Dizinin  15. elamani: ',data3[1,1,15]);
End.
```

Temel matematik işlemleri:

+ : İki ibarenin toplanıp bir sonuç değişkenine aktarılmakta kullanılır.
- : İki ibarenin çıkarılıp bir sonuç değişkenine aktarılmakta kullanılır.
* : İki ibarenin çarpılarak sonucun sonuç değişkenine aktarılmakta kullanılır.
DIV : İki ibarenin bölünerek sonuc değişkenine aktarılmakta kullanılır. (Div tamsayı bölme işlemlerinde kullanılır, / bölme işlemi ise küsüratlı bölme işlemlerinde kullanılır snuç mutlak olarak real değişken tipidir.)
MOD: Değişkenin belirtilen sayıya göre modunu alır.
INC: Değişkeni bir adet artırır.
DEC: Değişkeni bir adet azaltır.

Örneğin:

```
Var
    A,B,C:Integer;
    R:Real;
Begin
   Writeln('1. sayiyi giriniz: ');Readln(A);
   Writeln('2. sayiyi giriniz: ');Readln(B);
   C:=A+B;
   Writeln('2 sayiyinin toplamı: ');Writeln(C);
   C:=A-B;
   Writeln('2 sayiyinin Farkı ');Writeln(C);
```

```
            C:=A*B;
            Writeln('2 sayiyinin Carpımı  ');Writeln(C);
            C:=A div B;
            Writeln('2 sayiyinin Bolumu  ');Writeln(C);
            R:=A/B;
            Writeln('2 sayiyinin ondalikli Bolumu
');Writeln(R:0:0);
                                                {:0:0 ifadesi formatli yazma
işlemlerinde kullanılan ibaredir. 1 kademe noktadan öncesini, 2 hane ise
noktadan sonrasını ifade eder.}
            C:=A MOD B;
            Writeln('1. sayının 2. sayiya  Modu: ');Writeln(C);
          End.
```

Mantıksal işlemler ise:

AND: Lojik Ve karşılaştırması yapar.
OR: Lojik Veya karşılaştırması yapar.
NOT: Lojik tersini alır.
XOR: Lojik tersveya işlemini yapar.

Örnekler:

```
  Var
     A:byte;
  Begin
     A:=1;Writeln(A);
     A:=A AND 2;Writeln(A);
     A:=A OR  1;Writeln(A);
     A:=NOT A;Writeln(A);
     A:=A Xor 255;Writeln(A);
  End.
```

Temel tiplerin dışında tanımlanabilecek grup ve tanımsal ifadelerde pascalın içerdiği kullanışlı değişken türleridir. Gruplu yapıda *Record* ifadesiyle tanımlama bölümünde tanımlanabilinir, örneğin:

```
Type
  Kimlik= Record
              Ad:string[8];          {köşeli parantez içerisinde tanımlanan sayı
kaç karakterlik bir string olduğunu belirtir.}
              Soyad:string[8];
              Tel:longint;
            End;
 Var
    Data:Kimlik;
Begin
  write('Adi:');readln(data.ad);
  write('Soyadi:');readln(data.soyad);
  write('Telefonu:');readln(data.tel);
End.
```

Örnekte dikkat edilmesi gereken en önemli nokta Type kısmında tanımlanan değişken tiplerinin program parçasında kullanılabilmesi için mutlak olarak yeniden atanması gerektiğidir. Diğer noktayla yazılması ve Record-End; yapısı yazım ve tanım kurallarıdır. Diğer bir durum ise sabit atamalardır, bu atama türündeki bilgiler sabit olduğundan değiştirilemez yanı readonly(sadece okunabilir) yapısındadır, örneğin:

```
Const
  Hex:array[0..15] of
char=('0','1','2','3','4','5','6','7','8','9','A','B','C','D','E','F');
Var
  a:byte;
Begin
   Writeln('Bir sayi giriniz?');readln(a);
   Write('Hexedesimal karsiligi: ');
   Writeln(hex[a div 16],hex[a mod 16]);
End.
```

Yukarıdaki örnekte dışarıdan girilen bir byte'lık sayının hex karşılığı hex tiplemesi ile ortaya çıkarılmıştır, köşeli parantez burada birden fazla değeri içeren ifadelerde ifadelerden seçilecek olanı işaret etmekte kullanılır.

Mantıksal İşlemler:

If(Eğer) olarakta adlandırlan mantıksal karşılaştırma komutları gerek pascal olsun gerekse tüm diğer temel dillerde en önemli yapı taşıdır çünkü girilen veya elde edilen tüm veriler mutlak olarak bir karşılaştırma ve değişim sürecinde geçmelidirki sonuç işleme ulaşılabilsin bunun için eğer işlemi neredeyde her programda yer alan mutlak bir ifadedir, aşağıdaki örnekte dışarıdan girilen iki sayıdan hangisinin büyük olduğunu bulan bir program yer almaktadır;

```
Var
   A,B:Integer;
   S:string;
Begin
  write('Sayı 1:');readln(A);
  write('Sayı 2:');readln(B);
   If A>B then s:='Sayi 1 buyuktur sayi 2 den';
   If A<B then s:='Sayi 2 buyuktur sayi 1 den';
   If A=B then s:='Sayi 1 Sayi 2 ye esittir ';
   writeln(s);
End.
```

İşlemler sırasında birden fazla ifadeyi karşılaştırarak sonuca varmak gerekirse aralarına AND(Ve)- OR(Veya) ifadeleri yada olumsuzları için NOT özel durumlardada XOR kullanılabilinir önemli olan işlemleri doğru şekilde parantez içerisine alabilmektir

```
Var
    A,B:Integer;
    S:string;
Begin
   write('Sayı 1:');readln(A);
   write('Sayı 2:');readln(B);
    If (A=5) And (B=5) then s:='Her ikiside 5 dir';
    If (A=5) Or (B=5) then s:='Ikisinden biri 5 dir';
    If (A=5) Xor (B=5) then s:='Sadece biri 5 dir';
    If (A<>5) And (B<>5) then s:='İkiside 5 Değildir';
    If Not ((A=5) And (B=5)) then s:='İkiside 5 Değildir';
    writeln(s);
End.
```

Eğer istenmiyen bir sonuçla karşılaşılırsa karşılaştıma tersi yani doğru olmadığı durumda yapılması gerek bir ifade için ELSE deyimi kullanılabilir. Örneğin;

```
Var
    A,B:Integer;
    S:string;
Begin
   write('Sayı 1:');readln(A);
   write('Sayı 2:');readln(B);
    If (A=5) And (B=5) then s:='Her ikiside 5 dir' ELSE s:='Her ikiside 5
değildir';
    writeln(s);
End.
```

CASE OF:

Çoktan seçmeli şartlarda if satırlarının sayısıyla birlikte yazım güçlü oluşmasını önlemek için bir değişken üzerinde karar verilirken CASE ifadesi kullanılabilir. Kendinden sonra gelen blokta her satır bir değerine karşılık gelecek olan komut kümesini içermektedir;

```
Var
    A:Integer;
    S:string;
Begin
   write('Sayı:');readln(A);
   Case A of
       1:S:='Sayi birdir.';
       2,3:S:='Sayi iki yada üçtür.';
       4..100:S:='Sayi 4 ile 100 arasındadır.';
      else S:='Sayi yuzden buyuktur';
    writeln(s);
End.
```

Tekrarlama döngüleri:

Bazı işlev ve komutlar ard arda tekrarlanması gerektiğinde yapılması gereken bir döngü yani tekrarlama programının kurulması gerekir, bu tekrar işleminin bir olaya kadar yani bir şart ile soa ermesi veya değişmesi gerekmektedir işte bu yüzden döngü mutlak olarak bir şart bağlamında olmalıdır;

GOTO:

Temel makina dilinden tutun yüksek seviyeli dillere kadar bütün dillerde bu deyim temel atlama ifadesi olarak kullanılır, ifadeden sonra yazılan kısma gitmekte kullanılır. Örneğin;

```
Var A:Word;
Label Ana,Son;          {Goto ile kullaılacak atlama yerlerini tanımlamak
için kullanılır...}
Begin
  Writeln('Baslangic');
  Ana:          {Üst üste iki nokta label(Tanım) dan sonra yazım kuralı
olarak gereklidir.}
  Write('dongu');Readln(a);
  If A=0 then  Goto Son;
  Goto Ana;
  Son: Writeln('Son');
End.
```

FOR:

Döngü miktarı belirli ise yani 10 tane gibi hedefsel bir döndürme yapılacaksa bu ifade bloğu kullanılabilir;

```
Var  A:byte;
Begin
   For A:=1 to 10 do
       Writeln(A);
   For A:=10 downto 1 do          {Geri sayma}
       Writeln(A);
End.
```

WHILE DO:

Tekrarlanacak komut dizesi belirli bir şarta kadar tekrarlanacaksa While(Olduğu sürece) tekrarla işlevi kullanılabilir. Bu işlem aynen Goto ilede yapılabilmektedir ama yazım ve işleyiş açısından kolaylıkları

bulunmaktadır;

```
Var A:String;B:Byte;
Begin
  A:='';
  B:=0;
  While A<>'Ozgur Aksu' do
   Begin          {Yeni bir blok tanımı}
     Inc(B);
     Writeln(B,'.Bilgiyi giriniz??? ');Readln(A);
  End;           {Blok sonu}
End.
```

Blok Tanımından kasıt eğer bir ifadeden sonra yapılması gereken onun alt ifadeleri bir komuttan fazla ise yani komut kümesi ise komutlar yeni bir Begin-End yapısı içerisine alınarak blok tek satırlık bir komut gibi işletilir, end sonundaki noktalı virgül programın sonun bu end ile bitmediğini gösterir.

REPEAT UNTIL:

Aynen While gibi çalışmasına rağmen farklı olan yanı işlem neticesine 1. döngüden sonra bakması ve düz karşılaştırma yani Until(e kadar) yapar;

```
Var A:String;B:Byte;
Begin
  B:=0;
  Repeat
     Inc(B);
     Writeln(B,'.Bilgiyi giriniz??? ');Readln(A);
  Until A='Ozgur Aksu';
End.
```

Program Parçacıkları:

Genel kullanıma geçildiğinde ilk olarak zorlanılan konulardan biri bazı komut kümelerinin devamlı olarak kullanılması gerektiğiydi ve bu kullanım için GOTO kullanılamıyordu çünkü Goto ile çağrıldıktan sonra komut dizesi bitirildiğinde çağrıldığı noktadan bir sonraki komuta geri dönemiyordu, bu yüzden iki adet alt program parçacığı tanımı oluşturuldu, Procedure(Prosedür) ve Function(Fonksiyon). Procedure

istenilen komut dizesini parametreler ile yerine getirim geri kaldığımız yere geri dönüyor, Function ise procedure ile aynı şekilde çalışmasına rağmen tek farkı geriye bir sonuç verebilmesidir;

Örnek 1:

```
Var
  S:string;              {Global bir değişken}

Procedure OKU;
Begin                    {Procedure bloğu}
   Write('Adınız? ');
   Readln(S);
End;

Var A:byte;              {Local Değişken}
Begin
   For A:=1 to 100 do
         OKU;          {Procedure burada çağrılıyor . Procedure işleniyor .
ve buraya geri dönüyor}
End.
```

Global ve Local kavramları değişkenin yerine göre karar verilecek bir uygulamadır eğer değişken programın giriş kısmında yani tüm procedure, Function ve ana program tarafından kullanılabiliyorsa Global(Genel), sadece üstünde bulunduğu Begin-End bloğuna tanımlıysada Local(Yerel) değişken olarak tanımlanabilinir. Global değişkenler programın heryerinde değiştirilip kullanılabilen bir değişkendir, Local ise sadece kendi bloğunda tanımlıdır.

Örnek 2:

```
Funtion CARP(A,B:Integer):Integer;
Begin                    {Function bloğu}
   Carp:=A*B;
End;

Var A:byte;              {Local Değişken}
Begin
   For A:=1 to 100 do
         Writeln(Carp(A,A));
End.
```

Dosyalama İşlemleri:

Veriler bilgisayar ortamında RAM olarak bilinen hafıa kısmında bulunduklarından bilgisayar kapatıldığında içerilerindeki bilgileri kaybedeceklerdir. Bunu engellemek ve daha çok veri bloğunu kullanabilmek için daha geniş kapasiteli DİSKler kullanılmaktadır. İşte bu yüzden verileri diske yazmak için dosya diye tabir edilen bir havzaya yazmamız gerekir. İlk olarak işletim sistemi için dosyanın ismini tanımlamamız ve dosya tipinide belirtmemiz gerekir. Dosya ismi sekiz

karakter isim ve 3 karakter uzantılı bir şekilde MSDOS işletim sistemi için tanımlanabilir, karakterler temel setin yasaklanmış karakterlerini içermemelidir örneğin nokta,bölü vede çarpı gibi. Dosya tipi ise oluşturulacak dosyaya konacak verilerin tipine göre bir yapı seçilerek içerisine bilgi konması bir düzene sokulmalıdır. Aşşağıda temelbir örnek yer almaktadır.

```
Var
   A:Byte;
   Dosya:File of Byte;        {Byte sal dosya tipi}
   Dosyaismi:string;
Begin
   Dosyaismi:='BYTE.DAT';        {Dosya ismini Byte.dat olarak belirledik}
   Assing(Dosya,Dosyaismi);        {Assing dosya değişkenine dosya ismini
atadı}
   Rewrite(Dosya);                    {Dosyayı olusturduk}
   For A:=1 to 100 do
      Write(Dosya,A);                {Dosyaya A bilgisini yazdık}
   Close(Dosya);                      {Dosyayı kapatık}
   Writeln('Dosyaya yazdık ve dosyayı kapatık kapatık. Yeniden bilgileri
almak için Entere Basınız!!!');Readln;
   Reset(Dosya);                      {Dosyayı Okumak için Açtık}
   For A:=1 to 100 do
   Begin
      Read(Dosya,B);                {Dosyaya B bilgisini Okuduk}
      Writeln(A,'. bilgi ',B);
   End;
   Close(Dosya);                      {Dosyayı kapatık}
End.
```

Pascal editöründen F3 yardımıyla BYTE.DAT'ı açtığımızda karşımıza okuyamadığımız veriler çıkmaktadır çünkü veriler bytesal olarak yazılmıştır, okunma formatı TEXT yani stringsel moddur. Aşağıdaki örnekte kullanıcıdan alınan verilerle bir dosya oluşturup kontrol işlemi gerçekleştirilmekte;

```
Var
   S:String;
   Dosya:Text;        {Text(Yazı) dosya tipi}
   Dosyaismi:string;
Begin
   Dosyaismi:='BYTE.TXT';
   Assing(Dosya,Dosyaismi);
   Rewrite(Dosya);
   For A:=1 to 10 do
   Begin
      Readln(S);
      Write(Dosya,S);
    End;
   Close(Dosya);
```

```
   Writeln('Dosyaya yazdık ve dosyayı kapatık kapatık. Yeniden bilgileri
almak için Entere Basınız!!!');Readln;
   Reset(Dosya);
   For A:=1 to 10 do
   Begin
      Read(Dosya,S);
      Writeln(A,'. bilgi ',S);
   End;
   Close(Dosya);
End.
```

Dikkat edilmesi gereken dosyaya yazım işlemleri sırasında buffer(tampon) kullanıldığı için mutlak olarak dosyayı açtıktan ve işimiz bittikten sonra Close(Kapat) işlemi gerçekleştirilmelidir.

Tipli Dosyalar:

Kullanım gereksinimlerinden biride temel database yapısı olan tiplendirilmiş kayıt yapılarının ard arda kayıt numara sırasıyla dosya içerisine atılmasıdır. Burada dosyada yer alan değişkenler record yapısı ile tanımlanarak her bir kaydın tipi belirlenerek dosya oluşturulmaktadır. Aşağıdaki örnekte bir telefon fihrist uygulaması görülmektedir;

```
Const
    Disim='Tel.dat';
Type
   Tanim=Record
                Varmi:char;
                 Isim:string[8];
                Soyad:string[8];
                Tel:Longint;
                Adres:string;
            End;
 Var
    Dosya:File of  Tanim;
    Data:Tanim;
    Rno:Word;   {Record(Kayit) numarasi}
    Ch:char;

Function Sonkayitno;
Var A:word;
Begin
   A:=0;
   Repeat
      Inc(A);
      Seek(dosya,A);                {Kayit numarasina konumlanir}
      Read(dosya,data);
   Until eof(dosya) or (data.varmi=#0);    {Eof fonksiyonu dosya sonu
gelmişse olumlu cevap verir}
   Sonkayitno:=A;
```

```
End;

 Procedure Kayit;
 Var A:word;
  Begin
      A:=SonKayitno;
      Writeln('Yazılacak Kayit numarasi',A);
      Seek(Dosya,A);
      With Data do         {Data değişkenini kullanarak işlet}
       begin
          Write('Adi:');Readln(Isim);        {Isim aslında Data.isim with bunu
sağlıyor}
          Write('Soyadi:');Readln(Soyad);
          Write('Tel:');Readln(Tel);
          Write('Adresi:');Readln(Adres);
       end;         {Data değişkeni kullanımına son verdi}
       Writeln('Kayit Edilsinmi?(E/H)');
        Readln(ch);
       Data.Varmi:=#42;            {Varmi değişkeni recordun kullanılıp
kullanılmadığına işaret eder. #42 yıldız karakteridir.}
       if (ch=#69) or (ch=#101) then Write(Dosya,Data)
              else  Writeln('Kayit islenmedi IPTAL EDILDI!!!');
   End;

 Procedure Okuma;
 Var A:word;
 Begin
      Writeln('Okunacak Kayit numarasi?: ');Readln(A);
      Seek(Dosya,A);
       If data.Varmi<>#0 then
       With Data do
       begin
          Writeln('Adi',Isim);
          Writeln('Soyadi:',Soyad);
          Writeln('Tel:',Tel);
          Writeln('Adresi:',Adres);
       end  else  Writeln('Kayit Bulunamadı!!!');
   End;

 Procedure Silme;
 Var A:word;
 Begin
      Writeln('Silinecek Kayit numarasi?: ');Readln(A);
      Seek(Dosya,A);
       If data.Varmi<>#0 then
       With Data do
       begin
          Writeln('Adi',Isim);
          Writeln('Soyadi:',Soyad);
          Writeln('Tel:',Tel);
```

```
          Writeln('Adresi:',Adres);
          Writeln('Kayit Silinsinmi?(E/H)');
          Readln(ch);
          Data.varmi:=#0;
          if (ch=#69) or (ch=#101) then Write(Dosya,Data)
              else  Writeln('Kayit silme IPTAL EDILDI!!!');
       end  else  Writeln('Kayit Bulunamadı!!!');
   End;
  Var  A:word;
  Label Bas;
  Begin
     Bas:
       Writeln('Menu:');
       Writeln('1.Kayit');
       Writeln('2.Okuma');
       Writeln('3.Silme');
       Writeln('4.Cikis');
       Writeln('Secim??:');
       Readln(A);
       IF A=1 then kayit;
       IF A=2 then okuma;
       IF A=3 then silme;
       IF A<4 then goto bas;
  End.
```

Özel Komutlar ve Ünite kullanımı:

Pascalda gerek ve ihtayaç duyuldukça yeni işlevsel komutlar yapılmış ve kullanıcılarına sunulmuştur, en son versiyonu olan Borland Pascal 7.0'da da bu komut setleri hat safasına ulaşmıştır, bu komutlar ulaşmak ve gerkeli olanlarını kullanmak için yapılması gereken sadece bu komutların programımıza dahil edilmesi gerektiğidir işte bunun için USES (Kullan) ifadesi kullanılır. Uses kelimesi mutlaka programımızın en üst satırında yer almalıdır;

```
USES Crt;
Begin
   Sound(100);
   Delay(100);
   NoSound;
End.
```

Pascalda kullanılacak unite ve uygulamalar burada açıklanamayacak kadar çeşitli ve parametreseldir bu yüzden devamı için Borlan Pascal 7.0 Help-Units kısmına bakmanızya yarar vardır.

8.4 Delphinin Incelenmesi

Delphi temelde pascal ile aynı compile(derleyici) yapısına sahiptir bu yüzden değişken ve blok yapıları yeniden bahsetmek gereksizdir. Delphi'de farklı olarak object(nesnel) yapı kendini gösterdiğinden en önemli fark uygulama kurulma ve tasarlanma aşamasındadır. Hedef artık text veya koordinatsal bir alan değilde yapısal bir taban üzerindeki sıra yerleşim veya taşıma sıralamasıdır.

Aşağıda yapılan alet için tasarlanmış programın kaynak kodu yer almaktadır;
FORM1 formunun dökümü:

```
object Form1: TForm1
  Left = 166
  Top = 125
  BorderIcons = [biSystemMenu]
  BorderStyle = bsDialog
  Caption = 'Form1'
  ClientHeight = 446
  ClientWidth = 645
  Color = clBtnFace
  Font.Charset = DEFAULT_CHARSET
  Font.Color = clWindowText
  Font.Height = -11
  Font.Name = 'MS Sans Serif'
  Font.Style = []
  OldCreateOrder = False
  Position = poDesktopCenter
  PixelsPerInch = 96
  TextHeight = 13
  object Label1: TLabel
    Left = 8
    Top = 221
    Width = 89
    Height = 13
    Caption = 'Seçilen Dosya Ismi'
  end
  object Label2: TLabel
    Left = 180
    Top = 410
    Width = 74
    Height = 13
    Caption = '9600 baud N81'
  end
  object Label3: TLabel
    Left = 11
    Top = 404
    Width = 63
    Height = 24
    Caption = 'PORT:'
    Font.Charset = DEFAULT_CHARSET
```

```
  Font.Color = clWindowText
  Font.Height = -19
  Font.Name = 'MS Sans Serif'
  Font.Style = [fsBold]
  ParentFont = False
end
object Image1: TImage
  Left = 280
  Top = 8
  Width = 360
  Height = 425
end
object Gauge1: TGauge
  Left = 8
  Top = 248
  Width = 259
  Height = 22
  Color = clBtnFace
  ParentColor = False
  Progress = 0
  ShowText = False
end
object FileListBox1: TFileListBox
  Left = 160
  Top = 8
  Width = 105
  Height = 201
  FileEdit = Edit1
  ItemHeight = 13
  Mask = '*.plt'
  TabOrder = 0
end
object Edit1: TEdit
  Left = 101
  Top = 216
  Width = 166
  Height = 21
  TabOrder = 1
  Text = '*.plt'
end
object Button1: TButton
  Left = 74
  Top = 279
  Width = 129
  Height = 30
  Caption = 'Gönder:'
  TabOrder = 2
  OnClick = Button1Click
end
object DirectoryListBox1: TDirectoryListBox
```

```
    Left = 8
    Top = 33
    Width = 143
    Height = 174
    FileList = FileListBox1
    ItemHeight = 16
    TabOrder = 3
  end
  object DriveComboBox1: TDriveComboBox
    Left = 8
    Top = 10
    Width = 145
    Height = 19
    DirList = DirectoryListBox1
    TabOrder = 4
  end
  object ComboBox1: TComboBox
    Left = 83
    Top = 406
    Width = 89
    Height = 21
    ItemHeight = 13
    Items.Strings = (
      'COM 1'
      'COM 2'
      'COM 3'
      'COM 4')
    TabOrder = 5
    Text = 'COM 1'
  end
  object Button2: TButton
    Left = 74
    Top = 311
    Width = 129
    Height = 30
    TabOrder = 6
    OnClick = Button2Click
  end
  object CheckBox1: TCheckBox
    Left = 16
    Top = 360
    Width = 113
    Height = 25
    Caption = 'Ciziciyede Gonder'
    TabOrder = 7
  end
end
```

UNIT1 dosyasının dökümü:

```
unit Unit1;

interface

uses
    Windows, Messages, SysUtils, Classes, Graphics, Controls, Forms,
Dialogs,
  StdCtrls, FileCtrl, Grids, Outline, DirOutln, Gauges, ExtCtrls;

type
  TForm1 = class(TForm)
    FileListBox1: TFileListBox;
    Edit1: TEdit;
    Label1: TLabel;
    Button1: TButton;
    DirectoryListBox1: TDirectoryListBox;
    DriveComboBox1: TDriveComboBox;
    Label2: TLabel;
    Label3: TLabel;
    ComboBox1: TComboBox;
    Image1: TImage;
    Gauge1: TGauge;
    Button2: TButton;
    CheckBox1: TCheckBox;
    procedure Button1Click(Sender: TObject);
    procedure Button2Click(Sender: TObject);
  private
    { Private declarations }
  public
    { Public declarations }
  end;

var
  Form1: TForm1;

implementation

{$R *.DFM}
var
  cik:boolean;
  xx,yy,zz:longint;

Function Dparse(var s:string;sub:string):string;
begin
  if pos(sub,s)>0 then
  begin
    dparse:=copy(s,1,pos(sub,s)-1);
    delete(s,1,pos(sub,s)+length(sub)-1)
  end else dparse:=''
```

```
end;

Function Atrim(s:shortstring):shortstring;
const
  bos:string=#0#32#255;
var
  a,b:byte;
  dr:boolean;
begin
  dr:=true;
  if length(s)>0 then
  while dr do
  begin
    dr:=false;
    for b:=1 to 3 do
    begin
      a:=pos(bos[b],s);
      if a>0 then
      begin
        dr:=true;
        delete(s,a,1);
      end;
    end;
  end;
  atrim:=s
end;

Function Us(t,u:integer):extended;
var
  a:integer;
  s:extended;
begin
  s:=1;
  for a:=1 to u do s:=s*t;
  us:=s;
end;


Function Lval(s:string):longint;
const
  rkm=[48,49,50,51,52,53,54,55,56,57];
var
  x:byte;
  snc:longint;
  dr:boolean;
begin
  snc:=0;
  s:=copy(atrim(s),1,9);
  for x:=length(s) downto 1 do
    if ord(s[x]) in rkm then
```

```
    snc:=snc+(ord(s[x])-48)*round(us(10,length(s)-x));
  lval:=snc
end;

Procedure Sgit(x,y:longint;dr:boolean);
begin
  {SERI PORTTAN VERI YOLLAMA KISMI}
end;

procedure TForm1.Button1Click(Sender: TObject);
VAR
  dosya:textfile;
  s1,s:string;
  kmt:shortstring;
  dr:boolean;
  a,b,x,y:longint;
  gd,gd2,gdmax:longint;
begin
  {$I-}
  cik:=false;
  Button2.Caption:='IPTAL ET!';
  Button1.Enabled:=false;
  Button2.Enabled:=true;

  assignfile(Dosya,edit1.text);
  reset(dosya);
  for x:=1 to image1.height do
    for y:=1 to image1.Width do
      Image1.Canvas.Pixels[y,x]:=$ffffff;
  gauge1.Progress:=0;
  gd2:=0;
  gd:=0;
  if ioresult=0 then
  begin
    gdmax:=filesize(dosya) div 9;
    while not eof(dosya) and not cik do
    begin
      readln(dosya,s);
      while s<>'' do
      begin
        s1:=dparse(s,';');
        if s1>'' then
        begin
          kmt:=copy(s1,1,2);
          delete(s1,1,2);
          if (kmt='PU') or (kmt='PD') then
          begin
            dr:=(kmt='PD');
          end;
          if pos(',',s1)>0 then
```

```
        begin
          a:=lval(dparse(s1,','));
          b:=lval(s1);
          a:=a div 8;
          b:=b div 5;
          b:=image1.Height-b;
          if dr then Image1.canvas.LineTo(a,b)
            else Image1.canvas.moveTo(a,b);
          if checkbox1.Checked then sgit(a,b,dr);
          inc(gd2);
        end;
      end;
      if gd2>=gdmax then
      begin
        inc(gd);
        gd2:=0;
        gauge1.Progress:=gd;
      end;
    end;
  end;
 end;
 gauge1.Progress:=100;
 closefile(dosya);
 Button1.Enabled:=true;
 Button2.Enabled:=false;
 Button2.Caption:='';
 gauge1.Progress:=0;
 {$I+}
end;

procedure TForm1.Button2Click(Sender: TObject);
begin
 cik:=true;
end;

end.
```

Yararlanılan Kaynaklar:

Mikroislemciler Esaslar , Elemanlar ve Sistemler Milli Egitim Bakanligi Yayini (Çeviri)

The Intel Microprocessors BARRY B. BREY (4.BASKI)

Microchip Pic 16/17 series databook Microchip (1996/1997)

IBM PC/AT Technical Reference

Robot Teknolojisi Ders Notları Yazarı: A.Çoşkun Sönmez ITU

Robot Hands and the Mechanics of Manipulation Yazarı: M.T.Mason

Electronics In Industry Yazarı:George M. Chute 5.baskı
Robotics Yazarı: K.S.Fu, R.C. Gonzales
Hibrid Adım motorunun Modellenmesi ve Similasyonu Yazarı: Yüksek Lisans Tezi M.Aylin Tülay Usta
Robot tutucusu Tasarımı Yazarı: Bitime Ödevi İbrahim Oral Emül
Modern Control Engineering Yazarı: Katshiko Odaka
Robotics And Automated Systems Yazarı: Robert L. Hoekstra
Web üzerinden erişilen inanılmaz kaynaklar:
Web sayfamız altından Roboticle ilgili Linkler kısmında bulabilirsiniz....