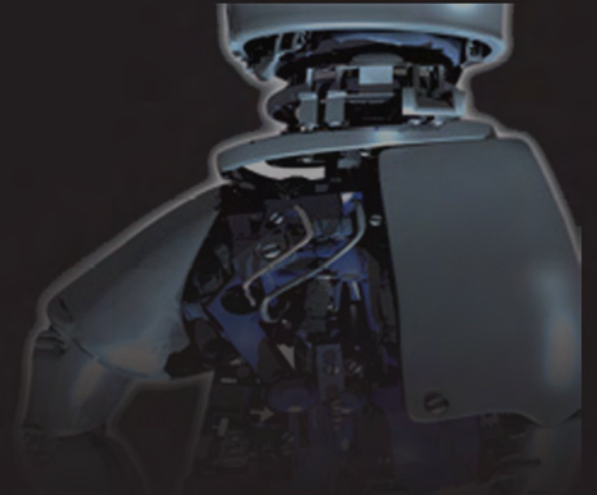


Eylül, 2008

PIC Mikroişlemciler ile Robotik ve Otomasyon Uygulamaları için Gömülü Elektronik Devre Tasarımı

DR E. FARUK KEÇECİ

İZMİR Yüksek Teknoloji Enstitüsü



ISBN 978-9944-89-618-4

TMMOB Elektrik Mühendisleri Odası

EMO E-Kitap

PIC Mikroişlemciler ile Robotik ve Otomasyon Uygulamaları için Gömülü Elektronik Devre Tasarımı

Dr. E. Faruk Keçeci

Eylül 2008

İÇİNDEKİLER

1. Giriş.....	7
2. Mikroişlemciler.....	10
3. PIC Mikroişlemcileri Programlamak için Kullanılan Programlar	14
3.1. Derleyiciler.....	14
3.2. Yükleme Programları.....	15
4. Mikroişlemci Kullanarak Motor Kontrolü.....	18
4.1. Servo Motor	18
4.2. Tarafsız Kutuplu Adım Motor Kontrolü	21
4.3. DC Motor Kontrolü.....	23
4.3.1. Röle Kullanarak.....	23
4.3.2. Entegre Motor Sürücü Devre Kullanarak.....	25
5. Sensörler.....	30
5.1. Dijital Sensörler.....	30
5.2. Analog Sensörler.....	32
6. Uzaktan Kumanda	36
7. Mikroişlemciler Arası İletişim.....	38
7.1. Paralel İletişim.....	38
7.2. Seri İletişim.....	41
8. Mikroişlemci ve Bilgisayar Arası İletişim.....	44
8.1. Seri İletişim.....	44
8.2. Paralel İletişim.....	50
9. Güç Kaynağı.....	54
9.1. Bataryalar.....	54
9.2. KA7805 Voltaj Regülatörü.....	54
10. Proteus Programı.....	57
11. Elektronik Kartların İmalatı ve Dikkat Edilecek Noktalar.....	65
11.1. Baskı Devre Hazırlama.....	65
11.2. Delikli Levha Üzerine Devre Hazırlama.....	66
11.3. Elektronik Kartların Tasarımı Sırasında Dikkat Edilecek Hususlar.....	67
12. Hata Bulma ve Kalibrasyon Cihazı.....	68
12.1 Paralel LCD.....	68
12.2 Kalibrasyon Cihazı.....	70

12.2.1.	Kalibrasyon Cihazının Yapısı	70
12.2.2.	Kartı Fonksiyonları ve Çalışma Şekli.....	73
13.	Sonuç.....	78
	Referanslar.....	79
Ek 1	PICBasicPro Komutları.....	80
Ek 2	MikroBasic Komutları.....	82



Şekil Listesi

Şekil 2.1 PIC16F84A pin diyagramı.....	10
Şekil 2.2 PIC16F877 pin diyagramı.....	11
Şekil 2.3. PIC18F452 pin diyagramı.....	12
Şekil 2.4. PIC16F877 işlemcinin temel bağlantı şeması.....	13
Şekil 3.1 Paralel port PIC programlayıcı.....	16
Şekil 3.2 Seri port PIC programlayıcı.....	16
Şekil 3.3 USB port PIC programlayıcı.....	17
Şekil 4.1. Parallax Standart hobi servo motoru.....	19
Şekil 4.2 Adım motoru iç yapısı ve uçların tetiklenme şekli.....	22
Şekil 4.3 ULN2003 entegresi kullanarak adım motoru kontrolü devre şeması.....	22
Şekil 4.4 Röle sürücü ile DC motorun kontrol şematığı.....	25
Şekil 4.5 LMD18200T entegresinin bacak numaraları ve fonksiyonları.....	27
Şekil 4.6 Darbe Genişlik Modülasyonu sinyali tanımı.....	28
Şekil 4.7 LMD18200T H köprüsü entegrenin DC motoru çift yönlü sürüş devresi.....	29
Şekil 5.1 Enkodör temel yapısı.....	31
Şekil 5.2 Analog bir sensörün çıkış sinyalinin karakteristiği.....	33
Şekil 5.3 GP2Y0A02YK mesafe sensörünün bacak bağlantıları.....	34
Şekil 6.1 Hitech uzaktan kumanda alıcı ve verici modülü.....	36
Şekil 7.1 İki mikro işlemcinin paralel haberleşmesi.....	39
Şekil 7.2 İki mikro işlemcinin seri haberleşmesi.....	42
Şekil 8.1 Servo motor kontrol programı kullanıcı arayüzü.....	45
Şekil 8.2 Servo motorların mikro işlemci üzerinden bilgisayar bağlantısı.....	46
Şekil 8.3 Mikro işlemci ve bilgisayar paralel bağlantı şematığı.....	51
Şekil 8.4 Mikro işlemci ile bilgisayar arası paralel haberleşme arayüz programı.....	51
Şekil 9.1 KA7805 voltaj regülatörü paketleri ve bacak fonksiyonları.....	55
Şekil 9.2 KA7805 voltaj regülatörünün kondansatörler kullanarak devreye bağlanış şematığı.....	56
Şekil 10.1 Proteus programının ana ekranı.....	57
Şekil 10.2 Elektronik eleman seçme ekranı.....	58
Şekil 10.3 Seçilen elektronik elemanın detaylarının görülmesi.....	59
Şekil 10.4 Tasarım ortamı.....	60
Şekil 10.5 Elektronik elemanların bağlantılarının yapılması.....	61
Şekil 10.6 İstenilen devrenin tamamlanmış hali.....	62
Şekil 10.7 Mikro işlemcinin özelliklerinin belirlenmesi ve dosyanın yüklenmesi.....	63
Şekil 10.8 Mikro işlemci programının çalıştırılması.....	64
Şekil 12.1 LCD ekranın mikro işlemciye bağlanması.....	68
Şekil 12.2 Hata bulma ve kalibrasyon cihazı devre şeması.....	72
Şekil 12.3 6 serbestlik dereceli bacak modülü.....	74
Şekil 12.4 Yılan robotun servo motorlarının konumlarının ayarlanması.....	75
Şekil 12.5 DC motor için frekans ve darbe genişliği modülasyonu ilişkisi.....	76

Tablo Listesi

Tablo 4.1 Parallax servo motoru kontrol için gerekli darbe genişlikleri ve PULSOUT değerleri.....	21
---	----



Önsöz

Bu kitap mikro işlemci kullanmak isteyen ama elektronik mühendisi olmayan kişiler için yazılmıştır.

Özellikle TÜBİTAK destekli 104M122 nolu “Bomba İmha Robotu Tasarımı ve Prototipi İmalatı” başlıklı ve İzmir Yüksek Teknoloji Enstitüsü destekli 2006 IYTE038 nolu “Deprem Sonrası Arama Çalışmaları için Kurtarma Robotu Tasarımı ve Prototip İmalatı” başlıklı yürütücüsü olduğum projeler gerçekleştirilirken edinilen elektronik bilgileri bu kitapta özetlenmiştir. TÜBİTAK, DPT ve IYTE kurumlarına robotik konusundaki araştırmalarımı finansal olarak destekledikleri için şükran borçluyum.

Bu çalışma sırasında lisans öğrencim Salih Ağrıdağ’ın sarf ettiği çabalar kelimeler ile anlatılamaz. Kendisine gayreti ve azmi için teşekkür ederim.

Bu kitabın yayına hazırlanmasında gerekli düzeltmeleri yapan İzmir Yüksek Teknoloji Enstitüsü öğretim üyelerinden Dr. Gürsoy Turan ve Dr. Serhan Özdemir’e zaman ayırdıkları ve çok faydalı yorumları için ve kitabın yazılması sırasında gösterdiği sabır ve anlayış için eşim Nina Keçeci’ye teşekkür ederim.

Bu kitabın bütün yazar hakları Türk Eğitim Vakfına bağışlanmıştır.

Dr. E. Faruk Keçeci

İzmir – Eylül 2008

1. Giriş

Bu kitap özellikle elektronik mühendisi olmayan ama mikro işlemci kullanarak robotik ve otomasyon sistemlerini kontrol etmek isteyen mühendisler için yazılmıştır. Elektronik mühendisliği alanında yazılan kitaplar detaylı olarak her bir elektronik elemanı ve mikro işlemcilerin yapısını ve çalışma şekillerini açıklar. Fakat ne yazık ki diğer mühendislik alanlarında uzmanlaşmış insanlar için bu kadar detaylı bir şekilde elektronik öğrenmek çok zaman alıcı olmaktadır.

Diğer bir tür kitap olan proje kitaplarında karşılaşılan sorun ise projelerin yapılan işe göre çok özelleşmiş olmalarıdır ve herhangi bir parçanın bulunamamasında proje gerçekleştirilemez. Projelerin çalışma şekli açıklanır ama çalışma prensipleri açıklanmadığı için okuyucu tarafından herhangi bir değişiklik yapılmak istenirse sistem tam anlaşılamadığından değişiklik yapılamaz.

Bu kitapta özellikle robotik ve otomasyon uygulamalarında gömülü elektronik sistemlerin çalışma prensipleri yapılmak istenen işlevler doğrultusunda açıklanmıştır. Pratik uygulamalara dönük olan bu kitap sayesinde okuyucu uygulamalarda kendi uygulaması için gerekli düzeltmeleri yapabilecek bilgi düzeyine erişecektir. Özellikle uygulamalarda kullanıcıların yapabilecekleri hatalar açıklanmış ve farklı mühendislik alanlarından gelen kişilerin çok fazla detay öğrenmek zorunda kalmadan mikro işlemci kullanabilme yeteneğine sahip olabilmeleri hedef alınmıştır.

Bu kitabın temel amacı robotik otomasyon uygulamalarında mikro işlemcilerin nasıl kullanıldığını açıklamaktır. Robotik ve otomasyon uygulamalarında yapılması hedeflenen işi gerçekleştirecek mekanik bir sistem bulunur. Bu mekanik sistemdeki motorların çalıştırılması, sensörlerden gelen bilgilerin okunması ve sensör bilgilerine göre motorlara gerekli kontrol komutlarının gönderilebilmesi için mikro işlemcilerin kullanılması gerekmektedir. Hesaplama işlemlerinin karmaşıklaştığı durumlar için mikro işlemcilerin birbirleri ile ve bilgisayarlar ile nasıl haberleşebilecekleri de açıklanmıştır.

Mobil robot uygulamaları sırasında robotun iç ve dış ortamı sensörler vasıtasıyla anlayabilmesi ve motorları kontrol edebilmesi için gerekli hesaplamaların ve kontrol

algoritmalarının çalıştırılabilmesi için diz üstü veya el bilgisayarları kullanılabilir. Fakat robotların boyut olarak küçük olması gereken durumlarda, örneğin enkaz içinde arama yapacak bir robotta, zaten faydalı yük taşıma kapasitesi düşük olan robotta elektronik kısmın hafifletilebilmesi için mikro işlemcilerin kullanılması gerekir.

Ağırlığın sorun olmadığı diğer durumlarda ise bazen bilgisayar kullanımı sırasında bilgisayarın çıkış portunda yeterince pin bulunmamaktadır. Böylesi durumlarda mikro işlemciler şifre çözücü elemanlar (decoder) gibi kullanılarak daha fazla pin çıkışı sağlanabilir.

Mikro işlemci kullanması gereken diğer bir durum ise mobil robot içi hesaplamalarda bilgisayar kullanılacak olsa bile bilgi transfer (data acquisition) kartlarının genel olarak masa üstü bilgisayarlar için tasarlanmış olmasıdır. Bu kartlar diz üstü veya el bilgisayarlarına takılamayacağı için mobil robot içine gömülemezler. National Instruments firmasının USB bağlantılı mobil bilgi transfer kartları mevcuttur fakat fiyatları bu kitapta tanıtılan mikro işlemcilerin on katına kadar daha pahalıdır.

Özellikle PIC serisi mikro işlemcilerin seçilmesinin sebebi, bu işlemcileri programak için gerekli yazılımların çok çeşitli olmaları ve ücretsiz olarak kullanılabilmesidir. Ayrıca işlemcileri programlamak için gerekli olan program ve kartlar diğer marka işlemcilere göre daha ucuz ve yaygındır.

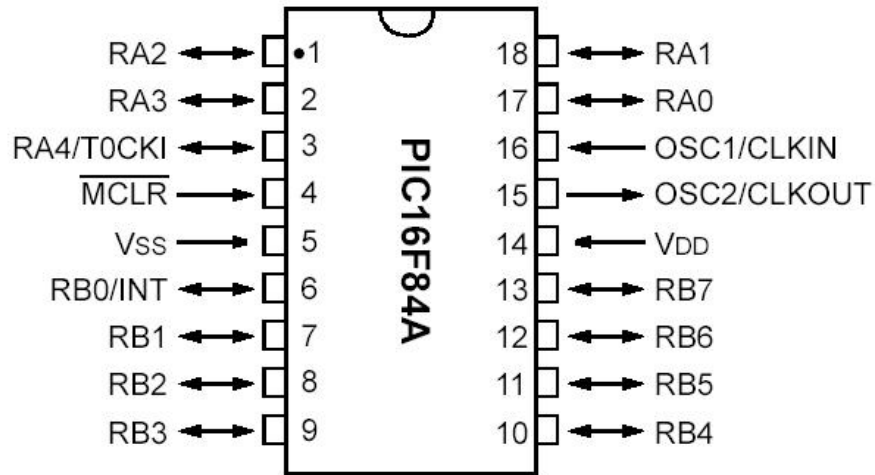
Bu kitapta en yaygın olarak kullanılan PIC16F84, PIC16F877 ve PIC18F452 mikro işlemcileri tanıtılmış ve bu işlemciler için program yazmak ve yüklemek için kullanılan PICBasicPro ve MikroBasic programları kısaca incelenmiştir. Daha sonra mikro işlemcilerle motor kontrolü, sensör kullanımı, kumanda kontrolü, mikro işlemciler arasında ve mikro işlemci ile bilgisayar arasında iletişim hakkında devre şemaları ve program kod örnekleri ile ilgili açıklamalar yapılmıştır. Bilgisayar ve mikro işlemci arasındaki iletişimin nasıl gerçekleştiğini açıklamak için Visual Basic programı ile yapılmış bir servo motor kontrol programının kodları ve PIC programı kodları verilmiştir. Ayrıca robotik ve otomasyon uygulamalarındaki elektronik kartlarda kullanılan voltaj regülatörü KA7805 ve bataryalar tanıtılmış, nasıl kullanılacakları anlatılmıştır. Ayrıca mikro işlemcili devrelerde hata bulma açıklanmış ve kalibrasyon

iřlemi iin gerekli mikro iřlemci kullanılarak tasarlanmıř ok fonksiyonlu bir  l m aletinin devre řeması ve kodları verilmiřtir.



2. Mikroişlemciler

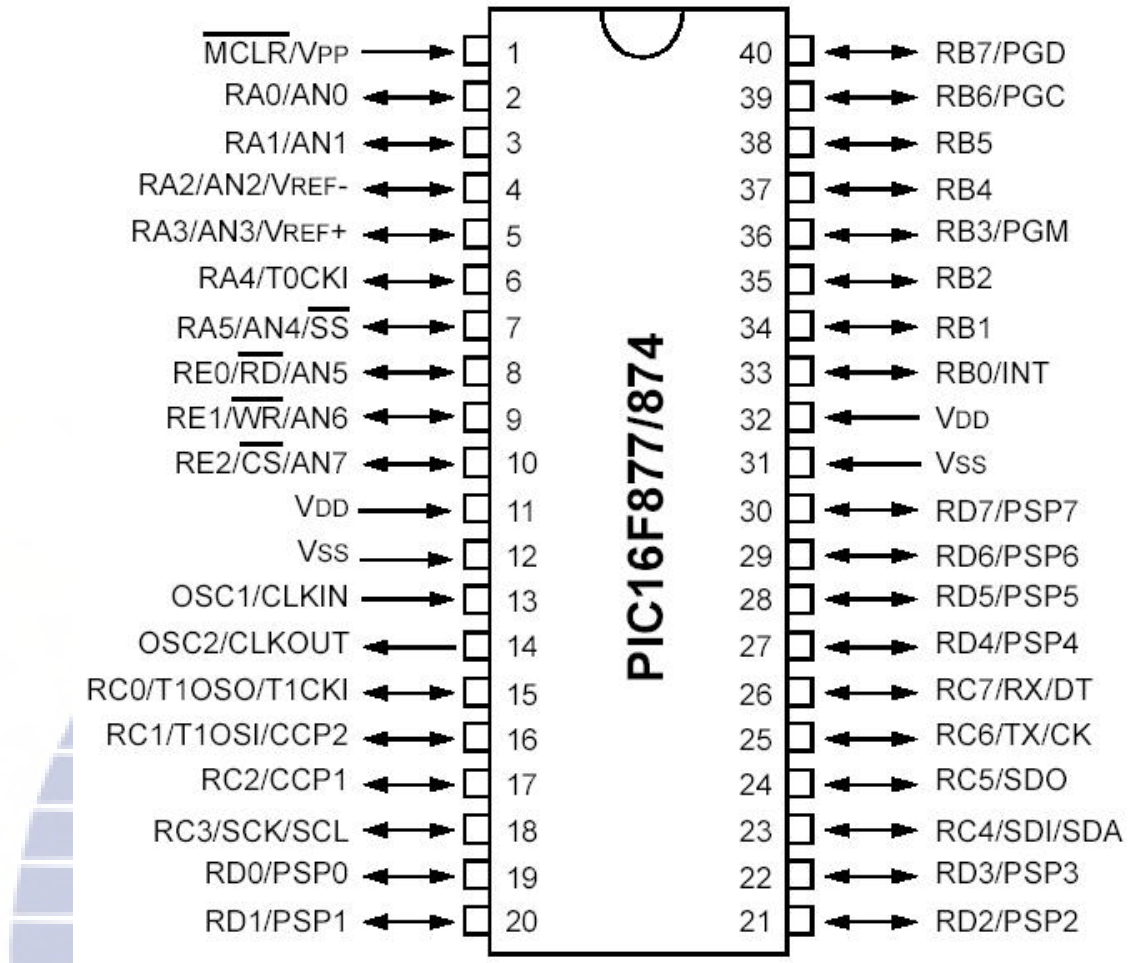
Bu kitapta temel olarak Mikrochip firmasının üretmiş olduğu 3 model PIC mikro işlemci açıklanmıştır. Özellikle PIC işlemcilerin tercih edilmelerinin sebebi fiyatlarının çok ucuz olmasıdır ve diğer marka işlemcilere göre neredeyse yarı fiyata mikro işlemci ve programlamak için gerekli kartların alınabilmesidir. Ayrıca PIC serisi mikro işlemciler için literatürün geniş olması programcının karşılaştığı sorunları kolaylıkla çözebilmesini sağlar. Bu kitapta anlatılan işlemciler, 1 Kb hafızaya sahip 18 pinli PIC16F84, 8 Kb hafızaya sahip 40 pinli PIC16F877 ve 32 Kb hafızaya sahip yine 40 pinli olan PIC18F452 dir. Bu mikro işlemcilerin hangisinin kullanılacağı yazılan programın uzunluğuna, çalışması istenen hıza, kaç adet giriş-çıkış (I/O) işlemi yapılacağına ve elektronik kartta kullanılacak özel işlemlere göre seçilirler. Örneğin basit giriş çıkış işlemleri için PIC16F84 yeterli olurken eğer tasarlanan sistemde analog dijital çevrim kullanılacaksa (analog bir sensor kullanılacaksa) bu işlemci kullanıcının işine yaramayacaktır. Çünkü PIC16F84'ün analog modülü bulunmamaktadır. Bu işlemci yerine, üzerinde analog sinyalleri dijital sinyallere çevrilmesini sağlayan analog modül bulunan PIC16F877 veya PIC18F452 kullanılmalıdır. Bahsedilen mikro işlemcilerin resimleri ve bacak fonksiyonları aşağıda gösterilmiştir.



Şekil 2.1 PIC16F84A pin diyagramı.

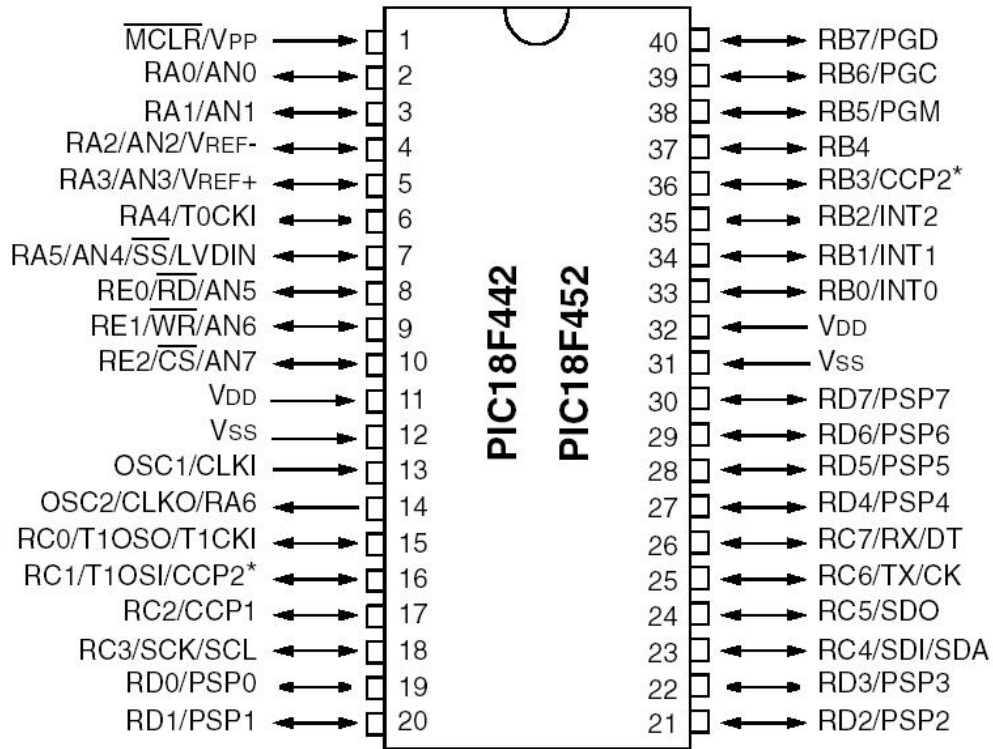
PIC16F84 işlemci çift yönlü giriş veya çıkış özelliğine sahip A ve B olmak üzere iki kısımdan oluşmaktadır, Şekil 2.1. 18 bacaklı bu işlemcinin 15 ve 16 nolu bacaklarına kristal devresi, 4 numaralı bacağına ise sıfırlama direnci bağlanır. A kısmında 5 pin, B

kısımında ise 8 pin yer alır. İşlemcinin çalışabilmesi için 5. bacak toprağa 14 nolu bacak ise pozitif güç kaynağına bağlanır.



Şekil 2.2 PIC16F877 pin diyagramı.

40 bacaklı PIC16F877 işlemcinin, Şekil 2.2, A, B, C, D ve E olmak üzere çift yönlü giriş veya çıkış özelliğine sahip 5 kısmı bulunur. 6 pinden oluşan A kısmı aynı zamanda analog sinyal girişine izin verir. B, C ve D kısımlarının herbiri 8 bacadan oluşmaktadır. E kısmı 3 bacadan oluşmakta, aynı zamanda A kısmı olarakta adreslenebilmektedir. Kiristal devresi 13 ve 14 numaralı bacaklara bağlanırken, 1 numaralı bacadan işlemci sıfırlanabilir. İşlemcinin çalışabilmesi için 12. bacak toprağa 11 nolu bacak ise pozitif güç kaynağına bağlanır.

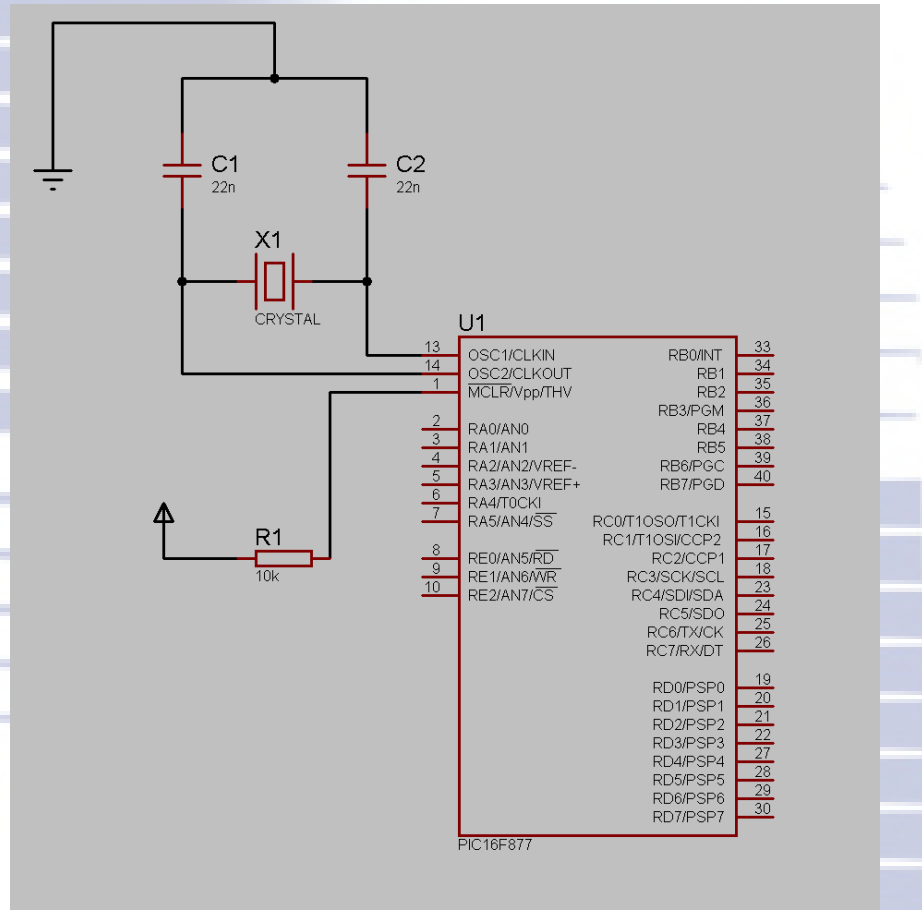


Şekil 2.3. PIC18F452 pin diyagramı

PIC18F452 mikro işlemcisinin kısımları ve bacak bağlantıları PIC16F877 ile tamamen aynı olmasına rağmen bu işlemcinin hafıza kapasitesi daha yüksektir, Şekil 2.3.

Devre tasarımı sırasında mikro işlemci seçilirken mikro işlemci dökümantasyonu mutlaka okunmalıdır. Buradan mikro işlemcinin bellek miktarı özel modülleri ve o mikro işlemci serisinin maksimum kaç Mhz hızda çalışacağı bilgilerine kolayca ulaşılabilir. Örneğin hızlı sinyal işleme işlemlerinin yapılacağı uygulamalar için PIC18F452 işlemcisi seçilmelidir çünkü PIC16F84 ve PIC16F877 maksimum 20Mhz de çalıştırılabilirken PIC18F452 işlemcisini 40Mhz hızda çalıştırmak mümkündür. Burada dikkat edilecek husus seçilen mikro işlemcinin bu çalışma hızını desteklemesidir. Örneğin PIC16F877 işlemcilerin piyasada 4, 8 ve 20 Mhz de çalışacak çeşitleri vardır. Burada bir diğer önemli nokta ise 4 Mhz için tasarlanan bir işlemci 20 Mhz de çalışamazken 20 Mhz'lik bir işlemci 4 Mhz de çalışabilir ve bu işlemciler arasında çok fazla bir fiyat farkı olmaması nedeniyle işlemci satın alınırken yüksek frekansta çalışabilecek bir işlemci seçmekte fayda vardır.

İlgili bacakların farklı kısımlarda olmasına rağmen bu işlemcilerin çalışması için gereken minimum bağlantı şekli hep aynıdır. Bu bağlantı bir sıfırlama devresinden ve bir kristal devresinden oluşmaktadır. Sıfırlama devresi de mikro işlemciyi çalışma sırasında sıfırlar ve işlemci içindeki programın başa dönmesini sağlar. Kristal devresinin görevi mikro işlemcinin çalışması için gerekli olan saat sinyali üretmektir. Kristal devresi bir adet kristal ve 2 adet kondansatörden oluşmaktadır. Kristal ve kondansatörler mikro işlemcinin OSC1 ve OSC2 bacaklarına bağlanır. Sıfırlama devresi ise bir ucu 5V'ta diğer ucu ise mikro işlemcinin MLRC bacağına bağlanan bir adet 10k'lık dirençten oluşmaktadır PIC16F877 için örnek bir sıfırlama ve kristal devresi Şekil 2.4'de görülmektedir. Kristal devresi kurulurken eğer mikro işlemci 2-20 Mhz arası hızlarda çalıştırılmak isteniyorsa 15-33 pF arasında herhangi bir değerde kondansatör seçilmelidir.



Şekil 2.4. PIC16F877 işlemcinin temel bağlantı şeması.

3. PIC Mikro İşlemcileri Programlamak için Kullanılan Programlar

Mikro işlemci kullanmanın temel amacı elektronik devre içinde bir kontrol uygulayabilmektir. Bu kontrol en basit anlamda sensörlerden gelen bilgilere tepki göstermek şeklinde veya belli bir hesaba göre kontrol parametrelerini (motorun hızı gibi) değiştirmek olabilir.

Mikro işlemcinin içine yazılan program sayesinde kontrol mantığı pratikte uygulanabilir. Programın yazılması sırasında derleyici programın komutları kullanılır ve program derlenerek ikilik sayı sisteminde mikro işlemciye yüklenecek .hex dosyası oluşturulur. Daha sonrasında ise bu dosya bir yükleme kartı ve programı kullanılarak mikro işlemciye aktarılır.

3.1 Derleyiciler

Mikro işlemciye program yazabilmek için öncelikle programın bir editör programda yazılıp derlenmesi gerekmektedir. Piyasada C ve Pascal gibi programlama dillerini kullanarak PIC programı yazabilen programlar mevcuttur. Bu kitapta ise uygulamalar PICBasicPro ve MikroBasic dilleri kullanarak yazılmıştır. Bütün bu programlar komutların yazılış şekilleri farklılık gösterse de temel olarak aynı fonksiyonları yerine getirebilmektedirler. Basic programla dili tabanlı PICBasicPro ve MikroBasic dilleri öğrenilmelerinin kolaylığından dolayı seçilmişlerdir. Kullanıcılar daha ileriki projelerinde daha karmaşık ama daha fazla özel fonksiyona sahip PIC C gibi dilleri öğrenme ihtiyacı duyabilirler.

Bu programlar kullanılarak yazılan programlar derlenip ikilik sayı sistemindeki formata dönüştürülürler. Bu iki programında birbirlerine göre üstün veya zayıf olduğu yönler vardır. Örneğin PICBasicPro dili geniş komut çeşitliliğine ve istikrarlı çalışan seri iletişim komutlarına sahip iken MikroBasic programının geniş bir matematik kütüphanesi vardır ve PICBasicPro dili kullanılarak işlemciye yaptırılmayan karmaşık matematiksel hesaplar MikroBasic dilinde kolayca çalışmaktadır. Fakat MikroBasic programının seri iletişim komutları çok kararlı çalışmamaktadır. Programlama dili seçilirken yapılacak işin gerektirdiği komutlar (trigonometrik fonksiyonlar, seri veya paralel iletişim veya darbe genliği okuma/yazma gibi) göz önünde bulundurulmalı ve

programların kullanma kılavuzlarında bulunan kütüphaneleri dikkatlice incelenmelidir. EK1 ve EK2 de MikroBasic ve PICBasicPro programlama dillerinin kütüphanelerinde bulunan komutlar verilmiştir. PICBasicPro programı

<http://www.mecanique.co.uk/products/compiler/pbp-demo-index.html>

adres ve MikroBasic ise <http://www.mikroe.com/en/download/> adresinden indirilip kurulabilir.

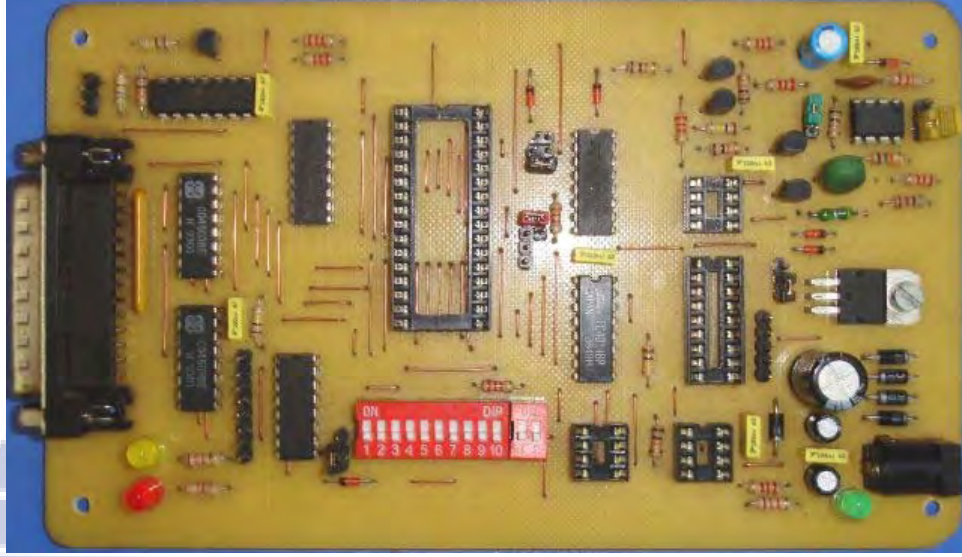
Program yazılırken kullanılacak mikro işlemcinin kristal değerinin programlama sırasında doğru seçilmesi çok önemlidir, aksi halde yazılan program işlemciye yüklendiğinde çalışmayacaktır. Program yazılırken dikkat edilecek diğer bir hususta kod koruma seçeneğinin açık veya kapalı konumunda olduğudur. Eğer bu seçenek açık ise işlemci sadece bir kere programlanabilir, programı yükledikten sonra işlemciyi tekrar programlamak veya içindeki programı okumak bir daha mümkün değildir. Eğer bu seçenek kapalı ise işlemciler birçok kez silinip programlanabilir.

Diğer bir önemli detay ise mikro işlemci için yazılacak kod içinde daha önce hiç kullanmadığınız komutlar varsa bu komutların kullanma kılavuzunda bahsedildiği gibi çalışıp çalışmadığını kontrol etmek için basit devreler kurularak kullanılmasında fayda vardır. Çünkü zaman zaman yapılan projelerde kullanma kılavuzunda bulunan komutların bahsedildiği gibi çalışmadığı gözlenmiştir. Örneğin MikroBasic programının seri iletişim komutları kılavuzda bahsedildiği gibi kararlı şekilde çalışmamaktadır. Bu yüzden seri iletişimin kullanılması gereken uygulamalarda PICBasicPro programı kullanılmıştır.

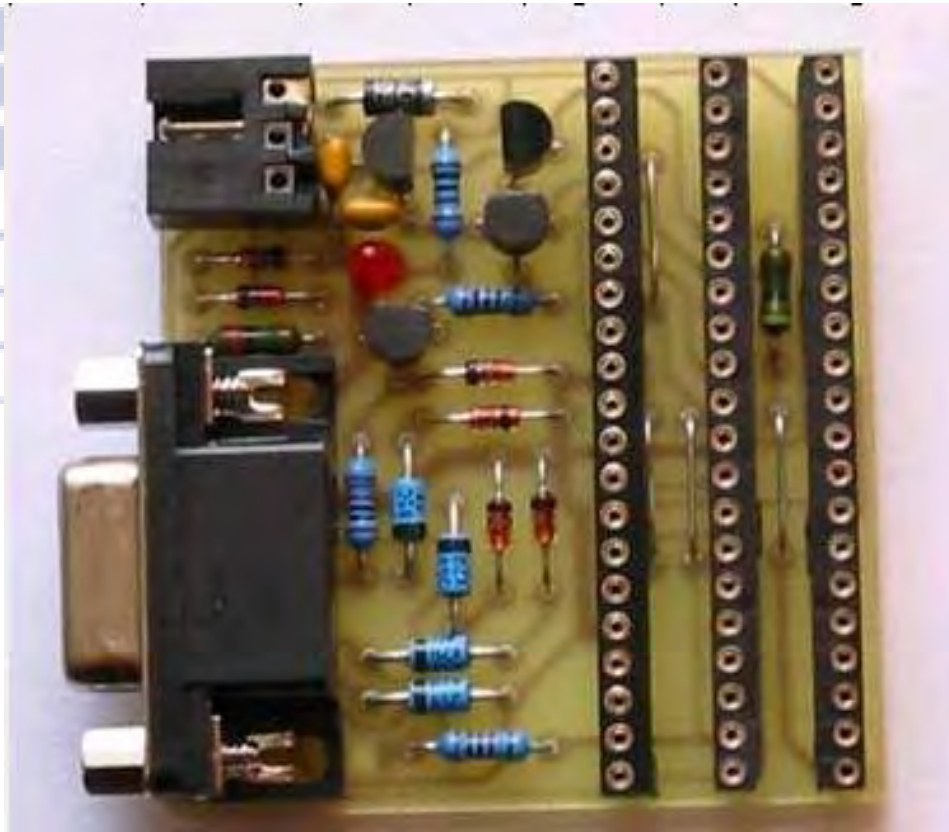
3.2 Yükleme Programları

Mikro işlemci için yazılan ve derlenip ikilik sayı sistemindeki formata dönüştürülen programların mikro işlemciye yüklenmesi için yine bir program ve programlama kartına ihtiyaç vardır. Piyasada paralel porttan, seri porttan ve USB portundan bilgisayara bağlanarak PIC serisi mikro işlemcileri programlayan birçok programlama kartı ve bunların programları bulunmaktadır. Hızlı programlama yapabilmesi ve sürücü programlarının rahatça bulunabilmesi nedeniyle programlayıcı seçilirken USB portundan çalışan kartlar tercih edilmelidir. USB den programlama yapan kartları kullanmanın diğer bir avantajı da kullanılan bilgisayarda port sıkıntısı çekilmemesidir

çünkü bu gün her bilgisayarda en az 2 adet USB port bulunurken özellikle diz üstü bilgisayarlarda seri veya paralel port bulunmayabilir. Aşağıda verilen Şekil 3.1 de paralel port programlayıcı, Şekil 3.2 de seri port programlayıcı ve Şekil 3.3 te ise dELAb firmasının üretmiş olduğu ve bizim projelerimizde kullandığımız USB port kullanan PIC programlayıcının resimleri görülebilir.



Şekil 3.1 Paralel port PIC programlayıcı.



Şekil 3.2 Seri port PIC programlayıcı.



Şekil 3.3 USB port PIC programlayıcı.

Programlama kartı seçilirken dikkat edilecek diğer bir nokta ise programlayıcının programlanmak istenilen mikro işlemciyi destekleyip desteklemediğidir. Unutulmamalıdır ki her programlama kartı her işlemciyi programlayamaz ve bu yüzden programlama kartı seçilirken bu kartların kullanma kılavuzları okunmalı ve hangi marka ve model işlemcileri programlayabileceği incelenmelidir.

Yazılan program yüklenirken programlayıcı kartın programında işlemcinin varsa kristal frekansının doğru seçilmesi gerekmektedir, aksi halde program çalışmaz. Ayrıca mikro işlemci programlama kartına yerleştirilirken bacakları doğru şekilde takılmalıdır işlemci üzerinde bulunan çıkıntı kart üzerinde bulunan çıkıntı ile aynı yönde olmalıdır aksi takdirde işlemci veya programlama kartı zarar görebilir.

4. Mikro İşlemci Kullanarak Motor Kontrolü

Robotik ve otomasyon uygulamalarında servo, adım ve DC motorlar kullanılır. Robotlar ve otomasyon cihazları her zaman fonksiyonları gerçekleştiren hareketli parçalardan ve bu parçaları bir arada tutan durağan parçalardan oluşur. Hareketli parçalar motorlar vasıtasıyla konum değiştirir.

Servo motorlar dönme momentinin çok olması gerekmeyen ve genel olarak yarım turdan daha az şaft dönmesi gereken uygulamalarda kullanılır. Bu tip motorların konum kontrolü kolaylıkla gerçekleştirilebilir. Örneğin, robot üzerine yerleştirilen kamera veya sensörlerin konumlandırılması için servo motorlar kullanılır.

Adım motorlarının genel olarak kullanım amacı motorun çok az dönmesi gereken uygulamalardır. Otomasyon uygulamalarında gereken küçük mesafeli hareketler için adım motorları idealdir.

DC motorlar yüksek döndürme momenti ve şaft hızı gerektiren uygulamalarda kullanılırlar. Motor hızı elektronik sürücü devre kullanıp darbe genişliği modülasyonu ile ayarlanabilir. Bu tür motorlar mobil robotların tahrik sistemlerinde rahatlıkla kullanılabilir.

Elektrik motorlarının dışında hidrolik ve pnömatik motorlar da kullanılabilir fakat bu tür motorların kontrol edilebilirliği düşük olduğu için robotik ve otomasyon uygulamalarında az yer bulurlar. Hidrolik motorlar daha yaygın olarak ağır yükleri, pnömatik motorlar ise hızlı tepki verme kabiliyetlerinden dolayı otomasyon uygulamalarında tetikleme mekanizmalarında kullanılabilirler.

4.1 Servo Motor Kontrolü

Servo motorların kontrolünden önce servo motorların yapısının ve çalışma şeklinin açıklanması gerekir. Servo motorlar temel olarak bir adet DC motor, dişli kutusu ve elektronik karttan oluşur. Dişli kutusu DC motordan gelen torku yükseltmek için kullanılır. Elektronik kart ise üzerinde bulunan motor miline bağlı bir potansiyometre

yardımla motor milinin konumunu kontrol eder ve böylelikle yüksek tork değerlerinde hassas kontrol uygulamayı mümkün kılar.

Bu kitapta anlatılan uygulamalarda kullanılan Parallax marka standart hobi servo motorlarda (Şekil 4.1) 3 adet kablo bulunmaktadır. Bunlar +5V (kırmızı), toprak (siyah) ve sinyal (beyaz) kablolarıdır. Servo motoru çalıştırmak için + 5V ve toprak uçları bağlandıktan sonra sinyal ucundan belli aralıklarla darbe uygulamak gerekir. Darbeler mikro işlemcinin servonun bağlı olan bacağına 1 ve 0 değerlerinin sıra ile gönderilmesi ile sağlanır. Uygulanan darbelerin uzunlukları ve iki darbe arasındaki bekleme süresi servo motorun kontrol edilmesini sağlar. Darbelerin uygulama süreleri servonun gideceği konumu belirlerken darbeler arasındaki bekleme servo motorun istenilen konuma gitme hızını belirler. Motorun kontrolü için yapılması gereken mikro işlemci içine belli aralıklarla sürekli olarak darbeler gönderecek bir program yazılmasıdır. Daha sonra sinyal ucu direk olarak mikro işlemcinin servo motoru çalıştırması istenilen pinine bağlanır. Eğer birden fazla servo motor aynı şekilde çalıştırılacak ise, servo motorların çektikleri sinyal akımların düşük olmasından dolayı iki servo motorun aynı pinden sürülebildiği görülmüştür.



Şekil 4.1. Parallax Standat hobi servo motoru.

MikroBasic programlama dilinde servo kontrolü için özel bir komut bulunmadığı için darbe üretecek kodun beklemler kullanılarak yazılması gerekir. Fakat PICBasicPro programlama dilinde servo motoru kontrol etmek için programın kütüphanesinde bulunan PULSOUT komutu kullanılabilir. Bu komut mikro işlemcinin istenilen pininden istenilen genişlikte ve istenilen sürede puls gönderebilmektedir. Bu şekilde servo motorlar rahatlıkla kontrol edilebilmektedir. İki programlama dilinde de yazılmış

örnek kodlar ve servo motorların hangi darbe değerlerinde hangi açı değerlerini yaptığı aşağıdaki Tablo 4.1 de görülmektedir.

Örnek 4.1.1

Bu kod MikroBasic dilinde B portuna bağlanan servo motorun milini 90° getirmek için kullanılır.

```
TRISB=0      'B portu çıkış olarak ayarlandı
PORTB=0      'B portunun ilk değeri sıfır

MAIN:
WHILE TRUE   'program sonsuz döngüye sokuluyor
PORTB.0=1    'ilk iki satırda 750us boyunca bir darbe gönderiliyor
DELAY_us(750)
PORTB.0=0    '750us lik darbe gönderildikten sonra 2ms bekleme yapılıyor
DELAY_us(2000)
WEND
END.
```

Örnek 4.1.2

Bu kod yukarıdaki darbe gönderme komutuna benzer şekilde darbe göndermektedir. Fakat PICBasicPro dilinde yazılmıştır ve PULSOUT komutu kullanılmıştır. LOOP ve FOR döngüleri ile darbeler seri olarak gönderilmektedir.

```
DEFINE OSC 20
ADCON1 = 7      'analog girişler dijital giriş çıkışa dönüştürülüyor

TRISB=0
PORTB=0

LOOP:
FOR CNT = 0 TO 50
PULSOUT PORTB.5, 100 'B portunun 5. pininden 10ms süreyle puls gönderiyor
PAUSE 20
NEXT CNT
GOTO LOOP      'programı sonsuz döngüye sokuyor
```

Hobi servolarını kontrol etmek için gerekli olan açı değerlerine karşılık gelen bekleme süreleri (darbe genişliği) ve PULSOUT değerleri Tablo 4.1'de gösterilmiştir. MikroBasic dili kullanılacaksa darbe genişliği Örnek 4.1'de ilk bekleme yerine yazılacaktır. Eğer PICBasicPro programlama dili kullanılacak ise PULSOUT değeri ikinci örnekte verilen program kodu içinde yazılmalıdır.

Tablo 4.1 Parallax servo motoru kontrol için gerekli darbe genişlikleri ve PULSOUT değerleri.

Açı değeri	Darbe genişliği	PULSOUT değeri
0°	2.5	1250
30°	2.17	1085
60°	1.83	915
90°	1.5	750
120°	1.17	585
150°	0.83	415
180°	0.17	250

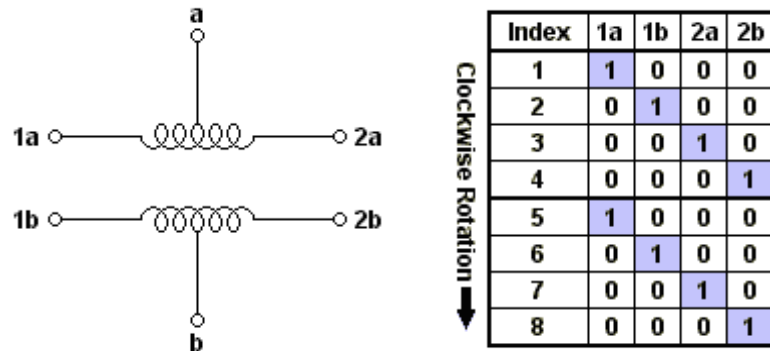
4.2 Tarafsız Kutuplu Adım Motor Kontrolü

Tarafsız kutuplu adım motorunun içyapısı Şekil 4.2’de görülmektedir. Bu tip adım motorlar içinde 2’ye bölünmüş 2 adet (yani toplam 4 adet bobin bulunur). Bu motorlar bobin uçlarına sırayla darbe uygulanarak çalıştırılırlar. Bu motorlarda genel olarak 6 adet kablo ucu bulunur. Bunlardan 2’si 2 şer bobinin ortak uçlarıdır. Bu uçlar şekilde a ve b uçları olarak gösterilmiştir. Diğer 4 uç ise diğer bobinlerin uçlarıdır ve Şekilde 1a, 2a, 1b ve 2b uçları olarak gösterilmiştir. Bu motorları çalıştırabilmek için a ve b uçları toprak veya pozitif güce bağlanır ve daha sonra diğer uçlar tabloda gösterilen sırayla aktif hale getirilerek motor çalıştırılır. Burada dikkat edilecek nokta ortak uçlar toprağa bağlandıysa diğer uçlara pozitif voltaj verilmelidir. Eğer ortak uçlar pozitif güce bağlandıysa diğer uçlar sırayla toprağa bağlanmalıdır.

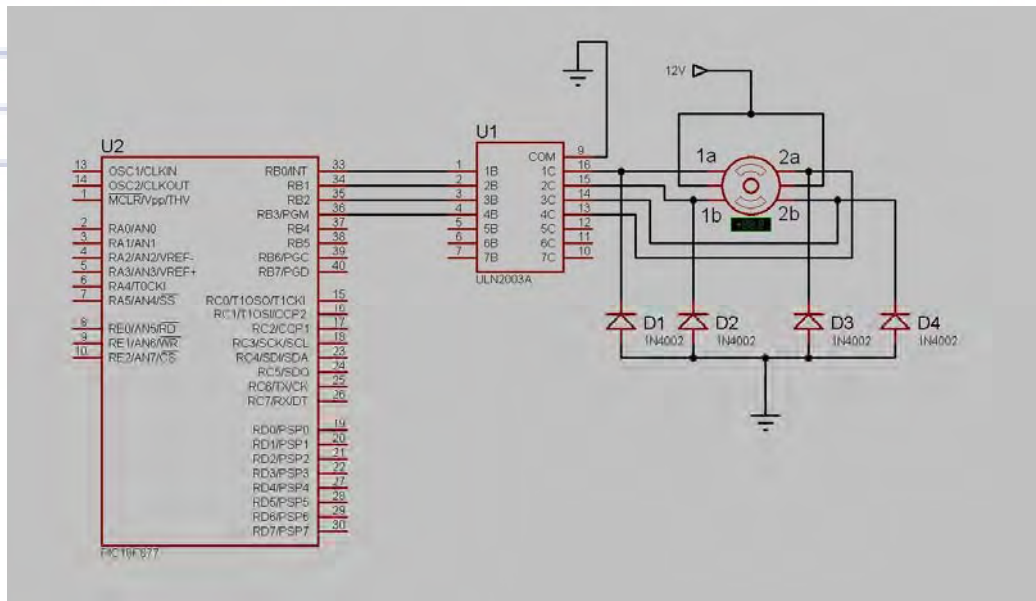
Bir diğer önemli nokta ise adım motorların mikro işlemcinin sağlayabileceğinden daha yüksek akım çektikleri için direk mikroişlemciye bağlanamamasıdır. Bu yüzden sürücü devreler kullanılmalıdır. Sürücü devreler transistörlerle yapılabilindiği gibi piyasada adım motor kontrolü için üretilmiş özel devreler de bulunmaktadır. Bu kitapta basitlik ve ekonomikliğinden dolayı ULN2003 entegre devresi kullanılmıştır. Piyasada yaygın olarak bulunan ULN2003 entegre devresi bir Schmit Triger tetikleyicisidir. Şekil 4.3’de görüldüğü üzere, hiçbir çevresel elemana ihtiyaç duymadan sadece com bacağı toprağa bağlanıp b bacaklarına +5V uygulandığında c bacaklarından negatif çıkış verir. Şekil 4.3’de görülen devrenin çalışması sırasında adım motorunun a ve b ortak bobin uçları +12V ta bağlanmıştır. Adım motorunun 1a, 1b, 2a, 2b uçları sırasıyla ULN2003 entegresinin 1c, 2c, 3c ve 4c uçlarına ve ULN2003 entegresinin 1a, 2a, 3a ve 4a uçları ise sırasıyla mikro işlemcinin PORTB.0, PORTB.1, PORTB.2 ve PORTB.3 uçlarına bağlanmıştır. Burada mikro işlemcinin PORTB.0 aktif olduğunda ULN2003

entegresinin 1c bacağında negatif çıkış oluşur ve adım motorun 1a ve a uçlarının bağlı olduğu bobin 12V ile enerjilenir. Bu işlem sırasıyla diğer uçlara da uygulanır ve motor çalışır. Portlar b0'dan b3 doğru sırayla aktif hale getirilirse adım motoru saat yönünde, b3 portundan b.0 portuna doğru sırayla aktif hale getirilirse motor saat yönünün tersine hareket eder. Şekil 4.3 de görülen devrenin PICBasicPro program kodları Şeklin altında Örnek 4.1.1 de verilmiştir.

Burada dikkat edilmesi gereken diğer bir nokta diyot kullanımıdır. Şekil 4.3'de şemada kullanılan diyotlar motor bobinlerinden oluşacak geri elektromotor kuvvetinin ULN2003 entegresine zarar vermesini önlemektedir.



Şekil 4.2 Adım motoru iç yapısı ve uçların tetiklenme şekli.



Şekil 4.3 ULN2003 entegresi kullanarak adım motoru kontrolü devre şeması.

Örnek 4.2.1:

Şekil 4.3’de gösterilen devrenin adım motorunu kontrol etmesi için mikro işlemciye yüklenmesi gereken PICBasicPro dilinde yazılmış kod örneği burada verilmiştir. Program PORTB.0’dan PORTB.3’e kadar olan pinleri 50 ms aralıklarla aktif hale getirir ULN2003 entegresi de mikro işlemcinin etkinleştirdiği bacağa karşılık gelen bobin uçlarına 12V gerilim uygulayarak motorun saat yönünde çalışmasını sağlar.

DEFINE OSC 4 *‘Kristal hızı tanımlanıyor*

TRISB=0 *‘B portu çıkış olarak tanımlanıyor*

PORTB=0 *‘B portunun bütün pinleri sıfırlanıyor*

LOOP:

PORTB.3=0 *‘Step motorun 2b ucunun enerjisi kesiliyor*

PORTB.0=1 *‘Step motorun 1a ucu enerjileniyor*

PAUSE 50 *‘Bobinin yeterince enerjilene bilmesi için 5 ms bekleme yapılıyor*

PORTB.0=0 *‘Step motorun 1a ucunun enerjisi kesiliyor*

PORTB.1=1 *‘Step motorun 1b ucu enerjileniyor*

PAUSE 50

PORTB.1=0 *‘Step motorun 1b ucunun enerjisi kesiliyor*

PORTB.2=1 *‘Step motorun 2a ucu enerjileniyor*

PAUSE 50

PORTB.2=0 *‘Step motorun 2a ucunun enerjisi kesiliyor*

PORTB.3=1 *‘Step motorun 2b ucu enerjileniyor*

PAUSE 50

GOTO LOOP *‘Program sonsuz döngüye sokuluyor*

END *‘Program sonu*

4.3 DC Motor Kontrolü

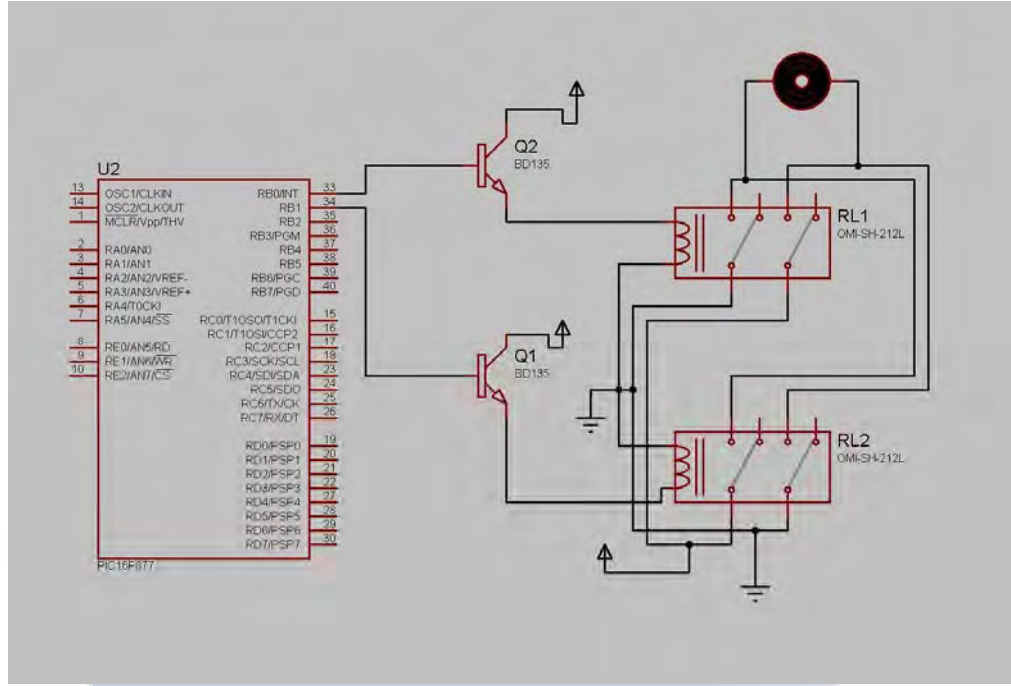
4.3.1 Röle Kullanarak DC Motor Kontrolü

DC motorların kontrolü oldukça basittir ve bu tip motorları kontrol etmek için birçok sürücü devre mevcuttur. Bu kısımda elektromekanik anahtarlar olan röleleri kullanarak DC motor kontrolü açıklanmıştır. Bu şekilde bir kontrol yalnızca aç kapa şeklindedir ve motor tam hızda dönmeye başlar. DC motorun hareket yönü motorun bobin uçlarına uygulanan akımın yönü değiştirilerek kontrol edilir. 2 adet röle ve bir adet mikro işlemci kullanılarak bir DC motor saat yönünde veya saat yönünün tersine hareket ettirilebilir. Bu sistemde birinci röle bobini enerjilendiğinde motorun birinci bobin ucu +12V, ikinci bobin ucu ise toprağa bağlanmış olur. Birinci rölenin enerjisi kesilip ikinci

röle enerji verildiğinde ise motorun birinci bobin ucu toprağa ikinci ucu ise +12V bağlanmış olur ve motor saat yönünün tersine hareket eder. Röleler ile kurulmuş böyle bir sistemin devre şeması Şekil 3.3'te verilmiştir ve bu devreyi çalıştıracak olan PICBasicPro dilinde yazılmış kod örneği de Örnek 3.3.1 te verilmiştir.

Burada programın çok dikkatli yazılması gerekir eğer bir hata sonucu iki röleye de aynı anda enerji verilirse oluşacak kısa devreden dolayı röleler ve mikro işlemci zarar görür. Bu sistemin en büyük dezavantajı ise sistemin tepki süresinin entegre devre ile yapılan motor sürücülere oranla daha uzun olmasıdır. Bu durum, uygulamada motorun daha yavaş açılıp kapanmasına sebep olmaktadır. Bu sebepten röleler ile kontrol edilen bir sistemde PWM uygulanamaz (PWM konusu bir sonraki kısımda daha detaylı olarak anlatılmıştır).

Dikkat edilmesi gereken ikinci önemli nokta ise motoru çalıştırmak için bağlanan rölelerin çektiği akımdır. Eğer yüksek akım çeken bir röle kullanılıyorsa transistörle bir sürücü devre kullanılmalıdır aksi halde sistem mikro işlemci üzerinden çok fazla akım çekerek işlemcinin yanarak zarar görmesine sebep olur. Şekil 4.3'te röleler BD135 NPN transistörler kullanılarak sürülmüştür. 1. röle PORTB.0'a ve 2. röle PORTB.1'e bağlanmıştır. PICBasicPro dilinde yazılmış olan program (Örnek 4.3.1) önce PORTB.0'ı aktif hale getirir Q2 tetiklenir ikinci röle aktif olur ve motor saat yönünde döner. Program yarım saniye bekledikten sonra PORTB.1'i aktif hale getirir ve PORTB.0'ı aktif durumdan çıkarır ve böylelikle Q1 tetiklenir, ikinci röle aktif olur ve motor saat yönünün tersine döner.



Şekil 4.4 Röle sürücü ile DC motorun kontrol şematiği.

Örnek 4.3.1

Şekil 4.4’de verilen devreyi çalıştırmak için gerekli olan PICBasicPro dilinde yazılmış kod aşağıda verilmiştir.

```
TRISB=0
PORTB=0
```

```
LOOP:
PORTB.0=1
PORTB.1=0
PAUSE 500
PORTB.0=0
PORTB.1=1
PAUSE 500
GOTO LOOP
```

4.3.2 Entegre motor sürücü devre kullanarak motor kontrolü

Piyasada birçok marka ve birçok özellikte entegre motor sürücü devreler bulunmaktadır. (A3953, A3955, L6204, L293C, L293D ve LMD18245T). Bunlar uygulamanın ihtiyaç duyduğu motorun çektiği akım ve voltaj değerlerine göre seçilirler ve bu devrelerin çoğunda PWM (pulse width modulation) desteği de bulunmaktadır. Bu kitapta LMD18200T H köprüsü entegre motor sürücü devresi ve PWM uygulama yöntemi açıklanmıştır.

LMD18200T entegre devresi 55V ta 3A kadar olan motorları sürebilmektedir. Ayrıca daha düşük voltaj ve amper değerlerinde de kullanılabilir. Yüksek voltaj ve amper değerlerinde çalışırken entegrenin sağlıklı çalışabilmesi için alüminyum soğutucular bağlanarak soğutulması gereklidir. Şekil 3.5 de görüldüğü üzere entegrenin 11 adet bacağı bulunmaktadır. Robotik ve otomasyon uygulamalarında kullanılan bacaklar ve fonksiyonları şöyledir;

2. Numaralı bacak OUTPUT1: Bu bacak 1. çıkıştır ve kontrol edilecek motorun 1 numaralı bobin ucuna bağlanır.

3. Numaralı bacak DIRECTION INPUT: Bu bacak yön seçimi için kullanılır bu bacağa mikroişlemci tarafından lojik 1 (+5V) değeri uygulandığında motor saat yönünde döner, lojik 0 uygulandığında (bacak topraklandığında) motor saat yönünün tersine döner.

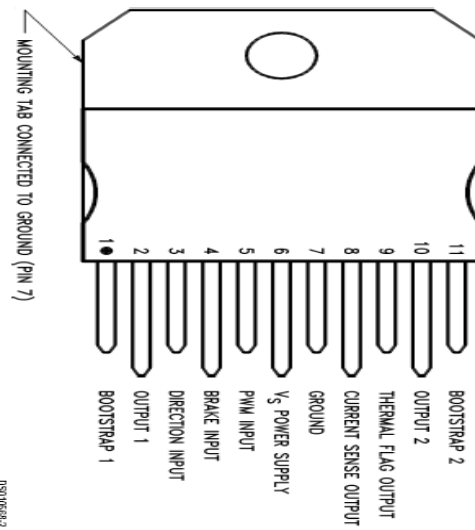
4. Numaralı bacak BREAK INPUT: Bu bacak motorun frenlenmesi için kullanılır lojik 1 değeri geldiğinde motoru frenler. Eğer frenleme kullanılmayacaksa bu bacak 10K'lık bir direnç ile toprağa bağlanmalıdır aksi takdirde entegre çalışmaz

5. Numaralı bacak PWM INPUT: Bu bacak PWM uygulamaları için kullanılır. Eğer kurulan devrede PWM uygulanmayacaksa bu bacak izin ver (enable) olarak kullanılabilir. Motorun çalıştırılması istendiğinde bu bacak aktif hale getirilir (mikro işlemciden +5V uygulanır).

6. Numaralı bacak Vs POWER SUPPLY: Bu bacağa motorun çalışma voltajı olan gerilim uygulanır. Entegrenin maksimum desteklediği 55V geçilmemelidir aksi takdirde entegre yanar.

7. Numaralı bacak GROUND: Bu bacak toprağa bağlanır.

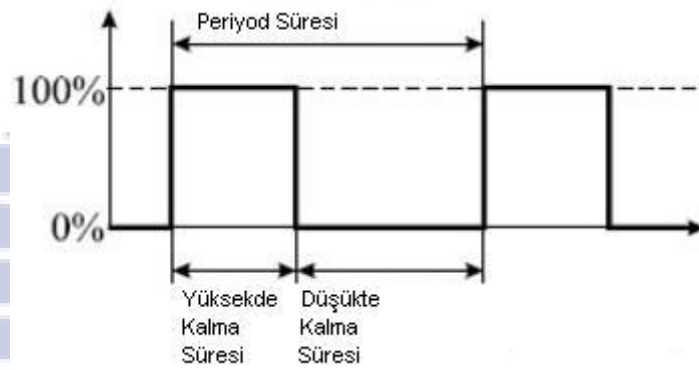
9. Numaralı bacak OUTPUT2: Bu bacak ikinci motor çıkışıdır. Motorun iki numaralı bobin ucuna bağlanmalıdır.



Şekil 4.5 LMD18200T entegresinin bacak numaraları ve fonksiyonları.

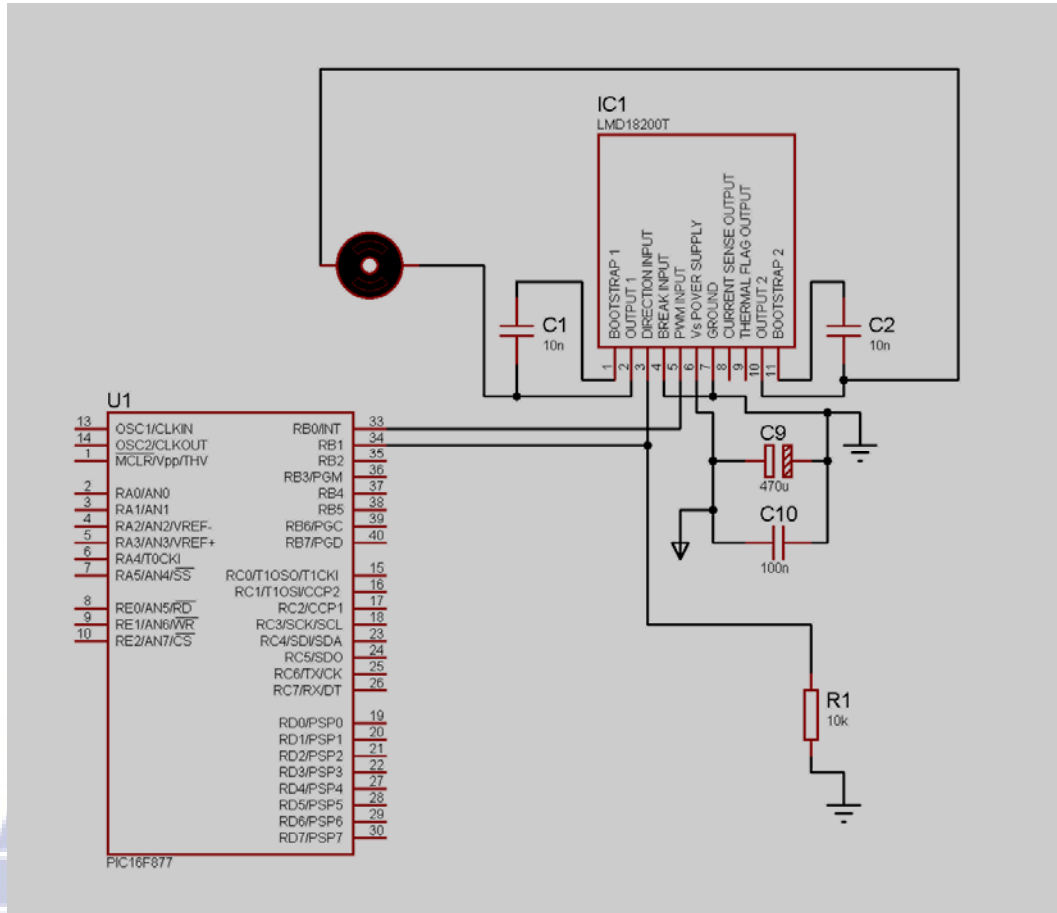
LMD18200T entegresinin kullanımı anlatılmadan önce PWM (Pulse Width Modulation-Darbe Genişlik Modülasyonu) kavramı açıklanmalıdır. PWM, (Darbe Genişlik Modülasyonu) üretilecek olan darbelerin, genişliklerini kontrol ederek, çıkışta üretilmek istenen analog elektriksel değerin veya sinyalin elde edilmesi tekniğidir. Bu kısımda PWM DC motor hız kontrolü için uygulanacaktır. Mikro işlemci belli aralıklarla darbeler göndererek motora giden enerjinin çok kısa zaman aralıklarında kesilmesini ve tekrar motorun enerjilenmesini sağlar ve motorun hızını bu şekilde kontrol eder. Burada motorun hızını belirleyen darbe gönderme süresi ve iki puls arasındaki bekleme süresidir. Şekil 4.6 de bir PWM sinyali görülmektedir. Şekilden de görülebileceği gibi, eğer yüksekte kalma süresi düşükte kalma süresine eşit ise motor %50 yavaşlar. Bu işlemi mikro işlemci ile nasıl gerçekleştirileceği düşünülürse, şekildeki %0 motor için 0V ve %100 motor için nominal çalışma voltajı (5V-bu uygulama için) olur. Mikro işlemcinin herhangi bir bacağından yüksekte kalma süresi, düşükte kalma süresine eşit bir darbe gönderilirse bu pine bağladığımız voltmetreden 2.5V değerini okunur ve teorik olarak yüksekte ve düşükte kalma değerlerini değiştirerek 0-5V arası istenilen voltaj değerine ulaşılır. Fakat özellikle motor hız kontrol uygulamalarında motor hızı %25 in altına düşürülmemelidir motorun içsel sürtünmelerinden dolayı motorun hızı %25 in altına düşürüldüğünde sorun yaşanmaktadır ve motorun hızı olması gerekenden daha da düşük olmaktadır.

Burada bir diğer önemli nokta ise uygulanacak PWM sinyalinin frekansıdır. Çok düşük ve çok yüksek frekanslarda üretilen PWM sinyali motorun çalışmasını olumsuz etkilemektedir. Çok düşük frekanslarda motor vurunutulu çalışırken, çok yüksek frekanslarda motorun titreme yaptığı gözlenmiştir. Önerilen PWM frekansı 1KHz-20KHz arasındır. Frekans değeri deneysel olarak bulunabilmektedir. Motorun aynı PWM ve farklı frekans değerleri için ürettiği tork değeri ölçülür. Doğru frekans değerinde motor maksimum torku üretir.



Şekil 4.6 Darbe Genişlik Modülasyonu sinyali tanımı.

Şekil 4.7 de LMD18200T entegresinin bağlantısı ve Örnek 4.3.2.1 de ise şemada verilen motoru %50 hızında çalıştıracak PICBasicPro kodu verilmiştir. Şekilde de görüleceği gibi entegrenin PWM bacağı PORTB.0'a yön bacağı ise PORTB.1'e bağlanmıştır. Ayrıca oluşabilecek elektronik gürültüleri engellemek için devreye 4 adet kondansatör şekilde gösterildiği gibi bağlanmıştır. Birer adet 10nF'lık kondansatör sırasıyla BOOTSTRAP1 ve OUTPUT1 bacakları arasına ve BOOTSTRAP2 ve OUTPUT2 bacakları arasına bağlanmıştır. Ayrıca toprak ve güç bacakları arasına 470uF ve 100nF'lık kondansatörler paralel olarak bağlanmıştır. BREAK bacağı devrede frenleme kullanılmayacağı için direk olarak toprağa bağlanmıştır.



Şekil 4.7 LMD18200T H köprüsü entegrenin DC motoru çift yönlü sürüş devresi.

Örnek 4.3.2

Aşağıda verilen program, motoru önce saat yönünde yarı hızında çalıştırır ve daha sonra saat yönünün tersine tam hızda çalıştırır.

```
TRISB=0
PORTB=0
```

```
LOOP:
```

```
PORTB.1=1      'Motor saat yönüne dönecek
```

```
FOR I=0 TO 150 'PWM sinyali üretiliyor %50
```

```
PORTB.0=1
```

```
PAUSE 10
```

```
PORTB.0=0
```

```
PAUSE 10
```

```
NEXT I
```

```
PAUSE 500      'Yarım saniye bekliyor
```

```
PORTB.1=0
```

```
PORTB.0=1
```

```
PAUSE 500
```

```
GOTO LOOP
```

5. Sensörler

Robotik ve otomasyon sistemlerinin motorları çalıştırıldığında, sistemin dinamik modellemesinde eksiklikler olduğu için yapılması istenen hareketler tam olarak gerçekleştirilemez. Örneğin dişli çarklar arasında yer alan boşluklardan dolayı sürücü motorun dönme miktarı ile sürüş takımına iletilen dönme miktarı arasında fark oluşur. Bazen de dış etkenler robotun istenilen görevi yerine getirmesini engelleyebilir (mobil bir robotun tekerleklerinin kayarak yönünün değişmesi gibi).

Otomasyon sistemlerinde imal edilen parçaların konumlarının ve oryantasyonlarının anlaşılması önem kazanır.

Sistemlerin kendi iç dinamiklerini ve etraflarını ölçmeleri üzerilerindeki sensörler vasıtasıyla olur. Bu şekilde uygulanacak kontrol yöntemleri için geri besleme sağlanabilir.

Sensörler verdikleri çıkış bigisine göre ikiye ayrılırlar bunlar dijital çıkış veren sensörler ve analog çıkış veren sensörlerdir. Dijital çıkış veren sensörlerin kullanımı analog sensörlere göre daha kolaydır çünkü bu tip sensörler lojik 1 veya 0 olarak çıkış verirler. Sensörden 0V veya 5V olarak gelen sinyal mikro işlemciye doğrudan bağlanarak sensör bilgisi okunabilir.

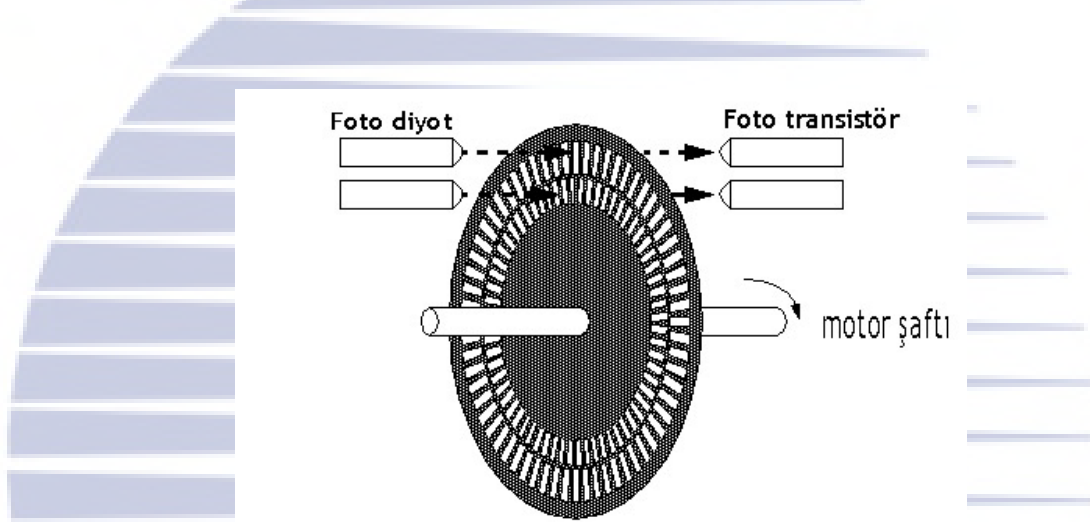
Analog sensörlerin kullanımı ise dijital olanlara göre daha karmaşıktır. Çünkü bu sensörler algıladıkları değeri 0-5V arası voltaj değeri olarak dışarıya verirler ve bu sensörleri okumak için analog/dijital çevrimin yapılması gerekir.

5.1 Dijital sensörler

Bu sensörlere en güzel örnek optik artmalı enkodırlardır. Motorların hızını ölçmek ve/veya konumunu kontrol etmek için kullanılan bu enkodırlar bir adet foto diyot bir adet foto transistor ve üzerine belli aralıklarla delikler açılmış bir diskten oluşurlar. Disk motor miline monte edilmiştir ve motor döndüğünde foto transistor ile foto diyot arasından kesikler geçer. Motorun dönmesi sırasında foto transistor ile foto diyot arasına disk üzerindeki delikli kısım geldiğinde foto transistor foto diyottan gelen ışık

sayesinde aktif hale gelir ve lojik 1 çıkışı verir. Eğer transistör ile diyot arasına deliksiz kısım gelirse transistör aktifleşmez ve 0V değeri çıkarır. Mikro işlemci, bu lojik 1 ve 0 değerlerini okuyarak motorun hızını ve/veya konumu kontrol etmek için kullanılır. Şekil 5.1’de bir enkodırın temel yapısı görülmektedir, Örnek 5.1.1’de ise mikro işlemcinin C0 pinine bağlı bir enkodırdan gelen verileri işleyip motorun kaç tur döndüğünü B portuna bağlı LCD de gösteren PICBasicPro dilinde yazılmış bir program verilmiştir.

Enkodırlarda genellikle 3 kablo bulunur. Bunlardan biri toprak, biri güç ve biride sinyal ucudur. Enkodırın sinyal çıkışı +5V veya -5V olabilir. Encoderin kaç volt çıkış verdiği voltmetre yardımıyla ölçülebilir. Bir diğer önemli nokta ise program yazılırken encoderin bağlanacağı bacağı program içinde giriş olarak tanımlanmasıdır, aksi takdirde program istenildiği gibi çalışmaz.



Şekil 5.1 Enkodır temel yapısı.

Örnek 5.1.1

Aşağıdaki program, mikro işlemcinin C0 portuna bağlı üzerinde 100 adet çizik bulunan bir enkodırdan gelen veriyi okuyup B portuna bağlı olan LCD ekranda motorun kaç dönüş yaptığını gösterir.

DEFINE LCD_DREG	PORTB	'LCD data bacakları hangi porta bağlı
DEFINE LCD_DBIT	4	'LCD data bacakları hangi bitten başlıyor
DEFINE LCD_EREG	PORTB	'LCD enable bacağı hangi porta bağlı
DEFINE LCD_EBIT	3	'LCD enable bacağı hangi bite bağlı
DEFINE LCD_RSREG	PORTB	'LCD RS bacağı hangi porta bağlı
DEFINE LCD_RSBIT	2	'LCD RS bacağı hangi bite bağlı
DEFINE LCD_BITS	4	'LCD 4 bit mi yoksa 8 bit olarak bağlı

```

DEFINE LCD_LINES      2           'LCD kaç sıra yazabiliyor

DEFINE OSC 4
ADCON1 = 7

LCDOUT $FE,1          'LCD temizler
PAUSE 200              'LCD açılması için gerekli süredir

W0 VAR WORD           'Encoder üzerindeki boşlukları sayan değişken
W1 VAR WORD           'Motorun dönüş sayısını tutan değişken
TRISB=%00000000
TRISC=%11111111

LOOP:                  'Program sonsuz döngüye sokuluyor

IF PORTC.0=1 THEN
WHILE PORTC.0=1: WEND   'Encoderin boşluğu geçmesini bekler
W0=W0+1                 'Foto transistor üzerine her ışık geldiğinde W0 değişkeni 1
                        'arttırılıyor
IF W0=100 THEN
W1=W1+1                 'W0 değişkeninin değeri her 100 olduğunda yani motor bir tur
                        'attığında W1 değişkeninin değeri 1 arttırılıyor
W0=0
END IF
END IF

LCDOUT $FE,2,"DONUS SAYISI=",#W1   'Motorun dönüş sayısı LCD'de
                                    'gösteriliyor
PAUSE 500                 'yarım saniyelik bir bekleme oluşturuluyor
GOTO LOOP

```

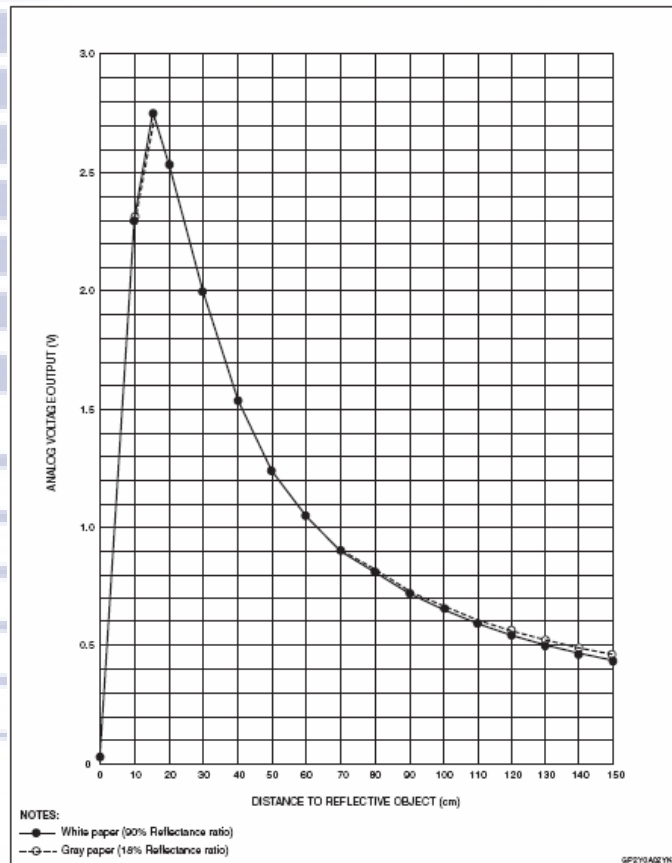
5.2 Analog sensörler

Analog çıkış veren sensörler algıladıkları değeri genellikle 0-5V arası değişen voltaj değerleri olarak verirler. Bu değerlerin işlenebilmesi için PIC16F877 veya PIC18F452 gibi üzerinde analog dijital çevirici modülü olan mikro işlemciler kullanılmalıdır. Yani PIC16F84 ile analog bir sensör direkt olarak okunamaz böyle bir mikro işlemci kullanılacaksa harici bir analog dijital çevirici kullanılmalıdır. Böyle bir entegre devreyi daha karmaşık hale getirir ve sistemin gürültüden etkilenmesine neden olur, ve buda sistem performansını düşürür. Uygulamada gürültüden dolayı mikro işlemci bazı çizikleri kaçırmaktadır.

Mikro işlemci aldığı voltaj değerini 10 bitlik sayısal değere çevirir yani 0-5V arası voltaj değerini 0-1024 arası sayısal değer olarak gösterir. Eğer işlemcinin analog

girişine 2.5V uygulanıyorsa, işlemci bunu 512 sayısal değeri olarak temsil eder. Başka bir deyişle mikroişlemcinin analog sinyal okurken çözünürlüğü 5000/1024 mv/bit'tir. Burada kullanılan sensörün analog çıkışının doğrusal olarak mı artıp azaldığını bilmek büyük önem taşır. Eğer artış veya azalış doğrusal değilse bir fonksiyon oluşturularak bu değişimler mikro işlemci tarafından işlenebilir. Bu bilgiler sensörün kullanma kılavuzundan kolaylıkla elde edilebilir.

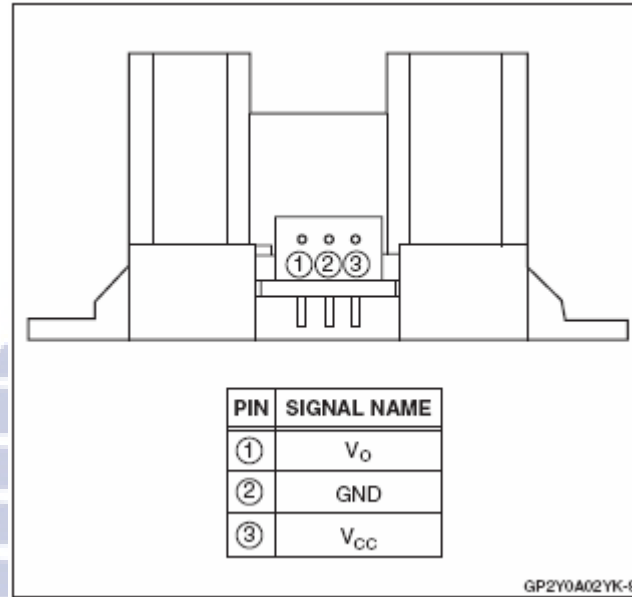
Burada Sharp firmasının üretmiş olduğu GP2Y0A02YK kodlu optik mesafe sensörü incelenmiştir. Bu sensör 20 ile 150 cm arası mesafe ölçümü yapabilmektedir ve bu mesafe aralığında 0-2.75 V arası voltaj değeri çıkış sinyali vermektedir. Şekil 5.2'de de görüleceği üzere bu sensörün çıkış değeri doğrusal olarak değişmemektedir ve sensor bazı birbirinden farklı uzaklık değerleri için aynı voltaj değerini çıkarabilmektedir ve buna programlama esnasında dikkat edilmelidir.



Şekil 5.2 Analog bir sensörün çıkış sinyalinin karakteristiği.

Sensörün bacak bağlantıları Şekil 5.2'de verilmiştir. Şekilde görülen 1 numaralı uç çıkış ucudur. Bu uç mikro işlemcinin analog girişine bağlanır, 2 numaralı uç toprağa bağlanır

ve 3 numaralı uç ise voltaj girişidir, bu uca +5V bağlanacaktır. Örnek 4.2.1’de bir PIC16F877’nin A0 girişine bağlanmış bir adet GP2Y0A02YK mesafe sensöründen gelen veriyi okuyan ve LCD de okuduğu sayısal değeri gösteren PICBasicPro programı verilmiştir.



Şekil 5.3 GP2Y0A02YK mesafe sensörünün bacak bağlantıları.

Örnek 5.2.1

Aşağıda verilen program PIC16F877 mikro işlemcisinin A0 analog girişine bağlanan Sharp GP2Y0A02YK sensöründen gelen analog mesafe bilgisini okuyup okunan ham değeri D portuna bağlı LCD de gösteren program verilmiştir. Program yazılırken dikkat edilecek nokta, kullanılacak analog girişin programda giriş olarak gösterilmesi ve gerekli yazmaçların program içinde ayarlanmasıdır.

```
TRISA=%00000001
```

```
TRISB=0
```

```
TRISC=0
```

```
TRISD=0
```

```
DEFINE LCD_DREG PORTD
```

```
DEFINE LCD_DBIT 4
```

```
DEFINE LCD_EREG PORTD
```

```
DEFINE LCD_EBIT 3
```

```
DEFINE LCD_RWREG PORTD
```

```
DEFINE LCD_RWBIT 2
```

```
DEFINE LCD_RSREG PORTD
```



```

DEFINE LCD_RSBIT      1
DEFINE LCD_BITS       4
DEFINE LCD_LINES      2

```

```

DEFINE OSC 4

```

```

DEFINE      ADC_BITS   10           'A/D çevirim sonucu kaç bit olacak
DEFINE      ADC_CLOCK  3           'Clock kaynağı (3=RC)
DEFINE      ADC_SAMPLEUS 10        'Örnekleme zamanı mikro saniye cinsinden

```

```

ADCON1=%10001110           '7. bit 1 yapıldı 10 bit sonuç almak için

```

```

HAM    var word           'ADC den okunan ham digital değer
LCDOut $FE,1
PAUSE 200

```

```

BASLA:

```

```

ADCIN 0,HAM    '0 nolu kanaldan analog değeri oku ve HAM değişkenine aktarır
BAK:
IF ADCON0.2=1 THEN BAK    'Çevirme işlemi tamamlanınca ADCON0.2=0 olacak
LCDOUT $FE,1," HAM =",# HAM    'Okunan değer LCD de gösteriliyor
PAUSE 500
GOTO BASLA

```

6. Uzaktan Kumanda

Mobil robot uygulamalarında kullanıcının gerekli komutları robota iletmesi gerekir ve bunun için kumanda ve alıcı devresi kullanılmaktadır. Bu kitaptaki uygulamalarda Hitech firmasının ürettiği kumandalar kullanılmıştır, Şekil 6.1.



Şekil 6.1 Hitech uzaktan kumanda alıcı ve verici modülü.

Bu kumandalar servo motorların kontrolü için kullanılırlar. Servo motor kontrolünde de anlatıldığı gibi servo motorlar darbelerin gönderilmesiyle kontrol edilirler. Fakat servo motorların güçlerinin her uygulamada yetmemesinden dolayı bu kumandanın DC motorları aç/kapa yapacak şekilde çalıştırılması gerekmektedir. Bu kumandaların alıcı modülleri servo motorları kontrol edebilmek için gerekli olan genişliklerde darbeler üretmektedir ve bu kumandaları aç/kapa olarak kullanabilmek içinde bu darbelerin genlikleri yani darbelerin uygulanma sürelerinin ölçülmesi gerekir. Bu işlemi yapabilmek için PICBasicPro programında bulunan PULSIN komutu kullanılmıştır. PULSIN komutu mikro işlemciye gelen sinyalin genliğini ölçmektedir. Kumandanın hangi butonundan hangi butonuna basıldığında nasıl bir sinyal çıktığı daha önce PULSIN komutu kullanılarak ölçülebilir. Daha önce yapılan ölçümlerde elde edilen değerler aşağıda sıralanmışlardır:

1. kanal butonunun 1. pozisyonu için 128 msn
1. kanal butonunun 2. pozisyonu için 200 msn
2. kanal butonunun 1. pozisyonu için 188 msn
2. kanal butonunun 2. pozisyonu için 111 msn

- 3. kanal butonunun 1. pozisyonu için 185 msn
- 3. kanal butonunun 2. pozisyonu için 130 msn
- 4. kanal butonunun 1. pozisyonu için 121 msn
- 4. kanal butonunun 2. pozisyonu için 194 msn

Burada dikkat edilecek nokta kumanda çıkışlarının mikro işlemci girişlerine 10K'lık dirençler ile voltaj düşürme (pulldown) yapılarak bağlanmasıdır. Kumandanın herbir kanalı fiziksel olarak alıcıda ayrı bir çıkışa denk düştüğü için sinyallerin karışması gibi bir durum söz konusu değildir. Daha sonra mikro işlemci tarafından çözülen sinyaller işlenerek istenilen komut robota gönderilebilir. Örnek 6.1'de PULSIN komutunun kullanımına örnek bir program görülebilir.

Örnek 6.1

```
@ DEVICE PIC16F877
@ DEVICE PIC16F877, WDT_OFF
@ DEVICE PIC16F877, PWRT_ON
@ DEVICE PIC16F877, PROTECT_OFF
@ DEVICE PIC16F877, XT_OSC

DEFINE LCD_DREG          PORTD
DEFINE LCD_DBIT          4
DEFINE LCD_EREG          PORTD
DEFINE LCD_EBIT          3
DEFINE LCD_RWREG         PORTD
DEFINE LCD_RWBIT         2
DEFINE LCD_RSREG         PORTD
DEFINE LCD_RSBIT         1
DEFINE LCD_BITS          4
DEFINE LCD_LINES         2
DEFINE OSC 4
LOW PORTD.2
LCDOUT $FE,1
PAUSE 200
FI    CON    4          'Frekansın okunacağı pin tanımlanıyor burada B4 pini
W0    VAR    WORD      'Okunan değerın atanacağı değişken

LOOP:
TRISB=%00000001      'Frekansın okunacağı pin giriş yapılıyor
PULSIN FI,1,W0        '10 usnlik aralıklarla frekans okunuyor
IF W0 = 0 THEN DISP   'Eğer okunan değer 0 ise DISP alt programına gidiyor
LCDOUT $FE,1," T(on) =",# W0
PAUSE 1000

DISP:
LCDOUT $FE,1," T(on) =0" 'Eğer okunan değer 0 ise ekrana 0 yazılıyor
GOTO LOOP
```

7. Mikro İşlemciler Arasında İletişim

Mikro işlemcilerin RAM ve ROM belleklerinin sınırlı olması veya bir işlemcinin bacak çıkış sayısının yetersiz olması nedeniyle çoğu robotik uygulamasında birden fazla mikro işlemci kullanmak gerekebilir. Bu işlemcilerin işlem yükünü paylaşabilmeleri için veya bir işlemcide yapılan işlemin sonucuna göre diğer işlemcinin çevresel araçları yönetebilmesi için işlemcilerin birbiriyle haberleşebilmeleri gerekir. Bu haberleşme işlemi seri veya paralel olmak üzere iki türlü gerçekleştirilebilir.

7.1 Paralel İletişim

Kullanması en kolay olan haberleşme yöntemidir. İki mikroişlemci arasına gönderilmek istenen bit sayısı kadar hat çekilir sinyal gönderecek olan işlemcinin haberleşme kısmı çıkış olarak ayarlanır, sinyali alacak olan işlemcinin haberleşme için seçilen kısmı giriş olarak ayarlanır. Ve veriler paralel olarak 1'ler ve 0'lar şeklinde gönderilir. Bu haberleşme şeklinin kolay olması ve sorunsuz çalışmasına karşın gönderilecek her bit için mikro işlemciden bir bacak ayrılması zorunluluğu en büyük dezavantajıdır. Yani bir bytelık veri göndermek için işlemcinin 8 pinini kullanmak gerekmektedir. Fakat seri iletişime göre işlemci belleğinde daha az yer kaplaması nedeniyle az miktarda verinin transfer edileceği basit uygulamalar için oldukça kullanışlı bir yöntemdir.

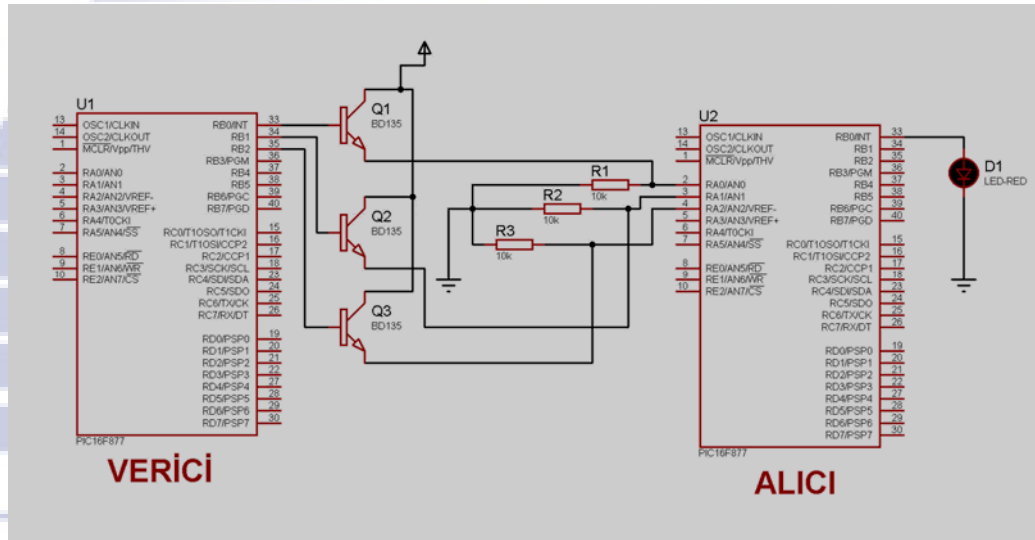
Burada dikkat edilecek nokta iletişimin sorunsuz çalışması için ve veri kaybını engellemek için alıcı mikro işlemcinin haberleşme pinlerinin 10K'lık dirençler kullanılarak voltaj düşürme (pulldown) yapılmasıdır.

Burada voltaj yükseltme (pullup) ve voltaj düşürme (pulldown) kavramları açıklanmalıdır. Voltaj yükseltme mikro işlemciye negatif voltaj seviyelerinde yani dijital 0 olarak giriş yapılmak istendiğinde kullanılan bağlantı yöntemidir. Bu yöntemde 10K'lık bir direnç mikro işlemcinin giriş yapılmak istenilen bacağına bağlanır, direncin diğer ucu ise +5V ta bağlanır. Negatif sinyal girişi direnç ve mikro işlemci pininin bağlandığı noktadan yapılır. Mikro işlemci sinyal gelmediği sürece 10K'lık direnç üzerinden pozitif voltaj aldığı için voltaj yükseltme yapılan pini dijital 1 olarak okur fakat negatif sinyal geldiğinde bu pindeki voltaj negatif seviyelere iner ve mikro işlemci artık bu pini lojik 0 olarak okur ve bu şekilde negatif sinyaller mikro işlemciye okutulmuş olur. Voltaj düşürme bağlantısı ise voltaj yükseltme bağlantısının tam tersi

şekilde olur. Burada direncin ucu +5V ta değil de toprağa bağlanır ve mikro işlemci bu pinden sürekli lojik 0 okur. Bu pine pozitif sinyal geldiğinde yani +5V uygulandığında pindeki voltaj seviyesi pozitif düzeye çıkar ve mikro işlemci bu pinden lojik 1 değerini okumaya başlar böylelikle pozitif sinyaller mikro işlemciye girilmiş olur.

Bazı uygulamalarda sinyal akımının yetersiz kalması voltaj düşürme akımını yenememesi nedeniyle bu pindeki değer lojik 1 değerine ulaşamaz. Bu durumda haberleşilecek pinlerin tetiklenmesi için transistor kullanılmalıdır.

Şekil 7.1’de 3 bitlik tek yönlü haberleşen iki adet mikroişlemcinin devre şeması verilmiştir. Örnek 7.1’de ise bu mikro işlemcilerin haberleşmesi için gerekli olan PICBasicPro dilinde yazılmış program verilmiştir.



Şekil 7.1 İki mikro işlemcinin paralel haberleşmesi.

Yukarıdaki şekilde görüleceği gibi alıcı mikro işlemcinin A0, A1 ve A2 pinleri gelen veriyi okumak için ayrılmış ve 10K’lık dirençler kullanılarak voltaj düşürme yapılmıştır. Vericiden sinyal gelmediği yani verici işlemcinin B0, B1 ve B2 pinleri düşük iken alıcı A pinlerinden 0 değerini okur. Verici sinyal gönderdiğinde yani B0, B1 ve B2 pinlerinden herhangi biri veya hepsi birden lojik 1 olduğunda yani bu pinlerden 5V çıkış oluştuğunda pine bağlı olan transistor aktif hale gelir ve bağlı olduğu alıcı işlemcisinin pinini aktif hale getirir ve alıcı işlemci bu pinden 1 değerini okur. Şekilde görülen bağlantı ile verici işlemciden istenilen komutlar ikilik düzende alıcıya gönderilir. Veriler ikilik düzende gönderildiği için bu 3 bitlik bağlantı ile verici işlemci

alıcı işlemciye toplam 8 adet komut gönderebilir. Bu komutlar ikilik sayı sisteminde şöyledir. 000, 001, 010, 011, 100, 101, 110 ve 111'dir. Alıcının içine yazılacak kod yardımıyla bu gelen değerler işlenerek alıcının istenilen işlemleri yapması sağlanır. Örnek 7.1.1'de verici mikro işlemcisinden 101 verisini yarım saniye aralıklarla gönderen bir program ve alıcı içinde de vericiden 101 değeri geldiği sürece B0 pinine bağlı ledi yakan fakat 000 değeri geldiğinde yani alıcıdan sinyal gönderilmediğinde ledi söndüren bir program örneği verilmiştir.

Örnek 7.1.1

Burada alıcı işlemcinin kodunda ikinci satır önemlidir. Bu satırda $ADCON1 = 7$ komutu yazılmıştır. Bu komut mikro işlemcinin analog girişlerini dijital giriş ve çıkışa çevirir. Eğer programda bu satır yazılmaz ise, A portu dijital giriş veya çıkış olarak kullanılamaz ve program çalışmaz.

Verici Mikro İşlemci Programı

```
DEFINE OSC 4      '4 Mhz lik kristal kullanılıyor

TRISB =%00000000  'B Portu çıkış olarak ayarlandı
PORTB=%00000000   'B Portunun başlangıç değeri 0 olarak belirlendi

LOOP:
PORTB=%00000101   'B0 ve B2 pinleri 1 yani 101 değeri yollanıyor
PAUSE 500          'Yarım saniyelik bekleme
PORTB=0            'B portu 0 yapıldı yani 000 değeri yollanıyor
PAUSE 500
GOTO LOOP
```

Alıcı Mikro İşlemci Programı

```
DEFINE OSC 4
ADCON1 = 7        'Normalde analog giriş olan A portu dijital olarak ayarlandı

TRISA =%11111111  'A portu giriş olarak ayarlandı
PORTA=%00000000   'A portunun başlangıç değeri 0 olarak belirlendi
TRISB =%00000000  'B portu çıkış olarak ayarlandı
PORTB=%00000000   'B portunun başlangıç değeri 0 olarak belirlendi

LOOP:
IF PORTA=%00101 THEN  '101 değeri gelmişse B0 pini aktif olur ve
PORTB.0=1              'B0 a bağlı led yanar
END IF
```



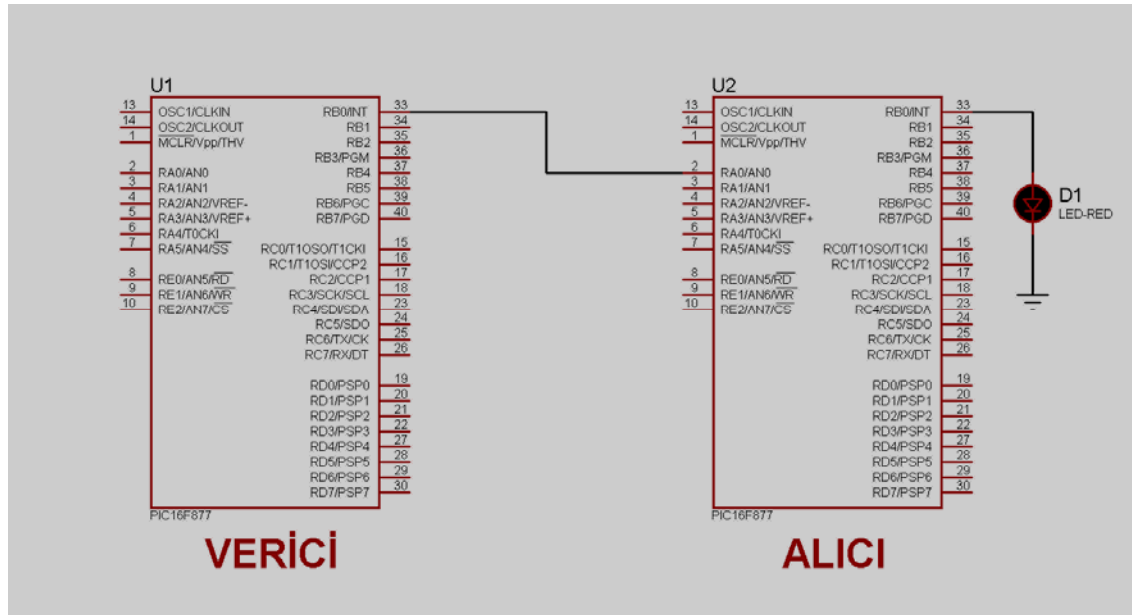
```
IF PORTA=0 THEN      'PORTA 0 ise yani veri gelmiyorsa led söner
PORTB.0=0
END IF
GOTO LOOP
```

7.2 Seri iletişim

Paralel iletişime göre biraz daha karmaşık bir iletişim şekli olmasına rağmen standartlaşmış bir iletişim şeklidir. İki mikro işlemci tek bağlantı üzerinden yani tek kablo üzerinden 8 bitlik veriyi gönderebilir ve yine tek bir kablo üzerinden 8 bitlik veriyi alabilir. Yani iki mikro işlemci sadece iki pin kullanarak 8 bitlik iletişim kurabilirler birbirlerinden veri alıp yollayabilirler.

Seri iletişim veri göndermek USART (Universal Sencron/Asynchronous Receiver/Transmitter) protokolünü kullanır. Bu protokol adından da anlaşılacağı gibi senkron veya asenkron iletişime izin verir. Bu kitaptaki uygulamalarda asenkron iletişim yöntemini kullanılmıştır. Bu yöntem ile mikro işlemciler birbirine sürekli veri gönderip alabilirler. Bu protokolde gönderilecek 8 bitlik veri; bir başlama biti 8 bitlik gönderilecek veri paketi ve bir stop biti olmak üzere toplam 10 bitlik veri paketleri halinde değişik hızlarda (boud rate) gönderilir. Uygulamalar sırasında bu bitleri tek tek çözmek yerine paket programların kütüphanelerinde bulunan komutların kullanılması programı basitleştirir ve iletişimi kolaylaştırır. Uygulamalarda PICBasicPro dili kullanılmıştır ve seri iletişim için bu programın kütüphanesinde bulunan ve gayet istikrarlı çalışan SERIN2 ve SEROUT2 komutlarını kullanılmıştır. Bu komutlardan SEROUT2 komutu ile mikro işlemcinin istenilen bir pininden seri olarak istenilen hızda veri göndermek mümkündür. Aynı şekilde SERIN2 ile mikroişlemcinin istenilen bir pininden istenilen hızda veri almak mümkündür.

Seri iletişim için mikro işlemcilerin bağlantısı oldukça basittir hiçbir çevresel eleman kullanmadan veri gönderilecek ve veri alınacak pinler birbirine bağlanır. Şekil 7.2 de bir biriyle seri olarak haberleşen mikro işlemcilerin devre şeması verilmiştir. Örnek 7.2.1 de bu işlemcileri seri olarak haberleştirecek ve verici işlemciden A harfi gelince B0 portuna bağlı ledi yakan B harfi geldiğinde ise ledi söndüren bir program örneği verilmiştir. Bu programda verici işlemcinin B0 portundan seri veri gönderilmektedir ve alıcı işlemcinin A0 portundan seri bilgi okunmaktadır.



Şekil 7.2 İki mikro işlemcinin seri haberleşmesi.

Örnek 7.2.1**Verici Mikro İşlemci Programı**

```
DEFINE OSC 4
```

```
TRISB = %00000000
```

```
PORTB = %00000000
```

```
LOOP:
```

```
SEROUT2 PORTB.0,2400,["A"]
```

'B0 pininden 2400 boud rate

```
PAUSE 500
```

'A harfine karşılık gelen 8 bitlik veri gönderiliyor

```
SEROUT2 PORTB.0,2400,["B"]
```

```
PAUSE 500
```

'B harfine karşılık gelen 8 bitlik veri gönderiliyor

```
GOTO LOOP
```

Alıcı Mikro İşlemci Programı

```
DEFINE OSC 4
```

```
ADCON1 = 7
```

```
TRISA = %11111111
```

'A portu giriş olarak ayarlandı

```
PORTA = %00000000
```

'A portunun başlangıç değeri 0 olarak belirlendi

```
TRISB = %00000000
```

'B portu çıkış olarak ayarlandı

```
PORTB = %00000000
```

'B portunun başlangıç değeri 0 olarak belirlendi

```
VERI VAR BYTE
```

'A0 portundan gelen verinin tutulduğu byte tipi değişken

```
LOOP:
```

```
SERIN2 PORTA.0,2400,[VERI]
```

‘mikro işlemci A0 portundan seri olarak aldığı 8 bitlik veriyi VERI adlı değişkenin içine atar daha sonra bu değişkenin aldığı değere göre programda işlem yapılır

IF VERI="A" THEN *‘VERI değişkeninin içindeki değer A harfinin ascı karşılığı ise*
PORTB.0=1 *‘B0 pinine bağlı led yakılır*
END IF

IF VERI="B" THEN *‘VERI değişkeninin içindeki değer B harfinin ascı karşılığı ise*
PORTB.0=0 *‘B0 pinine bağlı led söndürülür*
END IF
GOTO LOOP



8. Mikroişlemci ve Bilgisayar Arasında İletişim

Bazı robotik ve otomasyon uygulamalarda mikro işlemcilerin hafızalarının ve hızlarının yetersiz kaldığı görülür. Bazen de görüntü işleme gibi yüksek işlemci gücü gerektiren işlemler veya sensörlerden gelen bilgilerin saklanması istenir. Bu işlemlerin yapılması için mikro işlemciler yetersiz kaldığı zaman işlemci ve hafıza gücü için bilgisayar kullanmak gerekir. Fakat motorları kontrol etmek ve sensörlerden gelen verileri okuyabilmek için hala mikro işlemcilere ihtiyaç vardır. Bu durumda bilgisayar ve mikro işlemcinin birlikte çalışması gerekmektedir.

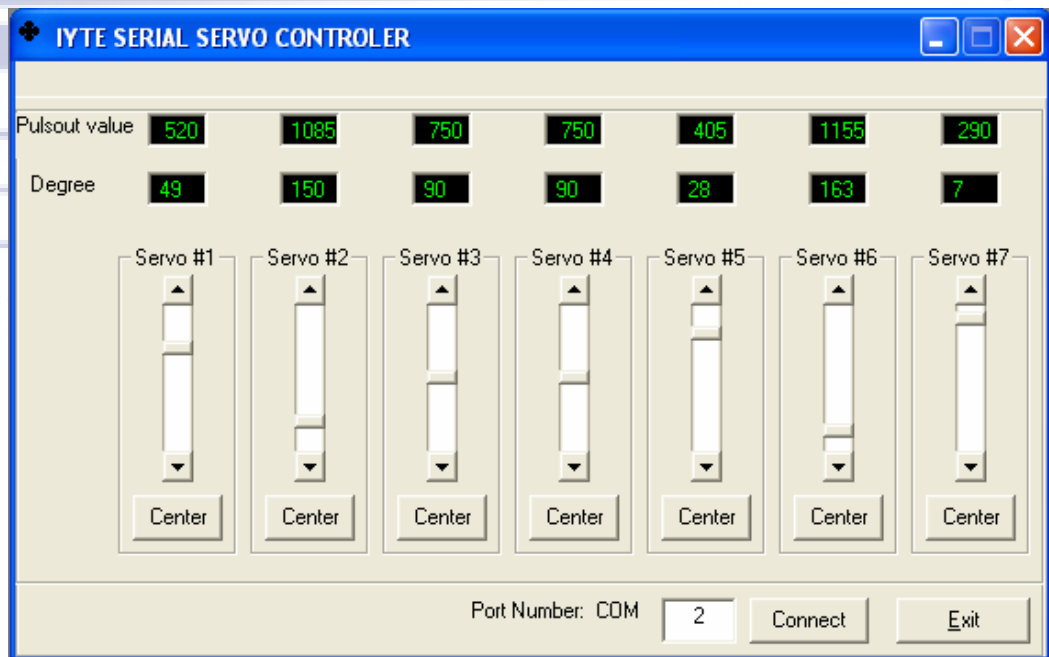
8.1 Seri iletişim

Bu kitaptaki uygulamalarda bilgisayarın RS232 portunu kullanarak bilgisayar ve mikro işlemciyi seri olarak haberleştirilmiştir. Mikro işlemcinin ve bilgisayarın daha önce bahsedilen USART protokolünü kullanmasına rağmen voltaj seviyelerinin farklı olması bu işlem için MAX232 gibi voltaj seviyesini dönüştüren devre elemanlarına ihtiyaç duyulur. Mikro işlemciler 0-5V arası işlem yaparken, RS232 portu -12V ile +12V arası çalışır. MAX232 entegre devresinin kullanılması ile mikro işlemciden gönderilen sinyalin voltaj değeri 12V'a yükseltir aynı şekilde bilgisayardan veri aktarıldığında 12V'luk sinyal mikro işlemcinin çalışma voltajı olan 5V seviyesine indirilir.

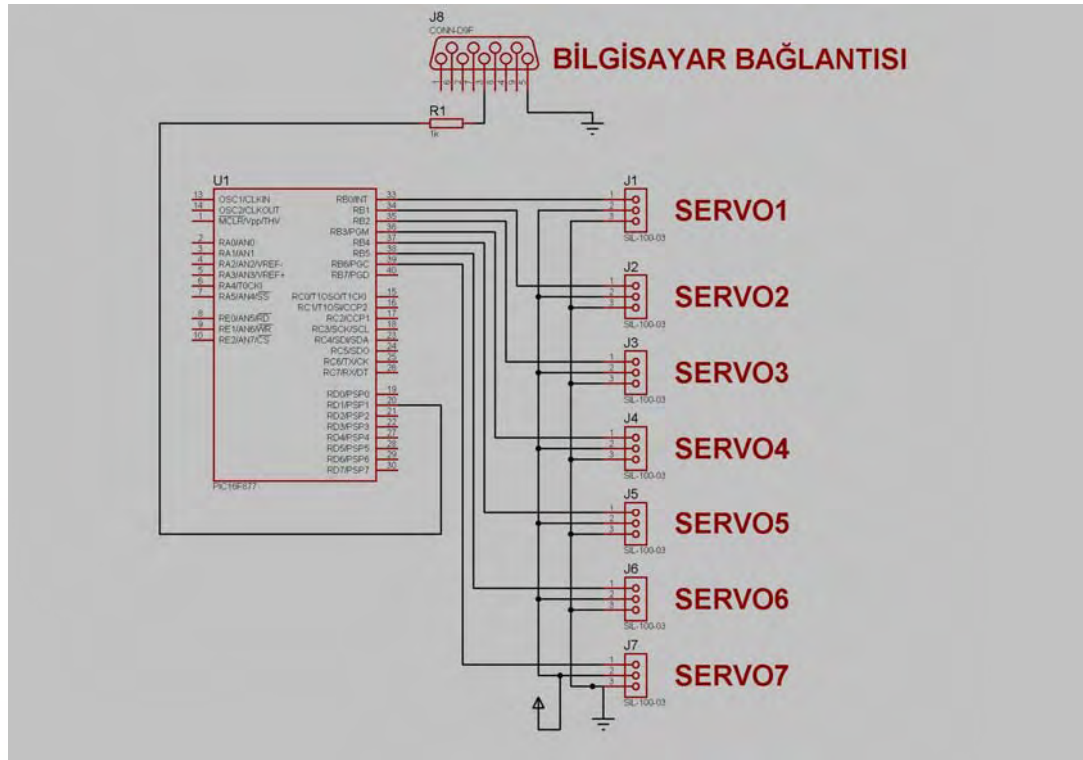
Burada mikro işlemci ve bilgisayarı birlikte çalıştırabilmek için ve kullanıcının işlemciye veri gönderebilmesi ve veri alabilmesi için bilgisayar programına ihtiyaç duyulur. Yani bir grafik ara yüz programı yazılmalıdır. Ara yüz programını yazmak için C veya Delphi gibi değişik programlama dilleri kullanılabilir bu kitaptaki uygulamalarda seri iletişim için özel kütüphanesi bulunan Microsoft firmasının geliştirdiği VisualBasic programlama dili kullanılmıştır.

Bilgisayar ve mikro işlemcinin birlikte çalışacağı sistemlerde en önemli noktalar, ara yüz programı yazılırken bilgisayar portunun doğru seçilmesi ve mikro işlemci ile bilgisayar programlarının aynı hızda çalıştırılmasıdır. Bilgisayarın hangi portunun o an için aktif olduğu başlat kısmından çalıştır komutu kullanılıp DEVMGMT.MSC komutu yazılarak görülebilir.

Bilgisayar ve mikro işlemcinin birlikte çalışmasını 7 adet sevoyu kontrol etmek için hazırlanan visualbasic programı ve PIC programı üzerinden inceleyebiliriz. Burada kullanıcı 7 adet sevoyu bilgisayara bağlı bir PIC16F877 mikro işlemci sayesinde kontrol eder. Çubuklar yardımıyla açı değerlerini ve hangi servo motorun çalışacağı bilgisini girer ve program bu değerleri mikro işlemciye gönderir. Mikro işlemci de bu değerlere göre darbeler üretir ve servo motorları çalıştırır. Kullanıcı ayrıca servo motorları hareket ettiren açı değerlerine karşılık gelen PULSOUT değerlerini bilgisayardan takip edebilir. Programın ekran görüntüsü Şekil 8.1’de gösterilmiştir. Ayrıca Şekil 8.2’de gerekli devre şeması görülmektedir. Bu programda mikro işlemciden veri okunmasına ihtiyaç olmadığı için tek yönlü bir iletişim kullanılmıştır, kullanıcı sadece bilgisayardan veri göndermektedir. Devre şemasında görüleceği üzere MAX232 entegresi kullanılmamış voltaj seviyesini ayarlamak için 1K’lık bir direnç bilgisayardan gelen bağlantının 3. pinine bağlanmıştır. Ayrıca, bağlantının 5. pini toprağa bağlanmıştır. Örnek 8.1.1’de mikro işlemci ve grafik ara yüz programları görülmektedir. Burada mikro işlemci programında SERIN komutu kullanılmıştır. Bu komut sayesinde istenilen pinden seri giriş alınabilir ve SERIN2 komutunun aksine bu komut “word” tipi değişkenler ile de çalışabilir. Yalnız dikkat edilmesi gereken nokta programa include **"modedefs.bas"** satırının eklenmesidir. Bu satır SERIN komutunu kullanmak için gerekli olan kütüphane bilgilerini programa dahil eder.



Şekil 8.1 Servo motor kontrol programı kullanıcı arayüzü.



Şekil 8.2 Servo motorların mikro işlemci üzerinden bilgisayar bağlantısı.

Örnek 8.1.1

Visual Basic Kodu

VisualBasic kodu yazılırken MsComm modülü kullanılmıştır. Kod yazılmadan önce VisualBasic ekranının sağ tarafında bulunan bu sembol alınarak formun üzerine sürüklenmelidir.

```
Private Sub Command1_Click() 'Connect butonu
If MSComm1.PortOpen = True Then 'Butona basıldığında port açıksa kapatıyor
MSComm1.PortOpen = False 'kapalı ise aşağıdaki değerlere göre portu açıyor
Else
MSComm1.Settings = "2400,N,8,1" '2400 boud rate No parity 8 bit ve 1 stop biti
MSComm1.CommPort = Text15.Text 'Com portu değerini Connect butonu
MSComm1.PortOpen = True 'yanındaki text box tan alıyor ve portu açıyor
End If
End Sub
```

```
Private Sub cmdCenter1_Click() '7. center butonuna basılırsa
MSComm1.Output = Chr$(7) 'Servo numarası gönderiliyor
MSComm1.Output = Chr$(75) 'Servoyu 90 ye getirmek için gerekli pulsout değeri
'bu değer mikro işlemci tarafından
'10 ile çarpılarak kullanılıyor
```


Text1.Text = Str(750) *'scroll barın üzerinde bulunan text boxta darbe değeri gösteriliyor*

VScroll2.Value = Text1.Text / 5 *'scroll bar 50 ile 250 arası değerde çalışıyor ve bu satırda ortaya getiriliyor*

```
Private Sub cmdCenter2_Click()  
    MSComm1.Output = Chr$(6)  
    MSComm1.Output = Chr$(75)  
    Text2.Text = Str(750)  
    VScroll2.Value = Text2.Text / 5  
End Sub
```

```
Private Sub cmdCenter3_Click()  
    MSComm1.Output = Chr$(5)  
    MSComm1.Output = Chr$(75)  
    Text3.Text = Str(750)  
    VScroll3.Value = Text3.Text / 5  
End Sub
```

```
Private Sub cmdCenter4_Click()  
    MSComm1.Output = Chr$(4)  
    MSComm1.Output = Chr$(75)  
    Text4.Text = Str(750)  
    VScroll4.Value = Text4.Text / 5  
End Sub
```

```
Private Sub cmdCenter5_Click()  
    MSComm1.Output = Chr$(3)  
    MSComm1.Output = Chr$(75)  
    Text5.Text = Str(750)  
    VScroll5.Value = Text5.Text / 5  
End Sub
```

```
Private Sub cmdCenter6_Click()  
    MSComm1.Output = Chr$(2)  
    MSComm1.Output = Chr$(75)  
    Text6.Text = Str(750)  
    VScroll6.Value = Text6.Text / 5  
End Sub
```

```
Private Sub cmdCenter7_Click()  
    MSComm1.Output = Chr$(1)  
    MSComm1.Output = Chr$(75)  
    Text7.Text = Str(750)  
    VScroll7.Value = Text7.Text / 5  
End Sub
```

```
Private Sub VScroll1_Change() '7. servonun scroll barı değişirse
```

```
Dim asil As Byte
Text1.Text = Str(VScroll1.Value * 5)    'darbe değeri scroll barın değeri
                                         '5 ile çarpılarak bulunuyor
asil = Round((Text1.Text - 250) * (180 / 1000)) 'Açı değeri hesaplanıyor
Text8.Text = Str(asil)                  'Açı değeri gösteriliyor
MSComm1.Output = Chr$(7)                'Servo numarası gönderiliyor
MSComm1.Output = Chr$(Text1.Text / 10) 'Pulsout değeri bilgisayara gönderiliyor
End Sub
```

```
Private Sub VScroll2_Change()
Dim asil As Byte
Text2.Text = Str(VScroll2.Value * 5)
asil = Round((Text2.Text - 250) * (180 / 1000))
Text9.Text = Str(asil)
MSComm1.Output = Chr$(6)
MSComm1.Output = Chr$(Text2.Text / 10)
End Sub
```

```
Private Sub VScroll3_Change()
Dim asil As Byte
Text3.Text = Str(VScroll3.Value * 5)
asil = Round((Text3.Text - 250) * (180 / 1000))
Text10.Text = Str(asil)
MSComm1.Output = Chr$(5)
MSComm1.Output = Chr$(Text3.Text / 10)
End Sub
```

```
Private Sub VScroll4_Change()
Dim asil As Byte
Text4.Text = Str(VScroll4.Value * 5)
asil = Round((Text4.Text - 250) * (180 / 1000))
Text11.Text = Str(asil)
MSComm1.Output = Chr$(4)
MSComm1.Output = Chr$(Text4.Text / 10)
End Sub
```

```
Private Sub VScroll5_Change()
Dim asil As Byte
Text5.Text = Str(VScroll5.Value * 5)
asil = Round((Text5.Text - 250) * (180 / 1000))
Text12.Text = Str(asil)
MSComm1.Output = Chr$(3)
MSComm1.Output = Chr$(Text5.Text / 10)
End Sub
```

```
Private Sub VScroll6_Change()
Dim asil As Byte
Text6.Text = Str(VScroll6.Value * 5)
asil = Round((Text6.Text - 250) * (180 / 1000))
Text13.Text = Str(asil)
```

```

MSComm1.Output = Chr$(2)
MSComm1.Output = Chr$(Text6.Text / 10)
End Sub

```

```

Private Sub VScroll7_Change()
    Dim asil As Byte
    Text7.Text = Str(VScroll7.Value * 5)
    asil = Round((Text7.Text - 250) * (180 / 1000))
    Text14.Text = Str(asil)
    MSComm1.Output = Chr$(1)
    MSComm1.Output = Chr$(Text7.Text / 10)
End Sub

```

```

Private Sub CmdExit_Click()      'Exit butonu
    MSComm1.PortOpen = False    'Port kapatılıyor
    End                          'Program sonlandırılıyor
End Sub

```

Mikro İşlemci Kodu

Burada SERIN komutunun kullanımı önemlidir. SERIN PORTD.1,N2400, SERVO,POSITION satırı bu komutun kullanımı detaylı bir şekilde açıklar. Bu satırda PORTD.1 pininden seri 2400 hızda eş olmayan (N harfi no parity belirler) alınan ilk veri servo adlı değişkene atanır aynı şekilde gelen ikinci veri pozisyon değişkenine atanır. Bu uygulamada, VisualBasic kodunda önce kaç numaralı servonun çalışacağı bilgisi gönderilir, ikinci olarak da PULSOUT değeri 10'a bölünerek gönderilir. 10'a bölmenin sebebi bilgisayardan byte tipi değişkenler gönderilebilmesidir. Aksi takdirde değişkenin değeri 256'yı geçerse program çalışmayacaktır.

Daha sonra bu alınan değerler PULSOUT komutu kullanılarak mikro işlemci tarafından servo motorlara gönderilir. PULSOUT SERVO,(POSITION*10) satırı incelenecek olursa burada servo motor değeri B portundan PULSOUT komutunun kullanılacağını gösterir. Yani eğer servo değişkenin değeri 1 olursa komut PULSOUT PORTB.1,(POSITION*10) şeklinde çalışır. Son olarak da bilgisayar tarafından hesaplanıp 10'a bölünerek gönderilen PULSOUT değeri tekrar 10 ile çarpılarak mikro işlemci tarafından servo motorlara gönderilir.

```

DEFINE OSC 20
ADCON1 = 7

```

```

INCLUDE "modedefs.bas"

```

```

TRISB=0
PORTB=0
TRISD=0
PORTD=0

```

```

GIRIS VAR PORTD.1
SERVO VAR WORD
POSITION VAR WORD
CNT VAR BYTE

```

```

LOOP:

```

```

SERIN PORTD.1,N2400,SERVO,POSITION

```

```

FOR CNT = 0 TO 50      'servoların gerekli pozisyona gitmesi
  PULSOUT SERVO,(POSITION*10)  'için gereken bekleme yaratılıyor
  PAUSE 20
  NEXT CNT
GOTO LOOP

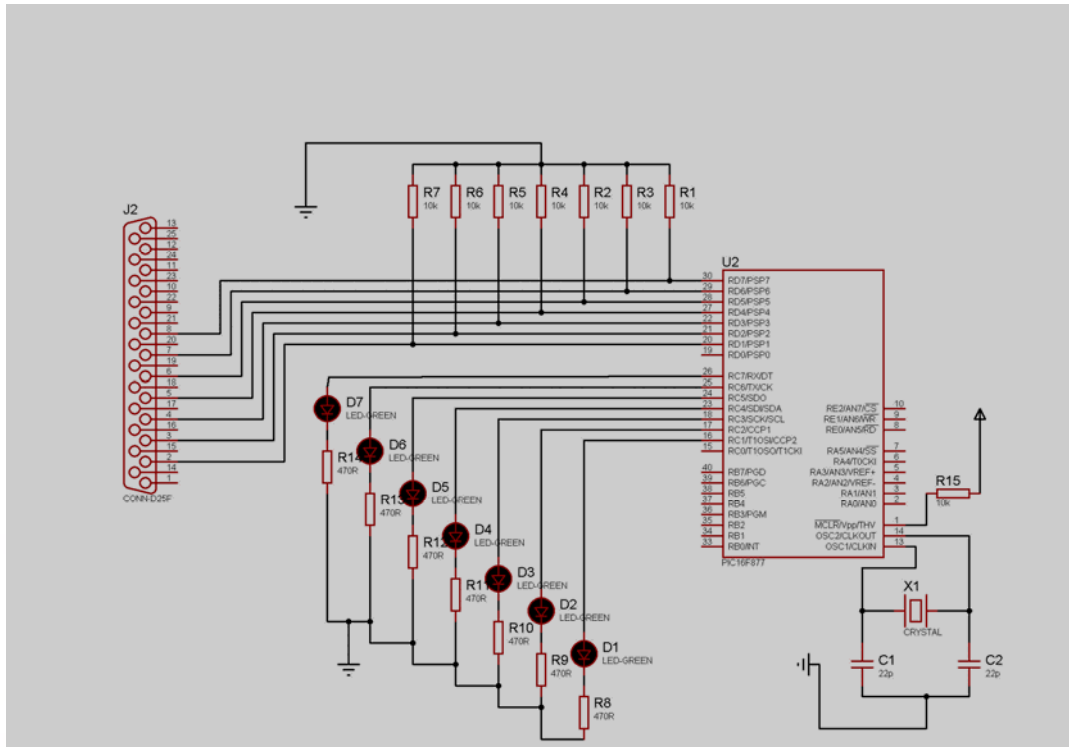
```

8.2 Paralel iletişim

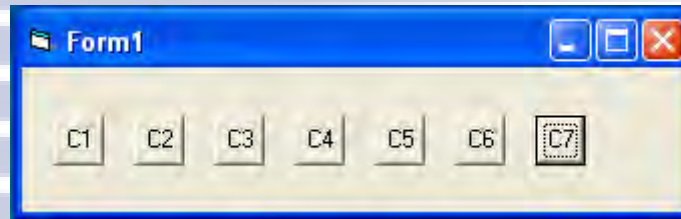
Bilgisayar ile çevresel cihazların kullanımı için kullanılan diğer bir yöntem ise paralel port kullanımı ile paralel iletişimdir. Paralel iletişimin kullanımı seri iletişime göre daha kolay olmasına rağmen paralel iletişimin eski bir teknoloji olması çoğu yeni bilgisayarda paralel portun bulunmaması bu yöntemin dezavantajlarıdır.

Bu yöntem basit uygulamalar için idealdir. Paralel iletişim bilgisayar üzerinde bulunan 25 pinlik LTP portu üzerinden yapılır. Bu portun 2, 3, 4, 5, 6, 7, 8 ve 9. bacakları dijital çıkıştır. Bu pinler herhangi bir programlama dilinde yazılan grafik ara yüz programı ile aktif hale getirildiğinde +5V çıkış vermektedir ve bu çıkış kolaylıkla mikro işlemciye giriş sinyali olarak aktarılıp bilgisayardan gelen komutlar mikro işlemciye aktarılabilir.

Örnek 8.2.1 de mikro işlemcinin C portuna bağlı led diyotları bilgisayardaki ara yüze konan butonlara basıldıkça yakıp söndüren bir devrenin mikro işlemci ve VisualBasic dilinde yazılmış kodları görülebilir. Ayrıca Şekil 8.3’de mikro işlemcinin ledlere ve bilgisayara bağlantısının devre şeması ve Şekil 8.4’te ise grafik ara yüzü programının ekran görüntüsü gösterilmiştir.



Şekil 8.3 Mikro işlemci ve bilgisayar paralel bağlantı şematiği.



Şekil 8.4 Mikro işlemci ile bilgisayar arası paralel haberleşme arayüz programı.

Örnek 8.2.1

Bu kısımda bilgisayarın paralel portuna bağlı olan bir adet PIC16F877 mikro işlemcisinin C portuna bağlı olan ledleri yakıp söndüren bir programı açıklanmıştır. Arayüz programı VisualBasic dilinde yazılmıştır. Kullanıcı ara yüz üzerinde bulunan butonlara bastığında mikro işlemciye bağlı ilgili led yanmaktadır. Yani kullanıcı C1 butonuna basarsa mikro işlemcinin C1 bacağına bağlı led yanmaktadır, aynı şey diğer butonlar için de geçerlidir. Windows güvenlik sebebiyle port erişimini engellediği için portlara erişimde sorun yaşanabilir. Bu sorunu aşmak için bu tür paralel iletişim uygulamalarında inpout.32 adlı bir .dll dosyası kullanılmaktadır. Bu .dll program çalıştırılmadan önce Windows içinde bulunan system ve system32 dosyalarına

kopyalanmalıdır. Ayrıca bu program VisualBasic programı yazılırken programın başında tanımlanmalıdır. Tanımlama şu şekilde yapılır;

```
Private Declare Function Inp Lib "inpout32.dll" _  
Alias "Inp32" (ByVal PortAddress As Integer) As Integer
```

```
Private Declare Sub Out Lib "inpout32.dll" _  
Alias "Out32" (ByVal PortAddress As Integer, ByVal Value As Integer)
```

Paralel porta bilgi göndermek için Out Val("&H378"),8 komutu kullanılır. Burada &H378 veri gönderilmek istenen port numarasıdır, 8 ise gönderilmek istenen veridir.

Programın çalışması oldukça basittir örnek olarak kullanıcı C5 butonuna bastığında Out Val("&H378"), 8 komutu aktif olur. Bilgisayar porta 8 bilgisini gönderir ve port üzerinde bulunan 5 numaralı pin aktif olur yani buradaki voltaj değeri +5V olur bu bacak şekil 7.2.1 de görülebileceği gibi mikro işlemcinin D5 bacağına bağlıdır. Bu bacak normalde voltaj düşümü olarak bağlıdır. Portun 5 numaralı pini aktif olunca normalde 0 değeri okunan bu bacadan 1 değeri okunmaya başlar ve mikro işlemci C portunun 5. pini aktif hale getirerek 5. pine bağlı ledi yakar. Diğer butonlara basıldıkça bu işlem tekrarlanır.

Visual Basic Kodu

```
Private Declare Function Inp Lib "inpout32.dll" _  
Alias "Inp32" (ByVal PortAddress As Integer) As Integer
```

```
Private Declare Sub Out Lib "inpout32.dll" _  
Alias "Out32" (ByVal PortAddress As Integer, ByVal Value As Integer)
```

```
Private Sub Command1_Click() ' C1 butonu  
Out Val("&H378"), 0  
End Sub
```

```
Private Sub Command3_Click() ' C2 butonu  
Out Val("&H378"), 2  
End Sub
```

```
Private Sub Command4_Click() ' C3 butonu  
Out Val("&H378"), 4  
End Sub
```

```
Private Sub Command5_Click() ' C4 butonu  
Out Val("&H378"), 8  
End Sub
```



```
Private Sub Command6_Click()‘ C5 butonu  
Out Val("&H378"), 16  
End Sub
```

```
Private Sub Command7_Click()‘ C6 butonu  
Out Val("&H378"), 32  
End Sub
```

Mikro İşlemci Kodu

```
@ DEVICE PIC16F877  
@ DEVICE PIC16F877, WDT_OFF  
@ DEVICE PIC16F877, PWRT_ON  
@ DEVICE PIC16F877, PROTECT_OFF  
@ DEVICE PIC16F877, XT_OSC  
  
DEFINE OSC 4  
  
TRISC=0  
PORTC=0  
TRISD=1  
PORTD=0  
  
LOOP:  
IF PORTD.1=1 THEN ‘C1’e bağlı LED yanar  
PORTC=%00000010  
ENDIF  
IF PORTD.2=1 THEN ‘C2’e bağlı LED yanar  
PORTC=%00000100  
ENDIF  
IF PORTD.3=1 THEN ‘C3’e bağlı LED yanar  
PORTC=%00001000  
ENDIF  
IF PORTD.4=1 THEN ‘C4’e bağlı LED yanar  
PORTC=%00010000  
ENDIF  
IF PORTD.5=1 THEN ‘C5’e bağlı LED yanar  
PORTC=%00100000  
ENDIF  
IF PORTD.6=1 THEN ‘C6’ya bağlı LED yanar  
PORTC=%01000000  
ENDIF  
IF PORTD.7=1 THEN ‘C7’ye bağlı LED yanar  
PORTC=%10000000  
ENDIF  
GOTO LOOP
```

9. Güç Kaynağı

Robotik ve otomasyon sistemlerinde mikro işlemcinin besleme akımı ile sistemde bulunan motorlar gibi diğer elemanların akımları farklı bataryalardan karşılanmalıdır. Eğer yalnızca bir batarya kullanılır ise, motorların yüksek akım çekmesinden dolayı mikro işlemciye giden akım çok kısa süreli olsa bile kesilmekte ve bu durum mikro işlemcinin sıfırlanmasına ve sistemin istenmeden kapanıp tekrar açılmasına sebep olmaktadır.

9.1 Bataryalar

Bataryalar kimyasal enerjiyi, elektrik enerjisine ve elektrik enerjisini tekrar kimyasal enerjiye çevirebilen bir doğru akım kaynağıdır. Yani bataryalar tekrar şarj edilip kullanılabilirler. Batarya seçilirken kullanılacak sistemin voltaj değeri, devre için gerekli akım değeri ve sistemin ne kadar süre çalışmasının istendiği göz önünde bulundurulmalıdır. Bataryalar şarj edilirken doğru akım kaynağının pozitif kutbu (+ kutup) bataryanın pozitif kutbuna (+ kutba) ve akım kaynağının negatif kutbu (– kutup) ise bataryanın negatif kutbuna (– kutba) denk gelecek şekilde bağlanmalıdır. Şarj değeri şarj edilecek bataryanın voltaj değerinin %10 fazlası olacak şekilde ayarlanmalıdır.

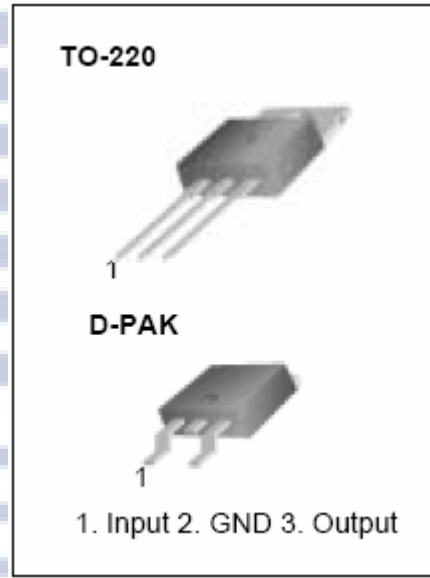
Birden fazla batarya aynı anda şarj edilmek isteniyorsa bataryalar paralel olarak bağlanmalıdır. Bu şekilde bir bağlantı yapılacaksa, bataryaların aynı voltaj değerinde olmalarına dikkat edilmelidir. Eğer şarj değeri %10 nun daha üzerinde seçilirse bataryaların patlama tehlikesi vardır. Bir diğer önemli nokta ise bataryaların tamamen boşalmadan şarj edilmesi ihtiyacıdır. Çok fazla enerji kaybeden ve çok uzun süre şarj edilmeyen bataryalar bozulurlar ve bir daha şarj olmazlar.

9.2 KA7805 Voltaj Regulatoru

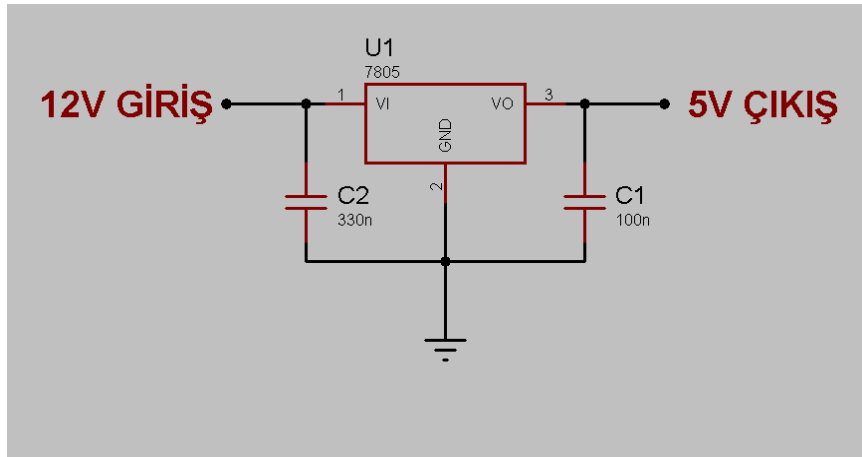
Mikro işlemciler ve çoğu devre elemanı 5V ile çalıştığı için bataryalardan alınan 12V değerinin 5V’ta düşürülmesi gerekmektedir. Bunun için voltaj düzenleyicileri kullanılır. Bu kitaptaki uygulamalarda üzerinden 1A kadar akım çeken KA7805 pozitif voltaj düzenleyicisini kullanılmıştır. Çok az çevresel elemana ihtiyaç duyan ve regüleli 5V çıkış verebilen bu devre elemanı üç bacağı sahiptir ve kullanması oldukça basittir. Şekil 9.1’de bu elemanın resmi ve bacak numaraları görülmektedir, 1. bacak voltaj girişi, 2. bacak toprak ve 3. bacak ise 5V çıkışıdır. Şekil 9.2’de ise 2 adet kondansatörle devrede

nasıl kullanılacağı gösterilmiştir. Burada kondansatörler voltajdaki dalgalanmayı azaltıp dengeli bir doğru akım elde etmek için kullanılmıştır. Şekil 9.2’de de görülebileceği gibi KA7805 entegresinin giriş bacağı ile toprak bacağı arasına 330nf’lık ve çıkış bacağı ile toprak bacağı arasına da 100nf’lık bir kondansatör bağlanarak bu işlem gerçekleştirilir.

KA7805 benzeri voltaj regülatörleri olarak LM78M05, LM340, KA7809, KA7824 ve MC7800 örnek olarak verilebilir. Bu voltaj regülatörlerinin seçimi uygulamanın gerek duyduğu voltaj ve akım değerine göre yapılır. Eğer yeterince yüksek değerde akım sağlayabilen regülatör seçilmemiş ise devre yüksek akım çekince voltaj regülatörünün yanmasına sebep olur. Bu durumdan kaçınmak için ya daha yüksek akım sağlayabilen bir eleman seçilmelidir, yada iki veya daha fazla eleman devreye paralel olarak bağlanarak daha yüksek akım üretilir.



Şekil 9.1 KA7805 voltaj regülatörü paketleri ve bacak fonksiyonları.

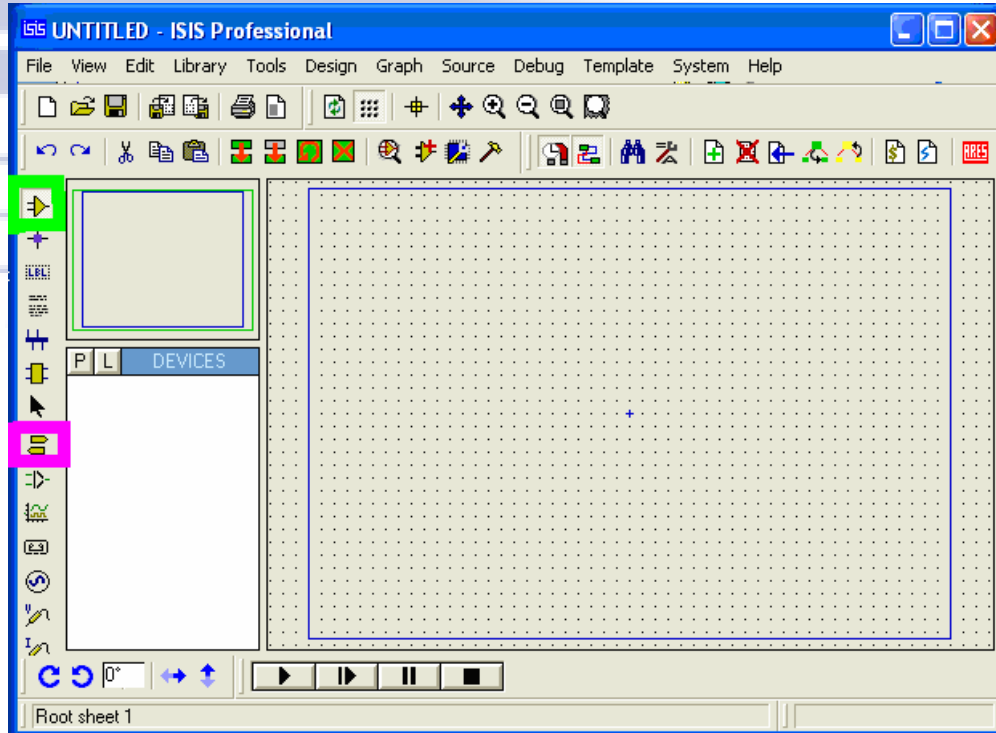


Şekil 9.2 KA7805 voltaj regülatörünün kondansatörler kullanarak devreye bağlanış şematığı.

10. Proteus Programı

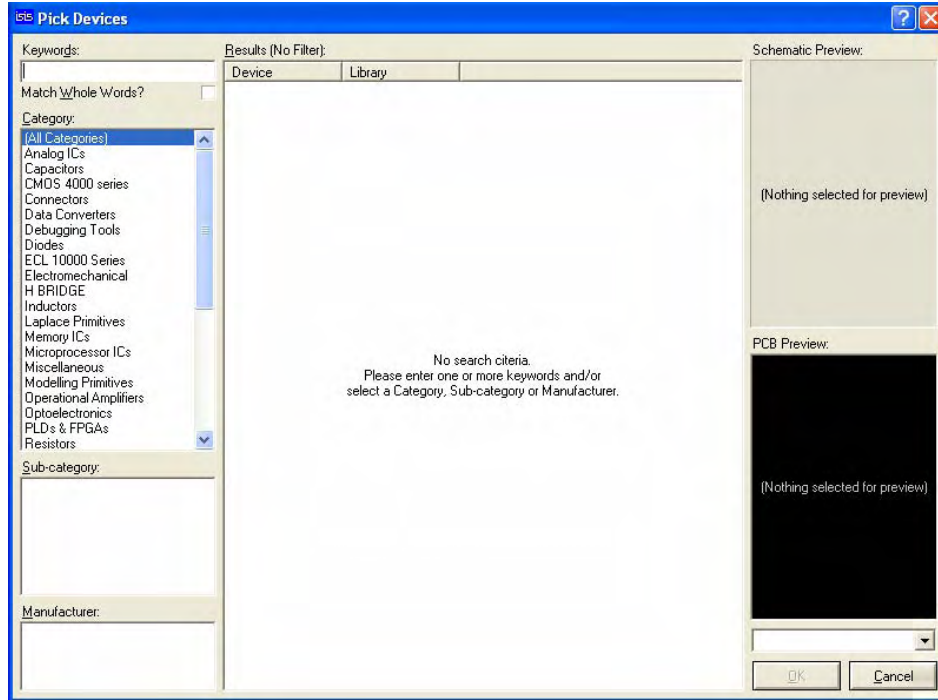
Elektronik devre çizimi ve simülasyon programı olan Proteus programı analog ve dijital devrelerin çizimini ve simülasyonunu yapabilmektedir. Çok geniş bir devre elemanları kütüphanesine sahip olması ve mikro işlemci simülasyonlarını da yapabilmesi prototip devre geliştirilmesi sırasında, yazılan kodların devreyi kurmadan bilgisayar ortamında denenmesine imkan sağlaması nedeniyle tasarım aşamasını kolaylaştırmakta ve olası hataların devre kurulmadan önce görülmesini sağlaması prototip devre geliştirme sürecini kolaylaştırmaktadır. Genellikle Proteus programı mikro işlemcilere yazılan kodların doğru olup olmadıklarını denemek amacıyla kullanılır. Programın çok kapsamlı olması nedeniyle burada sadece mikro işlemcilerin simülasyonu konusuna açıklanmıştır.

Şekil 10.1’de Proteus programı çalıştırıldığında kullanıcının karşısına çıkan ilk ekran görülmektedir. Simülasyona başlamak için öncelikle mikro işlemci seçilmelidir. Bunun için programın devre elemanlarının bulunduğu kütüphanesi açılmalıdır. Kütüphaneyi açabilmek için önce Şekil 10.1’de yeşil ile kare içine alınmış olan eleman (component) butonu tıklanmalı daha sonra ise ekranın sağ tarafında bulunan pencere üzerindeki Eleman Seç (Pick Device-P) butonuna basılmalıdır.



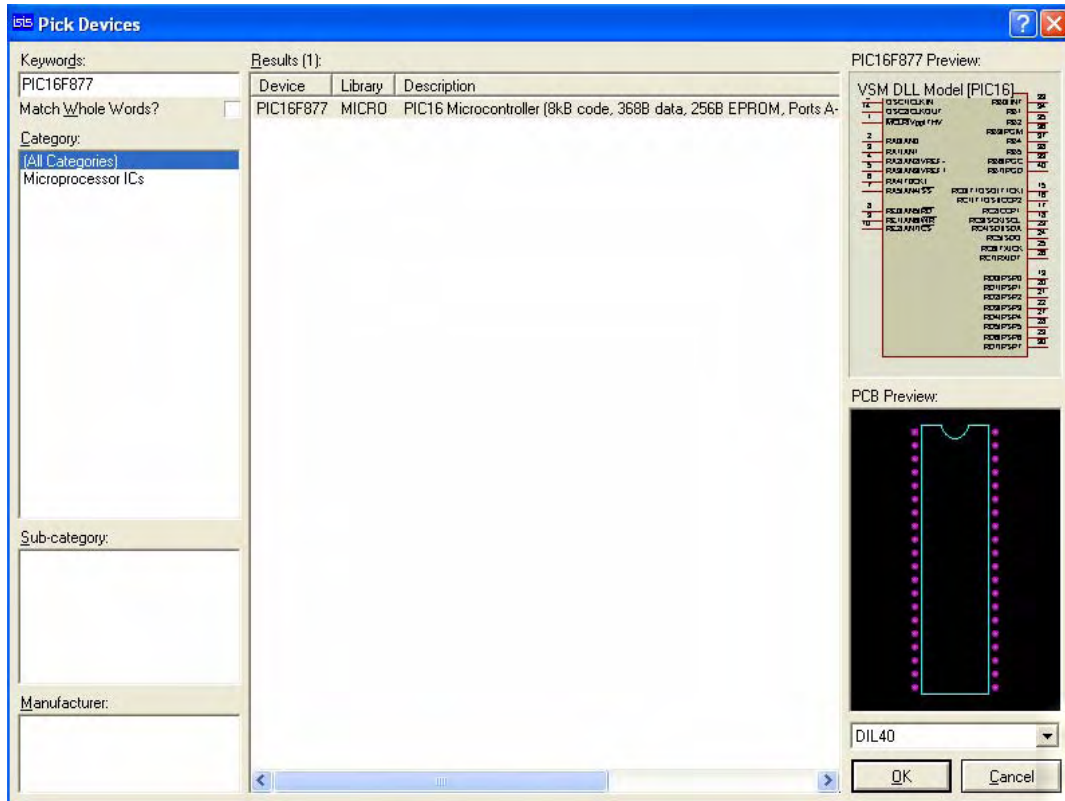
Şekil 10.1 Proteus programının ana ekranı.

Bu işlemler yapıldıktan sonra Şekil 10.2’de bulunan pencere ile karşılaşılır.



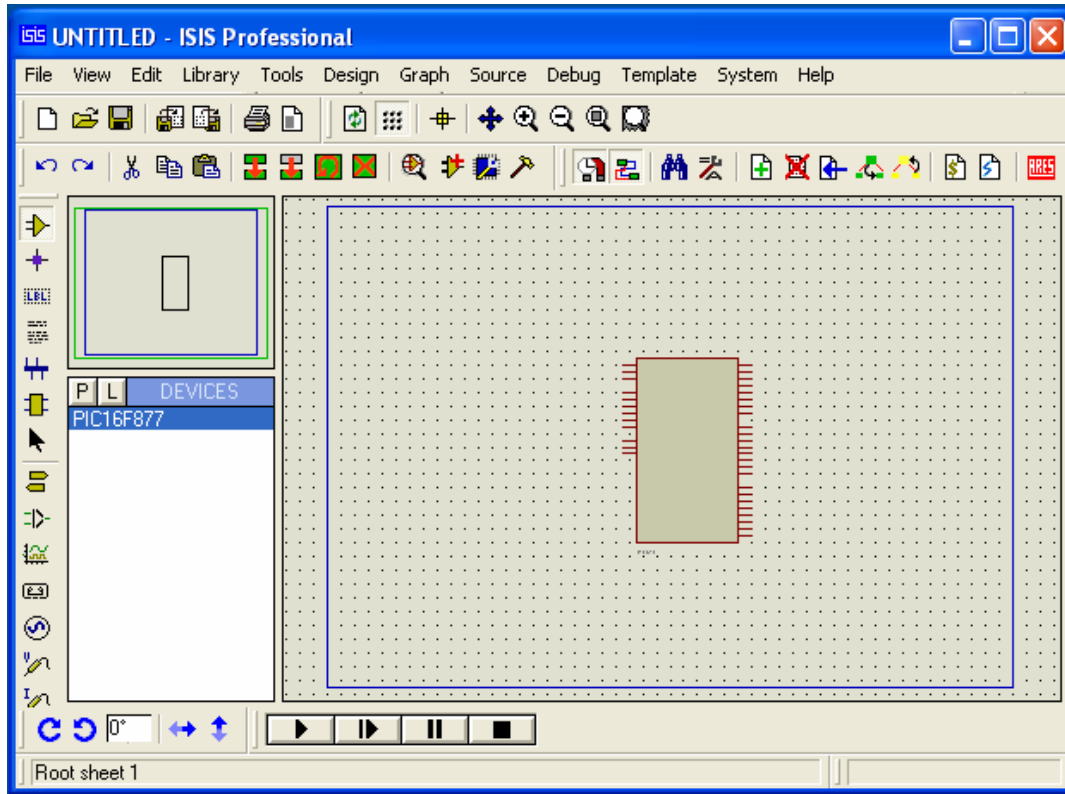
Şekil 10.2 Elektronik eleman seçme ekranı.

Bu pencereden anahtar kelimeler (keywords) kısmına kullanmak istenilen elemanın adı yazarak veya katagori (catagory) kısmından arayıp bularak istenilen devre elemanı seçilir. Burada PIC16F877 mikro işlemcisini kullanılmıştır. Bu yüzden anahtar kelimeler kısmına PIC16F877 yazılır. Bu işlemi yaptıktan sonra Şekil 10.3’te görülen pencere açılır.



Şekil 10.3 Seçilen elektronik elemanın detaylarının görülmesi.

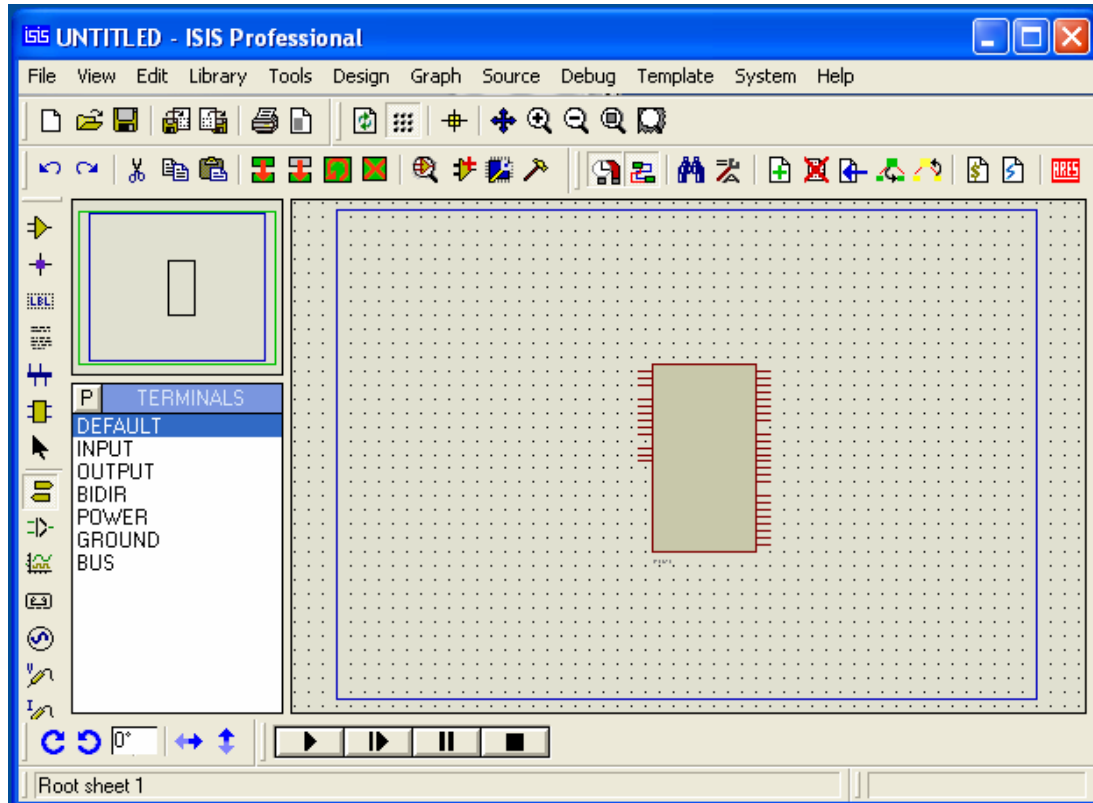
Bu pencerede görüleceği üzere seçilen devre elemanın adı, bazı teknik özellikleri ve kılıf şekli görülmektedir. Elemanı seçtikten sonra Tamam (OK) butonuna basarak Şekil 10.4'te görülen Proteus programının asıl tasarım ortamına dönülür ve seçilen eleman kullanıma hazır olur.



Şekil 10.4 Tasarım ortamı.

Şekil 10.4'te görülebileceği gibi ELEMENLAR (DEVICES) yazan pencerenin altında seçilen devre elemanı bulunmaktadır. Buradan önce bu elemana daha sonrada nokta nokta görülen tasarım alanına farenin sol butonuna tıklayarak, seçilen eleman tasarım alanına alınır. Diğer devre elemanlarını seçmek için de aynı işlem tekrarlanır. Seçilen her eleman ELEMENLAR penceresinin altında gösterilir ve kullanıcı bu elemanlardan istediği miktarda seçip kullanabilir. Elemanların bir birine bağlanması için önce bir birine bağlanmak istenen pinlerden birine farenin sol tuşu ile tıklanır daha sonra diğer elemana gidilerek bağlantı yapılmak istenen bacağı sol tıklanır program otomatik olarak bu iki bacağı birbirine iletken tel ile bağlar.

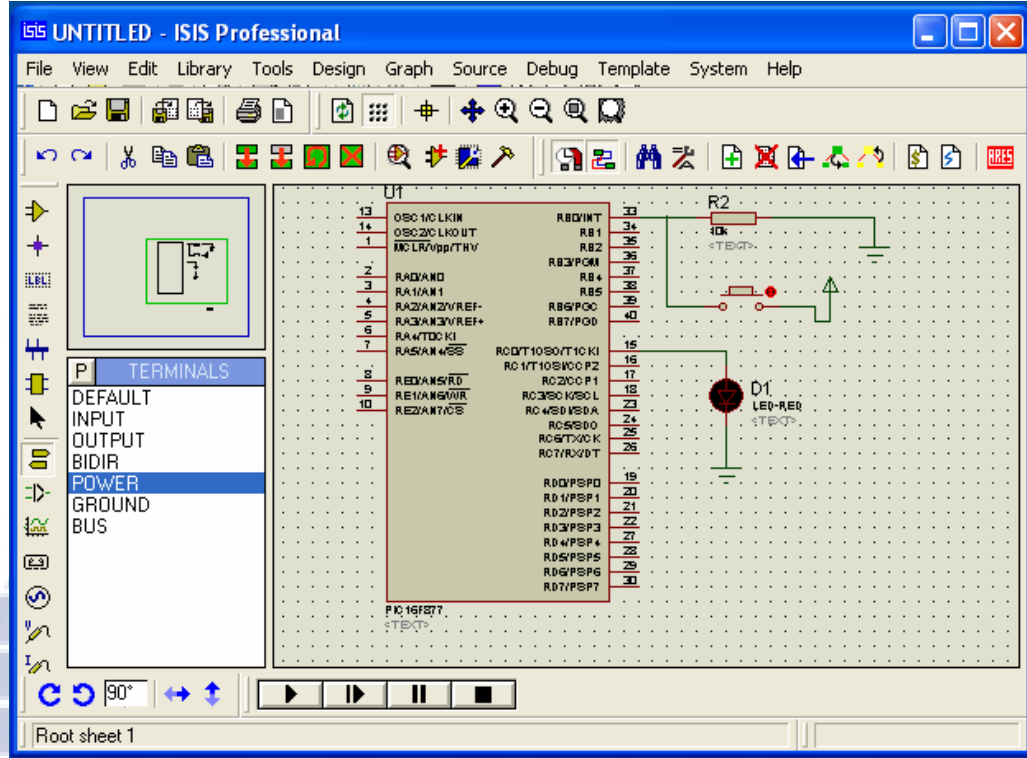
Tasarım aşamasında seçilen devre elemanlarına voltaj ve toprak bağlantısı yapılmak istenirse, Şekil 10.1'de pembe renk ile kare içine alınmış olan "inter-sheet terminal" butonuna tıklanır ve Şekil 10.5'te görülen pencere açılır.



Şekil 10.5 Elektronik elemanların bağlantılarının yapılması.

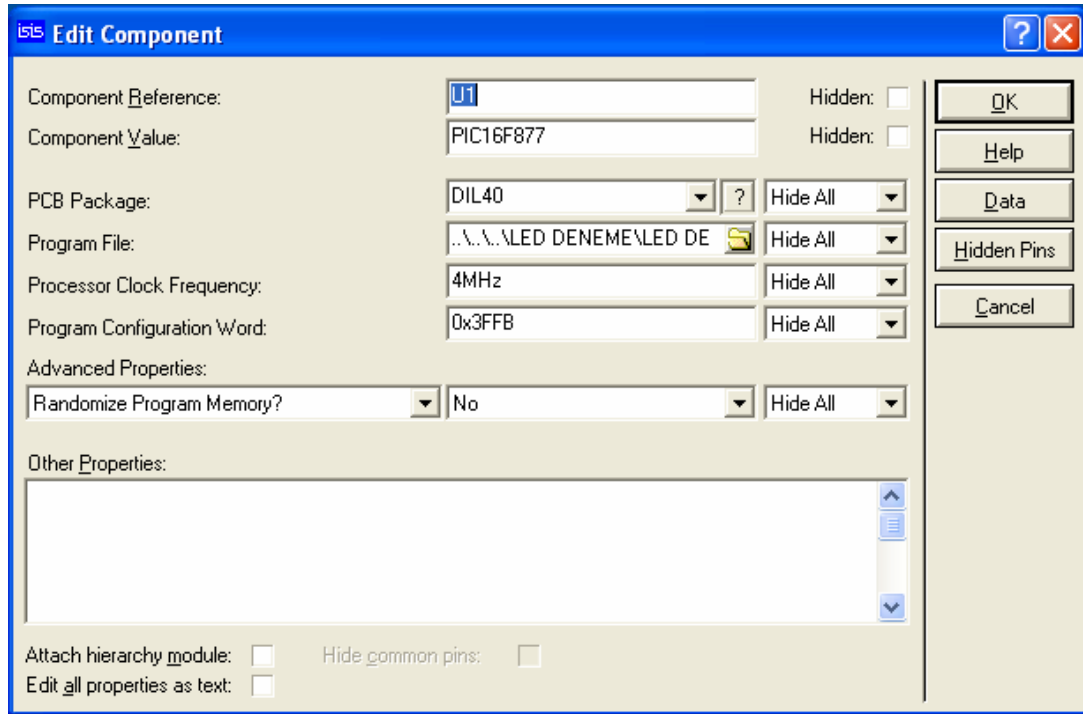
Şekil 10.5’de görüleceği gibi DEVICES penceresi TERMINALS penceresi olarak değişmiştir. Bu pencerede görülen POWER yazısına tıklanıp daha sonra tasarım ekranına tıklandığında, ekrana tıklanılan yere + voltaj kaynağı konulur. Aynı şekilde GROUND yazısına tıklanıp tekrar tasarım ekranına tıklandığında ekrana güç kaynağının negatif ucu yerleştirilir ve bu uçlar daha önce anlatılan birleştirilme yöntemiyle birleştirilebilir. Burada önemli bir noktaya dikkat etmek gerekmektedir, bu nokta Proteus programının kütüphanesi içinde bulunan elektronik devre elemanlarının gerilimlerini kendisinin vermesidir. Yani biz mikro işlemciyi tasarım ekranına aldığımızda işlemciye voltaj bağlantısı yapmamıza gerek yoktur. Bu işlem program tarafından otomatik olarak yapılır. Yine aynı şekilde mikro işlemci saat (clock) sinyali için bağlantı yapmaya gerek yoktur ama kullanıcı isterse bu saat bağlantısını da yapabilir. İşlemci simülasyonda her iki şekilde de çalışmaktadır. Güç ve toprak bağlantıları tasarım alanına konacak motor ve/veya led diyot gibi diğer devre elemanları için gerekmektedir. Devre elemanları yerleştirilip bağlantıları yapıldıktan sonra artık simülasyon işlemine geçilebilir. Mikro işlemci simülasyonu için önceden yazılmış olan mikro işlemci programının herhangi bir derleyicide .hex formatına çevrilip bir dosya içine kaydedilmesi gerekmektedir. Burada örnek olarak PIC16F877 mikro işlemcinin

B0 bacağına bir buton ve C0 bacağına bir adet led bağlanmıştır. Butona basıldığında ledi yakan bir programı mikro işlemciye yükleyip simülasyonunu yapılarak yazılan programın denenmesi gösterilmiştir. Bu işlem için öncelikle Şekil 10.6'da görülen devre kurulmuştur.



Şekil 10.6 İstenilen devrenin tamamlanmış hali.

Devre kurulduktan sonra mikro işlemciye daha önceden derlenen LEDDENEME adlı .hex dosyasının yüklenmesi gerekmektedir. Bu işlem için fare ile işlemcinin üzerine gelinip önce farenin sol ve daha sonra da sağ tuşuna tıklanır ve Şekil 10.7'de görülen pencere açılır.



Şekil 10.7 Mikro işlemcinin özelliklerinin belirlenmesi ve dosyanın yüklenmesi.

Şekil 10.7’de görülen pencerede Program Dosyası (Program File) yazan kısımdan .hex dosyasının kayıtlı olduğu klasör açılıp yüklenmek istenen .hex uzantılı dosya açılır. Daha sonra işlemcinin çalışma frekansı İşlemci Saat Frekansı (Processeor Clock Frequency) yazan kısma yazılır. Burada işlemci 4 MHz hızda çalıştırılmaktadır. Daha sonra Tamam (OK) butonuna basarak yükleme işlemi tamamlanır ve ana tasarım alanına geri dönlür.

Tasarım alanına dönlüldüğünde simülasyonu başlatmak ve çalıştırmak için, Şekil 10.8’de kırmızı çerçeve içine alınmış butonlar kullanılır. Bu butonlardan soldan sağa doğru en baştaki simülasyonu başlatmak için kullanılır, ikinci buton simülasyonu adım adım çalıştırır, üçüncü buton simülasyona ara verir ve son buton ise simülasyonu durdurur.

Şekil 10.8’de görüleceği üzere bu örnekte kurulan devre 1 adet butondan, 1 adet led diyottan, 1 adet 10K’lık dirençten ve bir adet mikro işlemciden oluşmaktadır. Örnek 10.1’de yazılan program işlemcinin C0 pinine bağlı butona basıldığında B0 pinine bağlı olan ledi yakmak için yazılmıştır. Bu programın çalışma şeklini görmek için, program derlenip .hex uzantılı dosya işlemciye yüklenir. Daha sonra simülasyonu başlatma

butonuna basılır ve son olarak tasarım ekranına konulan butona basılır ve Şekil 10.8’de görüldüğü üzere ledin yandığını ekrandan takip edilebilir.

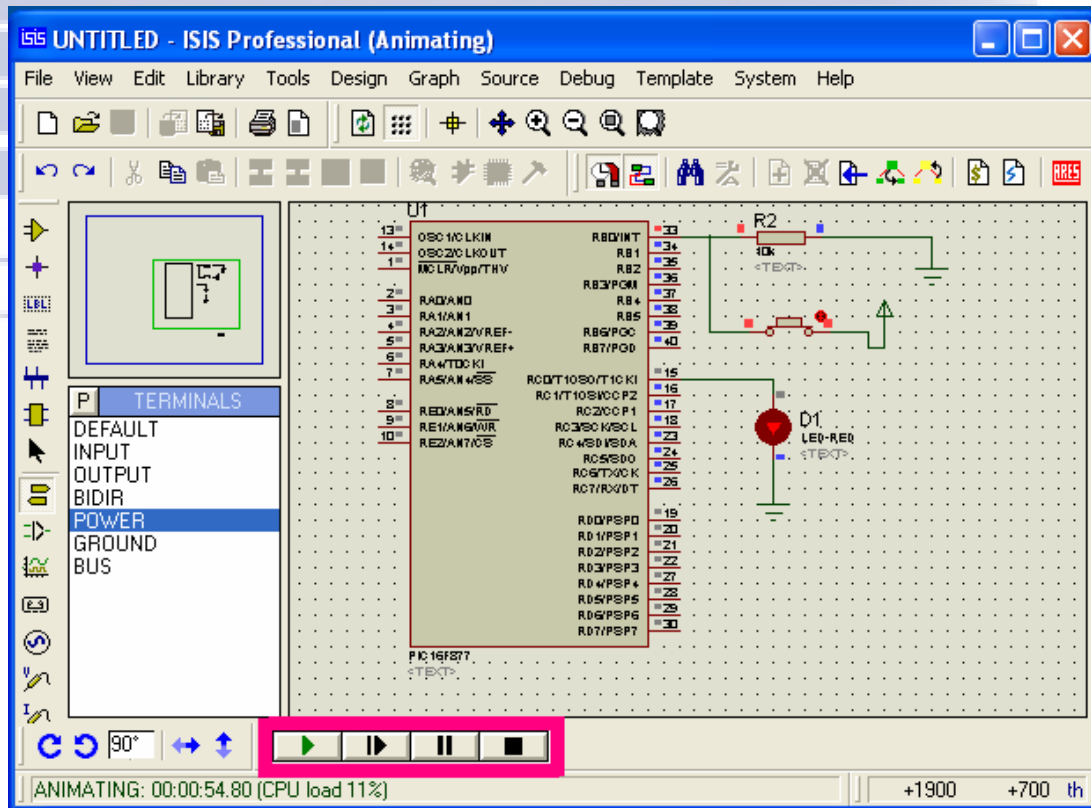
Örnek 10.1:

```
@ DEVICE PIC16F877
@ DEVICE PIC16F877, WDT_OFF
@ DEVICE PIC16F877, PWRT_ON
@ DEVICE PIC16F877, PROTECT_OFF
@ DEVICE PIC16F877, XT_OSC
```

```
DEFINE OSC 4
```

```
LOOP:
TRISB=1
TRISC=0
PORTB=0
PORTC=0
```

```
WHILE PORTB.0=1
PORTC.0=1
WEND
GOTO LOOP
```



Şekil 10.8 Mikro işlemci programının çalıştırılması.

11. Elektronik Kartların İmalatı ve Dikkat Edilecek Noktalar

Tasarımı ve simülasyonu yapılan kartlar iki yöntemle üretilebilir. Bunlardan birincisi baskı devre hazırlamak ve ikincisi ise delikli plakette üzerine devre kurmaktır. Aşağıda bu iki farklı yöntemin avantaj ve dezavantajları açıklanmıştır.

11.1 Baskı Devre Hazırlama

Bu yöntemde bilgisayar ortamında çizilen devrenin baskı devresi ARES veya AutoCAD gibi bir çizim programı kullanılarak hazırlanır. Daha sonra bu baskı devrenin lazer yazıcıdan çıktısı alınır. Alınan bu çıktı kurulacak devre boyutunda kesilmiş bir tarafı tamamen bakırlı olan pertiklas üzerine bakırlı kısım iyice temizlendikten sonra yerleştirilir. Daha sonra yazıcıdan alınan çıktı üzerindeki yollar plakette üzerine tamamen geçene kadar ütü ile ütülür. Ütüleme işlemi sırasında kağıt pertiklas üzerine yapışabilir. Bu yapışmanın kağıdı sökerken iletken yollara zarar vermemesi için kağıt sökülmeden önce iyice ıslatılmalıdır.

Daha sonra pertiklas üzerine çizilen yolların dışında kalan bakırlı kısmın temizlenmesi işlemine geçilir. Bu işlem için bir ölçek perhidrol içine 3 ölçek tuz ruhu konularak kimyasal bir karışım hazırlanır. Bu hazırlanan karışım zehirli ve yakıcı olduğu için bu işlemin açık havada ve çok dikkatli bir şekilde yapılması gerekir. Daha sonra üzerinde yollar çizili olan plakette bu hazırlanan asit karışımı içine atılır ve yollar haricindeki bütün bakırlı kısım temizlenene kadar beklenir ve plakette bu karışımın içinden çıkarılır. Eğer plakette asit içinde çok bekletilirse erimesi istenmeyen kısımda erimeye başlayacağı için asit içinde tutma süresine dikkat edilmelidir.

Plakette perhidrol ve tuz ruhu karışımından çıkarıldıktan sonra su ile iyice yıkanır ve daha sonra kalan mürekkep artıklarını temizlemek için selülozik tiner ile temizlenir. Bu işlemler bittikten sonra ince uçlu (1mm gibi) ufak bir matkap yardımıyla devre elemanlarının yerleştirileceği delikler delinir. Elemanlar yerleştirildikten sonra lehimleri yapılarak devre kartı tamamlanır.

Bu yöntemle hazırlanan kartlarda elemanları birbirine bağlamak için kablolar kullanılmadığı için kısa devre ihtimali ortadan kalkar ve elektronik gürültü büyük ölçüde önlenmiş olur. Fakat bu yöntem son ürün olan kartları üretmek için

kullanılmalıdır, çünkü bu yöntemle hazırlanan kartlar üzerinde değişiklik yapmak çok zordur. Kart üzerinde değişiklik yapmak gerektiğinde eski kart atılıp bu işlemler tekrarlanarak yeni bir kart yapılması gerekmektedir.

11.2 Delikli Levha Üzerine Devre Hazırlama

Bu yöntemde üzerinde delikler ve her deliğin altında bakırlı bir kısım bulunan pertiklaslar devre kurmak için kullanılır. Devre elemanları pertiklasın üzerinden deliklere yerleştirilir. Daha sonra alt kısımdan lehim ile pertiklasa tutturulur. Devre elemanlarının birbirleri ile bağlantısı ise pertiklasın alt veya üst kısmından çekilen kablolar ile yapılır. Bu bağlantılar yapılırken dikkat edilmesi gereken nokta kısa devre ihtimalinin yüksek olmasıdır. Bu tür devreler hazırlanırken üzeri lehim kaplanan tek damarlı kablolarla pertiklasın alt tarafından yollar çekilmesi iyi bir yöntemdir. Bu yöntemle çekilen yolların hem kısa devre ihtimalini ortadan kaldırır, hem de devrelerin uzun ömürlü olmalarını sağlar. Ayrıca tellerin lehim ile kaplanması, lehimleme işlemini kolaylaştırır ve devrenin daha hızlı ve temiz kurulmasını sağlar.

Her iki yöntemle hazırlanan kartlarda lehimleme büyük önem taşır. Her ne kadar lehim yapmak tecrübe ile geliştirilebilirse de bazı temel hususlara dikkat etmek sağlıklı lehim yapmayı ve devrelerin uzun ömürlü olmasını sağlar. Lehim yapılırken dikkat edilecek noktalar şöyle sıralanabilir.

1. Lehim yaparken mutlaka lehim pastası kullanılmalıdır. Lehim pastası hem lehim yapılacak malzemeyi temizler hem de lehimin kolayca malzeme üzerine yayılmasını sağlar.
2. Lehim yapılırken önce lehim yapılacak malzeme ısıtılmalıdır. Malzeme ısıtıldıktan sonra lehim teli ısıtılan malzemeye değdirilerek lehimleme işlemi tamamlanır.
3. Lehim yaparken lehimlenecek devre elemanın çok fazla ısıtılmasından kaçınılmalıdır. Elektronik devre elemanları ısıya karşı duyarlıdır ve dolayısıyla lehimleme esnasında çok fazla ısıtılırlarsa bozulabilirler.

4. Lehim yapılırken çok fazla lehim teli kullanılmamalıdır. Lehimlenen malzemeye çok fazla lehim yapılırsa fazla lehim zamanla çürüme yapabilir ve kısa devre ihtimalini artırır.

11.3 Elektronik Kartların Tasarımı Sırasında Dikkat Edilecek Hususlar

Temel olarak elektronik elemanların kartın üzerine yerleştirilmesi sırasında aşağıdaki genel prensiplere dikkat edilmesi ile hem imalatı hem de kullanımı daha kolay kartlar elde edilebilir.

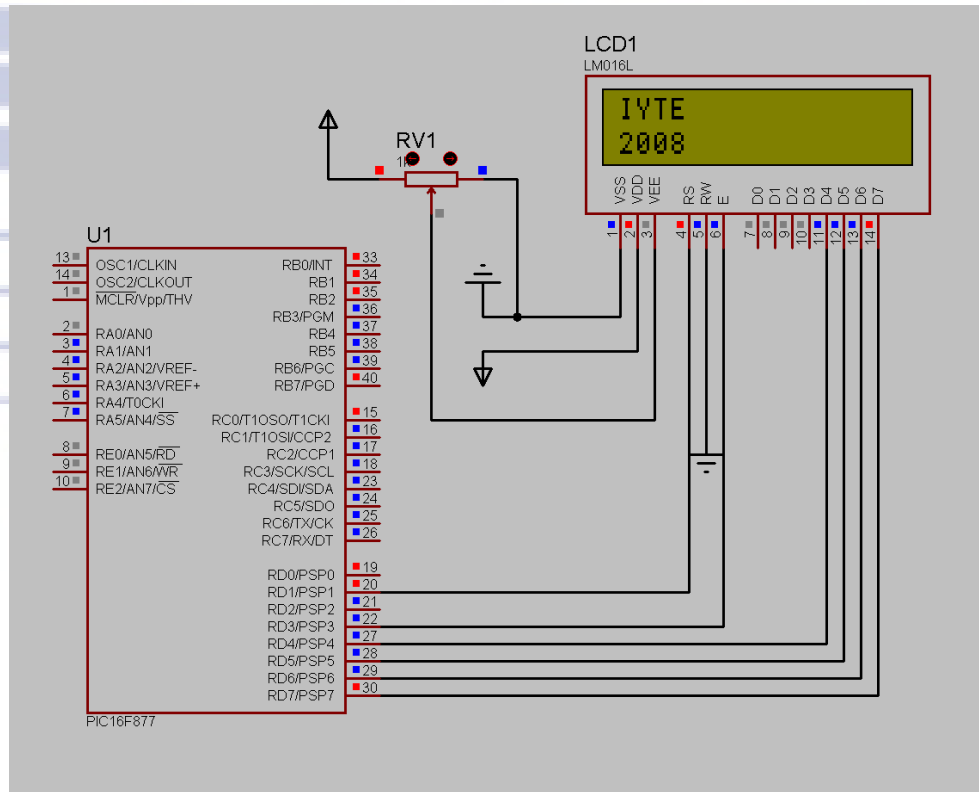
1. Kartların tasarımı sırasında eğer varsa soketler ve jaklar kartın kenarına yerleştirilmelidir. Böylelikle kullanıcıların kartlar üzerinden kabloları sökerken devreye zarar verme ihtimalleri azaltılır.
2. Mikro işlemci gibi devre elemanları kullanılacak ise, bunlar mutlaka soketler kullanılarak takılmalıdır. Bu şekilde bu elemanlar programlanmak için kart üzerinden sıklıkla sökülebilirler. Ayrıca diğer devre elemanları da bu malzemelerin kolayca sökülüp takılmasına izin verecek şekilde yerleştirilmelidirler.
3. Voltaj regülatörleri gibi arıza yapması muhtemel parçalar kolayca sökülebilecek şekilde kenarlara yerleştirilmelidir.
4. Kart daha sonra robot veya herhangi bir makine üzerine yerleştirilecek ise kartın kenarlarında montaj için gerekli boşluklar bırakılmalıdır.
5. Üzerlerine soğutucu takılacak elemanlar önceden belirlenmeli ve kart üzerinde bu elemanların yerleri soğutucunun büyüklüğü de hesaba katılarak ayrılmalıdır.

12. Hata Bulma ve Kalibrasyon Cihazı

Yapılan robotik uygulamalarında sistemde neler olduğunu izleyebilmek veya programın çalışması sırasında bazı verileri kullanıcıya görsel olarak iletmek için LCD'ler kullanılır.

12.1 Paralel LCD

Piyasada 2x8, 2x16, 4x16 gibi birçok LCD modeli bulunmaktadır, burada ilk rakam LCD'nin satır sayısını, ikinci rakam ise LCD'nin her satırda kaç karakter yazabileceğini gösterir. Yani 2x8 iki satırlı ve her satıra 8 karakter yazabilen bir LCD'yi tanımlar. Satır ve karakter sayıları farklı olsa da bütün bu LCD'ler mikro işlemciye aynı şekilde bağlanırlar. Örnek olarak 2x16 bir LCD'nin mikro işlemciye nasıl bağlanacağı Şekil 12.1 de görülmektedir. LCD'nin hangi bacağına mikro işlemcinin hangi pinine bağlanacağı ve LCD'nin kaç satır yazabildiği program içinde belirtilmelidir. Örnek 12.1.1'de D portuna bağlı LCD'nin birinci satırına IYTE ikinci satırına 2008 yazısını yazacak program görülebilir. Şekil 12.1'de LCD'nin mikro işlemcinin hangi bacaklarına bağlandığı gösterilmiştir.



Şekil 12.1 LCD ekranının mikro işlemciye bağlanması.

Örnek 12.1.1

PICBasicPro dilinde yazılmış bu programda LCD kullanımı için gerekli tanımlamalar görülebilir. Burada önemli olan nokta LCD'nin bağlanacağı portun çıkış olarak ayarlanmasıdır. LCD nin RW bacağı direkt olarak toprağa bağlanabilir veya işlemcinin belirtilen pinine bağlanabilir. Programda D2 portuna bağlanacağı belirtildiği halde doğrudan toprağa bağlanmıştır. Program iki farklı şekilde de çalışır.

```

DEFINE LCD_DREG    PORTD
DEFINE LCD_DBIT    4
DEFINE LCD_EREG    PORTD
DEFINE LCD_EBIT    3
DEFINE LCD_RWREG    PORTD
DEFINE LCD_RWBIT    2
DEFINE LCD_RSREG    PORTD
DEFINE LCD_RSBIT    1
DEFINE LCD_BITS    4
DEFINE LCD_LINES    2

DEFINE OSC 4
LCDOUT $FE,1
PAUSE 200

BASLA:

LCDOUT $FE,1,"IYTE"  'LCD nin 1. satırında 1. karakterden başlayarak IYTE yazar
LCDOUT $FE, $c0,"2008" 'LCD nin 2. satırında 1. karakterden başlayarak 2008 yazar

PAUSE 500 'Yarım saniyelik bekleme yapar
GOTO BASLA

```

12.2 Kalibrasyon Cihazı

Robotik uygulamalarının gerek mekanik gerekse elektronik ve programlama içeren sistemler olmalarından dolayı prototip imalatı sırasında mekanik, elektronik ve program hatalarından dolayı sistem çalışmamaktadır. Bu gibi durumlarda sistemin aşamalı olarak çalıştırılması ve sistem değişkenlerinin ölçülüp beklenen değişkenler ile karşılaştırılarak hatanın bulunması gerekir.

Sistem çalıştırılırken de gerek robot üzerindeki duyar elemanların gönderdiği verilerin gerekse sistemin mekanik kısmının tam olarak modellenememesinden dolayı oluşan hataların ölçülüp sistemin bir bütün olarak kalibre edilmesi gerekir. Prototipleri imal edilen robotik uygulamalarında sistem değişkenlerinin değişimini izleyebilmek veya programın çalışması sırasında gerekli verileri kullanıcıya görsel olarak iletmek için ekranlar kullanılır.

Daha gelişmiş uygulamalarda (mikro işlemcilerin birbirleri ile haberleşmeleri kablolu kumandadan veri gönderimi ve veri alımı gibi) sadece ekran kullanımı yetersiz kalmaktadır. Çünkü pasif bir eleman olan ekranlar sadece verileri gösterebilmektedir. Bu veriler programın ara sonuçları ve sensörlerden okunan değerler olabilir. Voltmetre ve osiloskop gibi diğer ölçüm cihazları da farklı elektriksel büyüklükleri ölçmek için kullanılabilir. Fakat robotun kalibrasyonu sırasında robotun ayarlanacak kısmının tetiklenmesi ve ölçümün buna göre yapılması gerekiyor. Bu işlem için robotun sürücü kartları veya ölçüm kartları kullanılmamalıdır, çünkü sistemin hatalı çalışması halinde istenildiği anda durdurulmadığından dolayı elektronik kartların yanarak zarar görme riski vardır.

Bu makalede açıklanan hata bulma ve kalibrasyon kartı ile robot kısım kısım çalıştırılarak sistemdeki hatalar bulunabilir ve gerekli düzeltmeler ve kalibrasyon yapılabilir.

12.2.1 Kalibrasyon Cihazının Yapısı

Hata bulma ve kalibrasyon kartının tasarımında ve gerçekleştirilmesinde özellikle mikro işlemci kullanımına dikkat edilerek bu kartın imalatı kolay, ucuz ve programı rahatlıkla

değiştirilebilir olması sağlanmıştır. Kart içinde yer alan mikro işlemci programı özellikle Basic programlama dili tabanlı PICBasicPro ile yazıldığı için anlaşılması ve geliştirilmesi kolaydır.

Şekil 12.2’de ayrıntılı devre şeması verilen hata bulma ve kalibrasyon kartı üzerinde bir adet PIC16F877 yer almaktadır. Kullanıcıya ölçülen değerleri göstermek için 2x16 (2 satır 16 karakterli) bir ekran kullanılmıştır. Eğer kullanıcı tarafından istenirse bu işlemci aynı yapıdaki ve daha yüksek kapasitedeki PIC18F452 ile değiştirilebilir.

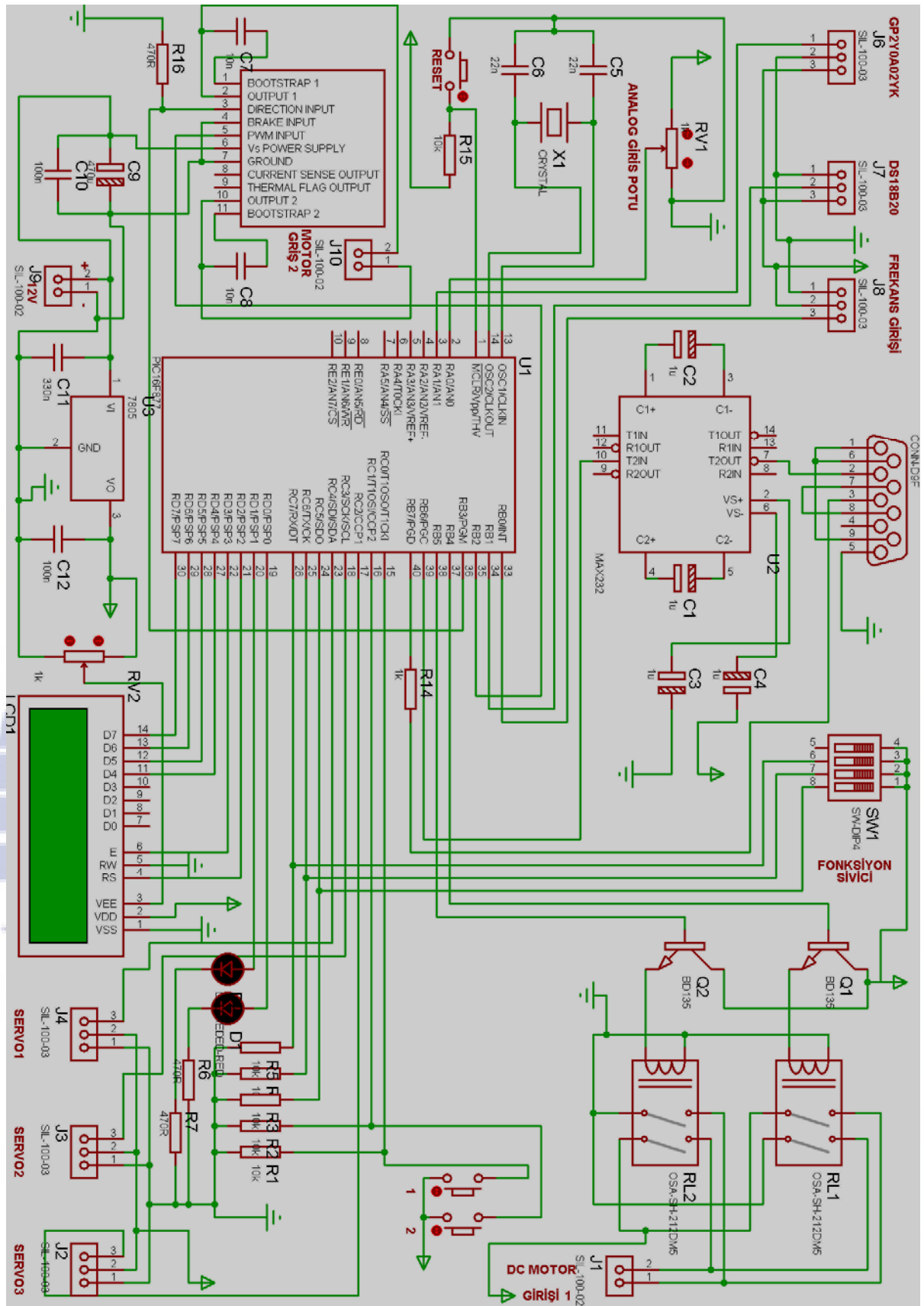
Mikro işlemcinin çalışması için gerek duyduğu kristal ve sıfırlama devreleri dışında mikro işlemci ve çoğu devre elemanı 5V ile çalıştığı için bataryalardan alınan 12V değerinin 5V düşürülmesi için 1A kadar akım çeken KA7805 pozitif voltaj düzenleyicisini kullanılmıştır.

2 röle yardımı ile devre açma kapama işlevi kazanılmıştır. Rölelerin yüksek akım çekip mikro işlemciye zarar vermesini engellemek için BD135 NPN transistörler kullanılmıştır. H köprüsü kullanılarak darbe genişlik modülasyonu (PWM) ile bir DC motorun istenilen hızda sürülebilmesi sağlanmıştır.

Bilgisayarın RS232 portunu kullanarak bilgisayar ve mikro işlemci arasında seri iletişim kurulabilmektedir. Voltaj seviyelerinin farklı olmasından dolayı MAX232 kullanılarak 0-5V arası işlem yapan mikro işlemci ile -12V ile +12V arası çalışan bilgisayar arasında veri aktarımı sağlanmıştır.

Mikro işlemcinin hem analog (analog-dijital dönüştürücü modülü sayesinde) hem de dijital sinyalleri işleyebilmesinden dolayı duyar elemanlar ve modülasyonlu çıkış sağlayabildiği için servo motorlar doğrudan bağlanabilmektedir.

Bu şekilde hata bulma ve kalibrasyon kartının yapısı detaylı bir şekilde açıklandığından bu kartın farklı uygulamalar için (paralel port üzerinden haberleşme veya yüksek amperli DC motoru modülasyonlu sürmek gibi) geliştirilmesi mümkündür.



Şekil 12.2. Hata bulma ve kalibrasyon cihazı devre şeması.

12.2.2 Kartın Fonksiyonları ve Çalışma Şekli

Hata bulma ve kalibrasyon kartı farklı robotik projelerinde doğan ihtiyaçları gidermek için tasarlanıp geliştirilmiştir. Temel olarak kart bir ölçü aleti gibi çalışırken; diğer ölçü aletlerinden üstün olarak aynı zamanda robotik sisteme farklı şekilde sinyallerde gönderebildiği için robotun farklı kısımlarını harekete geçirip ölçümler yapabilmektedir. Aşağıda detaylı olarak tasarlanıp imal edilen hata bulma ve kalibrasyon kartının işlevleri açıklanmıştır.

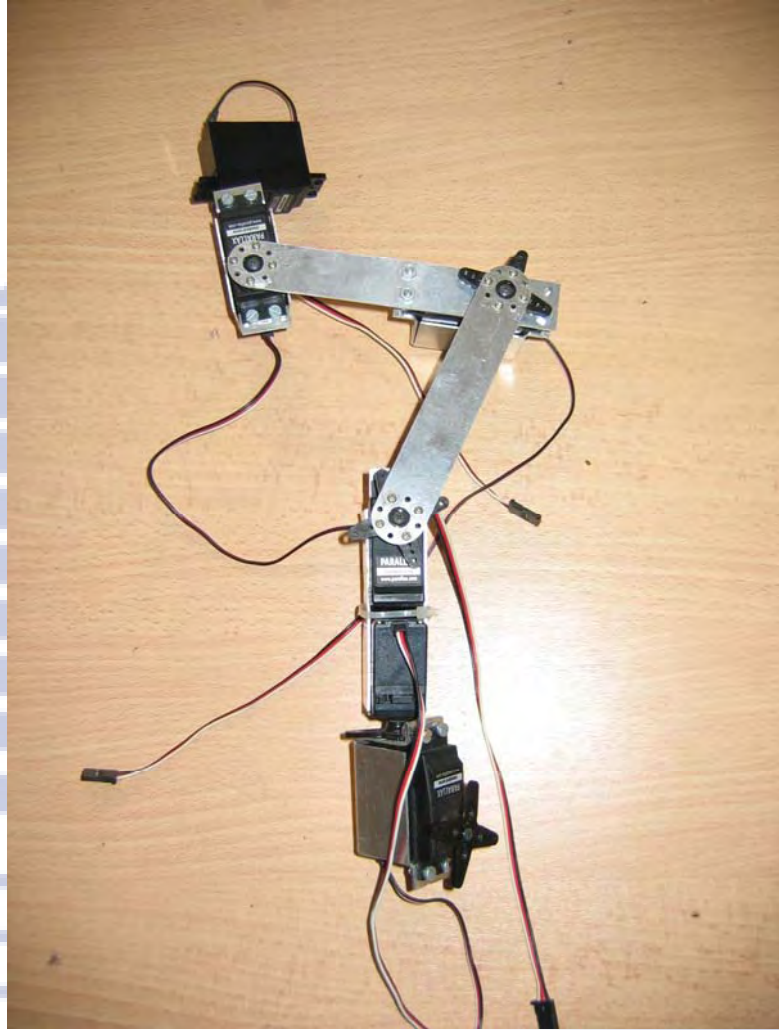
1. Frekans Okuma: Bu kart ile B0 pinine gelen frekansların genişlikleri okunabilir. Radyo kontrollü sistemler ile robotik sistemin uzaktan kablosuz kullanılması mümkündür. Radyo kontrolün herhangi bir verici kanalından sinyal gönderilirken alıcı tarafından servo motora iletmeye çalışılan sinyal kalibrasyon aletine bağlanırsa gelen sinyalin frekansı ölçülebilir. Verici kanalın 3 farklı konumuna göre (kolun en üste, ortada ve en altta olması) gönderilen farklı sinyaller alıcı tarafında bir mikro işlemci ile çözülerek robota farklı işlevler yaptırılabilir.

2. Dijital Sensör Girişi: B1 pinine bağlı DS18B20 sıcaklık sensörü ile 0,1 C° hassasiyetle sıcaklık ölçümleri yapılabilir. Ölçülen değer ekrana yazılarak kullanıcıya gösterilir.

3. Analog Sensör Girişi: A1 pinine bağlı Sharp sensör ile mesafe ölçümü yapılabilir. Bu tip bir duyar elemanın yapısı gereği mesafe ile sensör çıkış voltajı doğrusal değildir. Duyar elemanın önündeki engeli kaçırmaması için örnekleme işleminin yeterince hızlı olması gerekir.

4. Potansiyometre ile Servo Motor Kontrol: A0 pinine bağlı potansiyometre ile analog giriş yapılabilir ve bu analog değişime göre servo motor kontrol edilebilir. Özellikle eklemlerinde servo motorların kullanıldığı robotik uygulamalarında eklem montajı yapıldıktan sonra istenilen konuma eş düşen servo motor frekans değerinin bulunması için potansiyometre ile farklı frekanslarda sinyal gönderilir ve eklem istenilen açıda konumlandırılır. Şekil 12.3'de gösterilen bacak modüllerinin 4 tanesinin birleştirilmesi ile kedi

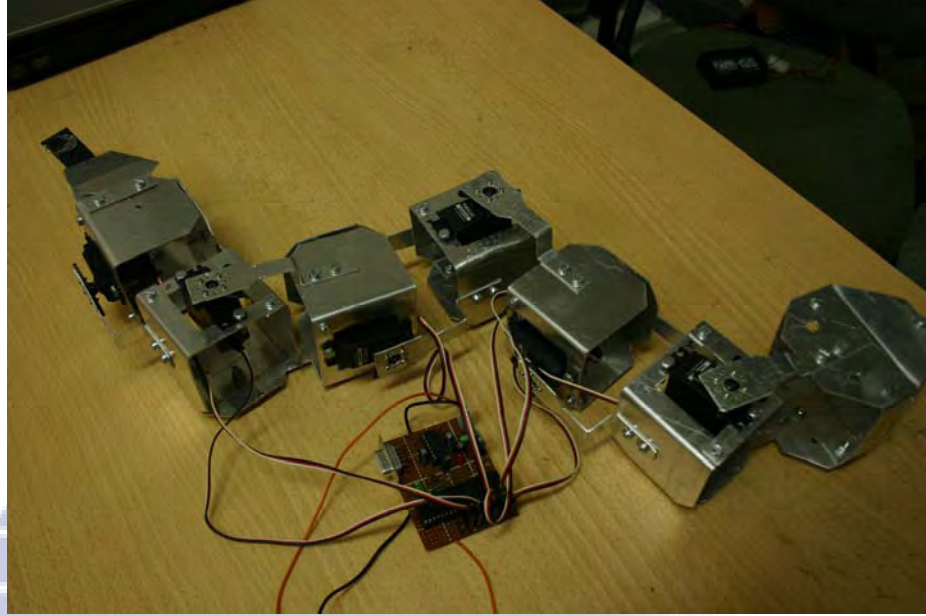
şeklinde bir robot imal edilmiştir. Bu robotun her bir bacağında 6 serbestlik derecesine sahip olmasından dolayı farklı adım hareketleri ile farklı şekillerde yürüyerek engebeli zeminlerde hareket edebilmektedir. Kalibrasyon kartı kullanılarak her bir eklemin istenilen konumu için servo motorlara gönderilmesi gereken sinyal ölçülebilmektedir.



Şekil 12.3 6 serbestlik dereceli bacak modülü.

- 5. Servo Motor Kontrolü:** C2, C3 ve C4 pinlerine servo motorlar bağlanıp, servo motor uygulamaları geliştirilebilir. Bu uygulama özellikle servo motorlar kullanılarak inşa edilen bir yılan robot uygulamasında kullanılmıştır. Bu uygulamada yılan robotun 3 ekleminin birbirleri ile uyumlu ve eş zamanlı olarak çalışması ve yılan robotun bir dalga hareketi yaparak ilerlemesi hedef alınmıştır.

Bu fonksiyon sayesinde robot test edilebilmektedir. Şekil 12.4’de yılan robotun servo motorlarının ayarlanarak robotun konumlandırılması gösterilmiştir.



Şekil 12.4 Yılan robotun servo motorlarının konumlarının ayarlanması.

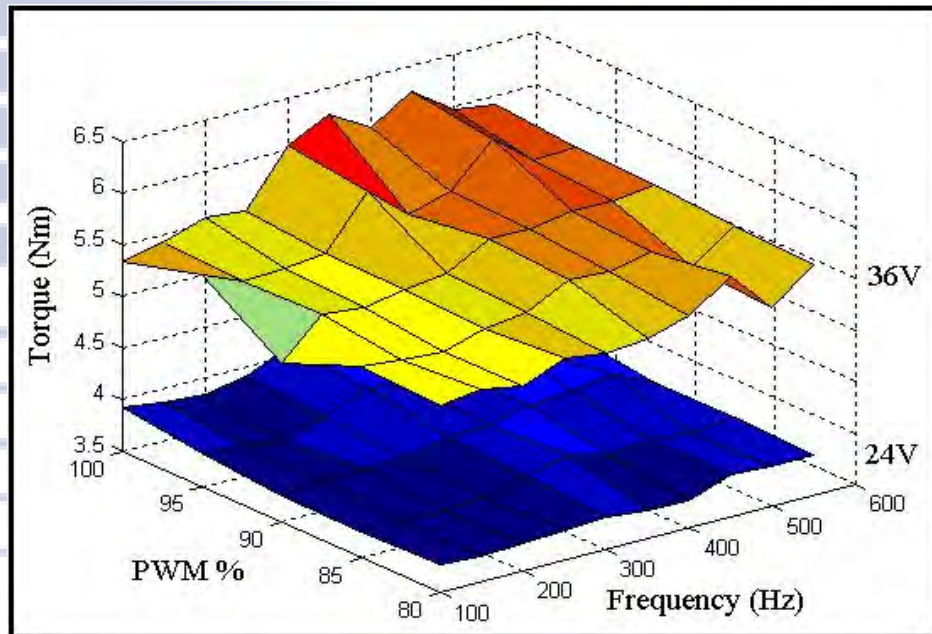
6. Dijital Tetikleme: C0 ve C1 pinlerine bağlı butonlar sayesinde 2 adet dijital giriş yapılabilir. Butonlar Şekil 12.2’de görüleceği gibi voltaj düşürme yapılarak bağlanmıştır. Mikroişlemci kodları yazılırken buna dikkat edilmelidir.

7. Led Çıkışı: D0 ve D1 pinlerine bağlı 2 adet led sayesinde yazılan programlardaki çıkışlar izlenebilir.

8. Röle ile DC Motor Kontrolü: B4 ve B5 pinlerine bağlı röleler ile DC motor kontrolü yapılabilir. B5 pini aktif yapıldığında motor saat yönüne, B4 pini aktif yapıldığında motor saat yönünün tersine döner. Burada program yazılırken dikkat edilecek nokta her iki pinin de aynı anda aktif duruma getirilmemesidir. Böyle bir durumda kısa devre olur ve devre zarar görür. Mekanik eklemler imal edilirken motorun şaftı dişli kutusuna bir kama vasıtasıyla bağlanır. Böyle bir bağlantıda kama yuvasının ve kamanın geometrik ve şekil toleransları birleşmenin kalitesini etkiler. Eğer sıkı geçme olarak bir birleşme sağlanmamış ise meydana gelen mekanik boşluktan dolayı eklemden bir boşluk oluşur. Bu

boşluk hareket yönünün değişmesi ile motorun boşa dönmesine sebep olur. Robotun eklemlerinde yer alan böyle bir boşluk burada anlatılan kalibrasyon aleti ile ölçülüp gerekli düzeltme robotun programına eklenir. Bu şekilde robotun mekanizmasında yer alan hata programlama ile giderilir.

- 9. PWM ile DC Motor Kontrolü:** B2 ve B3 pinlerine bağlı LMD18200T entegre motor sürücü devresi ile PWM uygulamaları geliştirilebilir. Burada Şekil 12.2’de gösterildiği gibi LMD18200T entegresinin PWM bacağı mikro işlemcinin B2 pinine, yön bacağı ise B3 pinine bağlanmıştır. DC motorlar darbe genişliği modülasyonu ile çalıştırılacak ise gönderilen darbenin frekansının belirlenmesi gerekir. Motorun yapısına bağlı olan bu frekans değerine göre motor en verimli olarak (minimum tork düşüşü) çalışır. Şekil 12.5’te farklı frekans değerlerine göre motorun en verimli olduğu frekans değeri bulunabilir. Kalibrasyon kartının programında frekans değeri değiştirilerek motorun ürettiği tork değeri ölçülür ve en verimli frekans değeri bulunur.



Şekil 12.5 DC motor için frekans ve darbe genişliği modülasyonu ilişkisi.

- 10. Bilgisayar ile Seri İletişim:** B6 ve B7 pinlerinden kart bilgisayara bağlanabilir ve seri iletişim uygulamaları geliştirilebilir. B6 pini ise bilgisayara veri göndermek için, B7 pini bilgisayardan veri almak için kullanılır. Visual Basic

gibi bir arayüz programı kullanılarak bilgisayar ile seri iletişim yapılabilir. Bilgisayar kullanımı ile hesaplama gücü artar, sensörlerden ölçülen bilgiler bilgisayarda saklanabilir ve kalibrasyon kartı bilgisayar üzerinden kullanılabilir.



13. Sonuç

Bu kitapta gömülü elektronik devrelerin özellikle PIC mikro işlemciler ile robotik ve otomasyon uygulamalarında karşılaşılan motor sürme, sensör bilgisi okuma, mikro işlemciler arası ve mikro işlemci bilgisayar arası haberleşmelerin nasıl yapılabileceği açıklanmıştır. Aynı zamanda devre tasarımı programı Proteus'un kullanımı açıklanmış ve devre imalatının nasıl yapılacağı ve karşılaşılabilecek sorunların nasıl giderileceği açıklanmıştır.

Özellikle son bölümde verilen kalibrasyon aleti yazarın kendi robotlarını imal ederken ihtiyaç duyduğu ve bu yüzden tasarlayıp geliştirdiği bir cihaz olarak robot imal edecek okurlara büyük kolaylıklar sağlayacaktır.

Mikro işlemci programlamada ve bilgisayar arayüz programının hazırlanmasında daha gelişmiş program kullanılabilir. Bu kitapta özellikle Basic tabanlı programlar seçilerek okurun kitabı daha kolay anlayabileceği ümit edilmiştir.

Özellikle mobil robotların kontrolünde mikro işlemci tabanlı gömülü elektronik sistemler büyük önem kazanmaktadır. PIC serisi mikro işlemciler kullanımı ve fiyatları ile öne çıkmaktadırlar. Kullanıcılar daha kapsamlı mikro işlemci programlama yazılımları (PIC C gibi) ve kapasiteleri daha yüksek bilgisayar arayüz programları (LabView gibi) kullanarak daha karmaşık ve kapasitesi daha yüksek robotik ve otomasyon sistemleri gerçekleştirebileceklerdir. Bu kitap ile uygulamalı olarak sistem parçalarını, elemanların çalışma prensiplerini, program kod örneklerini ve devre şemalarını kullanabilirler.

Referanslar

- [1] Microchip Technology Inc., *PIC16F84 18-pin Enhanced FLASH/EEPROM 8-bit Microcontroller*, Datasheet, DS35007B, 2001.
- [2] Microchip Technology Inc., *PIC16F877 28/40/44-Pin Enhanced Flash Microcontrollers*, Datasheet, DS39582B, 2003.
- [3] Microchip Technology Inc., *PIC18F452 High-Performance, Enhanced Flash Microcontrollers with 10-Bit A/D*, Datasheet, DS39564C, 2006
- [4] microEngineering Labs Inc., *PICBasicPro Compiler Manual*, 2002.
- [5] mikroElektronika Inc., *mikroBasic Users Manual*, 2006.



EK 1 PICBasic Pro Komutları

Komut	Açıklama
@	Bir satırlık assembly komutu PBP (PICBasicPro) içerisine yazılır.
ASM...AND ASM	Birden fazla assembly satırı PBP içerisine yazılır.
BRANCH	Koşullu GOTO komutu (ON...GOTO komutu benzeri).
BRANCHL	Koşullu GOTO komutu (2K'lık bellek aşıldığında).
BUTTON	Belirlenen bir pinden bir giriş almak için kullanılır.
CALL	Assembly dili alt programını çağırır.
CLEAR	Tüm değişkenlerin içeriğini sıfırlar.
COUNT	Bir pinden gelen darbe sayısını sayar.
DATA	On-Chip EEPROM belleğin içine sabit değerler yazar.
DEBUG	Sabit bir pin ve hızda asenkron seri çıkış yapar.
DISABLE	INTERRUPT kesmelerini geçersiz yapar.
DTMFOUT	Bir pinden telefon tuşu ton seslerini üretir.
EEPROM	On-Chip EEPROM belleğin içerisine veri yazar.
ENABLE	INTERRUPT kesmelerini geçerli yapar.
END	Programı durdurur ve düşük güç moduna geçer.
FOR...NEXT	Komut gurubunu istenilen sayıda tekrar eder.
FREQOUT	Bir pinden iki farklı frekans üretir.
GOSUB	Bir PBP alt programını çağırır.
GOTO	Belirlenen bir etikete dallanmayı sağlar.
HIGH	Belirlenen bir pin çıkışını (1) yapar.
HSERIN	Donanımı seri iletişimi destekleyen cihazlardan (USART) seri veri almak için kullanılır.
HSEROUT	Donanımı seri iletişimi destekleyen cihazlardan (USART) seri olarak veri göndermek için kullanılır.
I2CREAD	Bir seri EEPROM bellekten bir bytelık veriyi seri olarak okumak için kullanılır.
I2CWRITE	Bir seri EEPROM belleğe bir bytelık veriyi yazar.
IF..THEN...ELSE...ENDIF	Bir koşula bağlı olarak farklı komut veya komut gurupları çalıştırmak için kullanılır.
INPUT	Bir pini giriş yönüne yönlendirir.
{LET}	Bir ifadenin sonucunu değişkene atar (kullanımı isteğe bağlıdır).
LCD OUT	Bir LCD'ye 4-bit veya 8-bit ile karakter gönderir.
LOOKDOWN	Bir değer tablosundan sabit değer araştırmak için kullanılır.
LOOKDOWN2	Bir değer tablosundan sabit/değişken değer araştırmak için kullanılır.
LOOKUP	Bir değer tablosundan sabit değer almak için kullanılır.
LOOKUP2	Bir değer tablosundan sabit/değişken değer almak için kullanılır.
LOW	Bir pinin çıkışını LOW (0) yapar.
NAP	İşlemciyi kısa süre durdurur.
ONINTERRUPT	Bir kesme oluştuğunda kesme alt programının çağırılmasını sağlar.
OUTPUT	Bir pini çıkışa yönlendirir.

PAUSE	İstenilen bir zaman süresinde gecikme yapar (zaman birim aralıkları 1 ms dir).
PAUSEUS	İstenilen bir zaman süresinde gecikme yapar (zaman birim aralıkları 1 us dir).
PEEK	Bir registerdan 1 bytlık veriyi okur.
POKE	Bir registra 1 bytlık veriyi yazar.
POT	Belirlenen bir pindeki potansiyometreyi okur.
PULSIN	Bir pindeki darbe genişliğini ölçer.
PULSOUT	Bir pinden istenilen uzunlukta darbeler üretir.
PWM	Bir pinden PWM darbeleri üretir.
RANDOM	Bir değişken içerisine 1-65535 arasında rastgele sayılar üretir.
RCTIME	Bir pindeki darbe genişliğini ölçer.
READ	On-Chip EEPROM bellekten 1 byte lık veriyi okur.
RESUME	INTERRUPT alt programı bittikten sonra programı kaldığı yerden devam ettirir.
RETURN	Programı en son kullanılan GOSUB komutundan sonraki komuttan itibaren devam ettirir.
REVERSE	Bir pinin giriş/çıkış yönlendirmesini tersine çevirir. Giriş olan pinler çıkışa yönlendirilir.
SERIN	Asenkron seri giriş komutu (BS1 stilinde).
SERIN2	Asenkron seri giriş komutu (BS2 stilinde).
SEROUT	Asenkron seri çıkış komutu (BS1 stilinde).
SEROUT2	Asenkron seri çıkış komutu (BS2 stilinde).
SHIFTIN	Senkron seri giriş komutu.
SHIFOUT	Senkron seri çıkış komutu.
SLEEP	İşlemciyi belirlenen bir süre boyunca düşük güç durumuna sokar.
SOUND	Belirlenen bir pinden istenilen tonda ses üretir.
SWAP	İki değişken içerisindeki değerleri değiştirir.
TOGGLE	Bir pini çıkışa yönlendirir ve toggle darbe üretir. Yani pinin komumu 1 ise 0 yapar.
WHILE...WEND	Bir koşul doğru (true) olduğu sürece bir grup komutu tekrarlar.
WRITE	On-Chip EEPROM belleğe 1 bytelık veriyi yazar.

EK 2 Mikrobasic Komutları

Komut	Açıklama
ADC_READ	Belirli bir analog girişten 10 bit analog veriyi okur
EEPROM_READ	On-chip eepromun belirli bir adresindeki 1 byte'lık veriyi okur
EEPROM_WRITE	On-chip eepromun belirli bir adresine 1 byte'lık veriyi yazar
FLASH_WRITE	Mikro işlemcinin flash hafızasına veri yazar
FLASH_READ	Mikro işlemcinin flash hafızasından veri okur
FLASH_ERASE	Mikro işlemcinin flash hafızasında bulunan veriyi siler
KEYPAD_INIT	Tuş takımının hangi porta bağlı olduğunu belirler
KEYPAD_READ	Tuş takımından veri okur
KEYPAD_RELEASED	Tuş takımından herhangi bir tuşa basıldığında tuş bırakılana kadar bekler tuş bırakılınca veri okur
LCD_CONFIG	LCD'nin hangi pinlere bağlı olduğunu belirler
LCD_INIT	LCD'yi önceden belirlenen pinlere bağlamak için kullanılır
LCD_OUT	LCD'ye yazı yazdırmak için kullanılır
LCD_OUT_CP	İmlecin o an bulunduğu pozisyondan itibaren LCD'ye yazı yazdırır
LCD_CHR	LCD'ye karakter yazdırmak için kullanılır
LCD_CHR_CP	İmlecin o an bulunduğu pozisyondan başlayarak LCD'ye karakter yazdırır
LCD_CMD	LCD'ye komut göndermek için kullanılır
GLCD_INIT	GLCD'nin (Grafik LCD) hangi pinlere bağlanacağı belirlenir
GLCD_SET_SIDE	GLCD'nin yazım yönünü belirler sağdan mı soldan mı başlanacak
GLC_SET_PAGE	Yazma işleminde GLCD'nin hangi satırından başlanacağı seçilir
GLCD_SET_X	İstenilen satırda nokta olarak yazım işleminin nereden başlanacağı seçilir
GLCD_READ_DATA	GLCD'den veri okumak için kullanılır
GLCD_WRITE_DATA	GLCD'nin hafızasında istenilen bir adrese veri yazar ve bir sonraki adrese geçer
GLCD_FILL	GLCD üzerinde istenilen kısmı patern olarak doldurur
GLCD_DOT	GLCD üzerinde x ve y koordinatları verilen yere nokta koyar
GLCD_LINE	GLCD'ye (x1,y1) noktasından (x2,y2) noktasına kadar çizgi çizer
GLCD_V_LINE	GLCD'ye (x,y1) noktasından (x,y2) noktasına yatay çizgi çizer
GLCD_H_LINE	GLCD'ye (x1,y) noktasından (x2,y) noktasına dikey çizgi çeker
GLCD_RECTANGLE	GLCD'ye (x1,y1) sol üst köşenin (x2,y2) sağ alt köşenin koordinatları olacak şekilde dikdörtgen çizer
GLCD_BOX	GLCD'ye (x1,y1) sol üst köşenin (x2,y2) sağ alt köşenin koordinatları olacak şekilde kutu çizer

GLCD_CIRCLE	GLCDye (x,y) merkezi olmak üzere verilen yarıçapa göre çember çizer
GLCD_SET_FONT	Yazılan yazının karakterini belirlemek için kullanılır
GLCD_WRITE_CHAR	GLCDye karakter yazdırmak için kullanılır
GLCD_WRITE_TEXT	GLCDye yazı yazmak için kullanılır
GLCD_IMAGE	Bitmap formatındaki şekilleri GLCD ye yazdırmak için kullanılır. Nesneler matriks şeklinde tanımlanmalıdır
OW_RESET	Tek kablo iletişimi için kullanılır. Cihaz meşgul ise 1, veri göndermeye hazır ise 0 değeri alır.
OW_READ	Tek kablo ile iletişim kuran cihazlardan veri okumak için kullanılır
OW_WRITE	Tek kablo ile iletişim kuran cihazlara veri göndermek için kullanılır
PS2_INIT	PS2 klavyeleri PIC mikro işlemcilere bağlamak için kullanılır. Bu komut ile klavyenin mikro işlemcinin daha önce belirtilen pinlerine bağlı olduğu belirtilir.
PS2_CONFIG	Bu komut ile klavyenin hangi pinlere bağlanacağı kullanıcı tarafından seçilir
PS2_KEY_READ	Bu komut ile klavyeden basılan tuş okunur.
PWM_INIT	Üzerinde donanımsal olarak PWM modülü olan işlemcilerden PWM sinyali üretmek için kullanılır. Bu komut ile PWM modülü aktif hale getirilir.
PWM_CHANGE_DUTY	PWM iş süresini değiştirmek için kullanılır.
PWM_START	PWM sinyal üretimini başlatır.
PWM_STOP	PWM sinyal üretimini durdurur
SOFT_I2C_CONFIG	Harici EEPROMun hangi pinlere bağlandığı belirtilir
SOFT_I2C_START	Harici EEPROMa başlama sinyali gönderilir
SOFT_I2C_READ	Harici EEPROMdan veri okumak için kullanılır
SOFT_I2C_WRITE	Harici EEPROMa veri yazmak için kullanılır
SOFT_I2C_STOP	Harici EEPROMa durması için sinyal gönderilir.
SOFT_UART_INIT	SOFT_UART komut PIC-PC veya PIC-PIC arasında seri iletişim için kullanılır. Bu komut haberleşmenin hangi pinden ve hangi hızda yapılacağını belirtir.
SOFT_UART_READ	Bu komut daha önce belirlenen pinden seri olarak veri okumak için kullanılır
SOFT_UART_WRITE	Bu komut daha önce belirtilen pinden seri olarak veri göndermek için kullanılır.
SOUND_INIT	Ses frekansının hangi pinden gönderileceği belirtilir
SOUND_PLAY	Belirtilen pinden ses sinyali gönderilir.
USART_INIT	Üzerinde donanımsal olarak USART modulu olan işlemcilerde seri haberleşme için kullanılır. Bu komut ile haberleşme hızı belirlenir
USART_DATA_READY	İşlemciye gelecek veri hazır ise 1 değil ise 0 değeri döndürür.
USART_READ	Donanımsal USART modülünden veri okumak için kullanılır.
USART_WRITE	Donanımsal USART modulundan veri göndermek için kullanılır.
HID_ENABLE	HID (human interface device) komut USB portundan

	haberleşmek için kullanılır. Bu komut ile mikro işlemci üzerinde bulunan USB portu aktif hale getirilir.
HID_READ	USB portundan veri okumak için kullanılır.
HID_WRITE	USB portundan veri göndermek için kullanılır.
HID_DISABLE	USB portundan olan iletişim durdurulur.
BUTTON	Bu komut butona basıldığı anda oluşan arkların istenmeyen sinyaller göndermesini engellemek için kullanılır. Bu komut kullanılarak butonun hangi portun hangi pinine bağlandığı, buton basıldığında mı vrys bırakıldığında mı işlem yapılacağı seçilir.
BYTETOSTR	Byte tipindeki değişkenleri string tipine çevirir.
SHORTTOSTR	Short integer şeklinde olan değişkenleri string tipine çevirir.
WORDTOSTR	Word tipindeki değişkenleri string tipine çevirir.
INTTOSTR	Integer şeklindeki değişkenleri string tipine çevirir.
LONGINTTOSTR	Long integer şeklinde olan değişkenleri string tipine çevirir.
FLOATTOSTR	Float şeklinde olan değişkenleri string şekline çevirir.
STRTOINT	String şeklinde olan değişkenleri integer şekline çevirir.
STRTOWORD	String şeklinde olan değişkenleri word şekline çevirir.
BCD2DEC	8 bit Binary şeklinde olan değişkenleri decimal hale çevirir.
DECTOBCD	8 bit Decimal şeklindeki değişkenleri binary şekline dönüştürür.
BCD2DEC16	16 bit Binary şeklinde olan değişkenleri decimal hale çevirir.
DECTOBCD16	16 bit Decimal şeklindeki değişkenleri binary şekline dönüştürür.

Ayrıca Mikro Basic programının içinde Delay, Matematik, String, SPI Grafik LCD, Port Expander, RS485, Ses, SPI T6963 Grafik LCD, I2C, Ethernet ve SPI Ethernet kütüphaneleri vardır.