T.C.
MİLLÎ EĞİTİM BAKANLIĞI

MEGEP

(MESLEKİ EĞİTİM VE ÖĞRETİM SİSTEMİNİN
GÜÇLENDİRİLMESİ PROJESİ)

ENDÜSTRİYEL OTOMASYON TEKNOLOJİLERİ

MİKRODENETLEYİCİ 1

ANKARA, 2009

Milli Eğitim Bakanlığı tarafından geliştirilen modüller;

- Talim ve Terbiye Kurulu Başkanlığının 02.06.2006 tarih ve 269 sayılı Kararı ile onaylanan, Mesleki ve Teknik Eğitim Okul ve Kurumlarında kademeli olarak yaygınlaştırılan 42 alan ve 192 dala ait çerçeve öğretim programlarında amaçlanan mesleki yeterlikleri kazandırmaya yönelik geliştirilmiş öğretim materyalleridir (Ders Notlarıdır).
- Modüller, bireylere mesleki yeterlik kazandırmak ve bireysel öğrenmeye rehberlik etmek amacıyla öğrenme materyali olarak hazırlanmış, denenmek ve geliştirilmek üzere Mesleki ve Teknik Eğitim Okul ve Kurumlarında uygulanmaya başlanmıştır.
- Modüller teknolojik gelişmelere paralel olarak, amaçlanan yeterliği kazandırmak koşulu ile eğitim öğretim sırasında geliştirilebilir ve yapılması önerilen değişiklikler Bakanlıkta ilgili birime bildirilir.
- Örgün ve yaygın eğitim kurumları, işletmeler ve kendi kendine mesleki yeterlik kazanmak isteyen bireyler modüllere internet üzerinden ulaşılabilirler.
- Basılmış modüller, eğitim kurumlarında öğrencilere ücretsiz olarak dağıtılır.
- Modüller hiçbir şekilde ticari amaçla kullanılamaz ve ücret karşılığında satılamaz.

İÇİNDEKİLER

AÇIKLAMALAR

KOD	**523EO0365**
ALAN	**Endüstriyel Otomasyon Teknolojileri**
DAL/MESLEK	**Ortak Alan**
MODÜLÜN ADI	**Mikrodenetyeyici 1**
MODÜLÜN TANIMI	
SÜRE	40/32
ÖN KOŞUL	
YETERLİK	Yapılacak işe uygun mikro denetleyici seçerek program yüklemek.
MODÜLÜN AMACI	**Genel Amaç** Öğrenci, yapılacak işe uygun Mikro denetleyici seçerek program yükleyebileceksiniz. **Amaçlar** **1.** Bir mikro denetleyici seçerek kullanabilmek için gerekli olan temel bilgileri öğreneceksiniz. **2.** Bir mikro denetleyici seçerek kullanabilmek mikro denetleyici eğitim seti yapacaksınız. **3.** Seçilen mikro denetleyicinin programlanabilmesi için uygun editör, derleyici ve programlayıcıların seçerek kullandığınız bilgisayara yükleyeceksiniz. **4.** Mikro denetleyiciye örnek program yüklemek ve programı test edecektir.
EĞİTİM ÖĞRETİM ORTAMLARI VE DONANIMLARI	**Ortam:** Mikro denetleyiciler Laboratuarı **Donanım:** Mikro denetleyici katalogları, mikro denetleyici, Elektronik devre elemanları, Elektronik malzeme katalogları, multimetre, baskı devre ve lehimleme araç gereçleri, Mikro denetleyici programlama editörü, bilgisayar
ÖLÇME VE DEĞERLENDİRME	Modül içinde yer alan her öğrenme faaliyetinden sonra verilen ölçme araçları ile kendinizi değerlendireceksiniz. Öğretmen modül sonunda ölçme aracı (çoktan seçmeli test, doğru-yanlış vb.) kullanarak modül uygulamaları ile kazandığınız bilgi ve becerileri ölçerek sizi değerlendirecektir.

GİRİŞ

Sevgili Öğrenci,

Hızla gelişen dünya teknolojisinde kontrol, sistemlerde yerini çok uzun zaman önce almaya başlamıştır. Günümüzde kontrol teknolojisi hızla gelişmeye devam etmektedir. Dünya piyasalarında diğer şirketlerle rekabet edebilmek için kontrol teknolojileri yardımıyla daha ucuz ve daha kaliteli üretim yapmak gerekmektedir. Bu nedenle kontrol teknolojilerinin önemi giderek artmaktadır.

Kontrol teknolojileri, sistemlere göre farklılıklar gösterebilir ya da aynı sistemi farklı kontrol teknolojileri ile denetleyebiliriz. Mekanik olarak da kontrol edilebilen sistemler olduğu gibi aynı sistemleri yüksek veya düşük akım teknolojisi ile ya da programlanabilir bir denetleyiciyle de kontrol edebiliriz. Bu sadece bizim isteğimize ve beklentilerimize bağlıdır.

Biz bu kitapta, bir mikro denetleyici ile yapılan kontrol teknolojisinden bahsedeceğiz. Fakat mikro denetleyiciler de kendi arasında farklılıklar göstermektedir. Bu amaçla bu kitapta, kullanımı kolay ve ucuz olan PIC16F84 mikro denetleyicisi üzerinde çalışmalar yaparak yapısına ve kontrol teknolojisine ait bilgi ve beceri kazandırılacaktır. Yine bu kitapta bir mikro denetleyici ile kontrol yaparken kontrol edilen sistemin ve mikro denetleyicinin donanımsal özelliklerini bilmek gerekliliği ile birlikte programlamanın önemi de vurgulanacaktır.

PIC16F84 mikro denetleyicisi ile çalışmalar yaparken programlama dili olarak başlangıç seviyesi için ASSEMBLY dili kullanılacaktır.

ÖĞRENME FAALİYETİ–1

AMAÇ

Bir mikro denetleyici seçerek kullanabilmek için gerekli olan temel bilgileri öğreneceksiniz.

ARAŞTIRMA

- Mikro denetleyici bir kontrol arabirimidir. Fakat kontrol sistemleri farklılık gösterebilir. Acaba kontrol nedir? Hangi tür kontrol sistemleri vardır ve birbirlerine göre avantajları-dezavantajları nelerdir? Bu konu ile ilgili bir araştırma yapmanız gerekmektedir.

1. MİKRO DENETLEYİCİNİN TEMELLERİ

Mikro denetleyiceler Endüstriyel Otomasyonda kontrol sistemlerinde oldukça yaygın olarak kullanılmaktadır. Mikrodenetleyicilerin yapılarını anlamak için öncelikle tarihi gelişiminden başlayarak mikroişlemcilerden farkına kadar bazı temel bilgilere sahip olmamız gerekir. Aşğıda mikro denetleyicilere ait temel bilgileri detaylı olarak inceleyeceğiz.

1.1. Mikro Denetleyici Tanımı ve Çeşitleri

Mikrodenetleyiciler yapılarına göre farklıklıklar gösterirler. Mikrodenetleyiciler yaptıkları işlev olarak Mikroişlemciklere benzeselerde yapısal bazı farklılıkları bulunmaktadır.

1.1.1. Mikro Denetleyicinin Kısa Tarihçesi

Harvard mimarisindeki ilk mikro denetleyici ünitesi, General Instruments firması tarafından 1970'lerin ortalarında üretilen Signetics 8X300 modeliydi. Bu 16 bitlik CP1600 MPU için programlanabilen giriş/çıkış portu olmak üzere Peripheral Interface Controller (Çevrebirim arayüz denetleyicisi - PIC) olarak tasarlandı.

General Instruments firması mikro elektronik bölümünü sattı ve bu bölüm 1988 yılında Arizona Microchip Technology adıyla yeni bir firmaya dönüştü. Microchip'in ana ürünü, bugün de hâlâ öyle olan, PIC serisi mikro denetleyicilerdir. 1989'da ilk piyasaya sürülen aile PIC16C5X serisiydi. Bu Harvard mikro denetleyiciler 33 komutluydu. Bütün

komutlar 12-bit word olarak kodlanıyordu. Azaltılmış Komut Kümesi (Reduced Instruction Set Computer - RISC) temelli olan komut seti hızlı, etkili ve ucuz işlemci üretimini sağladı. PIC16C5XX 12-bit çekirdekli ailede 512 ve 2048 komutluk tek sefer programlanabilen (One Time Programmable (OTP)) EEPROM Program belleği, 25–73 bayt veri belleği,18 ve 28-pinli paketlerde 12 veya 20 giriş/çıkış pini ve 8-bit zamanlayıcı gibi özellikler bulunmaktaydı.PIC12CXXX ailesi bunların 8-pinlik eşdeğerleridir.

1992 yılında 14-bitlik çekirdeğe sahip PIC16CXXX ailesi daha fazla program alanının ve interrupt (kesme) işlemleri yanında A/D çeviriciler,16 bit sayıcılar gibi çevre birimlerinin kullanımına olanak sağladı. Bu ailedeki RISC komut seti de 12-bit çekirdektekilerle hemen hemen aynıydı ve 35 komuttan oluşuyordu.1997'de çarpma yapabilen bir ALU'e ve ileri arabirim yeteneklerine sahip 16-bit PIC17CXXX ailesi piyasaya sunuldu. Ardından 1999 yılında da genişletilmiş 16-bit çekirdekli PIC18CXXX ailesi sunuldu. Bu ailedeki işlemcilerde komut sayısı 77 idi ve bu yüksek-seviye dillerin derleyicilerin ihtiyaçlarını daha fazla karşılıyordu.

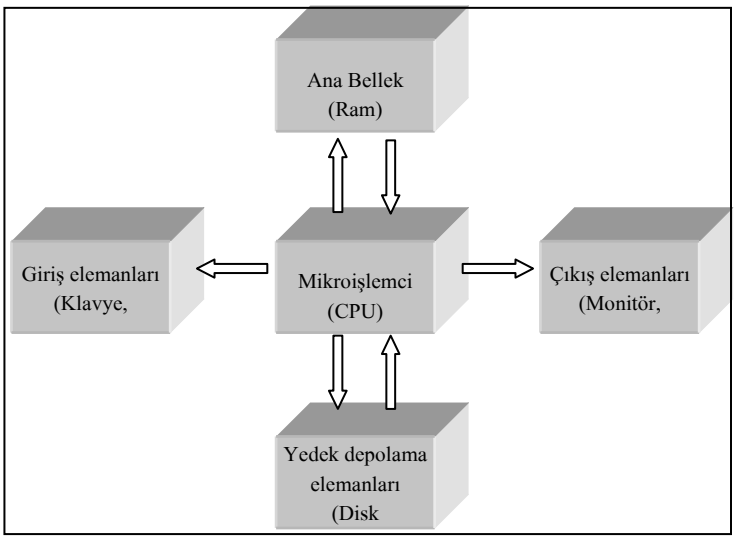
Bu 3 aile arasında, 14-bit çekirdekli olan aile hem kullanım kolaylığı hem de maliyet olarak en uygunudur. Burada ve birçok kaynakta hakkında bilgiler bulabileceğiniz PIC16F84, orta seviye ailesinin bir üyesidir. Yazılım açısından baktığımızda bütün cihazlar aynı çekirdeğe sahiptirler. Ancak donanım açısından birçok ortak noktaları olmakla birlikte farklı giriş/çıkış birimlerinin karışımıdırlar. Örneğin 16C74'te 8 kanal analog giriş portu, PIC16C66'da senkronize seri portu ve PIC16F84'de de kalıcı veri belleği bulunmaktadır. Bu üç cihaz da benzer paralel giriş/çıkış, sayıcı ve kesme idare birimlerine sahiptir.

1.1.2. Mikroişlemci ve Mikro Denetleyici Nedir?

Temelde yaptıkları işler aynı olarak görülsede mikroişlemciler ile mikro denetleyicler arasında yapısal olarak ve işlevsel olarak farklılıklar bulunmaktadır. Bu farklılıkları anlamak için öncelikle mikroişlemci ve mikro deneteleyici tanımlarını inceleyelim.

1.1.2.1. Mikroişlemci nedir?

Çok genel bir ifadeyle bir bilgisayarın beyni, esas işi yapan kısmı olarak isimlendirilebilecek olan mikroişlemciler hakkında biraz daha ayrıntılı bir açıklama şu şekilde yapılabilir: bir dijital bilgisayar üç temel kısımdan oluşmaktadır.



Şekil 1.1: Bir bilgisayarın temel blok diyagramı

- Merkezi İşlem Birimi - MİB (Central Processing Unit – CPU)
- Program ve Veri Hafızaları (Program and Data Memory)
- Giriş – Çıkış Birimleri (Input – Output Units)

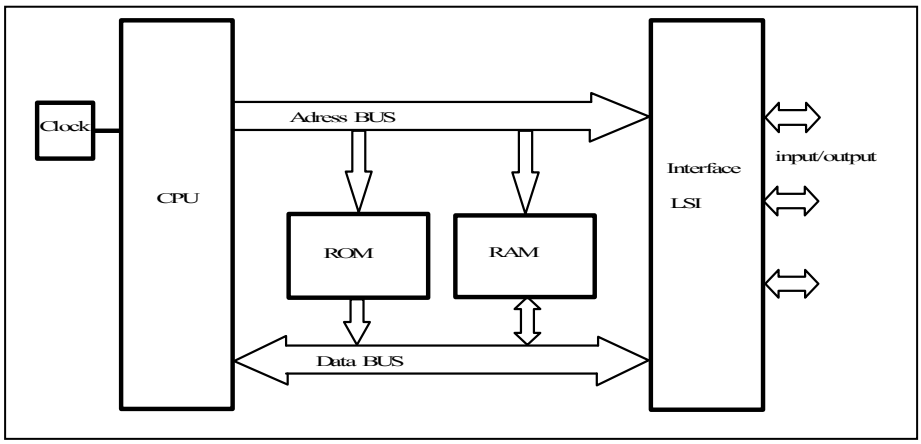
Merkezi İşlem Birimi (MİB / CPU), verileri işleme ve sistemi oluşturan çeşitli birimler arasında bilgi akışı kontrolü işlemlerini gerçekleştirir.

Veri işlemenin büyük çoğunluğu MİB'de yer alan Aritmetik Lojik Birim üzerinde gerçekleştirilir. Ancak bu işlemlerin gerçekleştirilmesi sırasında Kod Çözme Kontrol Birimleri ile çeşitli Saklayıcılar (Registers) da çok yoğun olarak kullanılır.

İşte bu merkezi işlem birimini oluşturan çeşitli alt birimlerin tek bir entegre devre üzerinde gerçekleştirilmiş – üretilmiş hâline Mikroişlemci (Microprocessor) adı verilir.

Bir mikroişlemci kullanılarak hazırlanmış bilgisayarlara mikrobilgisayar denilmektedir. Hafıza ve giriş-çıkış birimlerinin miktarı, türü ve kapasitesi uygulamaya bağlı olarak değişir.

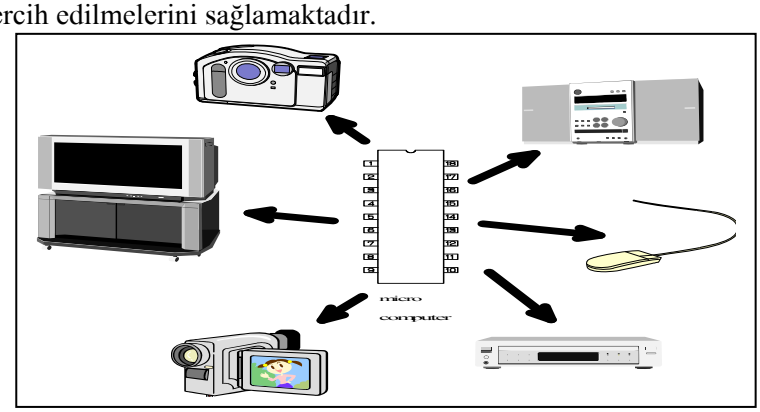
Mikroişlemciler, PC adını verdiğimiz kişisel bilgisayarlarda kullanıldığı gibi sanayi tezgâhlarına bağlı bilgisayarlarda da çok kullanılmaktadır. Tek başına bir mikro işlemci işlevini yerine getiremez. Yukarıda belirtilen elemanlara ihtiyaç vardır.



Şekil 1.2: Bir bilgisayarın mimarisi

1.1.2.2. Mikro Denetleyici Nedir?

Bir mikro denetleyici, komple bir bilgisayarın (MİB, hafıza ve giriş - çıkışlar) tek bir entegre devre üzerinde üretilmiş hâlidir. Kısıtlı miktarda olmakla birlikte yeterince hafıza birimlerine ve giriş – çıkış uçlarına sahip olmaları sayesinde tek başlarına çalışabildikleri gibi donanımı oluşturan diğer elektronik devrelerle irtibat kurabilir, uygulamanın gerektirdiği fonksiyonları gerçekleştirebilirler. Üzerlerinde analog-dijital çevirici gibi entegre devreler barındırmaları sayesinde algılayıcılardan her türlü verinin toplanması ve işlenmesinde kullanılabilmektedirler. Ufak ve düşük maliyetli olmaları gömülü uygulamalarda tercih edilmelerini sağlamaktadır.



Şekil 1.3: Mikro denetleyici uygulama örnekleri

Mikroişlemciler ve mikro denetleyiciler günlük hayatta kullanılan sayısız cihaz ve sistemin içinde yer almakta olup, bu ürünleri kullanarak gerçekleştirilebilecek uygulamalar insanın hayal gücü ile sınırlıdır. Motor kontrolünden fotoğraf makinesi ışık ve focus ayarına, cep telefonlarından merkezi klima sistemlerine, faks ve fotokopi makinelerinden radyo teyp ve TV lere, fabrika otomasyonundan hayat kurtaran biyomedikal cihazlara, oyuncaklardan askeri cihazlara, cebinizdeki elektronik bilet uygulamasından cüzdanınızdaki banka kartlarına varıncaya kadar akla gelebilecek her yerde mikro denetleyiciler yer almaktadır.

1.1.3. Mikro Denetleyicilerin, Mikroişlemcilere Göre Tercih Nedenleri

Mikroişlemci ve mikrodenetleyicilerin farkı: Mikroişlemciler hafıza veya giriş çıkış entegresi gibi çevre elemanlar ile birlikte aritmetik işlemler yapabilen, karar verebilen entegrelerdir. Mikro denetleyici ise mikroişlemcinin yanı sıra kendi program hafızası ve giriş çıkış özelliklerini içinde barındıran entegrelerdir. Bu özellikleri sayesinde yüklü bir programı uygulamaya dönüştürerek bir kontrol sistemi olarak kullanılabilir.

Mikro denetleyiciler, mikroişlemcilere kıyasla daha spesifik uygulamalar için hazırlanmıştır diyebiliriz. Çünkü gerekli elemanları kendi üzerinde barındırarak neredeyse yalnız başına çalışabilmektedir. Oysa mikroişlemciler bilindiği üzere BUS (veri yolu), IO (giriş/çıkış), RAM (bellek) vb. yapılara ihtiyaç duyarlar. Evet, sanırım aynı şeyi düşündünüz, mikro denetleyiciyi kendi işlemcisi, ana kartı, belleği, portları bulunan bir bilgisayar gibi düşünebiliriz.

İnsanı heyecanlandırmak için yeterli bir özellik; ancak boyutlarının küçülmesi, fiyatlarının ucuzlaması, daha az ısınmalarını sağlamak bazı özelliklerin bilgisayarlardan daha az güçlü olmasını gerektirebilmektedir. Daha önce bahsettiğim gibi biraz daha özel amaçlar için hazırlanırlar. Bu sebeple piyasada hayli fazla çeşit bulunmakta ve bunlardan birini seçmek tamamen yapılacak iş ile bağımlı duruma gelmektedir.

1.1.4. Mikro Denetleyici Çeşitleri ve Özellikleri

Günümüzde mikroişlemci ve mikro denetleyiciler üreten irili ufaklı pek çok firma bulunmaktadır. Bunlara örnek olarak INTEL, MOTOROLA, AMD, PHILIPS, SIEMENS, TEXAS INS., DALLAS, ATMEL, MICROCHIP, HITACHI, MITSUBISHI, SGS-THOMSON, ANALOG DEVICES, NATIONAL gibi firmalar sayılabilir. Bu firmaların bazıları sadece kendilerine özgü işlemcileri piyasaya sürerken bazıları da ilk üretimi ve patenti bir başka firmaya ait olmakla birlikte, orijinal işlemci ile uyumlu fakat çeşitli başka ek özelliklere de sahip türev ürünler (derivatives) üretebilmektedir. Neredeyse her mikroişlemci (CPU) üreticisinin ürettiği birkaç mikro denetleyicisi bulunmaktadır. Her firma ürettiği ürününe isim ve parça numarası verirken kendine özgü birtakım rakamlar ve karakterler kullanırlar. Bu rakam ve karakterler, ürünler hakkında kısa bilgiler de verebilmektedir.

Bir uygulamaya başlamadan önce hangi firmanın ürünü kullanılacağına, daha sonra da hangi numaralı denetleyicinin kullanılacağına karar vermek gerekir. Bunun için Mikro denetleyici gerektiren uygulamada hangi özelliklerin olması gerektiği önceden bilinmesi gerekir. Bu özellikler şu şekilde sıralanır.

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

Daha da ayrıntıya girecek olursak bu listede sıralanacak özellikler uzayıp gidecektir. Şimdi de bizim bu kitapta ele alacağımız Microchip'in ürünü olan PIC'i neden seçtiğimize değinelim. Microchip, 8-bit'lik Mikro denetleyici ve EEPROM üreten bir Amerikan şirketidir. Arizona eyaletinde iki, Tayland ve Tayvan'da da birer tane olmak üzere toplam dört fabrika ile kendi alanında dünyada söz sahibi olan bir chip üreticisidir.

1.1.5. Mikro Denetleyici İşletimi

Bir PIC mikro denetleyicisinde bir komutun işlenme süreci 4 aşamada gerçekleştirilir.

- **1.Aşama: Alma (Fetch)**

Hafızaya yüklenmiş olan program komutlarını alır.

- **2.Aşama: Kod Çözme (Decode)**

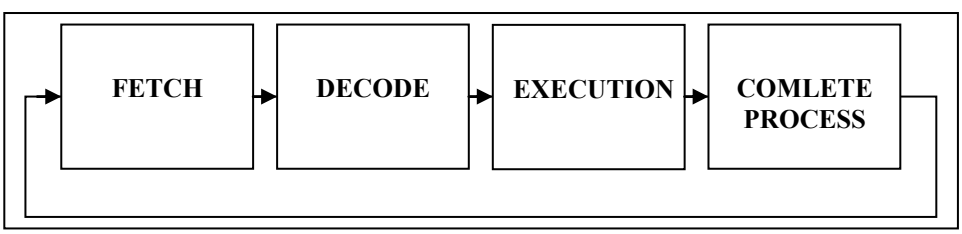
Yazmaçtaki komutları kod çözücü(decoder) yardımıyla çözer.

- **3. Aşama: Uygulama (Execution)**

Çözülen komutları uygular ve bu işlemi sürekli tekrar eder.

- **4. Aşama: İşlemi tamamlama (Comlete Process)**

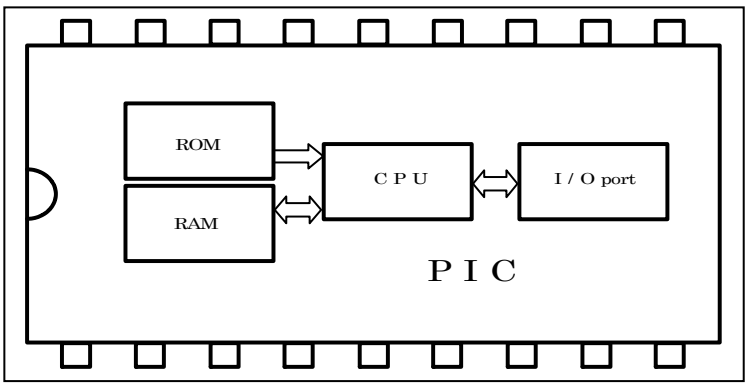
İşlemi tamamlama sürecidir. Bazı komutlarda işlem sonucunu W ya da file register'a yazama süreci olarak düşünülmüştür, bazı komutlarda ise bu süreç içersinde işlem yapılmaz.



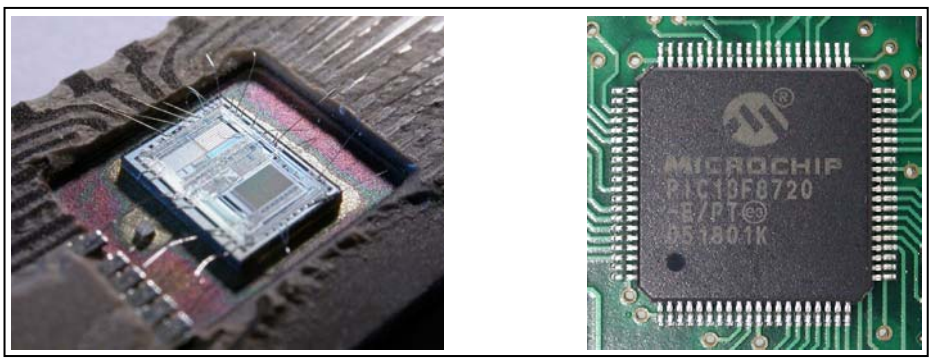
Şekil 1.4: Komut işleme döngüsü

1.1.6. Mikro Denetleyici Yapısı

PIC Mikro denetleyici daha önce de açıklandığı gibi İngilizce (Peripheral Interface Controller) kelimelerinin baş harflerinden oluşmaktadır. Anlamı ise dış üniteleri denetleyen arabirimdir.



Şekil 1.5: Bir mikro denetleyici sisteminin blok diyagramı



Şekil 1.6: Bir PIC'in içyapısının görüntüsü ve pin bağlantıları

Mikro denetleyiciler, çoğunlukla yer aldıkları uygulama devresinin içine gömülmüş, sadece oraya adanmış olarak kullanılırlar. Bu özellikleri nedeniyle bilgisayarlardaki kullanıcı uygulama programlarını çalıştırma gibi esneklikleri olmamakla birlikte kontrol ağırlıklı uygulamalarda alternatifsiz seçenek olarak karşımıza çıkarlar.

Onları böyle cazip kılan, çok düşük boyutlu olmaları (az yer kaplamaları), düşük güç tüketimleri, düşük maliyetlerine karşın yüksek performansa sahip olmaları gibi özellikleridir. Motor kontrolünden fotoğraf makinesi ışık ve focus ayarına, cep telefonlarından merkezi klima sistemlerine, faks ve fotokopi makinelerinden radyo teyp ve TV'lere, fabrika otomasyonundan hayat kurtaran biyomedikal cihazlara, oyuncaklardan askeri cihazlara, cebinizdeki elektronik bilet uygulamasından cüzdanınızdaki banka kartlarına varıncaya kadar akla gelebilecek her yerde mikro denetleyiciler yer almaktadır.

Bu tür uygulamalarda kullanıldıkları için hafıza ve paralel/seri giriş-çıkış birimlerinin yanı sıra zamanlayıcılar (timers), sayıcılar (counters), kesme kontrol birimleri (Interrupt Control), Analog-Sayısal dönüştürücüler (A/D Converters) gibi çeşitli çevre birimleri de mikro denetleyici bütünleşmiş devrelerinin içinde yer almaktadır.

Ayrıca genellikle gerçek zamanlı uygulamalarda çalışmalarıyla mikro denetleyiciler, mikroişlemcilerden ayrılmaktadırlar. Gerçek zamanlı uygulamalarda dış dünyadan (işlemcinin dışındaki elektronik ortamdan) gelen işaretler çok hızlı değişim gösterebilir ve bunları işleyip gereken çıkışları aynı hızlılıkta dış dünyaya uygulamak gerekebilir.

Böyle bir performansı, çok küçük boyutlarda ve çok daha az güç tüketerek sadece mikro denetleyiciler aracılığıyla gerçekleştirmek mümkündür.

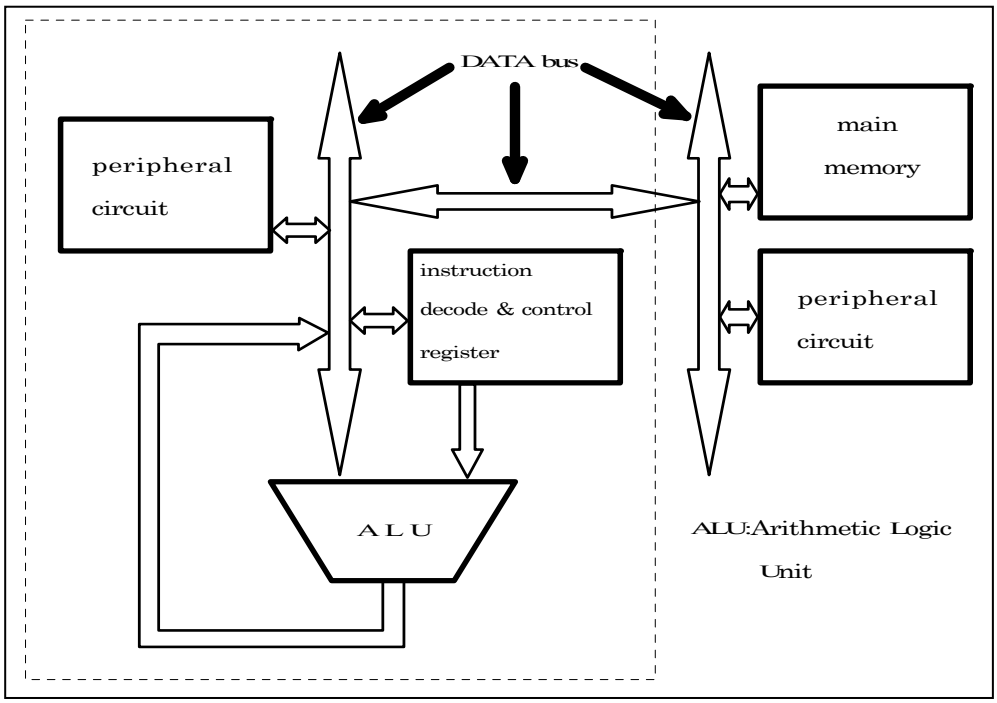
Diğer taraftan matematik işlem yapma yeteneklerinin kısıtlı oluşu, çok çeşitli on-chip çevre birimlerine sahip olmakla birlikte bunların kapasitelerinin de sınırlı olması nedeniyle bir mikroişlemcinin kullanıldığı (bir kişisel bilgisayar gibi) yerler için uygun bir seçenek oluşturmazlar.

Sonuç olarak mikroişlemciler ve mikro denetleyiciler temelde aynı alt yapı çalışma mantığına sahip olmakla birlikte kullanım yeri ve amacına göre iki ayrı grup ürün olarak değerlendirilebilir.

Mikroişlemciler ve mikro denetleyiciler günlük hayatta kullanılan sayısız cihaz ve sistemin içinde yer almakta olup bu ürünleri kullanarak gerçekleştirilebilecek uygulamalar insanın hayal gücü ile sınırlıdır.

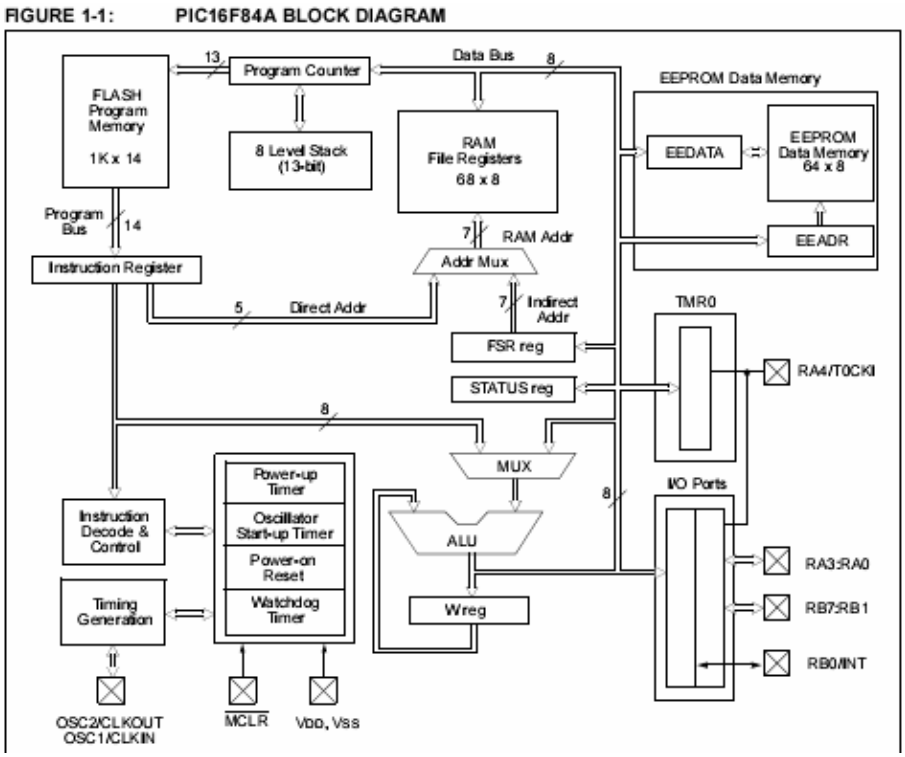
Genel mikro denetleyicilerin yapısı “Von Neumann mimarisi” olarak adlandırılır. Bilgi çıkışları birlikte bir bellektedir ve ALU bir veri bus ile bellek arasındadır (Şekil 1.6). Alışılagelmiş mikro denetleyiciler bu mimari yapıdadır. (Sinyal hattı Bus olarak adlandırılır. 8-bitlik bilgisayar veri bus, 8 sinyal hattına sahiptir).

PIC Harvard mimarisine sahiptir ve veri belleği ile program belleği vardır (Şekil 1.8). Data belleği bilgi yoluna (bus), program belleği ise program yoluna (bus) sahiptir. Bu iki yol birbirinden tamamen bağımsızdır. İçyapısı basittir ve işletimi hızlıdır. Program belleği bağımsız olduğundan dolayı kapladığı alan da bağımsızdır ve bu tanımlar bir tek kelimeyle ifade edilebilir.



Şekil 1.7: Von neumann mimarisi

FIGURE 1-1: PIC16F84A BLOCK DIAGRAM



Şekil 1.8: PIC16F84'ün yapısı

1.1.7. Hafızası ve Fonksiyonları

PIC Mikro denetleyicilerde farklı amaçlarla kullanılan bellek çeşitleri mevcuttur (Program belliği, EEPROM bellek, RAM bellek gibi). Mikro denetleyicinin ana hafızası olarak genellikle IC hafıza(RAM veya ROM) kullanılmaktadır.

1.1.7.1 ROM (Sadece okunabilir bellek)

Farklı özellikte program belleği bulunan PIC'ler microchip firması tarafından piyasaya sürülmektedir. Bunlar:

- Silinebilir ve programlanabilir bellek (Erasable PROgrammable Memory-EPROM).
- Elektriksel olarak silinebilir ve programlanabilir bellek (Electrically Erasable PROgrammable Memory-EEPROM). FLASH bellek olarak da adlandırılır.
- Sadece okunabilir bellek (Read-Only Memory-ROM).

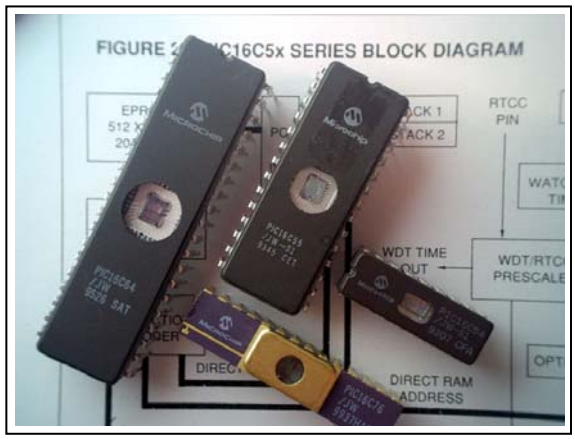
Her bir bellek tipinin kullanılacağı uygulamaya göre avantajları ve dezavantajları vardır. Bu avantajlar, fiyat, hız, defalarca kullanmaya yatkınlık gibi faktörlerdir.

EPROM bellek hücrelerine elektrik sinyali uygulayarak kayıt yapılır. EPROM üzerindeki enerji kesilse bile bu program bellekte kalır. Ancak silip yeniden başka bir program yazmak için ultra-violet ışını altında belirli bir süre tutmak gerekir. Bu işlemler EPROM silici denilen özel aygıtlarla yapılır. EPROM bellekli PIC'ler iki farklı ambalajlı olarak bulunmaktadır:

- Seramik ambalajlı ve cam pencereli olan tip, silinebilir olan tiptir.
- Plastik ambalajlı ve penceresiz olan tipler ise silinemez (OTP) tiptir.

Seramik ambalajlı ve pencereli olan bellek içerisindeki programın silinmemesi için pencere üzerine ışık geçirmeyen bir bant yapıştırılır. Ultra-Violet ışığı ile silinmesi istenildiğinde bu pencere açılır ve silici aygıt içerisinde belirli bir süre bekletilir. Plastik ambalajlı EPROM'lar ise programlandıktan sonra silinmesi mümkün değildir ve fiyatı silinebilen tipe göre oldukça ucuzdur. Silinemeyen tipe TP (One Time Programmable - Bir defa programlanabilir) olarak adlandırılır.

EEPROM belleği bulunan bir PIC içerisine program yazmak için PIC programlayıcı vasıtasıyla elektriksel sinyal gönderilir. EEPROM üzerindeki enerji kesilse bile bu program bellekte kalır. Programı silmek veya farklı yeni bir program yazmak istendiğinde PIC programlayıcıdan elektriksel sinyal gönderilir. Bu tip belleğe sahip olan PIC'ler genellikle uygulama geliştirme amacıyla kullanılırlar. Microchip bu tip belleğe çoğu zaman FLASH bellek olarak da adlandırmaktadır. Fiyatları silinemeyen tiplere göre biraz pahalıdır. Bellek erişim hızları ise EPROM ve ROM'lara göre daha yavaştır. PIC16C84 ve PIC16F84'ler bu tip program belleğine sahiptir.



Şekil 1.9: Değişik PIC görünüşleri

ROM program belleğine sahip PIC'lerin programları fabrikasyon olarak yazılırlar. EPROM ve EEPROM eş değerlerine nazaran fiyatları oldukça düşüktür. Ancak fiyatının düşüklüğünden dolayı gelen avantaj bazen çok pahalıya da mal olabilir. ROM bellekli PIC programlarının fabrikasyon olarak yazılması nedeniyle PIC'in elde edilme süresi uzundur.

Programda oluşabilecek bir hatanın PIC'e program yazıldıktan sonra tespit edilmesi, eldeki tüm PIC'lerin atılmasına da neden olabilir. Bu tip PlC'ler çok miktarda üretilecek bir ürünün maliyetini düşürmek amacıyla seçilir. Program hataları giderilemediği için uygulama geliştirmek için uygun değildir. Microchip, ROM program bellekli PIC'lere parça numarası verirken "CR"(PIC16CR62, PIC16CR84 gibi) harfleri kullanılır.

1.1.7.2. RAM (Rastgele Erişimli Bellek)

Ram kullanıcıların belleği özgürce okuyup yazabilmesi için bir IC belleğidir. Fakat elektrik kaynağı kesildiğinde bütün bilgiler bellekten silinir. Bu nedenle RAM aritmetik işlemlerin sonucunu geçici süreler içerisinde bellekte depolama işlemi yapar. İki tür RAM bellekten bahsetmek mümkündür. Birincisi Statik RAM, kullanışı basit ve kolaydır. Ancak pahalıdır. İkincisi ise Dinamik RAM, kullanışı zordur ancak pahalı değildir.

1.1.8. PIC Mikro Denetleyici Çeşitleri

Microchip, ürettiği mikro denetleyicileri aile diye adlandırılan 4 farklı gruba ayırarak isimlendirmiştir. Bahsedilen bu ailelere isim verilirken kelime boyu(Word lengt) dikkate alınmıştır. Acaba kelime boyu nedir? Mikroişlemciler veya mikro denetleyiciler kendi

içlerindeki dâhili veri saklama alanları olan registerleri arasındaki veri alış verişini farklı sayıdaki bitlerle yaparlar. Örneğin, 8051 mikro denetleyicilerde yonga içersindeki veri alış verişini 8~16 bit ile yaparken yeni çıkan Pentium işlemcilerde 64 bitlik dâhili veri yolları ile iletişim kurarlar. Bir CPU ya da MCU'nun dâhili veri yolu uzunluğuna kelime boyu denir.

Microchip, PIC'leri 12/14/16 bitlik kelime boylarında üretmektedir ve buna göre aşağıdaki aile isimlerini vermektedir.

PIC16C5XX ailesi 12-bit kelime boyu,
PIC16CXXX ailesi 14-bit kelime boyu,
PIC17CXXX ailesi 16-bit kelime boyu,
PIC12CXXX ailesi 12-bit/14-bit kelime boyuna sahiptir.

Bir CPU veya MCU'nun chip dışındaki harici ünitelerle veri alış verişini kaç bit ile yapıyorsa buna veri yolu bit sayısı denir. PIC'ler farklı kelime boylarında üretilmelerine rağmen harici veri yolu tüm PIC ailesinde 8-bittir. Yani bir PIC, I/O portu aracılığı ile çevresel ünitelerle veri alış verişi yaparken 8-bitlik veri yolu kullanır.

PIC programlayıcıları program kodlarını yazarken bir komutun kaç bitlik bir kelime boyundan oluştuğu ile pek fazla ilgilenmezler. Seçilen bir chip'i programlarken uyulması gereken kuralları ve o chiple ilgili özelliklerin bilinmesi yeterlidir. Bu özellikler PIC bellek miktarı, I/O portu sayısı, A/D dönüştürücüye sahip olup olmadığı, kesme (interrupt) fonksiyonlarının bulunup bulunmadığı, bellek tipinin ne olduğu (Flash, EPROM, EEPROM vb.) gibi bilgilerdir. Bu özelliklerinin en son değişikliklerini içeren güncel ve tam bir listesine microchip'in kataloglarından ulaşmak mümkündür. Aşağıdaki tablo da Nisan 1997 tarihine kadar üretilen PIC'lerin listesi verilmiştir. Kataloglardan alınan bu tablo orijinali bozulmadan özellikle İngilizce olarak verilmiştir. Çünkü bu gibi bilgilere ulaşmak istenildiğinde Türkçe kataloglar bulmak mümkün değildir.

Tablodaki bazı İngilizce kelimelerin anlamları aşağıda verilmiştir.

- **Family** (Aile): PIC'ler program belleğinin kelime genişliğine göre sınıflara ayrılmıştır. Bu sınıflarda her birine bir aile denir. Örneğin PIC17CXXX ailesi 16 bit kelime genişliğine sahipken PIC16CXXX ailesi 14 bittir.
- **Architectural Features** (Mimari Özellikler):Kelime genişliği, clock frekansı, interrupt özellikleri gibi bilgilerin verildiği sütun.
- **Name** (İsim): Bir aile içinde farklı özellikte üretilenlerin isimleri.
- **Technology** (Teknoloji): Program ve veri belleğinin tipini belirten sütun.
- **Products** (Ürün): PIC'in anma adı.

1996 Product Selection Guide

Family		Architectural Features	Name	Technology	Products
PIC17CXXX	8-bit High-Performance MCU Family	• 16-bit wide instruction set • Internal/external vectored • DC - 25 MHz clock speed • 160 ns instruction cycle (@ 25 MHz) • Hardware multiply	PIC17C4X	OTP program memory, Digital only	PIC17C42A, PIC17C43, PIC17C44
			PIC17CR4X	ROM program memory, Digital only	PIC17CR42, PIC17CR43
			PIC17C75X	OTP program memory with mixed-signal functions	PIC17C756 (Planned)
			PIC14CXXX	OTP program memory with A/D and D/A functions	PIC14C000
PIC16CXXX	8-bit Mid-Range MCU Family	• 14-bit wide instruction set • Internal/external interrupts • DC - 20 MHz clock speed (Note 1) • 200 ns instruction cycle (@ 20 MHz)	PIC16C55X	OTP program memory. Digital only	PIC16C554, PIC16C556, PIC16C558
			PIC16C6X	OTP program memory, Digital only	PIC16C62, PIC16C62A, PIC16C63, PIC16C64, PIC16C64A, PIC16C65, PIC16C65A
			PIC16CR6X	ROM program memory, Digital only	PIC16CR62, PIC16CR63, PIC16CR64, PIC16CR65
			PIC16C62X	OTP program memory with comparators	PIC16C620, PIC16C621 PIC16C622
			PIC16C7X	OTP program memory with analog functions (i.e. A/D)	PIC16C710, PIC16C71, PIC16C711, PIC16C715, PIC16C72, PIC16C73, PIC16C73A, PIC16C74, PIC16C74A
			PIC16F8X	Flash program and EEPROM data memory	PIC16C84 PIC16F83, PIC16F84
			PIC16CR8X	ROM program and EEPROM data memory	PIC16CR83, PIC16CR84
			PIC16C9XX	OTP program memory, LCD driver	PIC16C923, PIC16C924
PIC16C5X	8-bit Base-Line MCU Family	• 12-bit wide instruction set • DC - 20 MHz clock speed • 200 ns instruction cycle (@ 20 MHz)	PIC16C5X PIC16C5XA	OTP program memory, digital only	PIC16C52, PIC16C54, PIC16C54A, PIC16C55, PIC16C56, PIC16C57, PIC16C58A
			PIC16CR5X PIC16CR5XA	ROM program memory, digital only	PIC16CR54A, PIC16CR57 PIC16CR58A
XX		• 12-bit wide instruction set			

Tablo 1.1: Ürün seçim rehberi

PIC FAMILY	Architecture	Program Memory Type	KBytes	KWords	Data EEP ROM	RAM	I/O	ADC Channels	ADC Bit	ADC Sample Rate (ksps)	Timers/WDT	Interface	Max. Speed MHz	Int Oscillator (MHz)	ICSP™	ICD - # of Breakpoints	CCP / ECCP	PWM Channels	PWM bits	Parallel Port	Nanowatt	Vdd Min.	Vdd Max.	Pin Count	Packages
PIC12F508	8-bit	Standard Flash	0.75	0.50	0	25	6	0			1-8bit 0-16bit Yes		4	4	Yes	1	0/0			No	No	2	5,5	8	8/DFN 8/MSOP 8/PDIP 8/SOIC 150mil 40/CERDIP 600mil
PIC16F628A	8-bit	Standard Flash	3,5	2	128	224	16	0			2-8bit 1-16bit Yes	AUSART	20	4	Yes	1	1/0			No	Yes	2,00	5,50	18	18/PDIP 18/SOIC 300mil 20/SSOP 208mil 28/QFN
PIC16F84A	8-bit	Enhanced Flash	1,75	1	64	68	13	0			1-8bit 0-16bit Yes		20	0	Yes	0	0/0			No	No	2,00	5,50	18	18/PDIP 18/SOIC 300mil 20/SSOP 208mil
PIC16F877A	8-bit	Enhanced Flash	14,0	8	256	368	33	8	10-Bit	30	2-8bit 1-16bit Yes	AUSART MI²C Compatible/SPI	20	0	Yes	1	2/0	2	10	PSP	No	2,00	5,50	44	40/PDIP 40/PDIP 600mil 44/PLCC 44/QFN 44/TQFP
PIC16HV540	8-bit	OTP	0.75	.5	0	25	12	0			1-8bit 0-16bit Yes		20	0	No	No	0/0			No	No	3,50	13,50	18	18/CERDIP 18/PDIP 18/SOIC 300mil 20/SSOP 208mil
PIC18F6622	8-bit	Flash	64	32	1024	3936	54	12	10-Bit	100	2-8bit 3-16bit Yes	2 EUSART 2 MI²C Compatible/SPI	40	8	Yes	3	2/3	5	10	PSP	Yes	2	5,50	64	64/TQFP
PIC18F4585	8-bit	Enhanced Flash	48	24	1024	3328	36	11	10-Bit	100	1-8bit 3-16bit Yes	EUSART MI²C Compatible/SPI CAN 2.0B	40	8	Yes	3	1/1	2	10	No	Yes	2	5,50	44	40/PDIP 600mil 44/QFN 44/TQFP

Tablo 1.2: Bazı PIC mikro denetleyicilerin özellikleri

1.1.9. PIC Programlamak İçin İhtiyaç Duyulanlar

Pic programlayıcının geliştirilmesi şekil 1.10'de gösterilmiştir. PIC 16F84 mikro denetleyicisini programlamasını ve uygulamaların nasıl yapılacağını öğrenmek için bilinmesi ve sahip olunması gereken donanımlar aşağıda listelenmiştir:

1.1.9.1. Kişisel Bilgisayar

OS (İşletim sistemi) :Windows 95, 98, 2000, Me, XP,….
Özellikleri: Pentium 75MHz ve üstü, paralel port veya seri port veya USB portlardan birine sahip olmak.

1.1.9.2. Bir Metin Editörüne Sahip Olmak ve Kullanabilmek

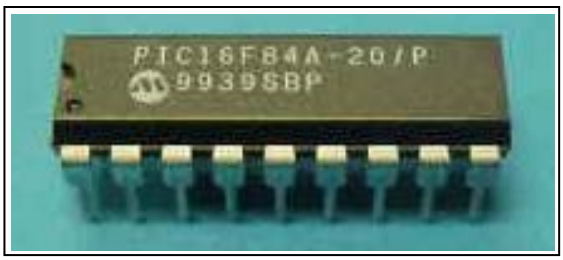
Metin editörü olarak txt uzantılı editörler kullanılabilir ya da PIC programlarını yazabilmek için **MPLAB-IDE** yazılımları kullanılabilir. Assembly dilinde yazılacak olan program kodlarının kayıt edilebileceği metin editörü "**.asm**" uzantılı olarak kayıt edilmelidir. **MPLAB** programı, Microchip Technology Co. (United States) firması tarafından geliştirilmiştir.

1.1.9.3. PIC Assembler Programına Sahip Olmak

Bir metin editöründe yazılmış olan pic assembly program kodları, İngilizce'deki bazı kelimelerin baş harflerinin birleştirilmesi ya da kelimelerin kısaltmalarından oluşur. Aynı zamanda sayı sistemlerinin ifadeleri de bulunur. Bu kodlamaların pic mikro denetleyicisi tarafından algılanması mümkün değildir. Bu nedenle bu program kodlarının PIC'in anlayabileceği makine diline dönüştürülmesi gerekir. Assembly dilinde yazılmış olan program kodlarını PIC'in anlayabileceği makine diline çeviren bir programa ihtiyacımız vardır. **MPASM** isimli program hem DOS hem de Windows altında, PIC Assembly dilinde yazılmış program kodlarını makine diline dönüştüren sürümleri bulunmaktadır. Microchip firması tarafından geliştirilen MPLAB programı içersinde hem bir metin editörü hem de MPASM assembler programı bulunmaktadır.

1.1.9.4. PIC Mikro Denetleyicisi (Bu Kitaptaki Programlar İçin Pic16f84)

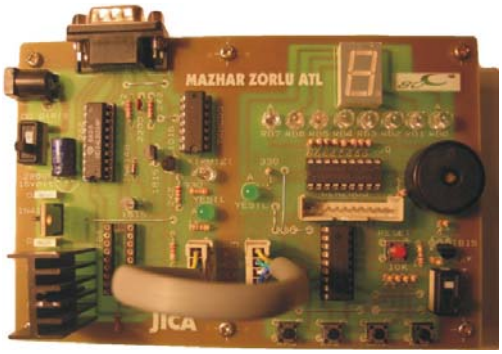
Şekil 1.10 da bu kitapda ve uygulamalarımızda bundan sonra kullancağımız PIC 16F84A entegresi resmi bulunmaktadır.



Şekil 1.10: PIC16F84

1.1.9.5. PIC Programlayıcı Donanımına Sahip Olmak

Makine diline çevrilmiş olan "**.hex**" uzantılı program kodlarının PIC içerisine yazdırılabilmesi için bir elektronik donanıma ihtiyaç vardır. Bu donanım PIC programlayıcı olarak isimlendirilir. Bizim PIC programlayıcı elektronik devremiz, PIC uygulama devresi ile birleştirilmiştir ve bu elektronik devrenin tümüne PIC kartı adını vermekteyiz. Bahsedilen PIC kartlarının aynısını ya da benzerini piyasada bulmak mümkündür.



Şekil 1.11: PIC16F84 yazıcı ve eğitim seti

1.1.9.6. PIC Programlayıcı Yazılımına Sahip Olmak

MPASM assemler programı ile derlenerek "**.hex**" uzantılı olarak oluşturulan PIC'in anlayabileceği makine dilindeki program kodlarının PC'den alınıp PIC içersine yazdırılabilmesi için gerekli olan yazılımdır. Bu programı çalıştırdığınızda bazı ayarlamalar yapmanız gerekmektedir. Bu ayarlamalar öğrenme faaliyeti–4 de açıklanmıştır.

1.2. Sayı Sistemleri

Mikro deneteleyicilerin programlanması ve işlemlerin yapılması için sayı sistemleri önemli bir konudur. Matematik dersinden de hatırlayacağımız sayı sistemlerini kısaca inceleyelim.

1.2.1. Sayıların Tipi

Microdenetleyiciler binary, decimal ve hexedesimal sayı sistemlerini kullanırlar. Binary sayı sistemi mikro denetleyicilerin anlayabildiği bir sayı kod sistemidir. Desimal sayı sistemi bize yabancı olmayan bir sayı sistemidir. Hexedesimal sayı sistemi ise bizim binary sayı sistemimizi kolayca anlamamıza yardımcı olan bir sistemdir. Bunun yanı sıra kullanıcının (programlayıcının) bu üç sayı sistemi arasındaki ilişkiyi çok iyi bilmesi gerekir ve bunlar arasında dönüştürmeleri kolayca yapabilme bilgisine sahip olmaları gerekmektedir.

Desimal sayılar	Binary sayılar	Hexadecimal sayılar
0	0000	0
1	0001	1
2	0010	2
3	0011	3
4	0100	4
5	0101	5
6	0110	6
7	0111	7
8	1000	8
9	1001	9
10	1010	A
11	1011	B
12	1100	C
13	1101	D
14	1110	E
15	1111	F

Tablo 1. 3. Decimal, binary ve hexedecimal karşılıkları tablosu

Binary ve decimal sayı sitemlerini öğrendikten sonra bunların birbirlerine dönüşüm metotlarını öğrenmemiz gerekmektedir.

Kelime Boyu	Maksimum	Kısaltma	Kelime Boyu	Maksimum	Kısaltma
1	2		17	131072	128K
2	4		18	262144	256K
3	8		19	524288	512K
4	16		20	1048576	1M
5	32		21	2097152	2M
6	64		22	4194304	4M
7	128		23	8388608	8M
8	256		24	16777216	16M
9	512		25	33554432	32M
10	1024	1K	26	67108864	64M
11	2048	2K	27	13217728	128M
12	4096	4K	28	268435456	256M
13	8192	8K	29	536870912	512M
14	16384	16K	30	1073741824	1G
15	32768	32K	31	2147483648	2G
16	65536	64K	32	4294967296	4G

Tablo 1.4: Dijital numara ve maksimum değerleri

1.2.2. Binary Dijit

0 ve 1'lerden meydana gelen ikili bir sayı sistemindeki her bir rakamı ifade eder. Diğer bir değişle her sayı bir dijit (bit) olarak tanımlanır. PIC içerisinde her bir desimal sayı 8 bit ile ifade edilir. Gerçekte desimal sayı sistemlerinde 9 ve 1 toplandığında 10 elde edilir ancak Binary sayı sistemlerinde 1 ve 1 toplandığında 2 olmaz, 1 ve 0 (10B) olur.

Mikro denetleyicilerin sinyal seviyeleri iki çeşittir. Burada "1" yüksek seviyeyi, "0" ise düşük seviyeyi belirtir. Mikrodenetleyicler 1 ve 0'lardan meydana gelen makine dili adını verdiğimiz elektriğin varlığından ve yokluğundan anlar. Makine dilinin kullanıcılar için anlaşılması ve program yazılması oldukça zordur.

Binary sayı sistemi kolayca hexedecimal sayı sistemine de dönüştürülebilmektedir. Sonunda B harfi olan tüm sayılar Binary sayı anlamını taşımaktadır. Örneğin, 21 olan decimal sayısı Binary de 10101B olarak veya B'10101' ifade edilir.

$12 = 1x2^4 + 0x2^3 + 1x2^2 + 0x2^1 + 1x2^0 = 10101B$

1.2.3. Hexadecimal Dijit

Hexadecimal sayı sistemi 16 tabanlı bir sayı sistemi olup 0~15 arasındaki sayılarla ifade edilir. 0~15 arasında 16 tane rakam ve harf ile gösterilir. 0'dan 9'a kadar rakam ile 9'dan sonra 15 dahil A - F arasıdaki İngilizce harfler ile gösterilir. Örneğin 21 olan decimal sayısı hexadecimal de H '15 ' veya 0x15 ya da (15h) olarak gösterilir. Hexadecimal sayının Binary ye ya da Binary sayının hezadecimal sayıya dönüştürülmesi çok kolaydır.

1.2.4. Binary, Desimal ve Hexadesimal Sayıların Dönüşümü

Sayı sistemleri ve birbirlerine dönüşümlerini daha önce 10. sınıf derslerinde öğretilmişti. Bu modülde sayı sistemleri ve dönüşümleri hatırlatıcı bilgi olarak verilecektir.

1.2.4.1. Decimal Sayıların Binary Sayılara Dönüşümü

Decimal sayı sürekli ikiye bölünerek Binary sayı elde edilir. İkiye bölme sonucunda kalanlar tersten yazıldığı zaman Binary sayı elde edilir.

Örnek 1.1:

19 decimal sayısını Binary sayıya dönüştürünüz.

Cevap 1.1:

19 ÷ 2 = 9 → kalan 1 → **LSB** (En Düşük Değerliğe sahip Bit)
9 ÷ 2 = 4 → kalan 1
4 ÷ 2 = 2 → kalan 0
2 ÷ 2 = 1 → kalan 0
1 ÷ 2 = 0 → kalan 1 → **MSB** (En Yüksek Değerliğe sahip Bit)

Decimal sayı sisteminden hexedecimal sayı sistemine dönüştürme işlemi de aynı yolla yapılabilir.

1.2.4.2. Binary Sayıların Decimal Sayılara Dönüşümü

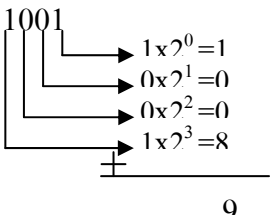
Binary sayıların decimal sayılara dönüşümü birkaç yöntemle yapılabilir. Aslında hepsi aynı yönteme bağlıdır (0.bit 2^0 1.bit 2^1 2.bit 2^2…). Bir örnekle açıklamak daha yerinde olacaktır.

Örnek 1.2:

1001 Binary sayısını decimal sayı sistemine dönüştürünüz.

Cevap 1.2:

Yöntem 1:



Yöntem 2:

3.bit	2.bit	1.bit	0.bit	Konum
2^3	2^2	2^1	2^0	Bitlerin decimal değeri
8	4	2	1	
1	0	0	1	Binary değer
8	0	0	1	Sonuç
		9		Genel sonuç

Yöntem 3:

$(1001)_2 = 1x2^3 + 0x2^2 + 0x2^1 + 1x2^0$

$= 8 + 0 + 0 + 1$

$= 9$

1.2.4.3. Decimal Sayıların Hexedecimal Sayılara Dönüşümü

Decimal sayıların Binary sayılara dönüşümü 2 sayısına bölünerek nasıl elde ediliyorsa decimal sayıların da hexedecimal sayılara dönüşümü aynı metotla fakat bu sefer 16'ya bölünerek elde edilir.

Örnek 1.3:

667 decimal sayısını hexedecimal sayı sistemine dönüştürünüz.

Cevap 1.3:

$677 \div 16 = 42 \rightarrow$ kalan 5

$42 \div 16 = 2 \rightarrow$ kalan 10

$2 \div 16 = 0 \rightarrow$ kalan 2

$(677)_{10} = (2A5)_{16}$

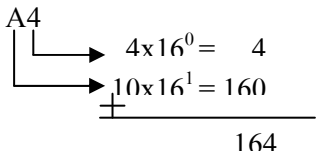
1.2.4.4. Hexadecimal Sayıların Decimal Sayılara Dönüşümü

Yine aynı şekilde Binary sayıların decimal sayılara dönüşümü hangi metotla yapılıyorsa hexedecimal sayıların da decimal sayılara dönüşümü aynı yöntemle yapılır fakat her bir sayı 16 sayının sıfırdan başlayan kuvvetleri ile çarpılır.

Örnek 1.4:

A4H sayısını decimal sayı sistemine dönüştürünüz.

Cevap 1.4:



1.2.4.5. Binary Sayıların Hexadecimal Sayılara Dönüşümü

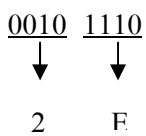
Binary sayıların hexedecimal sayı sistemine dönüştürebilmek için her bir Binary sayı 4'er bitlik gruplara ayrılır ve bu 4'erli grupları hexedecimal sayı sistemine dönüştürürüz.

Örnek 1.5:

$(101110)_2$ sayısını hexedecimal sayı sistemine dönüştürünüz.

Cevap 1.5:

4'erli gruplara ayırmayı LSB bitinden başlayarak yapmalıyız. Yüksek dört bitlere 4 bit düşmüyorsa "0" ilave edilir.



101110B = 2EH

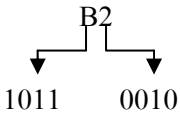
1.2.4.6. Hexadecimal Sayıların Binary Sayılara Dönüşümü

Her bir hexedecimal basamak 4'er bitlik Binary sayılarla ifade edilir. Bu Binary sayılar birleştirildiğinde hexedecimal sayı Binary biçimde yazılmış olur.

Örnek 1.6:

B2H hexedecimal sayısını Binary biçimde yazınız.

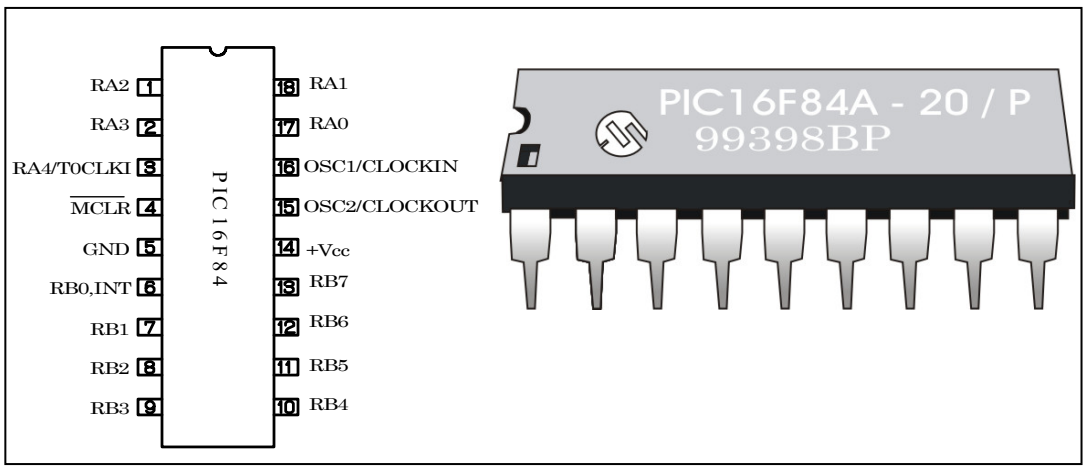
Cevap 1.6:



B2H = 10110010B

1.3. Mikro Denetleyici Yapısı

Aşağıda Pic 16F84 mikro denetleyicisinin bacak yapısı görülmektedir.



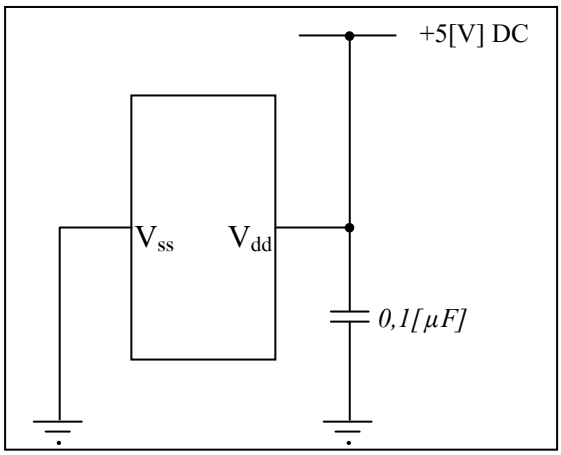
Şekil 1.12. PIC16F84'ün Pin ve Entegre Görünüşü

1.3.1. PIC16F84

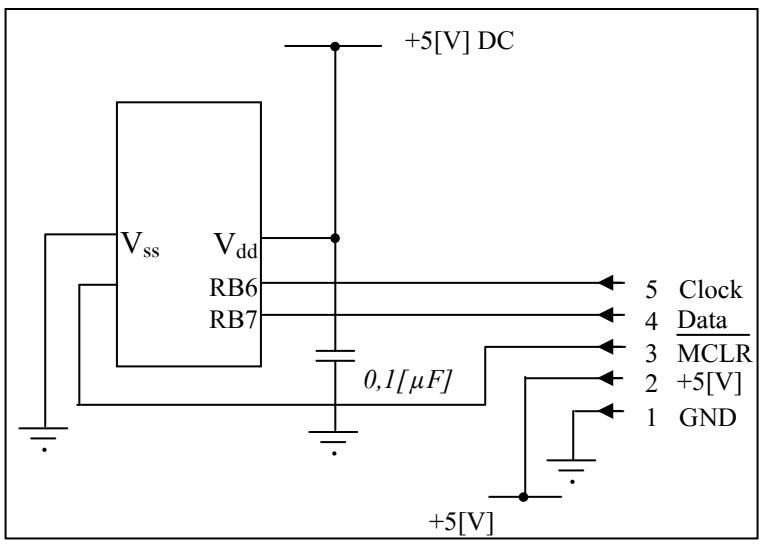
Her Bir Ucun Açıklanması:

OSC1/CLOCKIN : Osilatör girişi / External oscillator input
OSC2/CLKOUT : Osilatör girişi / OSC1 frekansının ¼ değerindeki çıkış clock ucu
MCLR (inv) : Reset girişi
RA0 – RA3 : Giriş Çıkış uçları
RA4/T0CKI : Giriş Çıkış ucu / TMR0 için clock puls giriş ucu
RB0/INT : Giriş Çıkış ucu / Dış kesmeler için giriş ucu
RB1-RB7 : Giriş Çıkış ucu
GND : Güç kaynağının eksi (—) ucu
Vcc : Güç kaynağının artı (+) ucu

Yazılan bir programı PIC'e kayıt ederken ise RB6 ucuna clock pulse, RB7 ucuna veri bilgilerini vermemiz gerekir. Ayrıca MCLR ucuna 12,5 Volt, GND ucuna (-) eksi, Vcc ucuna da +5 Voltu vermeyi unutmamalıyız.

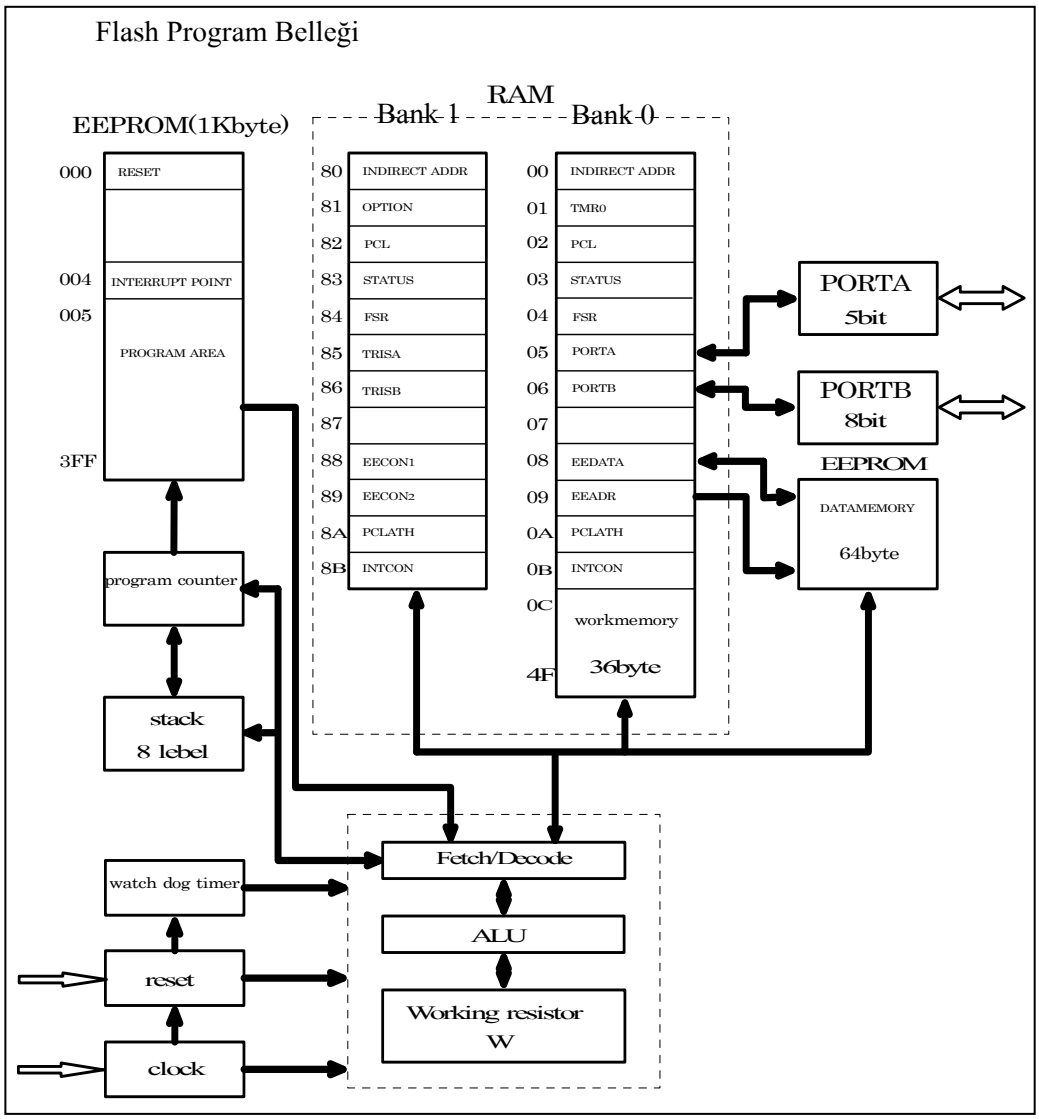


Şekil 1.13: PIC16F84'ün Besleme gerilimi bağlantısı



Şekil 1.14. PIC16F84'ün program aktarmak için kullanılan bacakları

PIC16F84 18 uçlu 1 Kbayt flaş program belleği, 68 bayt RAM bellek, 64 Bayt EEPROM belleğe sahiptir. PIC16F84'e program elektrik sinyalleri ile kolayca tekrar tekrar yazılabilir veya silinebilir. Ancak ROM bellek olanlarda ise tekrar yazabilmek için ultraviyole ışınlara ihtiyaç vardır.



Şekil 1.15: PIC16F84' ün blok diyagramı

1.3.2. PIC16F84' ün Yapısı

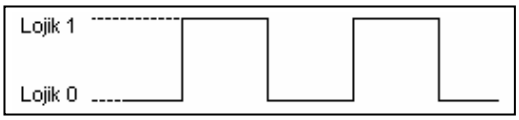
PIC RISC (**R**educed **I**ntruction **S**et **C**omputer) denilen azaltılmış komut sistemini kullanmaktadır. Bu sistem sayesinde komutlar daha sade ve daha azdır. Bir PIC'i programlamak için 35 komut kullanılır. PIC de yöntem bir komutu genellikle bir clock ile gerçekleştirmesidir. Program belleği (memory) (ROM) ve data belleği (memory) birbirinden bağımsızdır. (Harward mimarisine göre) Bu hafıza yapısı ile her iki hafızada aynı anda çalıştırılabilmekte ve böylece işletim çok daha hızlı olmaktadır.

Data belleğinin genişliği yapıya göre değişiklik gösterir (Program belleği 14 bit, veri belleği 8 bittir). Çünkü 1 kelimenin makineye tanıtımı 14 bit ile gerçekleştirilir. Örneğin

MOVLW B '01011111' komutunu makine 11000001011111 olarak tanır. Bunun 6 biti olan 110000 MOVLW komutunu ifade eder. 01011111 ise veri bölümünü tanımlar.

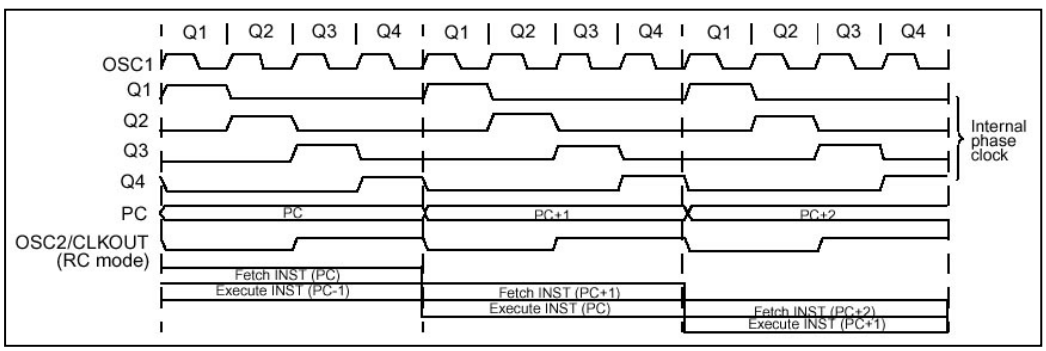
1.3.3. Clock Düzeni ve Komut Süresi

PIC'in program hafızasında bulunan komutların çalışması kare dalga (clock sinyali) ile olur.



Şekil 1.16: PIC' in Çalışabilmesi için OSC1 ucundan verilen kare dalga

OSC1 denilen 16 nolu PIC16F84 bacağı kare dalganın uygulandığı yerdir. Bacak yapısı görünüşünde (Şekil 1.12) **CLK IN** olarak ifade edilmiştir. Dışarıdan buraya uygulanacak kare dalga **OSC2/CLK OUT** dan dörde bölünmüş olarak (f/4) 15 numaralı bacaktan dışarı alınabilir. Q1, Q2, Q3, Q4 olan bu bölümler de kare dalga şeklindedir. Program sayıcı (PC), her Q bölümünde bir arttırılmakta ve komutlar program belleğinde işleme sokularak Q4'de sona ermektedir. Komutlar Q1 den Q4'e kadar çözülerek işlemin gerçekleşmesi sağlanır. Clock pulse ve bunun düzeni şekil 1.17'de görülmektedir.



Şekil 1.17: Clock / komut örneği

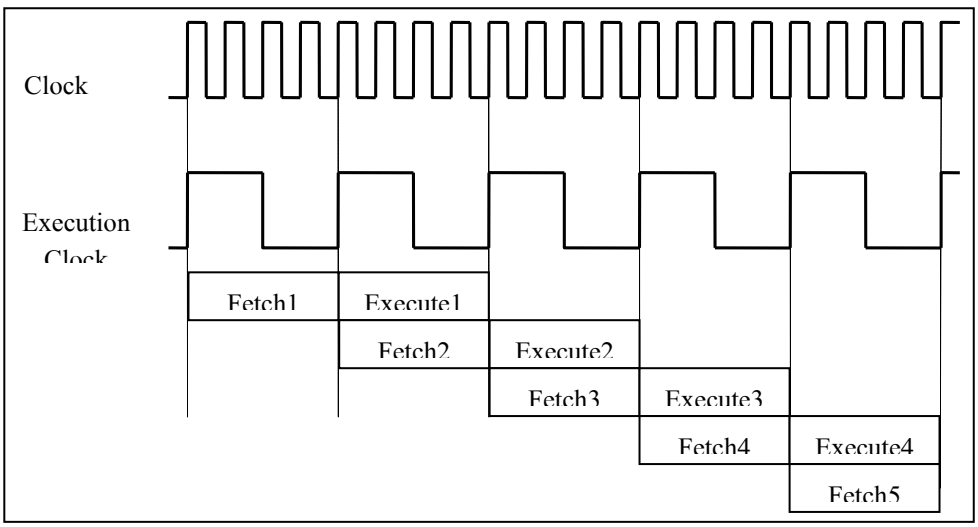
1.3.3.1. Komut Örneği ve Bilgi İletişim Kanalı

16 nu'lı bacaktan girilen osilatör sinyali, pic içinde 4'e bölündüğünü ve frekansın 1/4'nün 15 nolu bacaktan alınabileceğini söylemiştik. Bu 4'e bölünmüş saat frekansının karşılığı olan periyota "instruction cycle" yani komut süresi denir. Bu bir komutun işlenmesi için gereken zamandır. 16F84'de bu 4 Mhz'de 1 mikro saniye 10 Mhz'de ise 0,4 mikro saniyedir. Bu zaman, programlama esnasında çok önem arz eder. Bu komut sürelerinin toplamı ile zamanlar hesaplanır.

Bir komutun aktarılması ve işleyişi şu şekilde olmaktadır. Bir "Komut Döngüsü" dört Q döngüsünden (Q1, Q2, Q3 ve Q4) oluşmaktadır. Komut alımı ve gerçekleştirilmesi,

çözülüp gerçekleştirme diğer bir komut döngüsünü oluştururken bir komut döngüsünü oluşturmaktadır. Yine de bilgi iletişim hattı dolayısı ile her bir komut etkin bir şekilde tek döngüde gerçekleştirilmektedir. Eğer bir komut program sayacının değişmesine neden olursa (**GOTO**) bu durumda komutu tamamlamak için iki döngü gereklidir (Şekil 1.18).

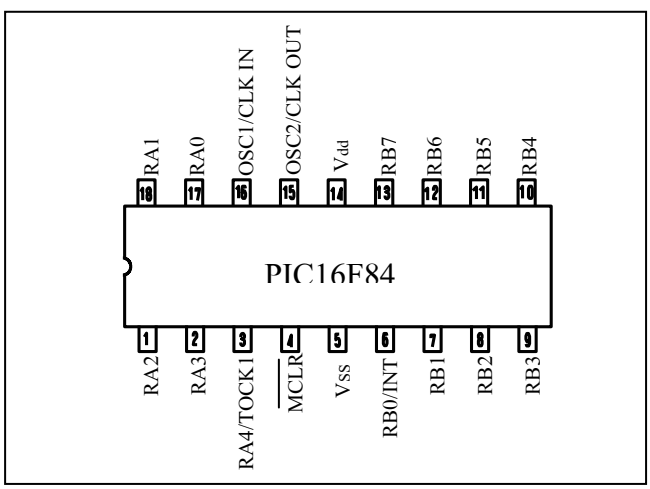
Bir alım döngüsü, Q1 dâhilinde artan Program Sayacı (PC) ile başlamaktadır. Gerçekleştirme döngüsünde, alınan komut, Q1 döngüsündeki "Komut Kaydı" içine kapatılmaktadır. Bu komut daha sonra Q2, Q3 ve Q4 döngüleri esnasında çözülür ve gerçekleştirilir. Veri hafızası Q2 esnasında okunmakta olup (bilgi okuması) ve Q4 esnasında yazılır (yazılacak hedef).



Şekil 1.18: Bilgi iletim kanalı (pipeline) yapısı

1.3.4. I / O port (input / output)

I/O portları sinyalin giriş ve çıkışlarıdır. PIC16F84 'e ait I/O portlar şekil 1.19'da görülmektedir. Buradan da anlaşılacağı gibi PIC16F84'ün en fazla uçları I/O için ayrılmış olup program kontrolü için kullanılmaktadır.



Şekil 1.19: I/O Portları

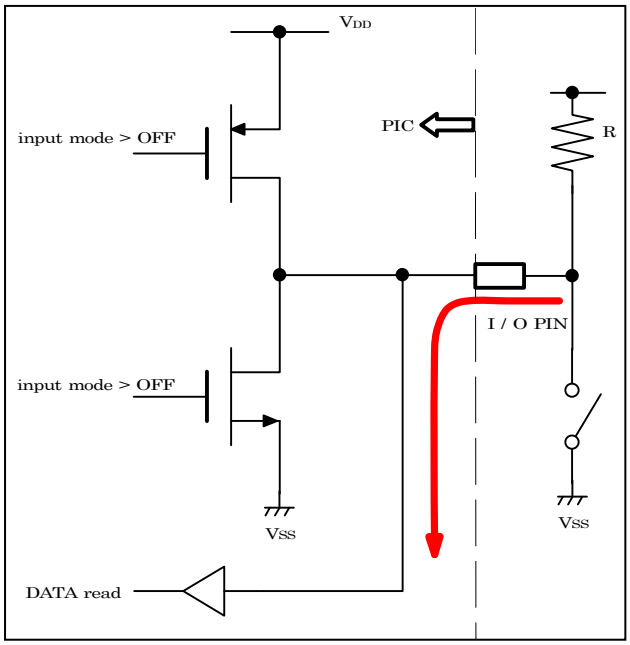
A portu 5 adettir (RA0, RA1, RA2, RA, RA4).

B portu 8 adettir (RB0, RB1, RB2, RB3, RB4, RB5, RB6, RB7).

Toplam 13 adet giriş veya çıkış olarak kullanılmak üzere uçları vardır.

1.3.4.1. Giriş İşlemi

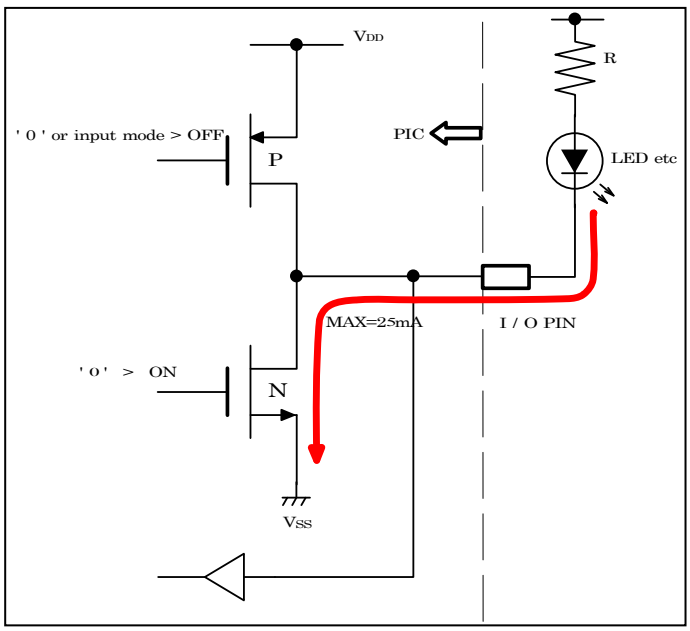
Eğer PIC giriş (input) modunda olursa FET çıkışları kapatır. Giriş sinyali tampona doğru akar (Şekil 1.20).



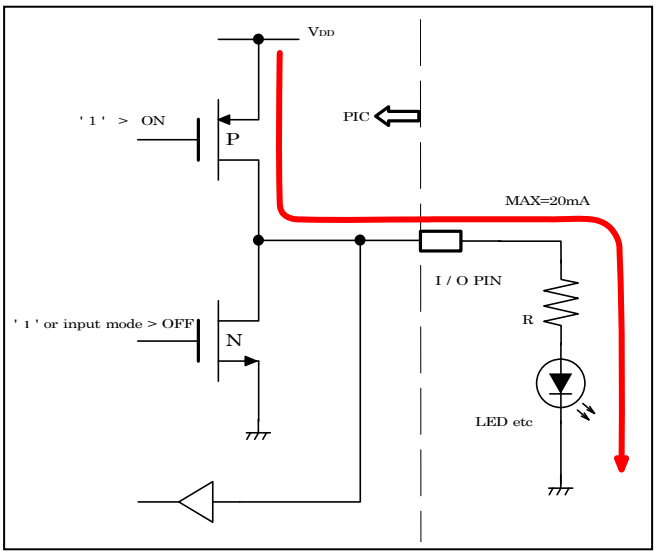
Şekil 1.20: Giriş işlemi

1.3.4.2. Çıkış İşlemi

PIC' in çıkışı FET' lidir. Eğer akım gerilim kaynağından çıkış portuna doğru ise buna SINK akım (Şekil 1.2), I/O pininden GND'ye doğru ise buna da KAYNAK (Source) akımı denir (Şekil 1.22). Kaynak akımı en fazla 20 [mA] iken sink akımı ise en fazla 25[mA] dir.



Şekil 1.21: Sink akımı



Şekil 1.22: Kaynak (source) akımı

1.3.5. Reset Devresi

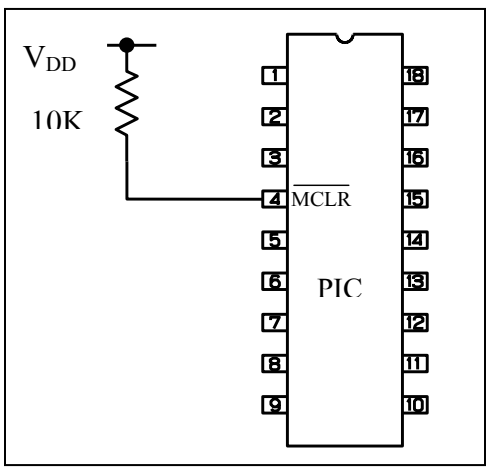
Burada Pic mikro denetleyicisinin programının baştan başlatılabilmesi için yapılması gereken Reset işlemi hakkında detaylı inceleme yapılacaktır.

1.3.5.1. Resetleme İşlemi (Power on Reset) (POR)

MCLR (**M**emory **Cl**ear) ucuna düşük gerilim (0[V]) uyguladığımızda, PIC16F84 reset edilmiş olur ve program başlangıçtaki adresine geri döner. MCLR ucu tekrar yüksek gerilim 5[V] olduğunda PIC16F84 programın çalışmasına ilk adresten itibaren devam eder. Kısaca MCLR ucu 0V olduğunda program çalışmaz sadece ilk adrese gider. Programın çalışabilmesi için MCLR ucunun tekrar 5[V] olması gerekir.

- **VDD'ye Direk Olarak Resetleme**

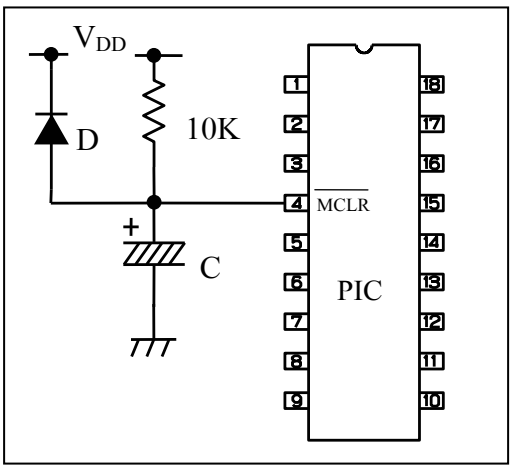
V_{DD} yüksek gerilimi bulduğunda (1,2[V] – 1,7[V]) reset çalışmaya başlar. PIC'in resetinden yararlanabilmek için MCLR ucunu direk olarak V_{DD} ye bağlayınız. Buna dirençte eklenebilir.



Şekil 1.23: V_{DD}' ye direk olarak resetleme

➢ **PIC'in Dışarıdan Resetlenmesi**

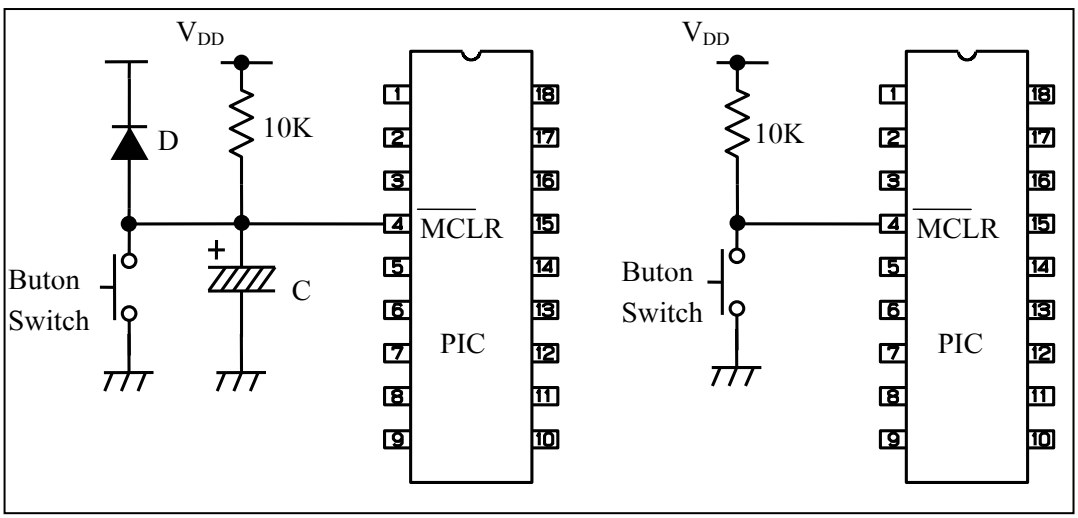
Eğer V_{DD} ile resetleme yavaş oluyor ve bunun hızlı gerçekleşmesini istiyorsak dış reset yapmamız gerekir. V_{DD} gerilimini hızlı bir şekilde 0 yapmak için LED ve kondansatör kullanmamız gerekir. LED kondansatörü hızlı bir şekilde boşaltır (deşarj) ve işlem hızlanmış olur (C =1 – *10 [µF])*.



Şekil 1.24: PIC' in dışarıdan resetlenmesi

➢ **Butonla Reset**

Şekil 1.25'de görüldüğü gibi MCLR ucunu düşük gerilime ulaştırmak için reset butonu vardır. Bu buton basılıp çekilir ve program ilk adresten itibaren çalışmaya başlar.



Şekil 1.25: Butonla reset

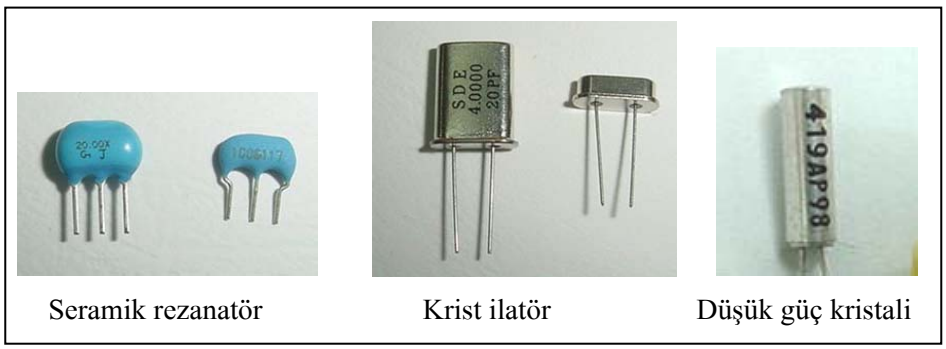
1.3.6. Osilatör Özellikleri

1.3.6.1. Osilatör Modelleri

PIC16F84 dört değişik osilatörleme ile çalışabilir. Bunlar

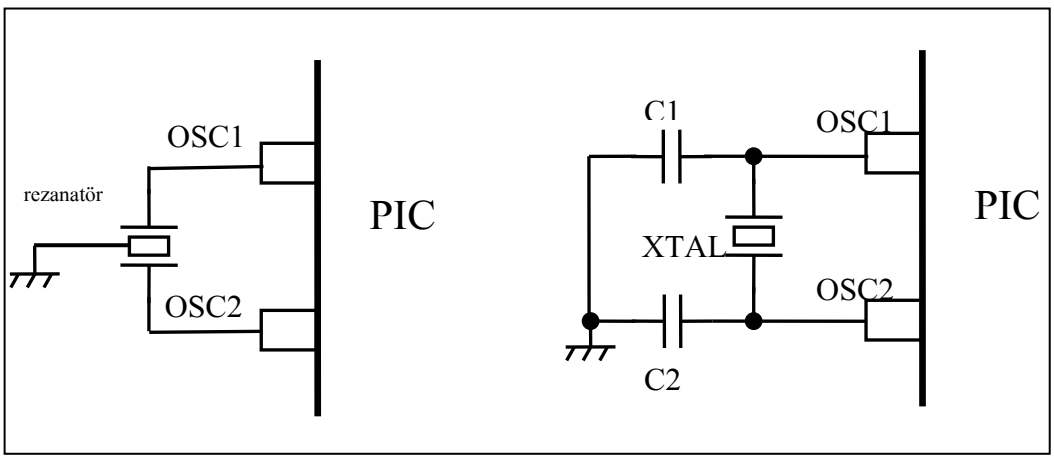
LP: Düşük güç kristal ile (Low Power Crystal)	(yaklaşık 40KHz)
XT: Kristal / Rezanatör ile(Crystal / Resonator)	(0 – 10MHz)
HS: Yüksek hız kristali / Resonatör(High Speed Crystal / Resonator)	(4 – 10MHz)
RC: Direnç / Kondansatör ile(Resistor / Capacitor)	(0 – 4MHz)

1.3.6.2. Kristal / Seramik Rezanatör İşlemi



Şekil 1.26: Kristal Çeşitleri

XT, LP veya HS modeller kristal veya seramik rezanatör ile OSC1/CLKIN ve OSC2/CLKOUT uçlarına bağlanırlar. Böylece osilatör sağlanmış olur (Şekil 1.27).



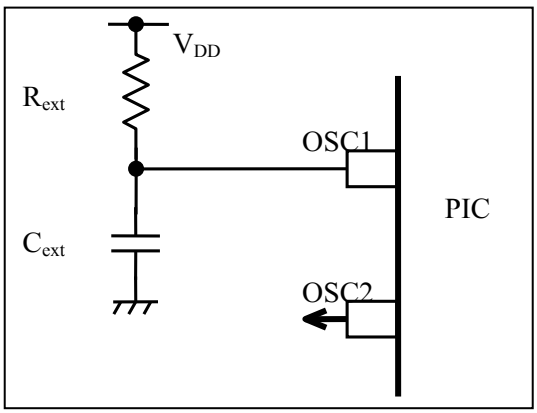
Şekil 1.27. Kristal ve rezanatör bağlantıları

Mod	Frekans	OSC1 / C1	OSC2 / C2
LP	32kHz	68-100pF	68-100pF
	200kHz	15-33pF	15-33pF
XT	100kHz	100-150pF	100-150pF
	2.0MHz	15-33pF	15-33pF
	4.0MHz	15-33pF	15-33pF
HS	4.0MHz	15-33pF	15-33pF
	10.0MHz	15-33pF	15-33pF

Tablo 1.5: Kristal osilatör için kondansatör seçimi

1.3.6.2. RC Osilatör

Zamanın çok hassas olmadığı durumlarda RC osilatör kullanılarak maliyet düşürülür. RC osilatör frekansı gerilim kaynağının özelliğine, direncin değerine, kondansatörün değerine ve işlem ortamının sıcaklığına bağlıdır. Buna ek olarak osilatör frekansı normal işlem parametrelerine göre sapmalar gösterir. Bu sapma % 20 civarındadır. Şekil 1.28'de bu osilatör tipinin PIC16F84'e nasıl bağlandığı görülmektedir. Direnç değeri 4k-ohm'un altında olan osilatör işlemlerinde osilasyon sabit olmayabilir veya tamamen durabilir. Çok yüksek değerde dirençler ise (yaklaşık 1M-ohm), gürültüye, neme ve sızmaya çok hassaslaşır. Bu nedenle direnç değerini 5 k-ohm ve 100k-ohm arasında kullanılmalıdır.



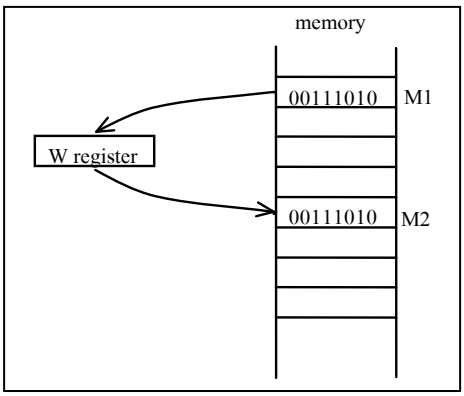
Tavsiye edilen değerler
5k < Direnç < 100k
Kondansatör > 20pF

Şekil 1.28: RC Osilatör

Her ne kadar osilatör dış bir kondansatörle çalışmıyor olsa bile biz gürültü ve sabitliliği sağlamak için 20 pF değerinin üzerindeki değerde bir kondansatörün kullanılmasını tavsiye ederiz.

1.3.7. W Yazmacı (Egister)

W (working) yazmacı bilgilerin geçici olarak depolandığı ve bilgilerin aktarılmasında kullanılan bir kısımdır. Direk olarak ulaşamayız. PIC'te yapılan tüm işlemler ve atamalar bunun üzerinden yapılmak zorundadır.

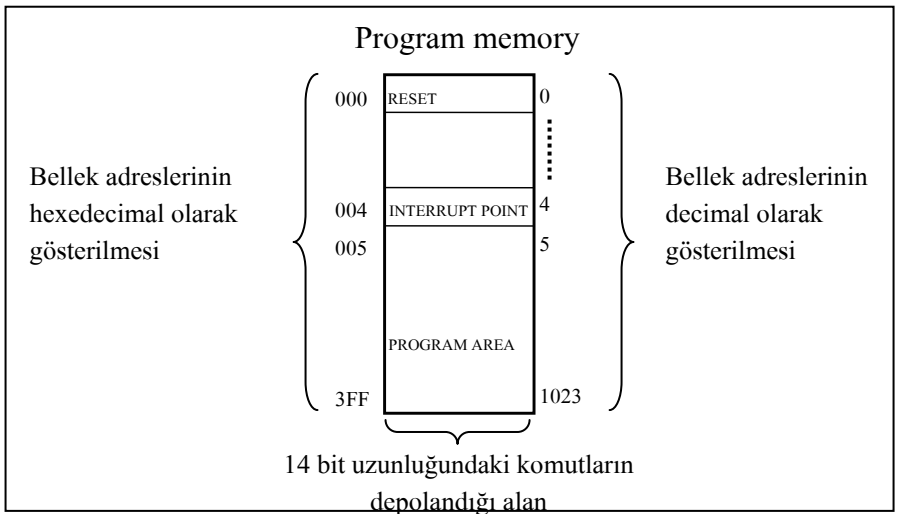


Şekil 1.29: W Yazmacı (register)

Örneğin M1'deki data bilgilerini M2 ye aktarmak istersek once M1'deki bilgileri W yazmacına (register) aktarırız daha sonrada geçici alanda bulunan (w registeri) bu bilgileri M2 kısmına aktarırız. M1'deki bilgiler M2'ye direk olarak aktarılamaz mutlaka W yazmacını kullanmak zorundayız.

1.3.8. Program Belleği (Memory)

Assembly de kullanılan komutlarla yazılmış programın yüklendiği alandır. Program belleği (memory) (ROM). Bu alan PIC16F84 de EEPROM şeklindedir. Program yazıcısı kullanarak ROM'a programımızı rahatlıkla yazabiliriz. Çünkü elektrik dalgasıyla yazılıp silinebilme özelliği vardır. Mikro denetleyici uygulayacağı komutları ve işlem sırasını bunun ilgili adreslerine bakarak uygular. İlgili adresler ise PC (Program **C**ounter) program sayıcıda saklanır. Bir PIC'te ROM belleğe program yaklaşık 1000 defa yazılabilir. (Program belleğinin genişliği 14 bittir. PIC16F84 program belleğinin 1024 (1K) alanı 000 dan 3FF kadar olan adrestedir ve 3FF de 1024 demektir (Bölüm 1'deki Tablo 1–2'de daha detaylı bilgi verilmiştir.).



Şekil 1.30: ROM belleğin haritası (program memory)

1.3.9. Veri Belleği (Data Memory)

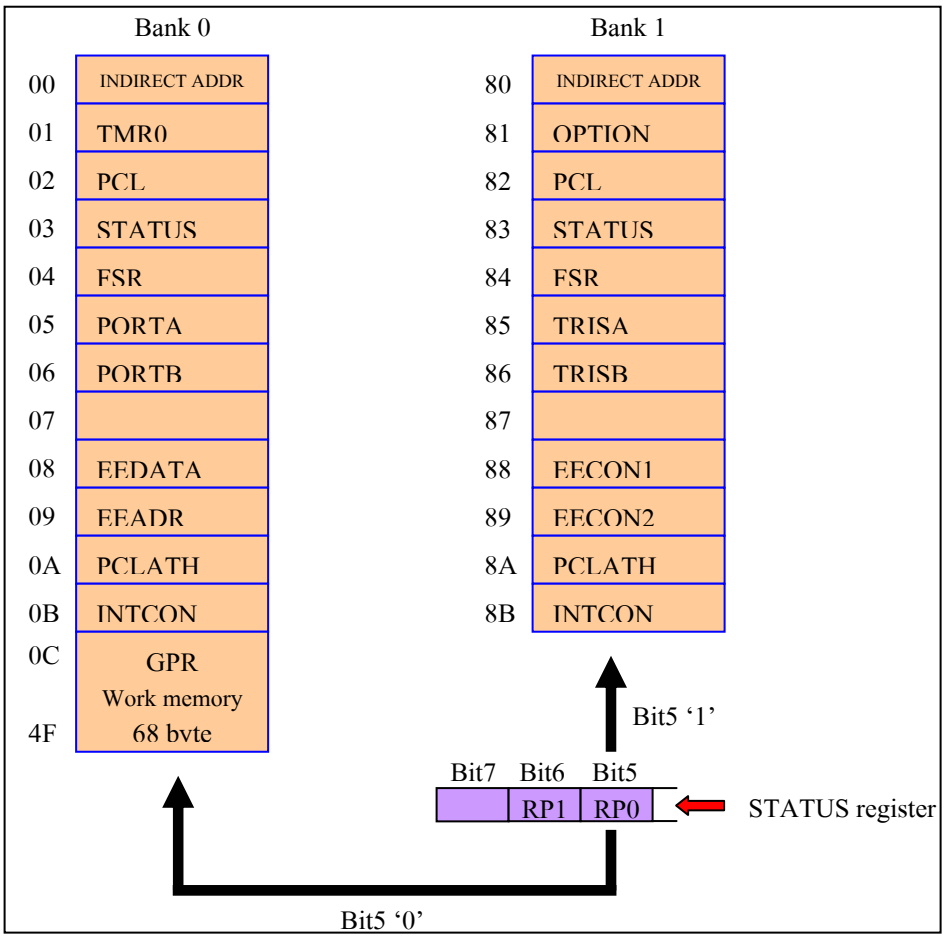
Mikro denetleyicilere bilgilerin ve programların saklandığı farklı hafıza bölgeleri bulunmaktadır. Bu hafıza bölgeleri yapısal olarak farklılık gösterebileceği gibi o hafıza bölgesine ulaşımda farklılık gösterebilir. Bu konuda detaylı olarak alt başlıklarda bilgi verilecektir.

1.3.9.1. Veri Belleği (Ram Bellek)

Veri belleği iki kısımdan meydana gelmektedir. Bunların her birine bank adı verilir. bank 0 ve bank 1 dir. Bank 1 dediğimiz ilk bölüm özel fonksiyon (**S**pecial **F**unction **R**egister) (SFR) yazmaç alanıdır. Bank 0 ise ikinci bölümdür ve buna da genel amaçlı yazmaç (**G**eneral **P**urpose **R**egister) (GPR) denir.

GPR alanı genel amaçlı RAM'ın 16 baytından daha fazlasına olanak sağlamak amacıyla bölünmüştür. SFR ise özel fonksiyonları kontrol eden kayıtlara aittir. Bunların seçimleri için kontrol bitleri gerekmektedir. İşte bu kontrol bitleri de STATUS yazmacında bulunmaktadır. Şekil 1.31'de veri belleğinin harita organizasyonu göstermektedir.

Veri belleği genel amaçlı kayıt ve özel fonksiyon kaydını içeren iki bölümü vardır. Bank0' ı seçmek için RP0 bitini (ki bu STATUS'un 5. biti oluyor) temizlemek gerekir. Aynı bitin kurulması (set) ile de BANK1 seçilmiş olur. Her iki bankın ilk onikisinin yerleşimi özel fonksiyon kaydı için ayrılmıştır. Kalanı ise statik RAM olarak genel amaçlı kayıtları yürütmektedir.



Şekil 1.31. PIC16F84'ün kayıt dosya haritası

- **Genel Amaçlı Kayıt Dosyası**

Bütün cihazlar belli bir miktar genel amaçlı kayıt alanlarına sahiptirler. Bu PIC 16F84 de 68 bayttır. Her bir GPR 8 bit genişliğindedir ve dolaylı ya da doğrudan SFR üzerinden erişilmektedir.

- **Özel Fonksiyon Kayıtları**

Özel fonksiyon kayıtları aygıtın işlemini kontrol etmek için CPU ve özel fonksiyonlar tarafından kullanılmaktadır. Bu kayıtlar statik olan RAM'dadır.

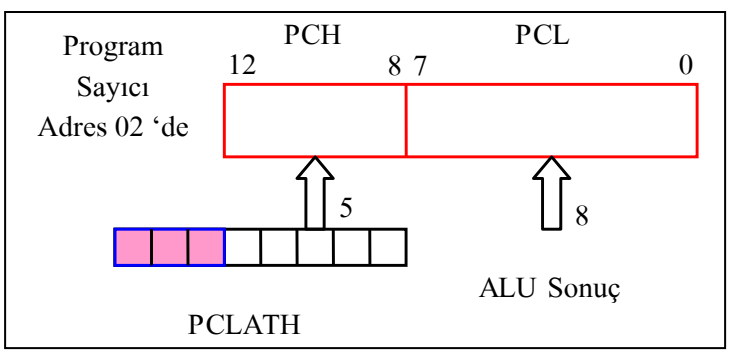
1.3.9.2. Yığın Hafıza (Stack Memory)

PIC16F84 8 derinliğinde ve 13 bit genişliğinde yığına (stack) sahiptir (Şekil 1.11). Bu yığın alanı program veya bilgi (data) yeri değildir ve direk olarak okunamaz veya yazılamazlar. CALL komutu veya kesme ile 13 bitlik (PC) program sayıcı yığının üzerine konur bu işleme "pushed " denir. Yığındaki bu adres alt programlar çalışıp bittikten sonra en sonunda bulunan 'RETLW, RETFIE' komutları ile yığından atılır bu işlemede "popped" denir.

Burada dikkat edilecek iç içe en fazla 8 altprogram veya kesme kullanabiliriz. Fazla kullandığımız takdirde yığın taşması dediğimiz (stack overflow) hatası belirir.

1.3.9.3. Program Sayıcı (Program Counter)(Adres 02h, Pclath Adres 0ah)

Program Sayıcı (**P**rogram **C**ounter) (PC) 13-bit genişliğindedir. Yürütülecek komutun program belleğindeki adresini tutar. Bu nedenle kendisi bir gösterge olarak görev yapar. Program sayıcının alt 8 bitine PCL düşük bayt denir PCL okunabilen - yazılabilen bir yazmaçtır. Üst 5 bitine ise PCH yüksek bayt denir ki bu (PC <12:8>) arasındadır ve direk olarak okunup yazılamaz. RAM bellekte H'0A' adresinde bulunan özel PCLATH (PC latch high) yazmacından okunup yazılabilirler. PCLATH'ın içeriği program sayıcının üst bitlerine transfer olur ve PC yeni bir değerle yüklenmiş olur. Bunu şekil 1.28'de görebilirsiniz. CALL, GOTO gibi komutlar PCL ye yazılırlar çünkü 256 bayttan küçüktürler. RESET durumunda ise üst bitler temizlenir.



Şekil 1.32: Program sayıcı

1.3.9.4. STATUS Yazmacı (Register) (Adresi 03h,83h)

STATUS yazmacı (register) aritmetik mantık ünitesine (**A**rithmetic **L**ogic **U**nit) ALU sahiptir. RESET durumu içeren veriler ve bank seçimi sağlayan bilgilerde mevcuttur.

b7	b6	b5	b4	b3	b2	b1	b0
R/W-0	R/W-0	R/W-0	R-1	R-1	R/W-x	R/W-x	R/W-x
IRP	RP1	RP0	TO	PD	Z	DC	C

R=Okunabilir bit W=Yazılabilir bit -n=Reset değeri

İlk değer (enerji verildiği andaki değeri) 0 0 0 1 1 x x x

bit7 **:** IRP: Bank seçme bit'i (Register Bank Select bit)

IRP bit'i PIC 16F84A'larda kullanılmaz. .IRP sıfır olarak kalmalıdır.

bit 6-5 **:** RP1:RP0: Bank seçme bit'i (Register Bank Select bit)

00 = Bank 0(00h-7Fh)

01 = Bank 1(80h-FFh)

Her bir bank 128 bayt'dır. PIC16F84A 'da sadece RP0 kullanılır. RP1 sıfır olmalıdır.

bit 4 **:** TO: Zaman aşım bit'i (Time-out bit)

1= PIC' e enerji verildiğinde ve CLRWDT ve SLEEP komutu çalışınca

0 = WDT zamanlayıcısında zaman dolduğunda

bit 3 **:** PD: Enerji kesilme bit'i (Power-down bit)

1 = PIC'e enerji verildiğinde ve CLRWDT komutu çalışınca

0 = SLEEP modu çalışınca

bit 2 **:** Z: Zero bit (Sıfır bit'i)

1 = Bir aritmetik işlem veya mantıksal işlem sonucu 0 (sıfır) olduğunda

0 = Bir aritmetik işlem veya mantıksal işlem sonucu 0 (sıfır) ol**ma**dığında.

bit1 **:** DC: Taşma ve Ödünç bit'i (Digit carry/borrow bit) (ADDWF ve ADDLW komutları için)

1 = Alt dört bitin 4. bitinde taşma meydana geldiğinde

0 = Alt dört bitin 4. bitinde taşma meydana gel**me**diğinde

bit 0 **:** C: Taşma ve Ödünç bit'i (Carry/borrow bit)(ADDWF ve ADDLW komutları için)

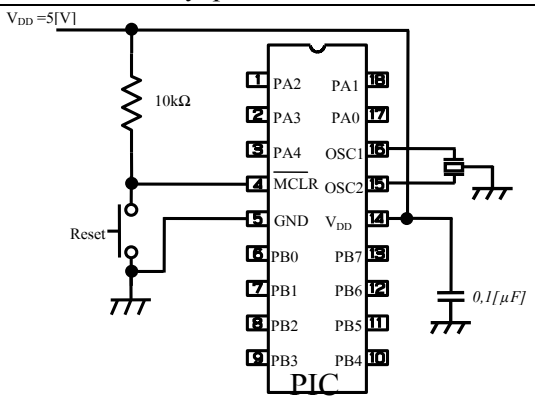
1 = En soldaki 7.bitte taşma olduğunda

0 = En soldaki 7.bitte taşma ol**ma**dığında

Not: RLF ve RRF komutları çalıştığında en sol bit veya en sağ bitin değeri carry bitine yüklenir.

UYGULAMA FAALİYETİ

Aşağıdaki işlem basamaklarına göre uygulama faaliyetini yapınız.

DENEYİN ADI	PIC16F84 Mikro denetleyicisinin program yazılarak çalıştırılabilmesi için gerekli ön şartlarının sağlanması
DENEYİN AMACI	Bir mikro denetleyiciyi çalışır hâle getirebilmek için besleme, reset, osilatör vb. devrelerini oluşturmak.
ARAÇ-GEREÇ	Delikli pertinaks, PIC16F84, kristal 4MHz, kondansatör (2x22pF-*0,1* μF), 18 bacaklı entegre soketi, iletken, lehim teli ve havya
İŞLEM BASAMAKLARI	**1.** 5x7 ebatlarında delikli pertinaks temin ediniz. **2.** 18 bacaklı soketi pertinaksın uygun bir yerine monte ediniz. **3.** İlk önce osilatör devre elemanlarını uygun şekilde yerleştiriniz. **4.** Daha sonra besleme kondansatörünü yerleştiriniz. **5.** Kristal osilatörün PIC'e yakın olmasına dikkat ediniz. **6.** Yerleştirilen elemanların yerleşim şekline tümüyle bakınız. Herhangi bir kusur yok ise elemanların bacaklarını bağlayınız. **7.** Lehimleme yaparken lehimleme kurallarına dikkat ediniz.
ŞEMA	
SONUÇ	Uygulamadan elde edilen sonuçları aşağıya maddeler hâlinde yazınız.

ÖLÇME VE DEĞERLENDİRME

Aşağıdaki soruları dikkatlice okuyunuz ve doğru seçeneği işaretleyiniz.

1. PIC16F84 mikro denetleyicisinin kaç adet I/O portu vardır?
A) 8 **B)** 5 **C)** 13 **D)** 18

2. PIC16F84 mikro denetleyicisinin program bellek kapasitesi ne kadardır?
A) 1kB **B)** 1MB **C)** 64KB **D)** 64MB

3. PIC16F84 mikro denetleyicisinde osilatör frekansı 4 [MHz] ise 1 clock süresi ne kadardır?
A) $4\,\mu s$ **B)** $1\ \mu s$ **C)** 1 ms **D)** Hiçbiri

4. PIC16F84 mikro denetleyicisinde, kaç adet bank vardır?
A) Yok **B)** 1 **C)** 2 **D)** 4

5. PIC16F84 mikro denetleyicisinde RAM bellek üzerindeki yazmaçlara yazılacak veri genişliği kaç bittir?
A) 16 **B)** 1 **C)** 32 **D)** 8

6. PIC16F84 mikro denetleyicisinde, çıkış pinlerinden geçirilecek maksimum sink akımı kaç mA'dir?
A) 20 **B)** 50 **C)** 8 **D)** 25

7. Status yazmacı kaç numaralı adreste yer alır?
A) 0x02 **B)** 0x08 **C)** 0x03 **D)** 0x09

8. PIC16F84 mikro denetleyicisinde, besleme uçları hangi numaralı pinlerdir?
A) 6-13 **B)** 1-14 **C)** 5-14 **D)** 1-10

9. Bir mikro denetleyici ile bir mikro işlemci arasındaki en büyük fark, mikro denetleyicide ……. üniteleri tek bir çip içersinde yer almaktadır.

10. PIC16F84 …….. bitlik bir mikro denetleyicidir.

DEĞERLENDİRME

Cevaplarınızı cevap anahtarıyla karşılaştırınız. Yanlış cevap verdiğiniz ya da cevap verirken tereddüt ettiğiniz sorularla ilgili konuları faaliyete geri dönerek tekrarlayınız.

ÖĞRENME FAALİYETİ–2

AMAÇ

Bir mikro denetleyici seçerek kullanabilmek mikro denetleyici eğitim seti yapacaksınız.

ARAŞTIRMA

- Sevgili öğrenci, mikro denetleyici bir kontrol arabirimidir. Fakat kontrol sistemleri farklılık gösterebilir. Acaba kontrol nedir? Hangi tür kontrol sistemleri vardır ve birbirlerine göre avantajları-dezavantajları nelerdir? Bu konu ile ilgili bir araştırma yapmanızı bekliyoruz.

2. PROGRAMLAMA VE UYGULAMA DEVRESİ

Mikro deneteleyici uygulamalarının gerçekleştirilmesi için mikro denetleyiciye özel olarak tasarlanmış programlayıcı araçılığı ile yazılmış olan programın aktarılması gerekmektedir. Programı yüklenmiş mikro denetleyici gerekli işlemleri gerçekleştirebileceği uygulama devresinde çalışmak üzere hazırdır.

2.1. Mikro Denetleyici Programlama Çeşitleri

Mikro denetleyicilerin programları farklı dillerde yazılabilir. Yazılmış programam belli derleme işlemlerinden geçerek entegreye yüklenmesi gerçekleştirilir.

2.1.1. Pic Nasıl Programlanır

PIC'i programlamak için ilk önce yazdığımız program komutlarının makine diline (1 ve 0 lardan oluşan biçime) yani HEX koduna çevrilmesi gerekir. Bunun için bir derleyiciye (compiler) ihtiyacımız vardır. Eğer programı yazmak için Assembly dilini kullanıyorsak her hangi bir editör kullanmamız yeterli olacaktır. Editörde yazılan program dosyası kaydedilirken uzantısının mutlaka ".asm" olması gerekmektedir. Dos ortamında dosyanın uzantısı ".asm" olarak değiştirilebilir. MPASM (assembler) derleyicisi ile HEX dosyası oluşturulur. Öğrenme faaliyeti–1 de anlatıldığı gibi hem editör hem de derleyici arayüzene sahip MPLAB programını kullanmak kolay olacaktır. Şimdi yapılması gereken HEX

biçimindeki dosyanın PIC mikro denetleyicinin program belleğine gönderilmesidir. HEX biçimindeki program kodlarının, PIC'in program belleğine gönderebilmek için programlayıcı yazılım ve donanım ara yüzüne ihtiyacımız vardır. Yazılım olarak bu modülde IC-PROG programını kullanacağımızı açıklamıştık. Bu yazılım birçok programlayıcı donanımını, bilgisayarımızın seri veya paralel portlarını kullanarak desteklemektedir. Bilindiği gibi PIC içerisine program yazdırabilmek için programlama donanım ve yazılımın uyumlu olması gerekmektedir. Programlayıcı donanımları bilgisayarımızın seri, paralel veya usb portlarını kullanmak üzere tasarlanmış olabilir. Bu nedenle seri, paralel veya usb programlayıcı olarak da isimlendirilebilir. Dolayısı ile programlayıcı yazılımı da aynı port üzerinden iletişim kurmalıdır. Bu modülde kullandığımız programlayıcı donanımını AN589 programlayıcı protokol şartlarına göre tasarlanmıştır ve PIC16F84'e RB6 ve RB7 portları üzerinden seri iletişim protokolü ile veri aktarımı sağlanır.

2.2. Pic Eğitim Seti Elemanları

Temel endüstriyel elektronik dersinde ana elektronik parçaların özelliklerini öğrendik. Burada PIC yazıcı ve deney setini öğrenmiş olacağız.

2.2.1. Elektronik Elemanların Tanıtımı

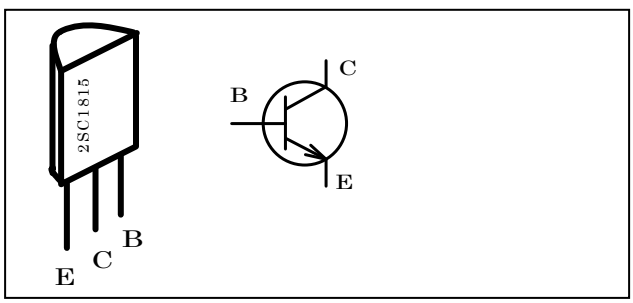
Pic programla devresinde farklı elektronik elemanlar bulunmaktadır. Şimdi bunların yapılarını ve özelliklerini inceleyelim.

2.2.1.1. Transistör

Programlayıcı devresindeki anahtarlama işlemleri için transistorler kullanılmışlardır.

a) 2SC1815 (NPN Transistör)

2SC1815 npn tipi küçük sinyal transistorü.

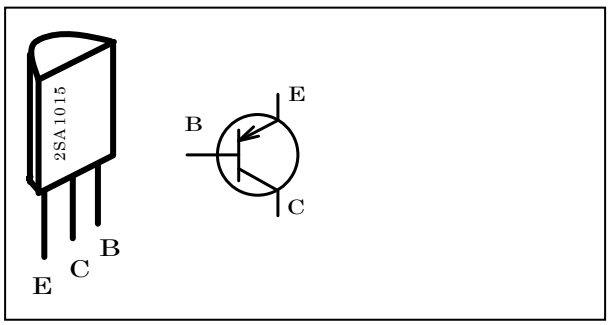


E: Emiter
C: Kolektör
B: Beyz

Şekil 2.1: 2SC1815 Transistörünün bacak bağlantısı

b) 2SA1015 (PNP Transistör)

2SA1015 pnp tipi küçük küçük sinyal transistorü.

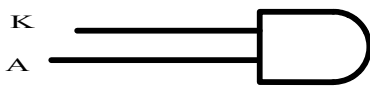


E: Emiter
C: Kolektör
B: Beyz

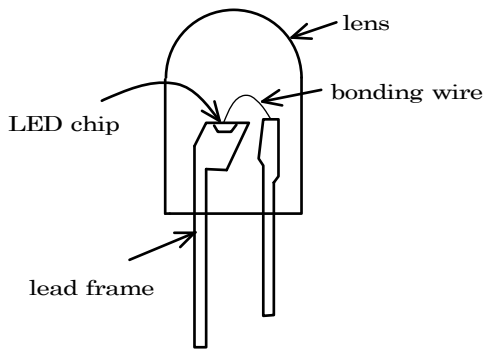
Şekil 2.2: 2SA1015 Transistörünün bacak bağlantısı

2.2.1.2. LED (Light Emitting Diode) (Işık Yayan Diyot)

Güç kaynağında enerjinin olup olmadığını öğrenmek için ve eğitim setinde yazılan programa gore işevini yapıp yapmadığını kontrol amacıyla LED kullanacağız. Görünen ışığın dalga boyu LED (**L**ight **E**mitting **D**iode) 380 – 760 nm arasındadır.



Uzun uç anot u göstermektedir. (+)



LED aşağıdaki özellliklere sahiptir:

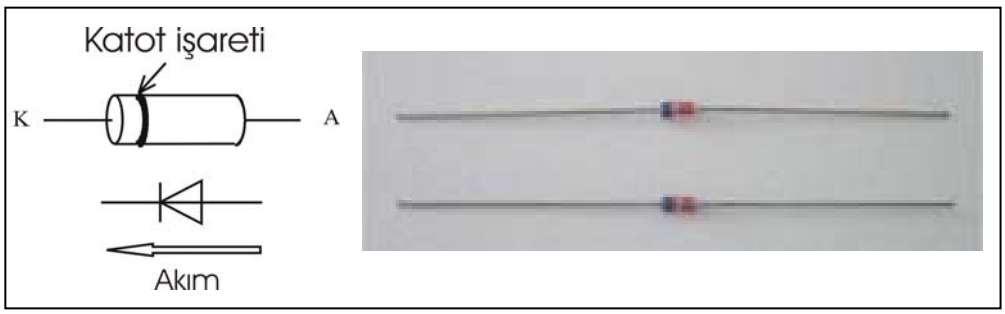
- Uzun ömürlüdür
- Düşük güç harcarlar
- Sahip oldukları ısı çok azdır.
- Yüksek hızda çalışma gösterirler
- Çizimi kolaydır
- Işıklandırma kontrolü kolaydır.

Şekil 2.3: LED'in görünüşü ve yapısı

Mavi LED 1993'te japonyada bir şirket tarafından geliştirildi. Günümüzde birçok alanda kullanılmaktadır çünkü 3 ana renge sahiptir (kırmızı, yeşil ve mavi) (örneğin, dot matrix ekran tam renkli ekrandır).

2.2.1.3. Diyot (Anahtar Diyot)

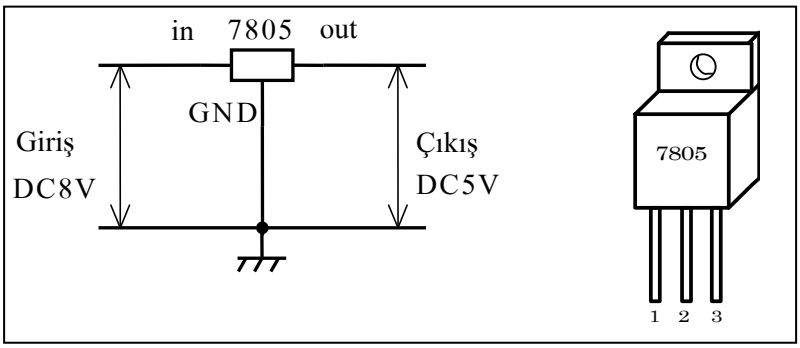
1N4148 küçük bir anahtar diyottur. Düz polarma durumunda 0,6 V. gerilime sahiptir.



Şekil 2.4: Diyot görünüşü ve sembolü

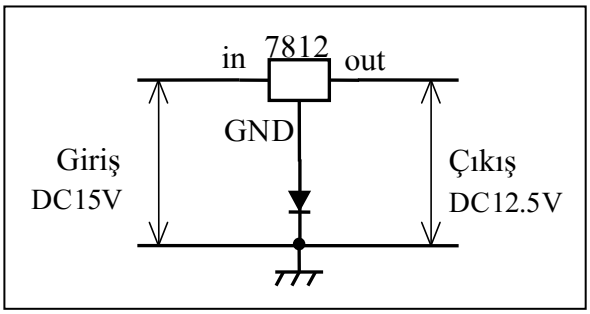
2.2.1.4. Seri Regülatör (Gerilim Düzenleyici)

Sabit bir gerilim elde etmek seri regülatör ile çok kolaydır. Örneğin, eğer 7805 seri regülatör şekil 1.33'teki gibi kullanılırsa 5V düzenli ve sabit bir gerilim elde edilmiş olur. Şayet 7812'de aynı şekilde kullanılacak olursa bu defada 12 volt sabit gerilim elde edilmiş olur.



Şekil 2.5: Seri regülatör

PIC programını yazma devresi için gerekli olan voltaj 12,5 volttur. 12,5 voltu elde edebilmek için 7812 seri regülatörün toprak (GND) kısmına diyot elde etmemiz yeterli olacaktır (Şekil 1.34).



Şekil 2. 6: 12,5 volt elde edilmesi

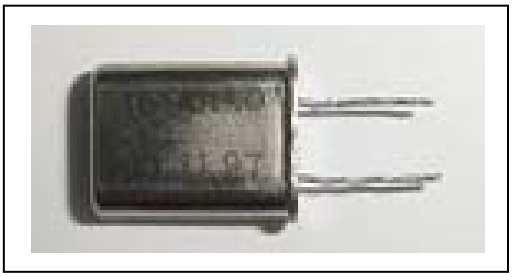
Çıkış gerilim 12,5 volt değerindedir ve diyotun her iki ucunda da 0,5 voltluk gerilim oluşmuş olur.

Kutup	Model numarası	Çıkış akımı	Uçların karşılıkları		
			1	2	3
Pozitif	78M serisi	500mA	Giriş	GND	Çıkış
	78 serisi	1A	Giriş	GND	Çıkış
Negatif	79M serisi	500mA	GND	Giriş	Çıkış
	79 serisi	1A	GND	Giriş	Çıkış

Tablo 2.1: Seri regülatör

2.2.1.5. Kristal Osilatör

Kristal osilatör mikro denetleyicilere pals üreterek çalışmalarını sağlayan elemandır. 4MHz. lik kristal osilatörler kullanmaktayız.

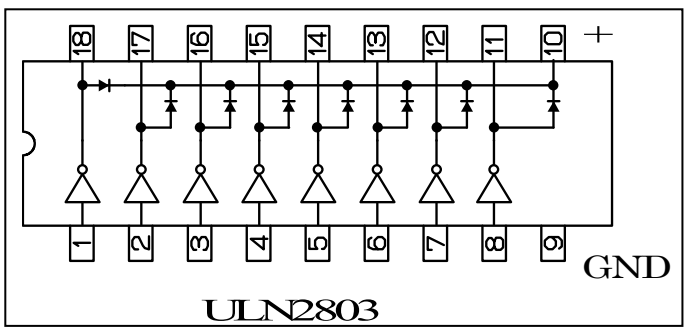


Şekil 2.7: Kristal osilatör

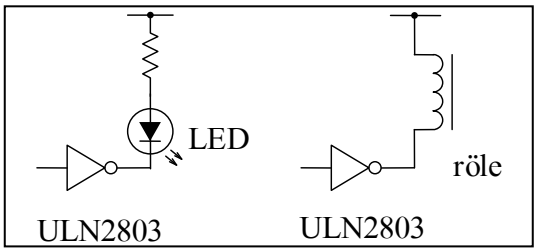
PIC'e göre en doğru kristali seçmemiz gerekmektedir. Eğer uygun kristal seçersek en doğru pals elde etmiş oluruz. PIC' ler 4 – 10 – 20 MHZ gibi hızlara sahiptir ve kristali yüksek hızda da seçsek PIC in en yüksek hızına göre çalışma sağlanır.

2.2.1.6. Darlington Bağlı Transistor Grubu (ULN2803)

Transistor düzeninde 8 adet darlington transistor kullanılmıştır. Diyotlarda bu transistor düzeni içinde bulunmaktadır ve röle veya benzerleri elemanları kontrol edilebilme özelliği vardır. ULN2803 50 Volt ve 500 mA yükü kontrol edebilir. Mikro denetleyici kontrolleri için transistor düzeneği çok kullanışlıdır.



Şekil 2.8: ULN2803 Entegresinin bacak yapısı



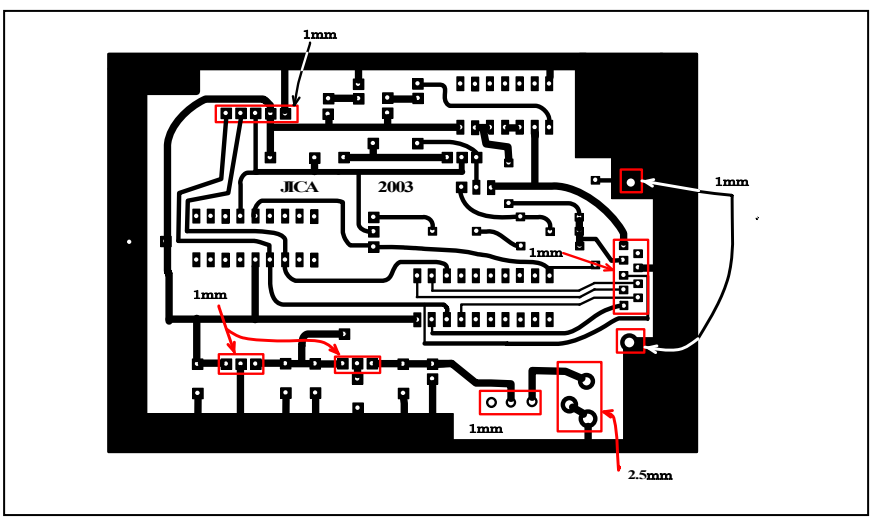
Şekil 2.9: Transistor grubu (ULN2803)

2.3. Pic Eğitim Setinin Yapılışı

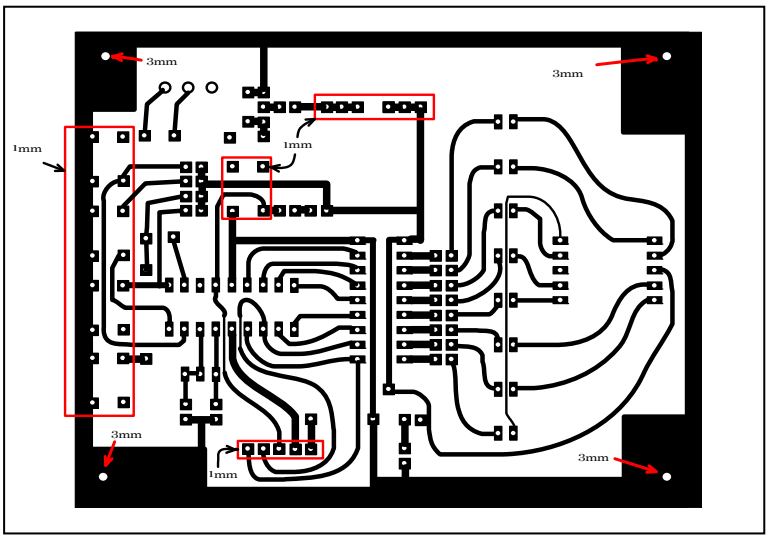
Konumuzun başındada belirtiğimiz gibi programlanmış mikro denetleyicinin programına göre yapacağı işlemleri gerçekleştireceği bir uygulama devresi bulunması gerekmektedir. Dersimizde birçok işlem için kullanabileceğimiz üzerinde led çıkışları ve düğme girişleri bulunan uygulama devresini berebar yapımını gerçekleştireceğiz.

2.3.1. Devre Kartı Üzerindeki Deliklerin Delinmesi

Delik delme işlemi her iki devre içinde yapılır. 0.8 mm'lik matkap ucu kullanılmalıdır. Şekil 2.10 ve Şekil 2.11'de gösterildiği gibi de 1mm, 1.5 mm, 3 mm ve 6 mm çaplarında da deliklerde mevcuttur.



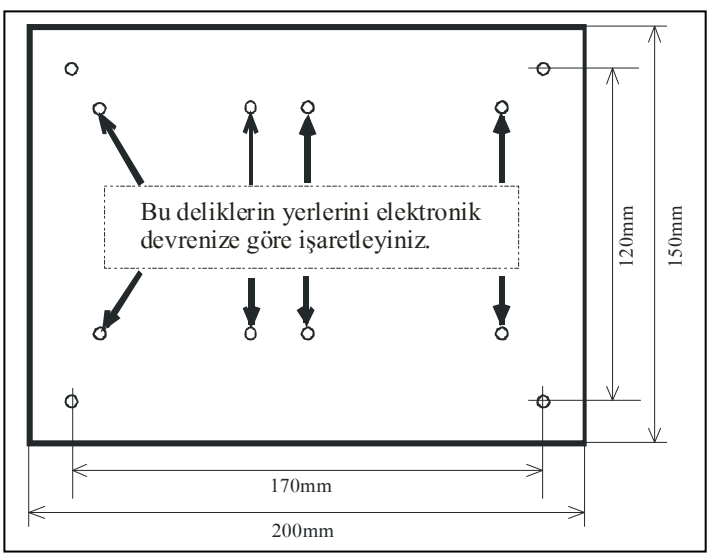
Şekil 2.10: PIC yazıcı



Şekil 2.11: PIC deney seti

2.3.2. Alt Taban (Şeffaf Pertinaks) Deliklerinin Açılması

Alt taban yapılan elektronik devrenin biraz havada kalmasını sağlayan ve yerle temasını gerçekleştiren bölümdür. Şeffaf malzeme üzerindeki koruma yapışkanı delikler açıldıktan sonra çıkartılmalıdır. Bütün delikler 3 mm olmalıdır.



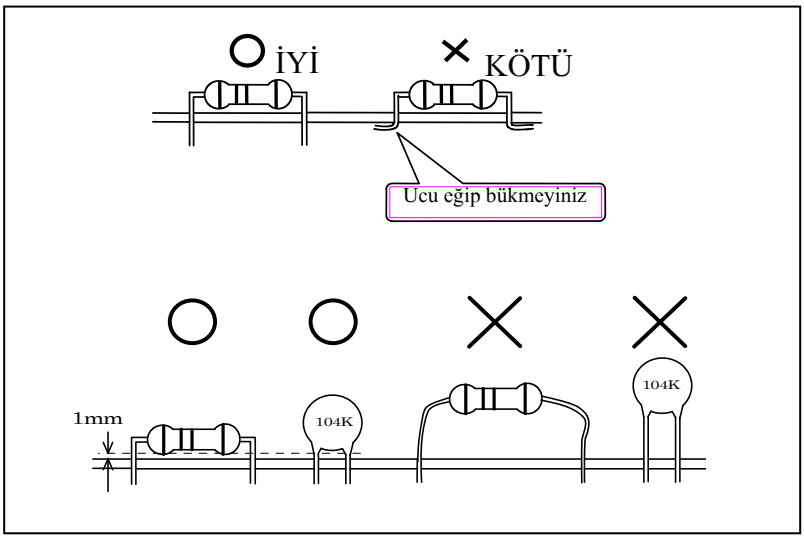
Şekil 2.12: Şeffaf alt taban delikleri

2.3.3. Lehimleme İşlemleri

Tanıtımları yapılan elemanların işlevlerini gerçekleştirebilmeleri için elektronik karta yerleştirilerek lehimle montajlarının yapılması gerekir.

2.2.3.1 Elektronik Elemanların Yerleştirilmesi

Direnç ve kondansatörler şekilde gösterildiği gibi montajı yapılır.

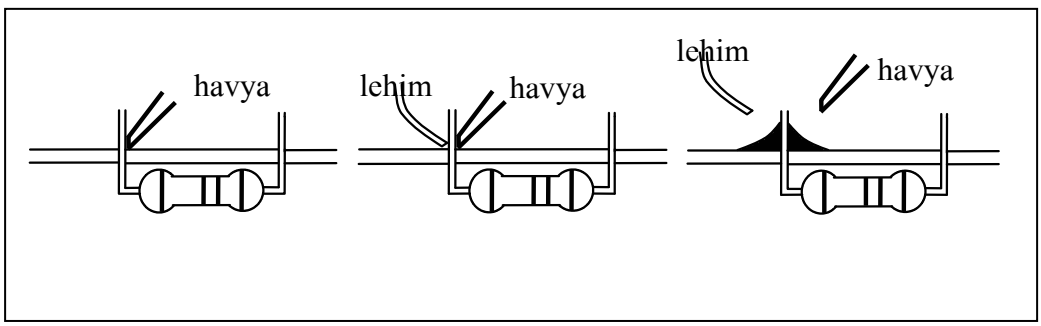


Şekil 2.13: Elektronik parça elemanlarının yerleştirilmesi

2.3.3.2. Lehimlemenin Yapılması

Elektronik devrelerin montajında lehimleme çok önemli bir aşamadır.

- Havyanın ucunu lehimlenecek olan yere tut (yaklaşık 1 saniye).
- Sonra lehimi tut ve lehim erir.
- Daha sonra lehim ve havya ucunu aynı zamanda lehim yapılan yerden uzaklaştır.

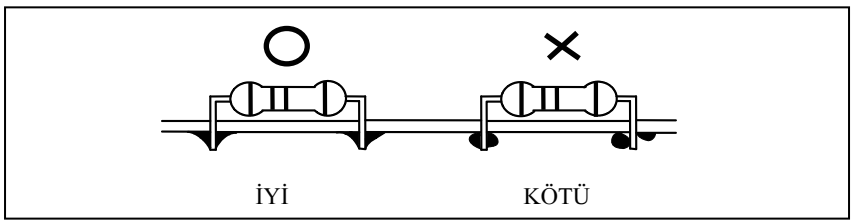


(a) (b) (c)

Şekil 2.14: Lehimleme Yöntemi

2.3.3.3. Lehimlemeyi Kontrol Et

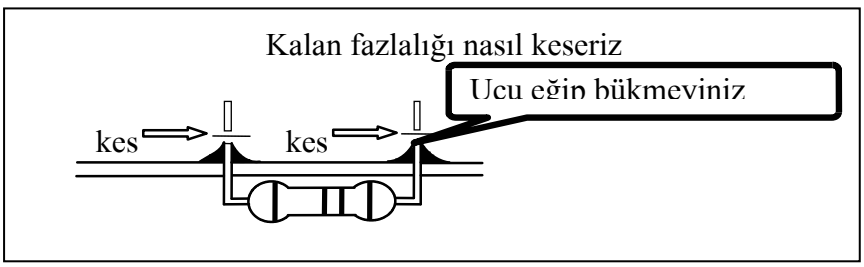
Devremizin sağlıklı ve hatasız çalışması için lehimlerimizin muntazam ve kurallarla uygun şekilde yapılması gerekir. Lehimlerinizin aşağıda doğru olarak gösterilen şekildeki gibi olması gerekir.



Şekil 2.15: Lehimleme

2.3.3.4. Kalan Fazla Kablo Başlarının Kesilmesi

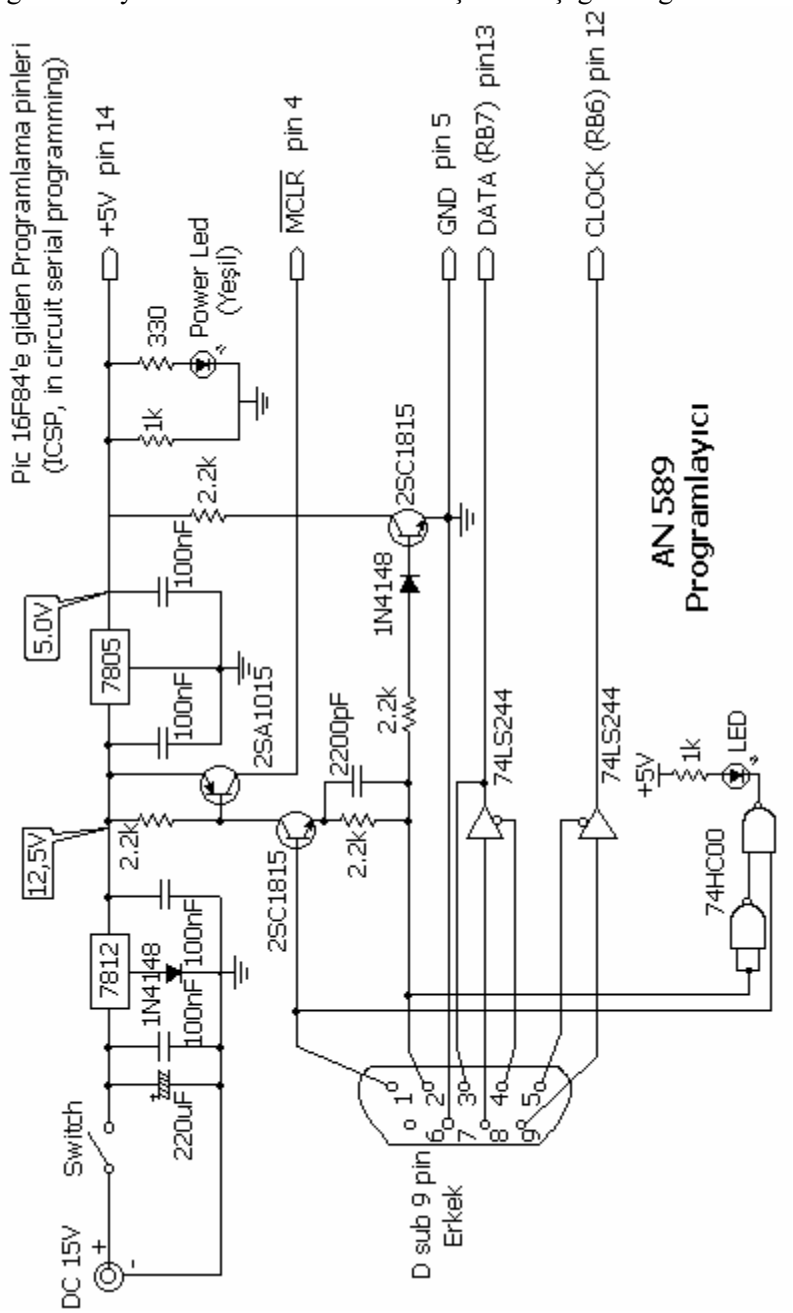
Lehimleme işleminin kontrolleri yapıldıktan sonra devredeki elemanların fazla ucları yan keski yardımı ile kesiniz.



Şekil 2.16: Kablo fazlalıklarının kesilmesi

2.3.4. Pic Yazıcı Devre Şeması

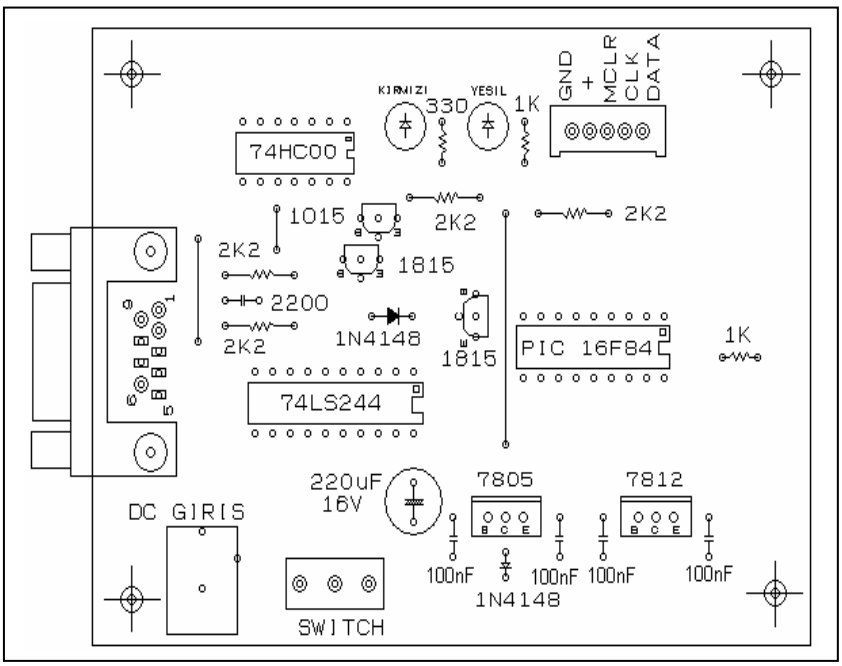
Pic programının yazıcı devresinin elektronik şeması aşağıdaki gibidir.



Şekil 2.17: PIC yazıcısının devresi

2.3.5. PIC Yazıcı Devre Elemanlarının Düzeni

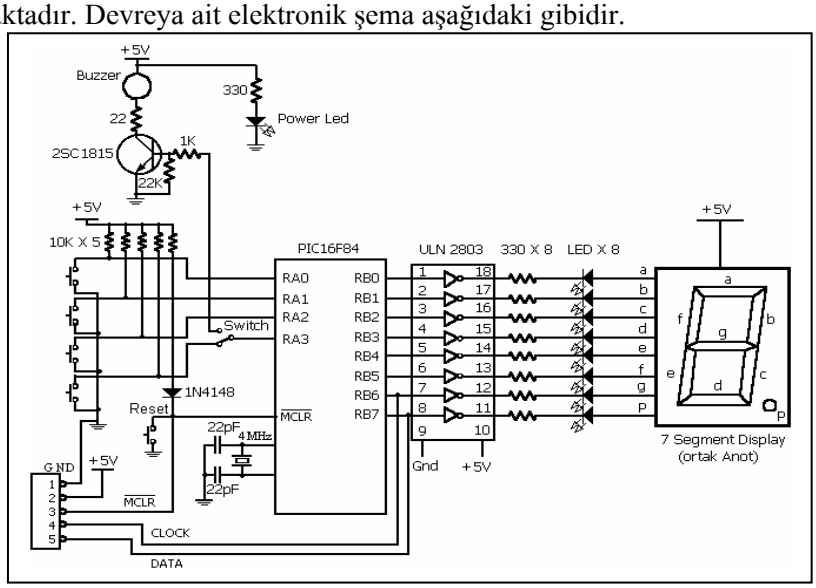
Pic yazıcı devresinin montajı yapılırken elektronik elemanları aşaıdaki düzene göre yerleştiriniz.



Şekil 2.18: PIC yazıcı devre yerleşim şeması

2.3.6. PIC Uygulama Seti Devre Şeması

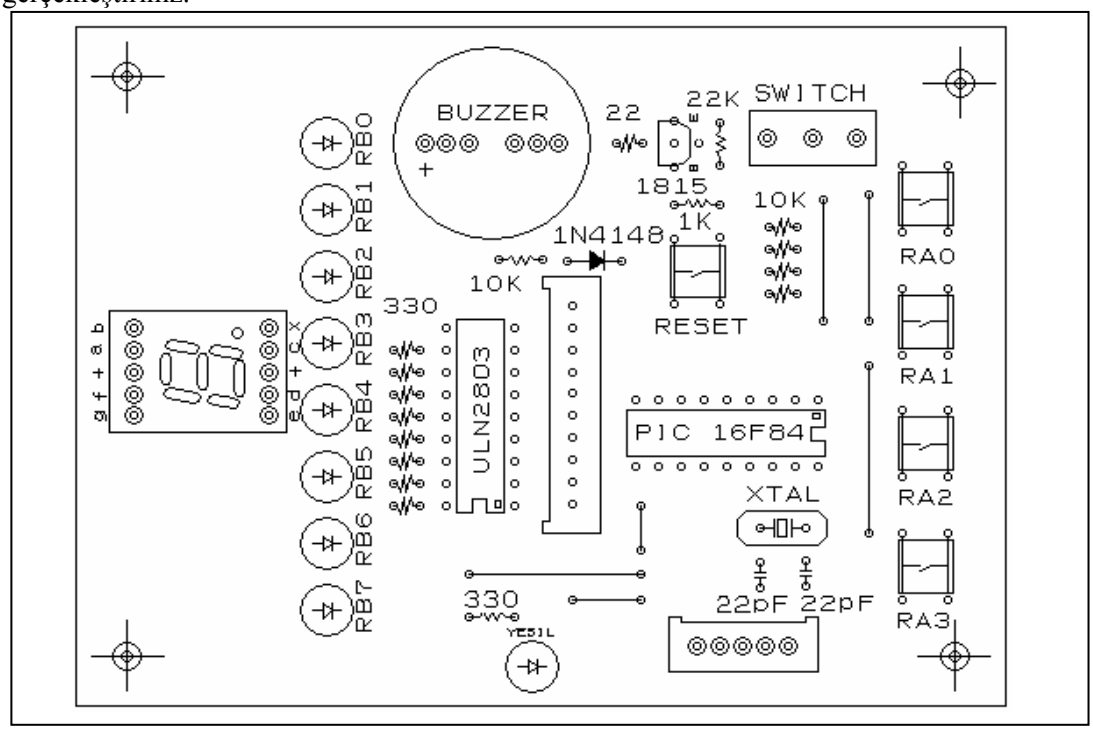
Pic programlarımızın uygulamalarını yapacağımız devrede 4 adet giriş ve 8 adet çıkış kullanılmaktadır. Devreya ait elektronik şema aşağıdaki gibidir.



Şekil 2.19: PIC Uygulama devresi

2.3.7. PIC Uygulama Seti Devre Elemanlarının Düzeni

Pic uygulama devremizdeki malzemeleri aşağıdaki düzene göre yerleştirip montajını gerçekleştiriniz.



Şekil 2.20: Uygulama devresi yerleşim planı

2.3.8. Parça Listesi

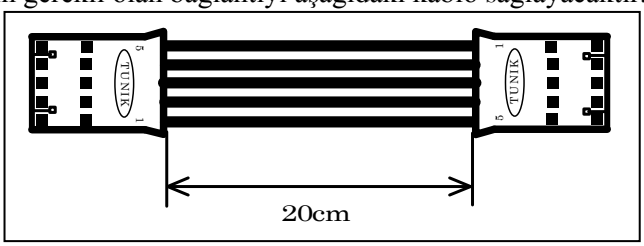
Pic programlayıcı ve uygulama devresinin gerçekleştirilebilmesi için aşağıda adetleri ve özellikleri yazılı parçaların temin edilmesi gerekir.

Sıra Nu	Parçanın ismi	Özelliği	Adedi
1	Baskı devre plaketi	Yazıcı ve Eğitim seti için	1
2	IC soket	14pin	1
3	IC soket	18pin	3
4	IC soket	20pin	1
5	IC	74HC00	1
6	IC	74LS244	1
7	PIC microdenetleyici	PIC16F84A	1
8	Transistör düzenleyici	ULN2803	1
9	Regülatör IC	7805	1
10	Regülatör IC	7812	1
11	LED	kırmızı	9
12	LED	yesil	2
13	Transistör	2SA1015	1
14	Transistör	2SC1815	3
16	Direnç	*300Ω 1/4W*	9
17	Direnç	*2.2kΩ 1/4W*	4
18	Direnç	*1kΩ 1/4W*	3
19	Direnç	*10kΩ 1/4W*	5
20	Direnç	*22Ω 1/4W*	1
21	Direnç	*22kΩ 1/4W*	1
22	Kondansatör	0.1uF	4
23	Kondansatör	2200pF	1
24	Kondansatör	22pF	2
25	Elektrolitik kondansatör	220uF/16V	1
26	Anahtar	3P üç uclu	2
27	Konnektör (PC için)	D sub 25P erkek	1
28	Kablo konnektörü	Dsub 25P	1
29	Konnektör	D sub 9P disi	1
30	Kablo konnektörü	Dsub 9P	1
31	Konnektör (set için)	D sub 9P erkek	1
32	Konnektör	5P erkek devre için	2
33	Konnektör	5P disi	2
34	Konnektör	10P erkek devre için	2
35	Güç giris soketi	2P	1
36	AC adaptör	15V 300mA	1
37	Kablo	8'li 1.2m	1
38	Kristal	4MHz	1
39	Basmalı buton	Mikro buton	5
40	Hoparlör	Piezoelektrik	1
41	7 segment LED	Ortak anot	1
42	Seffaf alt taban	100*150	2
43	Vida	M3*30	8
44	Vida somunu	M3	8
45	Yükselti elemanı	20 mm	8

Tablo 2.2: Malzeme listesi

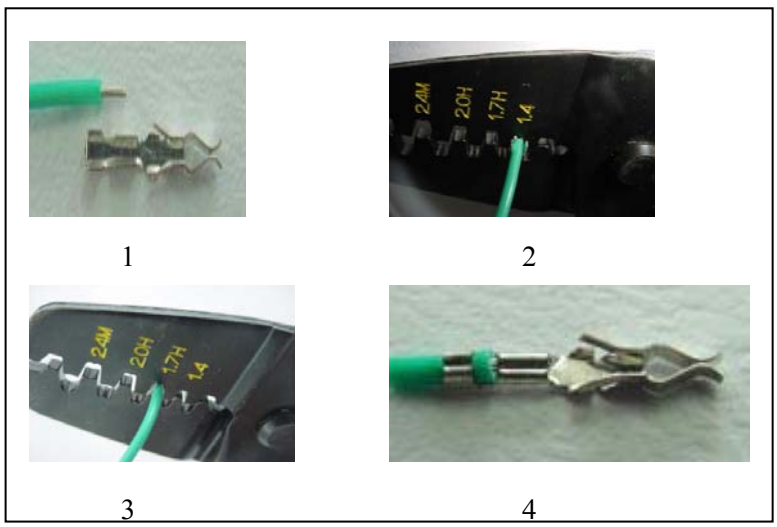
2.3.9. Ara Bağlantı Kablosunun Yapımı

Programlayıcı ile uygulama devresinin arasında pic mikro deneteleyicisinin programlanması için gerekli olan bağlantıyı aşağıdaki kablo sağlayacaktır.



Şekil 2.21: PIC Yazıcı ve uygulama devrelerinin ara bağlantı kablosu

Şekil 2.22: Açık pinleri kapatma aleti

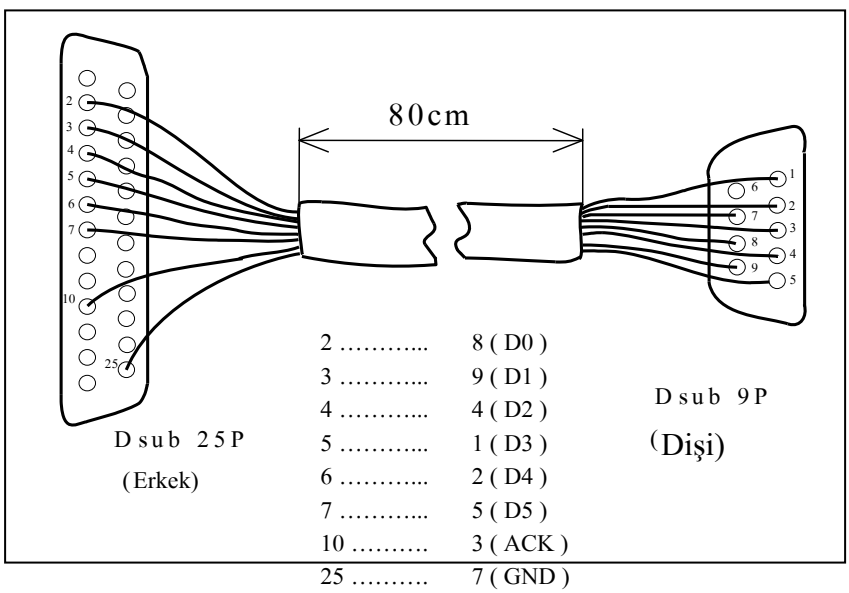


Şekil 2.23: Aletin kullanılması

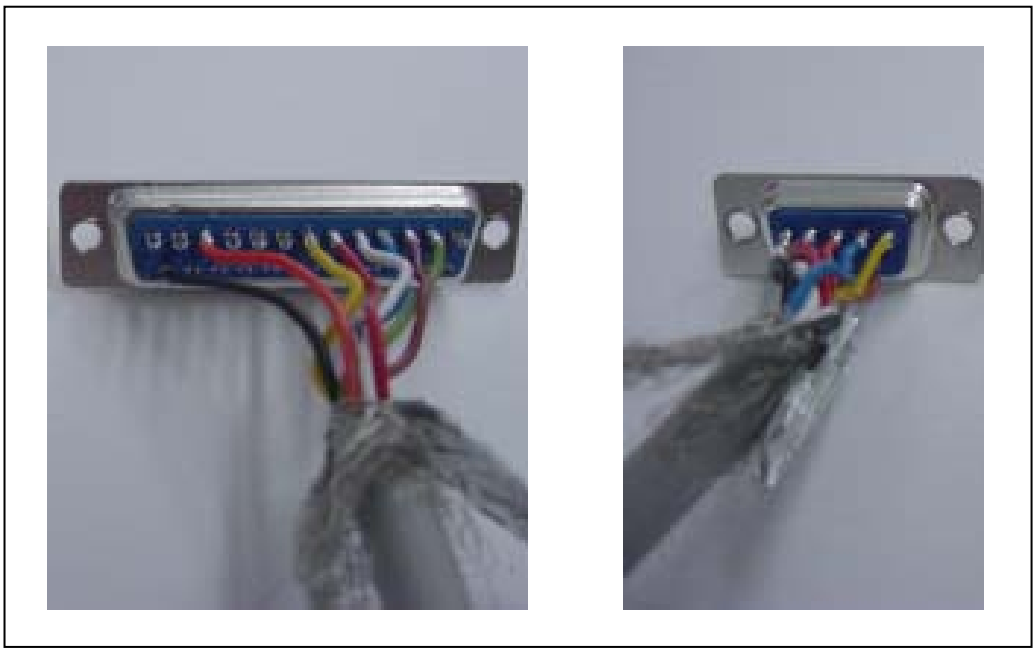
Şekil 2.24: Uçları plastiğe geçirme işlemi

2.4. Bilgisayar Bağlantı Kablosunun Yapımı

Pic programlayıcı ile bilgisayar arasında bir bağlantı kablosu yardımıyla bağlanır. Bu kablo için 25 pin erkek ve 9 pinlik dişi iki adet soket ve 8li cok damarlı kabloya ihtiyac vardır.



Şekil 2.25: Bilgisayar bağlantı kablosunun hazırlanması



Şekil 2.26: Kişisel bilgisayar için kablo

UYGULAMA FAALİYETİ–1

Aşağıdaki işlem basamaklarına göre uygulama faaliyetini yapınız.

- Bilgisayarda çizilmiş PIC yazıcı ve uygulama devresinin plaketinin hazırlanıyınız.
- Hazırladığınız plaket üzerine malzemelerin yerlşetiriniz.
- Lehimleme işlemlerini gerçekleştirerek montajı tamamlayınız.

İşlem Basamakları	Öneriler
➢ Şekil 2.17 ve şekil 2.19'da PIC yazıcı ve uygulama devresi verilmiştir. Bu devre şemalarını, bilgisayarınızda elektronik devre tasarımı programı kullanarak, baskı devresini hazırlayınız.	
➢ Baskı devresini hazırladıktan sonra bir bakırlı plaket üzerine aktarma işlemlerini gerçekleştiriniz.	➢ Elektronik devre kartını aşındırma işlemini yaparken asit karışımını ve aşındırma işlemini açık havada, laboratuarın dışında yapılması sağlık açısından uygundur.
➢ Plaketi aşındırma işlemine tabi tutarak, plaket üzerindeki bakır yolların oluşmasını sağlayınız.	➢ Öğrencilerin tamamı aşındırma işlemine hazır hâle geldikten sonra topluca aşındırma işlemi yapılmalıdır. Diğer durumda her seferinde çözelti hazırlamak gerekir ki atık çözeltinin çevreye çok zararı vardır.
➢ Bakır yolların üzerindeki boyalı maddeyi alkol vb. maddelerle temizleyiniz ve bakır yolların tamamen ortaya çıkmasını sağlayınız.	➢ İşi biten kullanılmaz hale gelen çözeltinin, toprak üzerine dökülmesi ya da kanalizasyona gönderilmesi çevre sağlığını etkileyecektir. Kesinlikle yapılmamalıdır. Çözelti içinde metallerin nötrleştirilmesi gerekmektedir.
➢ Plaket üzerinde açılması gereken delikleri uygun matkap ucu ile açınız (direnç vb. elemanları 0,8 mm, daha kalın eleman bacakları için 1mm uc kullanınız).	➢ Devre kartı üzerine açılacak deliklerin mutlaka eleman bacaklarının çapına uygun delinmesi gerekmektedir.

➢ Plaket üzerine elektronik devre elemanlarını lehimleyiniz.	➢ Lehimleme yaparken uygun güçte, temiz uçlu bir havya kullanınız. Lehimleme kurallarına dikkat ediniz.
➢ Lehimleme yaparken elektronik elemanların boy sırasına dikkat ediniz. Boyu kısa olan elemanları önce lehimleyiniz.	➢ Gerilim düzenleyici entegreler için soğutucu kullanabilirsiniz. Entegreler için soket kullanınız.

UYGULAMA FAALİYETİ–2

Aşağıdaki işlem basamaklarına göre uygulama faaliyetini yapınız.

- PIC yazıcı ve deney devresinin devre üzeri bağlantıları yapan kabloları için uygun kabloları eşit boylarda hazırlayınız.
- Hazırladığınız kabloların uclarını uygun şekilde soyunuz.
- Konektörlere yerleştirip sıkıca sıkıştırınız.

İşlem Basamakları	Öneriler
➢ Şekil 2.21, 2.22 ve 2.23'teki şekillere bakarak ara bağlantı kablosu yapılacaktır. Bunun için 21 cm uzunluğunda 5 adet 0,5mm^2 kesitinde çok telli kablo, 2 adet 5'li dişi konektör ve 10 adet konektör için pin gerekmektedir.	➢ Uçlarına dişi pin bağlanmış kablolar hazırlandıktan sonra dişi konektörün 1 nolu yuvasına birinci kabloyu bağlarken kablonun diğer ucunu diğer dişi konektörün 5 nu.lı yuvasına bağlanmalıdır. Bu sıra dikkat edilerek konektörlerin yuvalarına kabloları bağlayınız.
➢ Şekil 2.22'deki pin sıkma pensi ile şekil 2.23'te gösterildiği gibi kablo ucu 1mm kadar açınız ve pin içersine yerleştirilerek sıkma pensi ile sıkınız.	
➢ 5 adet kablo bu şekilde her iki ucuna pin bağlantısı yapıldıktan sonra dişi konektörlerin içine şekil 2.23'te gösterildiği gibi yerleştiriniz.	

UYGULAMA FAALİYETİ–3

Aşağıdaki işlem basamaklarına göre uygulama faaliyetini yapınız.

- PIC eğitim seti ile bilgisayar arasıdaki iletişimi sağlayan bilgisayar bağlantı kablosu için uygun kabloyu seçiniz.
- Kablonun uclarını soyarak lehimleme işlemi için hazırlayınız.
- Renk tablolarını kontrol ederek soketlere yerleştirip lehimleyiniz.
- Soketleri plastik kılıflarına yerleştirerek kabloyu tamamlayınız.

İşlem Basamakları	Öneriler
➢ Şekil 2.25'e bakarak bilgisayar bağlantı kablosu yapılacaktır. Bunun için yaklaşık 1,2 m uzunluğunda 8'li veri kablosu, D Sub DB–9 platik ekonomik başlık –dişi ve D SUB DB–25 platik ekonomik başlık-erkek temin edilmelidir.	➢ Bilgisayar bağlantı kablosunu hazırlarken lehimleme basamağında bir arkadaşınızın yardım etmesi sağlıklı olacaktır.
➢ Şekil 2.25'e bakarak 4 cm uzunluğunda veri kablosunun uçlarını açınız. Örgülü bakır teli ve blendajı dış kılıfın bittiği yere dolayarak tutma montaj elemanı ile sıkarak bağlayınız.	
➢ Lehim yapılacak noktaları ön lehimleme yaparak hazır hâle getiriniz.	
➢ Kablo uçlarını yaklaşık 2 mm açarak ön lehimleme yapınız.	
➢ Şekil 2.25'e bakarak pinlere lehimlemeleri gerçekleştiriniz.	
➢ DB–9 dişi ve DB–25 erkek konektörleri plastik kılıfların içine yerleştirerek montajı bitiriniz.	

ÖLÇME VE DEĞERLENDİRME

Aşağıdaki soruları dikkatlice okuyunuz ve doğru seçeneği işaretleyiniz.

1. Uygulamalarımızda kullanacağımız pic programlayıcı hangisidir?
A) AN589 **B)** JDM **C)** Fun_Card **D)** ProPic 2

2. Programlayıcıda data(veri) ucu pic'in hangi pininine bağlıdır?
A) RA2 **B)** RB3 **C)** RB7 **D)** RB6

3. Kullandığımız pic mikro denetleyicisini programlarken kaç volt gerilime ihtiyaç vardır?
A) 5V **B)** 18V **C)** 12,5V **D)** 3V

4. PIC16F84 mikro denetleyicisinde giriş veya çıkış olarak ayarlanabilecek toplam kaç pin vardır?
A) 7 **B)** 8 **C)** 5 **D)** 13

5. Pic eğitim setinin çalıştırılabilmesi için kullanılabilecek adaptör gerimi kaç volt olmalıdır.
A) 12V **B)** 5V **C)** 9V **D)** 50V

6. Derlenmiş bir programı pic'e yazdırırken hata oluşma nedeni hangisi **olamaz.**
A) Programın yanlış yazılması **B)** Data kablosu arızalı
C) Besleme gerilimi yok **D)** Pic ters takılmış

7. Pic'e yazdırılmış bir programın uygulama setinde çalışması izlenemiyorsa nedeni hangisi **olamaz.**
A) Program doğru yazılmamış **B)** Devre beslemesi yok
C) Pic ters takılmış **D)** Data kablosu takılı değil

8. Deney setinde buzzer pic'in hangi pinine bağlanmıştır.
A)RA0 **B)** RA3 **C)** RB6 **D)** RB7

DEĞERLENDİRME

Cevaplarınızı cevap anahtarı ile karşılaştırınız. Doğru cevap sayınızı belirleyerek kendinizi değerlendiriniz. Yanlış cevap verdiğiniz ya da cevap verirken tereddüt yaşadığınız sorularla ilgili konuları faaliyete geri dönerek tekrar inceleyiniz.

ÖĞRENME FAALİYETİ–3

AMAÇ

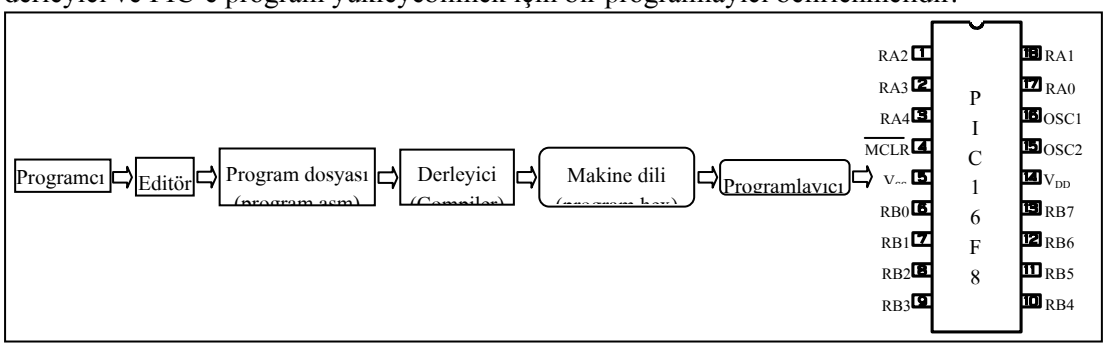
Seçilen mikro denetleyicinin programlanabilmesi için uygun editör, derleyici ve programlayıcıların seçerek kullandığınız bilgisayara yükleyeceksiniz.

ARAŞTIRMA

- Sevgili öğrenci, PIC programlamak için değişik editör, derleyici ve programlayıcılar mevcuttur. Siz kullanıcılar olarak seçtiğiniz PIC'i programlamak üzere kullanacağınız editör, derleyici ve programlayıcı türlerini ve özelliklerini araştırarak arkadaşlarınıza sunum yapınız.

3.1. Mikro Denetleyici Program Editörünün Kurulumu

Modül 1'de açıklandığı üzere bir PIC entegresine program yükleyerek giriş-çıkış terminallerini kontrol edebilmek için şekil 3.1'de gösterilen aşamaların tamamlanması gerekmektedir. Bu aşamalardan önce tabiki PIC mikro denetleyicisi ile neyi ve nasıl kontrol edileceği çözülmelidir. Ancak bu öğrenme faaliyetinde seçilen programların bilgisayarımıza yüklenmesi, ayarları ve PIC'e küçük bir program yükleyerek çalıştırılabilmesi için adı geçen programları nasıl kullanmak gerektiği anlatılacaktır. İlk önce ihtiyaçların belirlenmesi gerekmektedir. İhtiyacımız, bir editör, kullandığımız PIC'e ve programlama diline uygun bir derleyici ve PIC'e program yükleyebilmek için bir programlayıcı belirlenmelidir.

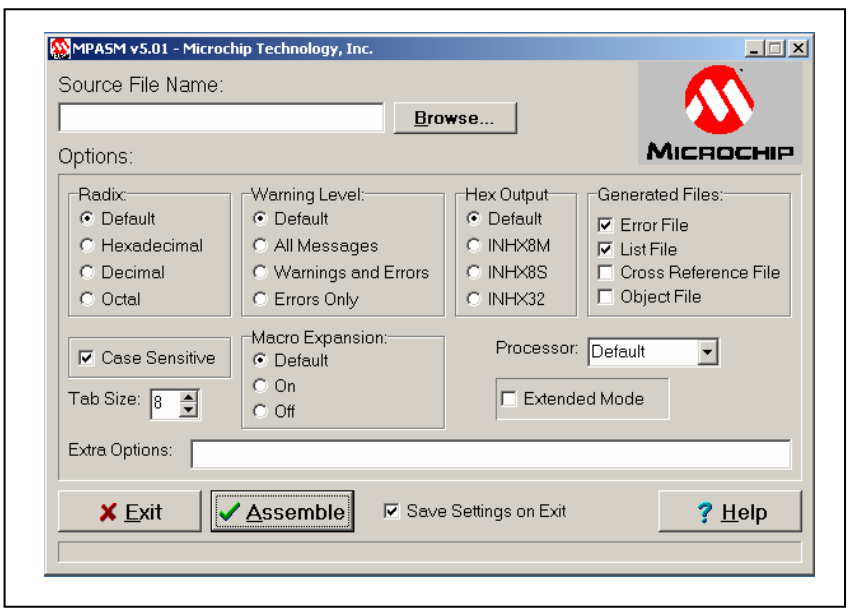


Şekil 3.1. Bir kullanıcı ile mikro denetleyici arasındaki işlem sıralaması

Assembly dili ile program yazarken derleyicinin kabul edebileceği hangi editör kullanırsa kullanılsın mutlaka dosya uzantısı olarak ".asm" verilmelidir. Editör olarak Windows not defteri kullanılabilir. Derlemek için ise Microchip firması tarafından

geliştirilen MPASM derleyicisi (assembler) kullanılabilir. Derleme sonucunda "hex" uzantılı dosya elde edilir ve bu dosya içerisinde makine dilinde yazılmış program kodları bulunur. Bundan sonra yapılacak işlem programlayıcı ile mikro denetleyiciye hex dosyasını göndermek olacaktır.

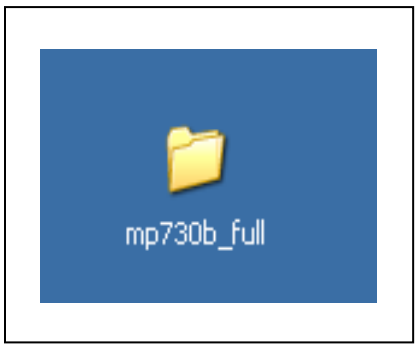
Şekil 3.2: MPASM exe program ikonu



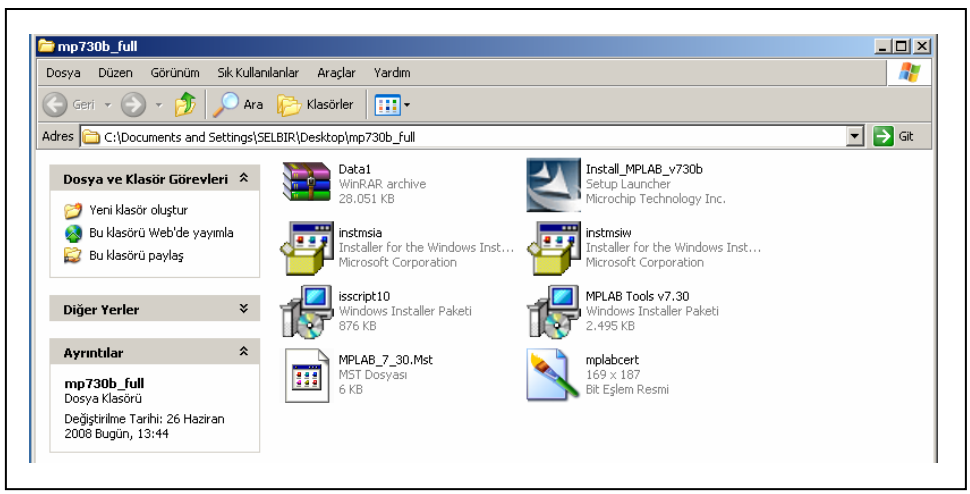
Şekil 3.3. MPASM v5.01 derleyicisi

Şekil3.3 de Browse butonundan derlenecek olan asm uzantılı olan dosya araştırılır ve seçilir. Dosya seçildiğinde metin kutusu içerisine yolu gelir. Burada yapılacak ayarlama olarak PIC seçimi processor etiketindeki açılır metin kutusundan belirlenir. Başka herhangi bir ayarlamaya yapmaya gerek yoktur. Yalnız MPASM derleyicisi ile derleme yapıldığında kullanılan PIC'e ait tanımlamaların yapıldığı ".INC" uzantılı dosyanın aynı dizin içerisinde olması gerekir

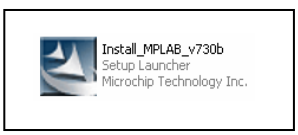
Bu yöntemin dışında Microchip firması tarafından geliştirilen MPLAB programı kullanılabilir. MPLAB, Windows uyumlu bir paket programdır. Microchip firmasının ürettiği PIC'lere uygun program yazmak ve derlemek üzere geliştirilmiştir. Editör ve derleyiciyi içinde barındırır. C derleyici eklentisi ile PIC'e uygun C kaynak kodlu programlar yazılıp, derlenebilir. Microchip firmasının ürettiği PIC mikro denetleyicilerini destekler. Ayrıca pic Simulatöre sahiptir. Bu modülde, avantajlarından ötürü MPLAB programının kullanımı anlatılacaktır.



Şekil 3.4. Mplab730b_full versiyon program dizini



Şekil 3.5: Mplab730b_full dizin açılımı

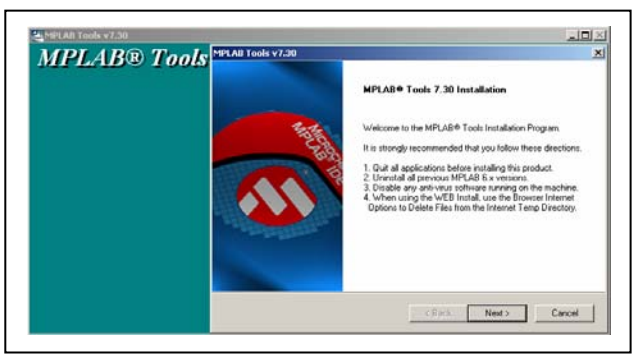


Şekil 3.6: MPLAB v7.30b setup exe dosyası

Bu bölümde MPLAB v7.30b kurulumu anlatılacaktır. Şekil3.6 de MPLAB v7.30b kur dosyası iki defa tıklandığında kurulum başlar.

Şekil 3.7. MPLAB v7.30b kurulumun başlaması

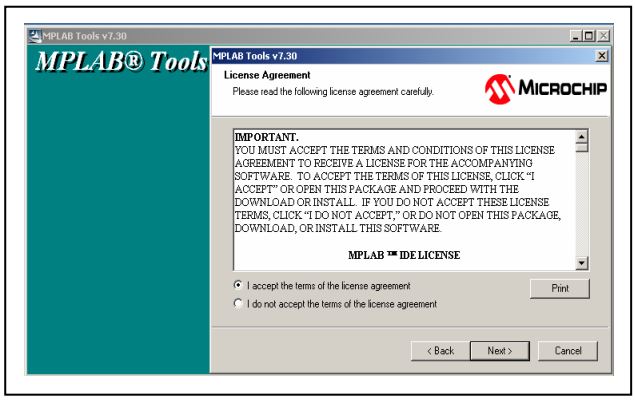
Yükleme sihirbazının yardımı ile sizin rehberliğinizde hazırlanacağını belirten bir ibare bulunmaktadır.



Şekil 3.8. MPLAB Tools v7.30b programını yüklemek için tavsiyeler

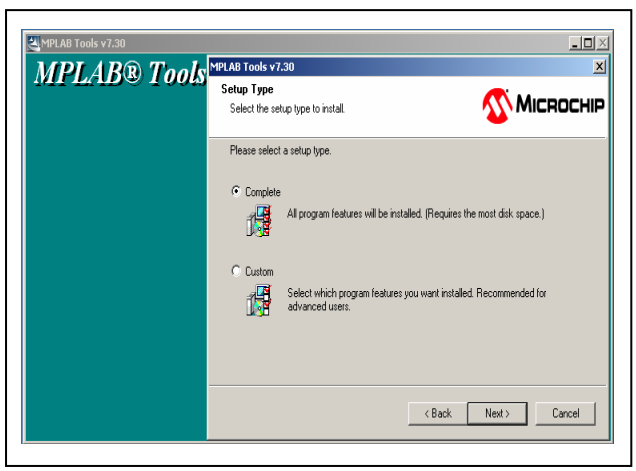
- Bu programı yüklemeden önce tüm uygulamaları durdurunuz.
- Bir önceki sürüm olan MPLAB6.x sürümünü kaldırınız.
- Bilgisayarınızdaki çalışan anti-virüs yazılımlarını etkisizleştiriniz.
- Programı WEB den yüklerken internet geçici klasörü dosyalarını siliniz ve bu dosyaları silmek için internet tarayıcı özelliklerini kullanınız.

Şekil 3.9 da, lisans sözleşmesinin şartlarının kabul edildiği seçilerek, next butonu tıklanır.



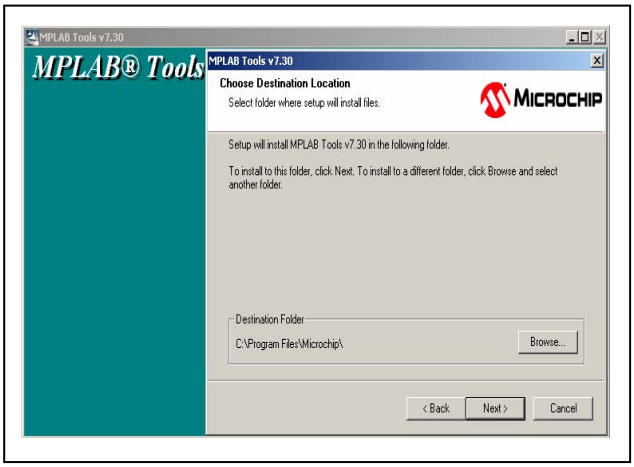
Şekil 3.9: Lisans sözleşmesi

Şekil 3.10'da görüldüğü gibi, kur yükleme seçimi belirlenir. Karşımızda iki seçenek mevcuttur. Birinci seçenek olan Comlete seçeneğinde bütün program özelliklerinin yüklenmiş olacağını belirtir. Ancak bu seçeneğin daha fazla disk alanı gerektirdiği belirtilmektedir. Diğer seçenek Custom ise kullanıcının yüklemek istediği program özelliklerinin belirlenebileceği seçenektir. İleri seviye kullanıcılar için tavsiye edildiği belirtilmiştir.



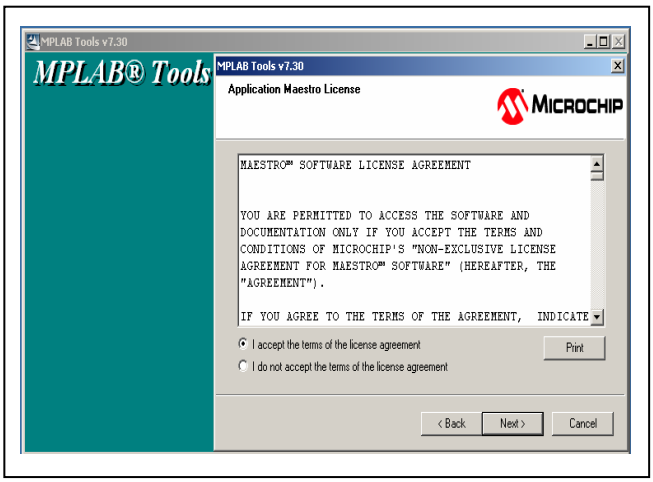
Şekil 3.10. Kur tipinin belirlenmesi

Şekil 3.11'de MPLAB v7.30b programının hangi dizin içerisine yükleneceği, hedef dizin(destination folder) etiketi içerisinde gösterilmiştir. Eğer kullanıcı programın bu hedef dizin içerisinde olmasını isterse next butonunu tıklaması yeterlidir. Farklı bir hedefteki dizine programı yüklenmek isterse o zaman göz at (Browse) butonunu tıklayarak seçim yapabilir.



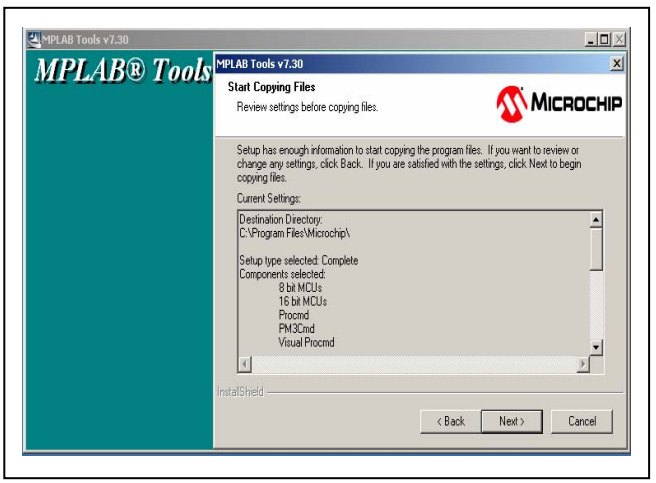
Şekil 3.11. Programın kurulacağı hedef dizinin belirlenmesi

Şekil 3.12'de maestro yazılım lisans anlaşması seçeneği sunulmuştur. Lisans anlaşmasının koşullarını kabul ediyorum seçenek düğmesi işaretlenerek next butonu tıklanır. Print butonu tıklanır ise varsayılan yazıcınızdan lisans anlaşmasının ayrıntılarını yazılı olarak alabilirsiniz.



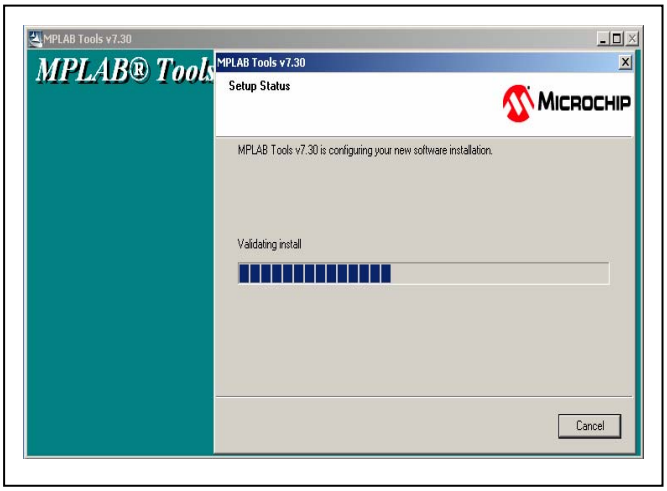
Şekil 3.12. Uygulama uzmanlık lisansı

Belirtilen hedef dizin içerisine program dosyalarını kopyalamadan önce kopyalanacak olan dosyaları gözden geçirerek next butonu tıklanır (Şekil 3.13). Eğer eksik dosya var ise geri (back) butonu tıklanarak geri adımlara dönülebilir ve varsa hatalar düzeltilebilir.



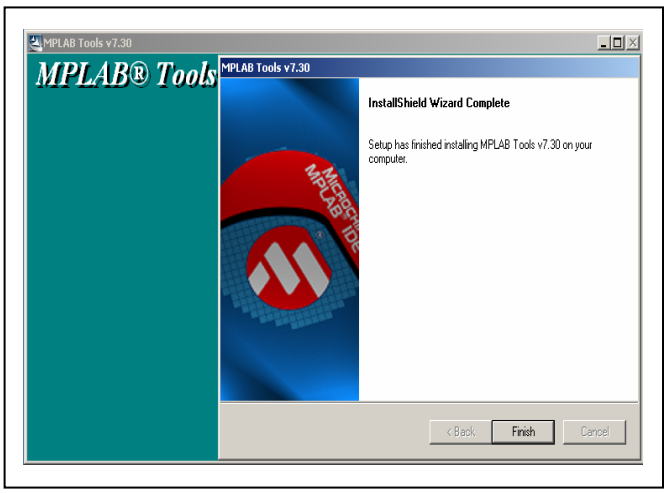
Şekil 3.13: Kopyalanacak program dosyalarını gözden geçirme

Şekil 3.13'te next butonu tıklanarak yükleme başlar ve şekil 3.14'te görülen pencere ekranda yerini alır. Ekrandaki pencerede MPLAB Tools v7.30 programı yeni yazılım kuruluşunuz ile yapılandırılıyor açıklaması yer almaktadır.



Şekil 3.14. MPLAB Tools v7.30 araçlarının yapılandırılması

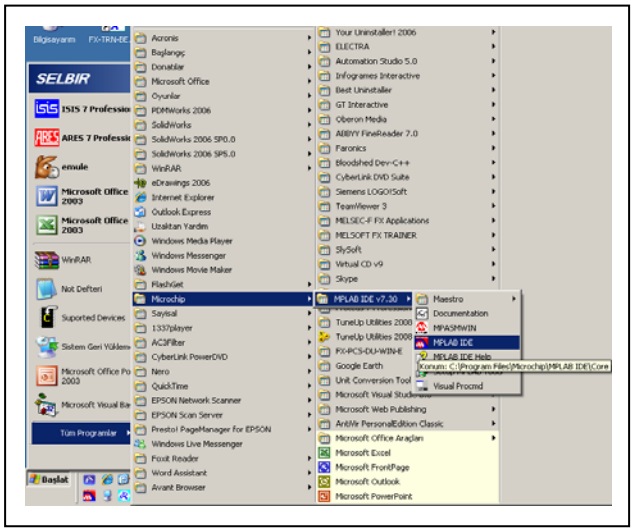
En son aşamada şekil 3.15'te, pencerede kurulumun tamamlandığını belirten bir açıklama görülecektir. Finish butonu tıklanarak kurulum bitirilerek sona erdirilir.



Şekil 3.15: InstallShield Sihirbazının Tamamlanması

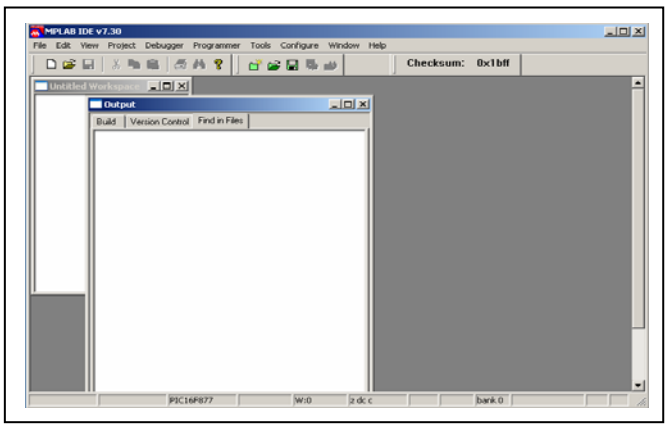
3.2. Mikro Denetleyici Program Editörünün Ayarları

Başlat menüsünden, Tüm programlar→Microchip→MPLAB IDE v7.30→MPLAB IDE seçildiğinde, MPLAB IDE v7.30 programı çalıştırılır(Şekil 3.16). Ayrıca masa üstünde de MPLAB IDE programının kısa yolu atanmıştır.



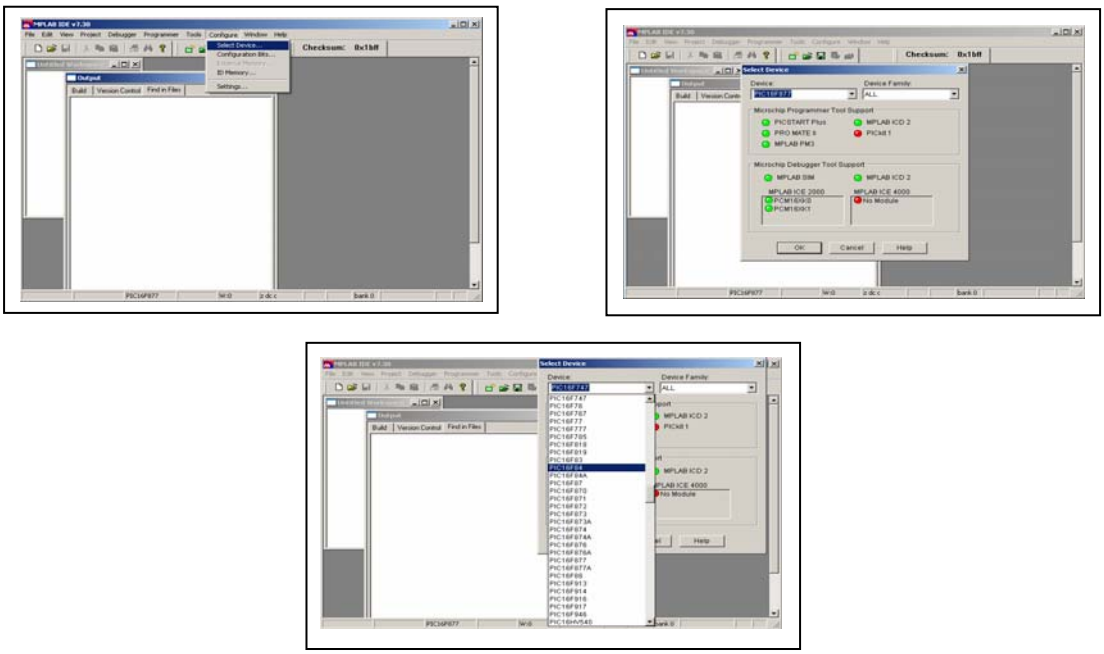
Şekil 3.16: MPLAB IDE programın başlatma

MPLAB IDE programı açıldığında program editörünü kullanmaya başlamadan önce bazı ayarlamalarının yapılması gerekir (Şekil 3.17).



Şekil 3.17. MPLAB IDE v7.30 programı

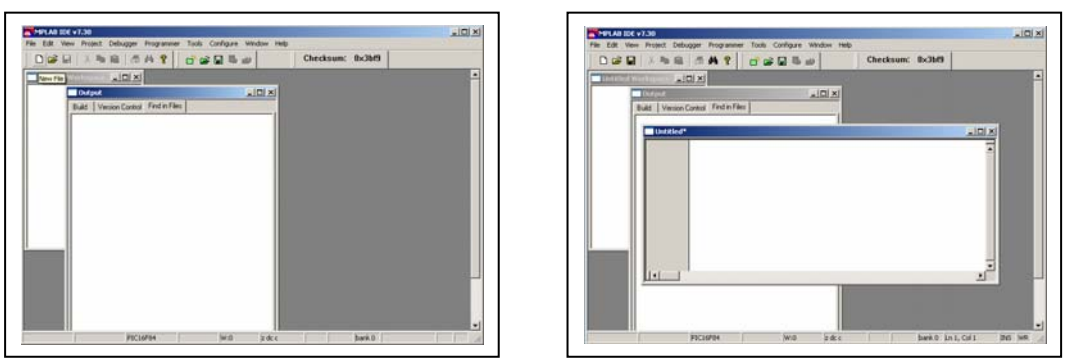
Yapılacak olan ilk ayarlama, kullanacağımız PIC entegresinin seçimi olmalıdır. Bunun için Configure menüsünden Select device tıklanarak açılan pencereden device (eleman) olarak kullanacağımız PIC seçilmelidir (Şekil 3.18).



Şekil 3.18: PIC seçimi

3.2.1. Programlama Editörü Ayarları

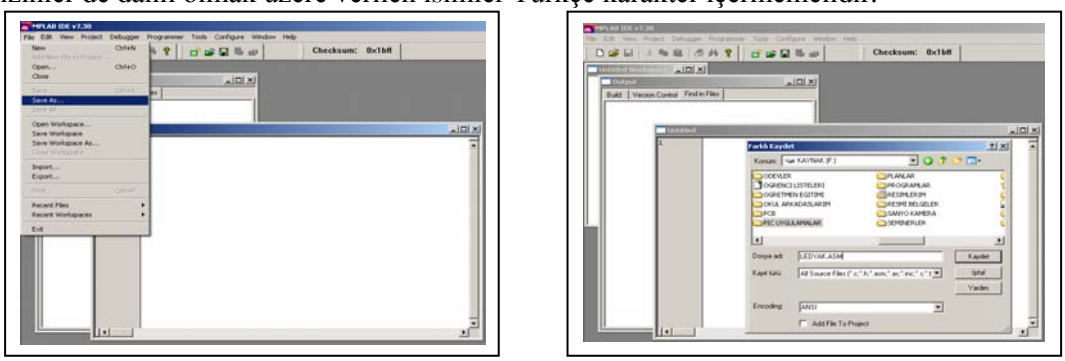
Program yazmaya başlamadan önce standart araç çubuğundan New File (Yeni dosya) tıklanarak yeni bir editör sayfası açılır (Şekil 3.19).



Şekil 3.19. Yeni sayfa açma

Çalışma sayfamızı kayıt edebiliriz. Şekil 3.20 de file (dosya) menüsünden save as (farklı kaydet) alt menüsü tıklanır. Açılan pencereden kayıt yerinin, dosya adının ve kayıt türünün belirlenmesi gerekmektedir. Kayıt yeri olarak yazdığınız programları belirlediğiniz ana dizin içerisinde alt dizin olarak saklayabilirsiniz. Dosya adı verilirken mutlaka program içeriğini açıklayıcı kısaltmalar kullanılması, dosyanın diğer zamanlarda aranması ve açmadan içeriğinin anlaşılabilmesi açısından önemlidir. Dosya adı verilirken uzantısının da

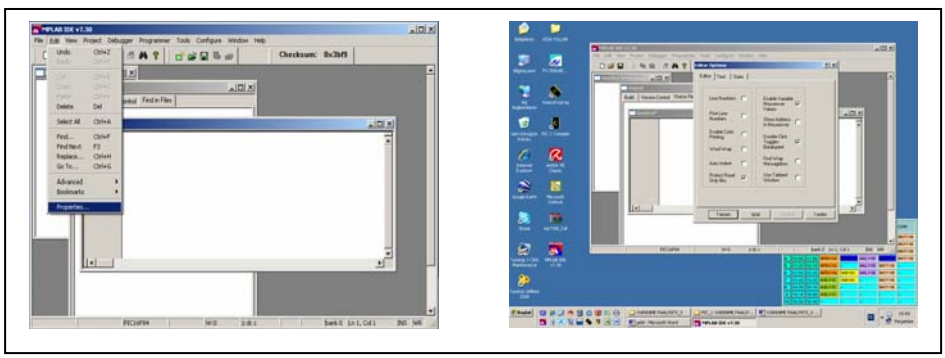
".ASM" olarak verilmelidir. Kayıt türü, "ASM" uzantılı kayıt türü seçilmelidir. Dosyayı derlerken hata mesajı verilmemesi için yazılan programlar kayıt edilirken ana dizin ve alt dizinler de dâhil olmak üzere verilen isimler Türkçe karakter içermemelidir.



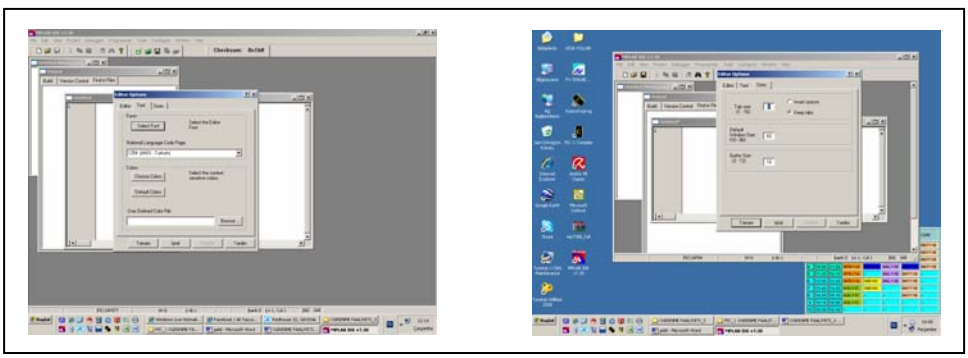
Şekil 3.20: Kayıt işlemi

Kayıt yeri, dosya adı ve uzantısı yazıldıktan sonra kaydet butonu tıklanarak kayıt işlemi tamamlanır.

Program yazarken satır numaraların görülmesi hatalı satır veya satırların bulunmasını kolaylaştırır. Ayrıca yazı tipi, sitili ve boyutu ile tab ayarları ile renk ayarları yapılabilir. Çalışma sayfamız açık iken edit menüsünden properties (özellikler) alt menüsü tıklanır. Açılan pencereden Editör-Text Sizes sekmelerinden ayarlarınızı yapabilirsiniz.



Şekil 3.21. Editör Ayarları



Şekil 3.22: Editör seçenekleri

3.2.2. Basit Bir Program

```
INCLUDE  "P16F84.INC"     ;Adresleri belirten dosyayı yükle                    (1)
LIST     P=16F84          ;16F84 ün tanıtılması                                (2)
BSF      STATUS,5         ;Bank 1'e geç                                        (3)
CLRF     TRISB            ;PORTB'nin hepsini çıkış yap                         (4)
BCF      STATUS,5         ;Bank 0'a geç                                        (5)
CLRF     PORTB            ;PORTB'nin bütün bitlerini 0 yap                     (6)
BSF      PORTB,0          ;PORTB'nin 0.bitini 1 yap(Led On)                    (7)
DEVAM    GOTO             DEVAM    ;Sonsuz döngü ile program sonlandırılır.    (8)
END                                                                            (9)
```

Bu program eğitim setindeki portb'ye bağlı 8 ledden 0.bit'e karşılık gelen led'i yakan diğerlerinin sönük kalmasını sağlayan bir programdır. Bu programı daha ayrıntılı olarak incelersek,

(1) PIC'lerin RAM belleğindeki özel yazmaç adreslerinin tanımlamalarının yapıldığı başlık dosyasıdır. Kullanılmaz ise programda her kullanılan özel isim verilmiş yazmaç adreslerinin tanımlanması gerekir. Başlık dosyaları Microchip firması tarafından geliştirilmiştir ve ".INC" uzantılı olarak kullanılmaktadır. Program yazarken en başa INCLUDE "P *mikro denetleyicinin kodu*.INC yazmak suretiyle tüm RAM bellekte bulunan adresleri tek satırda tanıtabiliriz.

(2) Hangi PIC çeşidinin kullanılacağını tanımlar. LIST bir talimat dilidir ve derleyici bildirisi olarak kullanılacak olan PIC özellikleri derleyiciye bildirilir.

(3) Status özel yazmacının 5. bit'i 1 yapılırsa Bank1'e geçilir. Bank1'de giriş-çıkış portlarının tanımlamaları yapılır. BSF komutu ile STATUS'un 5.bit'i 1 yapılır.

(4) Bank1'deki özel yazmaç olan TRISB, 8 bit'lik bir yazmaçtır. Bu yazmacın her bir bit'inin içine 1 yazılırsa PORTB giriş, 0 yazılırsa çıkış yapılır. Bir kısmı 1 ve diğerleri 0 yapılırsa, 1 yapılanlar giriş 0 yapılanlar da çıkış olarak belirlenmiş olur. CLRF komutu ile tümü 0 yapılır ve PORTB çıkış yapılır. Böylece PORTB pinlerine çıkış elemanları bağlanabilir.

(5) Programın çalışması Bank0'da gerçekleşir. Bank0'a geçmek için yine STATUS yazmacının 5.bit'i 0 yapılmalıdır. BCF komutu ile bu yazmacın 5. bit'i 0 yapılır.

(6) Daha önceki çalışmalardan ya da bilgisayar portundan gelen bir sinyal ile ledlerden biri yanık olabilir. CLRF komutu ile PORTB'ye bağlı olan ledlerin sönmesi sağlanır.

(7) BSF komutu ile PORTB'nin 0. bit'ine bağlı olan led yanar konumuna getirilir.

(8) GOTO komutu ile hedef etikete yönlendirilir. Böylece sonsuz bir döngü elde edilir.

(9) END komutu ile programın sonlandığı tanımlanır.

3.2.3. Sayıların İfade Edilmesi

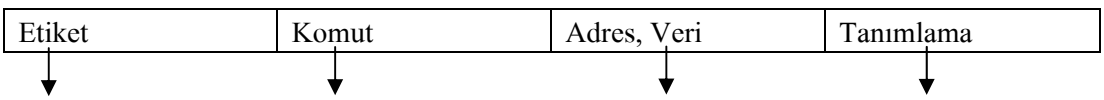
Program yazarken kullanılacak olan sayı ve karakterleri ifade edebilmek için aşağıya bir tablo çıkarılmıştır. Bu tabloda desimal, heksadesimal, oktal, binary ve ascii kodlarla ifade biçimleri verilmiştir.

Kodlama	Biçim	Örnek
Desimal	D '<digit>'	D '125'
Heksadesimal	H '<digit>' 0x <digit>	H '9F' 0x9F
Oktal	O '<digit>'	O '67'
Binary	B '<digit>'	B '00111011'
ASCII Kod	A '<digit>'	A 'C'

Tablo 3.1: MPASM için sayıların ifade edilmeleri

3.2.4. Programın Temel İfade Şekli

MPLAB editöründe assembly dili ile program yazarken uyulması gereken bazı kurallar vardır. Bu kurallara eğer dikkat edilirse program daha sonraki zamanlarda tekrar yazan kişi ya da başka bir programcı tarafından incelendiğinde daha anlaşılır ve açıklayıcı olacaktır. Program aşağıda belirtilen biçimde 4 sütun olarak ve tab tuşu ile ilerleyerek yazılması tavsiye edilir.



Etiket: Satırın en başında bulunur. Program bölümlerini birbirinden ayırmak için kullanılabilir. Aynı zamanda GOTO komutu ile hedefte belirtilen etiketlere program akışı yönlendirilebilir.

Komut: Belirtilen adresteki veriyi işler.

Adres, veri: Ram bellekteki özel fonksiyona sahip yazmaçlar ile programcının kullanımına bırakılmış yazmaçların adreslerinin yer aldığı sütundur. Aynı zamanda yazmaçlara aktarılması istenilen verilerde bu sütunda bulunur. Adres, veri sütununa yazılan etiket ismi, GOTO komutu ile aynı etiket isim satırına yönlendirilir.

Tanımlama: Tanımlama bölümüne hatırlatıcı açıklamaların yazıldığı bölümdür. Hatırlatıcı açıklamaların olduğu bölüm ";" ile ayrılmalıdır.

Program yazarken dikkat edilmesi gereken bazı noktalar vardır. Bunlar;

- İki alt çizgi ile başlamamalıdır.
- Bir alt çizgi ve sonrasında rakam yazılmamalıdır (2 gibi).
- Assembly dili komutları ve kullanıldığı kelimeler etiket olamaz.

- Bir etiket en fazla 32 karakter olabilir.
- Etiket yazımında büyük/küçük harf ayırımı olup büyük ve Türkçe olmayan harflerin kullanılması tavsiye edilir.
- Etiket yazılmayacağı zaman etiket sütunu boş bırakılmalıdır.
- Açıklama yazılacağı zaman ";" den sonra yazılmalıdır.
- Bir satırda en fazla 200 karakter olmalıdır.
- Her paragraf bir veya daha fazla boşlukla ayrılmalıdır.

Etiket	Komut	Adres, Veri	Tanımlama
	INCLUDE	"P16F84.INC	; Adresleri belirten dosyayı yükle
	LIST	P=16F84	; 16F84'ün tanıtımını yap
	BSF	STATUS,5	; Bank1'e geç
	CLRF	TRISB	; PORTB'nin hepsini çıkış yap
	BCF	STATUS,5	; Bank0'a geç
	CLRF	PORTB	; PORTB'nin hepsini 0 yap
	BSF	PORTB,0	; PORTB'nin 0. bitini1 yap(Led yak)
DEVAM	GOTO	DEVAM	; Sonsuz döngü ile programı sonlandır
	END		

3.2.4.1. Noktalı virgül (;)

Noktalı virgül ile başlayan satırlar assembler derleyicisi tarafından derlenmez. Bu satırlar programın başka programcılar tarafından da kolay bir şekilde anlaşılmasını sağlar. Program bölümlerini ayırmak için de kullanılabilir. Örneğin başlık bölümü ile atama bölümünü ayırmak istediğimizde satırın başına noktalı virgül yazarak ,"--------" veya "======" çizgileri ile ayırabiliriz.

3.2.4.2. Atama İfadesi (EQU)

EQU ifadesi İngilizcedeki "equal" kelimesinden türetilmiştir. Kullanılan PIC16f84'ün RAM belleğindeki heksadesimal adresleri, belirlenen etiketlere atamak için kullanılır.

Örnek 3.1:

STATUS	EQU	H '0003'
Etiket	Atama İfadesi	Heksadesimal Adres

3.2.4.3. Sabit Sayılar

PIC assembly dilinde sabit sayılar heksadesimal, decimal, binary ya da oktal olarak ifade edilebilir. Tablo 3.1'de sabit sayıların nasıl ifade edileceği açıklanmıştır. Sabit sayılar MOVLW, bazı mantıksal ve aritmetiksel işlemlerde kullanılırlar.

Örnek:

ADDLW B ‘00001111’

Komut Binary Sabit

3.2.4.4. ORG ifadesi

İngilizcedeki “origin” kelimesinden türetilmiş ve başlangıç noktası anlamına gelmektedir. ORG ifadesi iki nedenle kullanılır;

Program komutlarının hangi adresten itibaren başladığını derleyiciye bildirmek için kullanılır.

ORG 0x000

İlk program komutunun bellek adresi

PIC 16F84’ün kesme(interrupt) alt programlarının başlangıç adresini belirlemede kullanılır. Yani bir kesme olduğunda program döngüsünün belirtilen adresten itibaren başlamasını bildirmek için kullanılır.

ORG 0x004

Kesme alt programlarının başlangıç adresi

3.2.4.5. Sonlandırma

PIC 16F84’de duraklatma ya da aynı işlemi tekrar ettirme komutu yoktur. Programı istediğimiz bir satırda bekletmek istiyorsak, o satırda programı sonsuz bir döngüye sokabiliriz. Bunu yapabilmek için aşağıdaki komut satırı kullanılabilir.

```
DEVAM   GOTO    DEVAM
        END
```

Yukarıdaki sonsuz döngüde DEVAM etiketine assembler otomatik olarak bir adres atar. GOTO DONGU komutu ile sürekli aynı adrese yönlendirme yapılarak sonsuz döngü sağlanır. END ifadesi ise program komutlarının sona erdiğini derleyiciye bildirir. Her program sonuna END ifadesini yazarak bu satırdan sonra işlenecek komut bulunmadığı belirtilmek zorunluluğu vardır. END ifadesi kullanılmadığında program derlenirken dosya sonunun belirtilmediğini belirten bir hata mesajı verecektir.

3.2.4.5. Büyük/Küçük Harf Kullanımı

PIC assembly komutlarının büyük ya da küçük harfle yazılma zorunluluğu yoktur. Programcı isterse büyük ya da küçük harf kullanabilir. Hatta programcı büyük/küçük harf karışık olarak yazılmış komutları da kullanabilir. Ancak bu etiketler için geçerli değildir.

3.2.4.6. PIC Assembly Komutlarının Yazım Formatı

PIC 16F84'ün programlanmasında toplam 35 adet komut kullanılabilir. Bu komutların yazılış biçimini dört gurup şeklinde ifade edebiliriz.

- Bayt işleyen komutlar
- Bit işleyen komutlar
- Sabitleri işleyen komutlar
- Kontrol komutları

Assembly programlama komutları kullanılırken komutlarla beraber bazı semboller kullanılacaktır. İlgili sembolik değerler ve anlamları aşağıda sıralanmıştır.

W : 8 bitlik çalışma saklayıcısı (working register)

F : Bellek haritasındaki özel veya genel amaçlı saklayıcı (file register)

d : İşlem sonucunun W saklayıcısına (d=0) mı, F saklayıcısına (d=1) mı kaydedileceğini belirtir.

b : İşlenecek F saklayıcısının ilgilenilen bit numarasını gösterir (3 bitlik ikili sayıdır)

k : 8 bitlik sabit değerleri ve dallanma ve alt programlar için 10bitlik sabit adresleri gösterir.

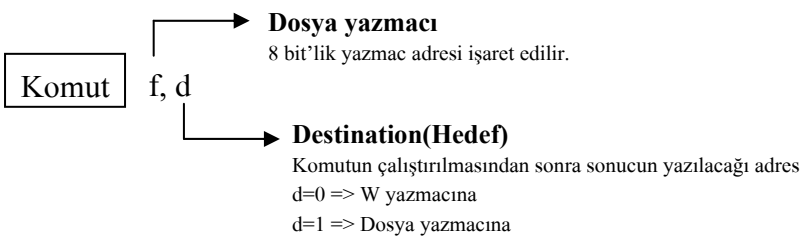
C : Elde bayrağı (toplama işleminden gelen elde ve çıkarma işleminden gelen borç)

DC : Ondalık elde bayrağı, düşük anlamlı 4 bitten gelen elde ve borç

Z : Sıfır bayrağı, işlem sonucunun sıfır olduğunu gösterir

- **Bayt işleyen komutlar**

8 bit'lik bir yazmaçlar üzerinde işlem yapan komutlardır. Yazmaç adı yerine bellek adresi de kullanılabilir.



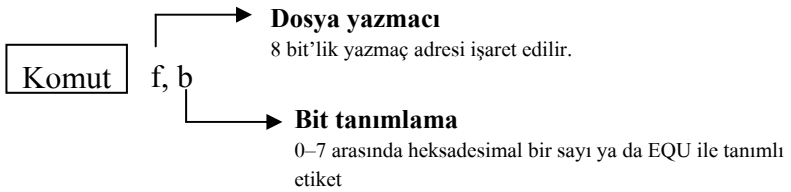
Örnek 3.2:

```
INCF    0x06, 0    ;0x06 adresindeki yazmaç içeriğini bir arttırır ve sonucu W
yazmacının içerisine kopyalar.
INCF    PORTB, 0   ;PORTB yazmacının içeriğini bir arttırır ve sonucu W
yazmacına kopyalar.
INCF    PORTB, 1   ;PORTB yazmacının içeriğini bir arttırır ve sonucu yine
kendi yazmacı içerisine kopyalar.
```

Bayt yönlendirmeli komutlarda hedef olarak belirtilen d ifadesi 0 ise W yazmacı 1 ise dosya yazmacı anlamını taşımaktadır. Fakat d ifadesi yerine 0 ya da 1 yazmak akılda kalıcı değildir. MPASM derleyicisi bu nedenle 0 yerine W,1 yerine de F ifadelerinin kullanımına izin verir.

- **Bit işleyen Komutlar**

8 bit'lik bir yazmacın sadece 1 bit'i üzerinde işlem yapan komutlardır. f ifade yerine heksadesimal adres ya da EQU komutu ile RAM bellekte tanımlanmış adres kullanılır. b ifade yerine 0–7 arasında heksadesimal bit değeri ya da EQU komutu ile tanımlanmış adresi olmalıdır.

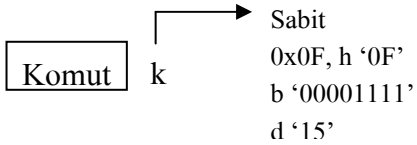


Örnek 3.3:

```
BCF     0x03,5          ;Ram bellekteki 0x03 adresindeki yazmacın 5. bit'ini
sıfırlar.
BSF     STATUS, Bit5    ;Ram bellekteki STATUS etiketi ile EQU komutu
kullanılarak tanımlanmış özel yazmacın yine Bit 5 etiketi ile tanımlanmış bit değerini
1 yapar.
```

- **Sabit işleyen komutlar**

8 bit ile ifade edilebilen sayılar üzerinde işlem yapan komutlardır.

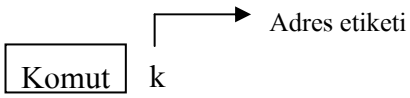


Örnek 3.4:

ADDLW 0x0F ;W yazmacı ile 0x0F sabit heksadesimal sayısını toplar.

MOVLW b ‘00001111’ ;b ‘00001111’ sabit binary sayısını W yazmacı içerisine kopyalar.

- **Kontrol komutları**



Örnek 3.5:

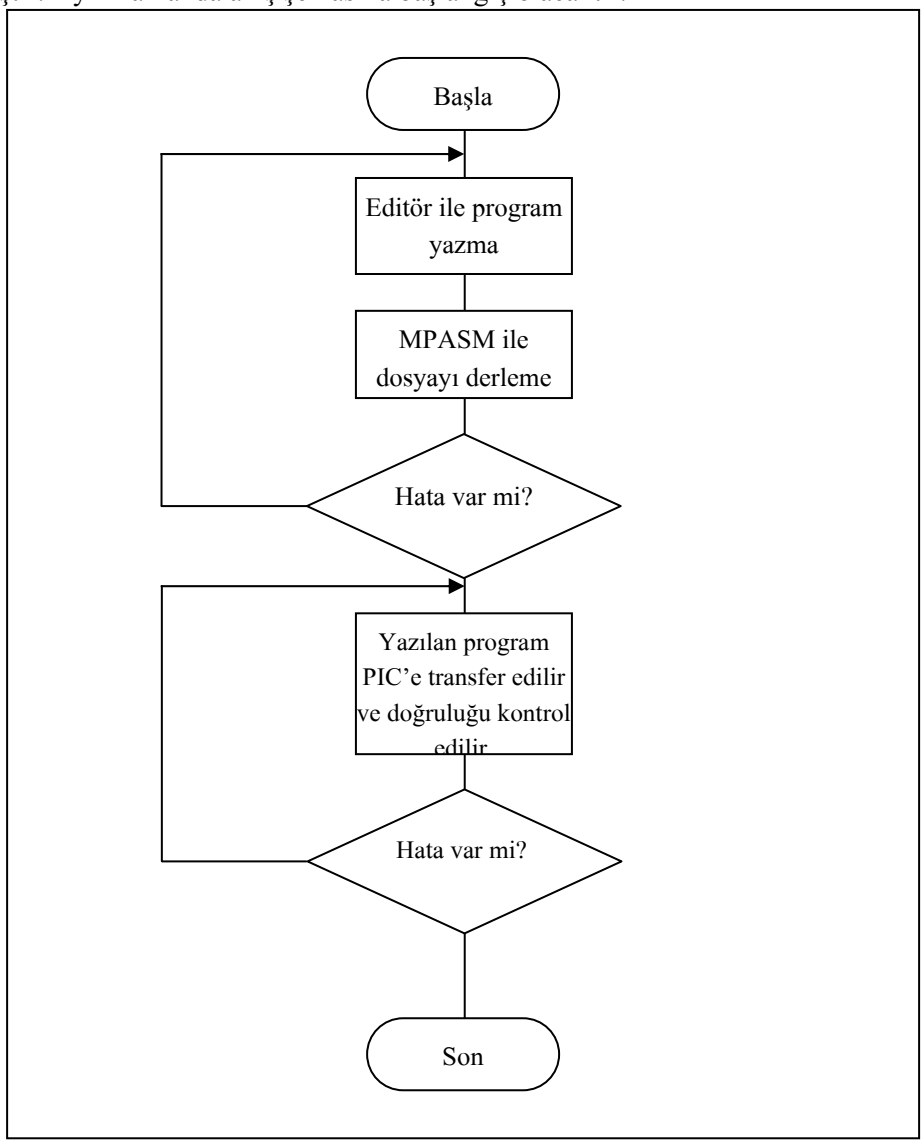
GOTO DEVAM ;Program akışı DEVAM isimli etikete yönlendirilir.

CALL TIMER ;Program akışı TIMER isimli etiketi ile belirlenen adresteki alt programa yönlendirilir.

NOT: Program içerisinde yazılan etiketlere derleyicinin (assembler) otomatik olarak adres verdiğini unutmayınız. Bu nedenle etiket adreslerini program içerisinde tanımlamaya gerek yoktur.

3.2.5. Program Yazmanın Yöntemi

Program yazmanın yönteminin akış diyagramı ile ifade edilmesi şekil 3.23'te verilmiştir. Aynı zamanda akış şemasına başlangıç olacaktır.

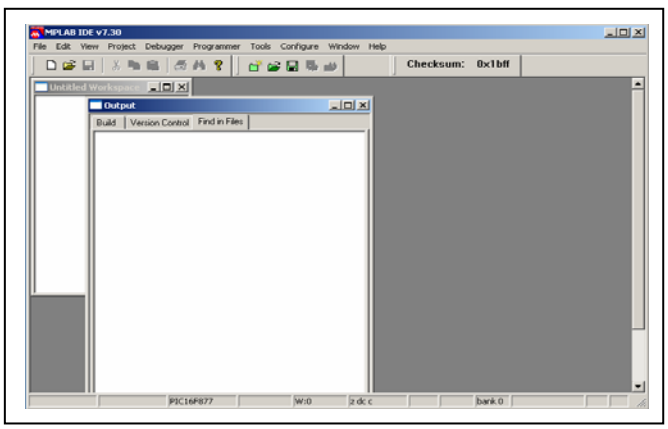


Şekil 3.23. Program yazmanın yöntemi

3.2.6. Programlama Editörünün Kullanımı

3.2.6.1. MPLAB programının başlatılması

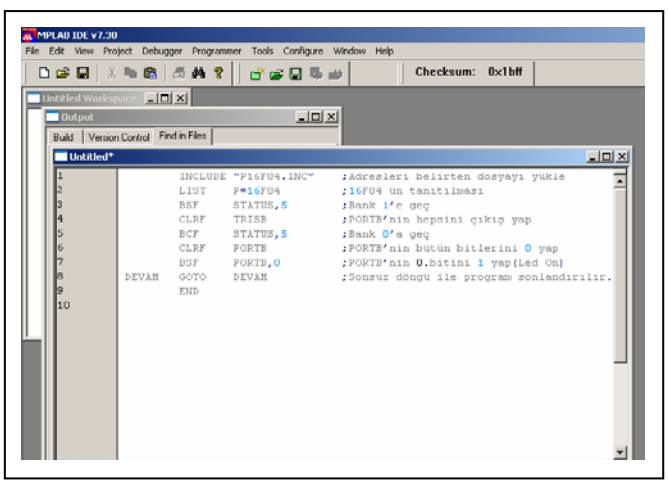
MPLAB programının başlatılması için masa üstü kısa yol ikonu çift tıklanarak ya da başlat → tüm programlar → Mikrochip → MPLAB IDE v7.30 → MPLAB IDE tıklanarak program başlatılır.



Şekil 3.24. MPLAB IDE başlama penceresi

3.2.6.2. MPLAB IDE Yazı Editörüne Program Yazılması

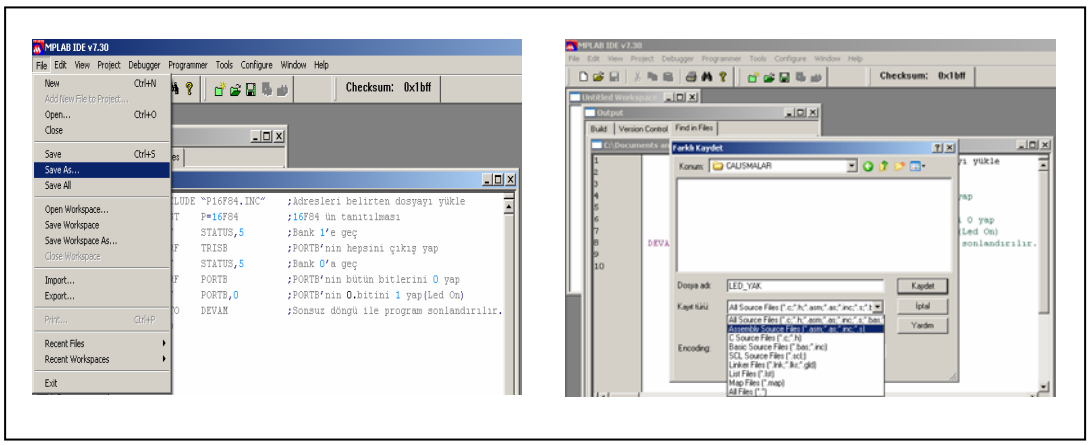
Açılan pencereden standart araç çubuğu içersinden yeni butonu tıklanarak boş bit editör sayfası açılır. Sayfa 12'deki basit programı editör sayfasına açıklama bölümlerini de ekleyerek yazınız.



Şekil 3.25: Program yazımı

3.2.6.3. Dosyanın Kayıt Edilmesi

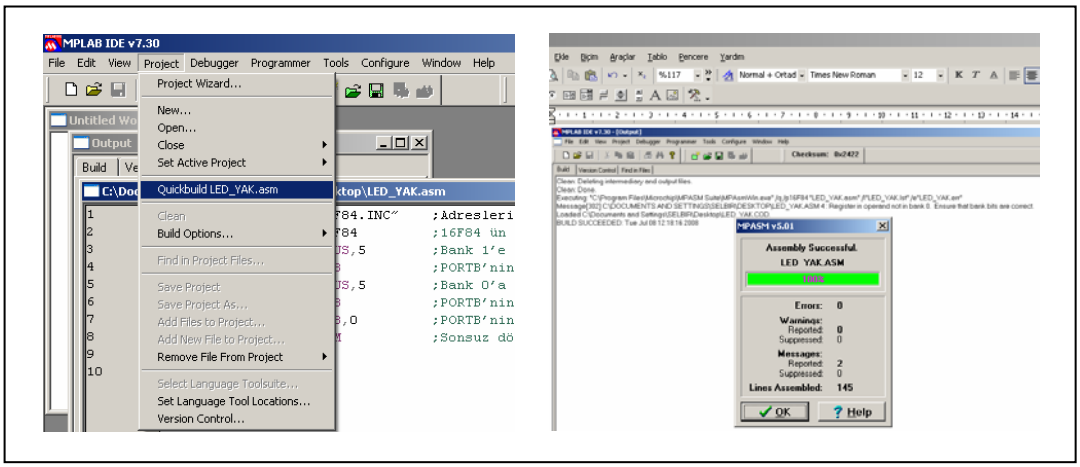
File → Save as tıklanarak açılan pencereden kayıt yeri, kayıt türü ve dosya adı belirlenerek dosya kayıt edilir. Sayfa 11'de dosya kaydı ile ilgili açıklamalar yapılmıştı. Bu açıklamalar dikkate alınarak dosya kaydı yapılmalıdır. Kayıt türü Assembly Source Files seçeneği işaretlendiğinde dosya adı verilirken uzantı verilmesine gerek kalmamaktadır. Program otomatik olarak ASM uzantısı verecektir.



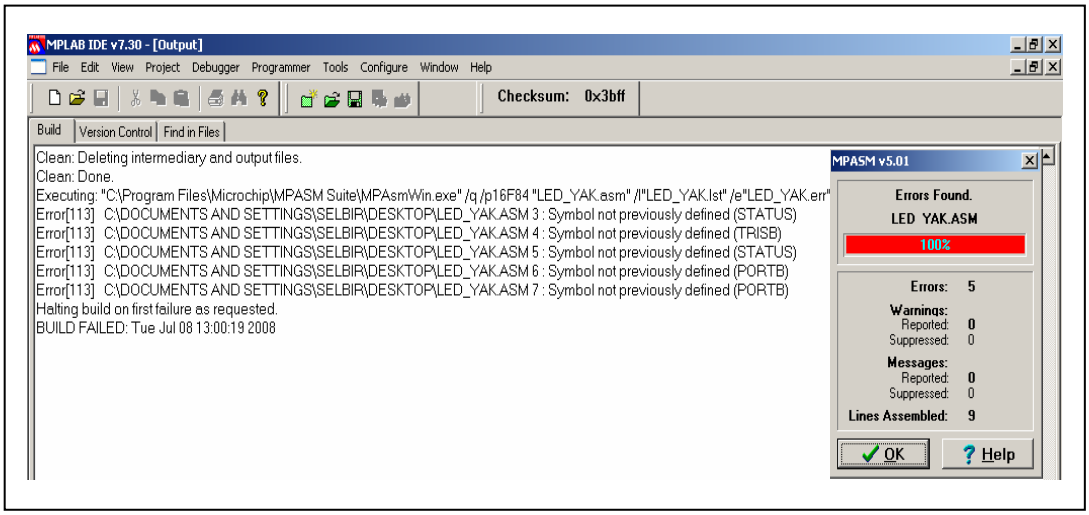
Şekil 3.26: Dosyanın Kayıt Edilmesi

3.2.6.4. Dosyanın Derlenmesi

Kayıt edilen programın derlenmesi gerekmektedir. Derleme işlemini iki yöntemle yapabiliriz. Menü çubuğundan Project menüsü tıklanır ve açılan alt menüden Quickbuild (dosya adı.asm) tıklanarak ya da ALT+F10 tuşlarına birlikte basılarak derleme işlemi başlatılır. Program içersinde yapılan ufak değişiklikler derleme işlemi ile kayıt da edilmiş olur. Bu nedenle önce tekrar kayıt işlemi yapılmasına gerek yoktur.



Şekil 3.27: Dosyanın derlenmesi ve başarılı olma penceresi



Şekil 3.28: Dosyanın derlenmesinde başarısız olma penceresi

Derleme işleminin başarılı olması programın doğru çalışacağı anlamına gelmemelidir. Derleme süresince sadece komut yazılım hatası olup olmadığı kontrol edilir, mantık hatası kontrolü yapılmaz.

Program yazılırken iki tür dosya vardır. Giriş dosyalarından olan kaynak dosyayı kendimiz oluştururuz. Yükleme dosyasını ise hazır olandan kullanırız.

(P16F84.INC gibi) Bu tür dosyalar MPLAB programı yüklenirken otomatik olarak gelmektedir. Çıktı dosyaları ise derleme sonucunda oluşan dosyalardır ve kaynak dosyanın bulunduğu klasörün içerisindedirler.

- **GİRİŞ DOSYALARI**
  - Kaynak dosya (dosya adı.ASM)
  - Yükleme (Include) dosya (dosya adı.INC)
- **ÇIKTI DOSYALARI (Derleme esnasında assembler tarafından üretilir)**
  - Liste dosyası (dosya adı.LST)
  - Hata dosyası (dosya adı.ERR)
  - Sembol ve debug dosyası (dosya adı.COD)
  - Hex formatında dosya (dosya adı.HEX, dosya adı.HXL, dosya adı.HXH)
  - **Liste dosyası**

Dosyanın hangi yazılım tarafından üretildiği, versiyonu, tarih ve saati, sayfa numarasından sonra;

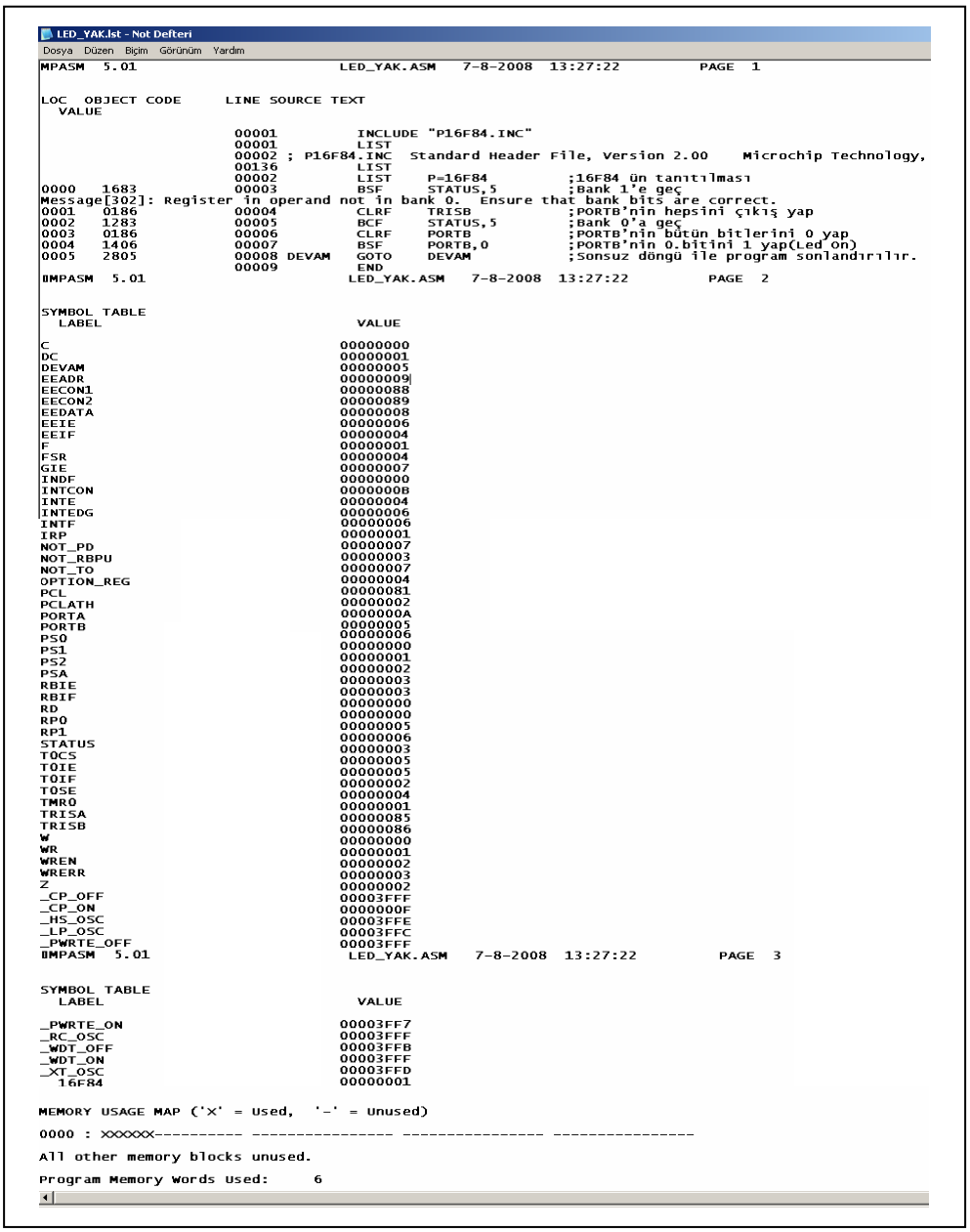
Birinci kolonda komutların hafızadaki adreslerini, (LOC)

İkinci kolonda set, equ, variable gibi emirlerin ve komutların oluşturduğu 32 bit'te heksadesimal karşılıklarını, (OBJECT CODE)

Üçüncü kolonda kaynak program ve bunun satır numaraları ve varsa açıklamalar (LINE SOURCE TEXT)

Programda kullanılan etiketler (SYMBOL TABLE) ve adresleri (VALUE)
Hafıza kullanım haritası (MEMORY USAGE MAP)
X Kullanılan kısım - Kullanılmayan kısmı gösterir
PROG1.ASM örnek programında kullanılan alan 6 kullanılmayan alan 1018' dir.

Hata (Error) sayısı ve Uyarı (Warning) sayısı da verilmektedir. Bu uyarı ve hataların nerelerde olduğu ise açıkça komutların altlarına yazılmıştır.



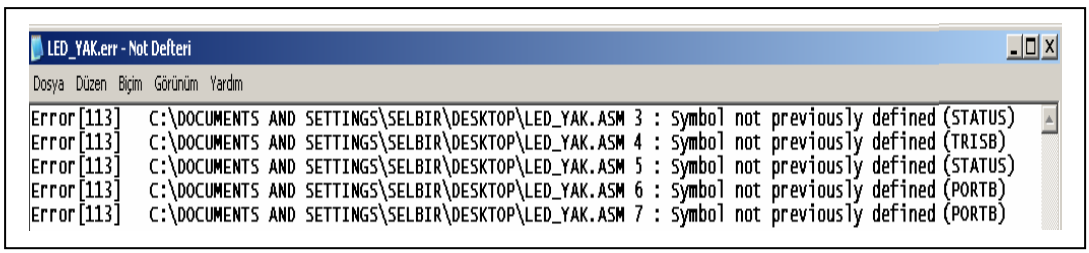
```
LED_YAK.lst - Not Defteri
Dosya  Düzen  Biçim  Görünüm  Yardım
MPASM  5.01                    LED_YAK.ASM   7-8-2008  13:27:22         PAGE  1


LOC  OBJECT CODE     LINE SOURCE TEXT
  VALUE

                     00001         INCLUDE "P16F84.INC"
                     00001         LIST
                     00002 ; P16F84.INC  Standard Header File, Version 2.00    Microchip Technology,
                     00136         LIST
                     00002         LIST    P=16F84          ;16F84 ün tanıtılması
0000   1683          00003         BSF     STATUS,5         ;Bank 1'e geç
Message[302]: Register in operand not in bank 0.  Ensure that bank bits are correct.
0001   0186          00004         CLRF    TRISB            ;PORTB'nin hepsini çıkış yap
0002   1283          00005         BCF     STATUS,5         ;Bank 0'a geç
0003   0186          00006         CLRF    PORTB            ;PORTB'nin bütün bitlerini 0 yap
0004   1406          00007         BSF     PORTB,0          ;PORTB'nin 0.bitini 1 yap(Led On)
0005   2805          00008 DEVAM   GOTO    DEVAM            ;Sonsuz döngü ile program sonlandırılır.
                     00009         END
MPASM  5.01                    LED_YAK.ASM   7-8-2008  13:27:22         PAGE  2


SYMBOL TABLE
  LABEL                             VALUE

C                                 00000000
DC                                00000001
DEVAM                             00000005
EEADR                             00000009
EECON1                            00000088
EECON2                            00000089
EEDATA                            00000008
EEIE                              00000006
EEIF                              00000004
F                                 00000001
FSR                               00000004
GIE                               00000007
INDF                              00000000
INTCON                            0000000B
INTE                              00000004
INTEDG                            00000006
INTF                              00000006
IRP                               00000001
NOT_PD                            00000007
NOT_RBPU                          00000003
NOT_TO                            00000007
OPTION_REG                        00000004
PCL                               00000081
PCLATH                            00000002
PORTA                             0000000A
PORTB                             00000006
PS0                               00000000
PS1                               00000001
PS2                               00000002
PSA                               00000003
RBIE                              00000003
RBIF                              00000000
RD                                00000000
RP0                               00000005
RP1                               00000006
STATUS                            00000003
T0CS                              00000005
T0IE                              00000005
T0IF                              00000002
T0SE                              00000004
TMR0                              00000001
TRISA                             00000085
TRISB                             00000086
W                                 00000000
WR                                00000001
WREN                              00000002
WRERR                             00000003
Z                                 00000002
_CP_OFF                           00003FFF
_CP_ON                            0000000F
_HS_OSC                           00003FFE
_LP_OSC                           00003FFC
_PWRTE_OFF                        00003FFF
MPASM  5.01                    LED_YAK.ASM   7-8-2008  13:27:22         PAGE  3


SYMBOL TABLE
  LABEL                             VALUE

_PWRTE_ON                         00003FF7
_RC_OSC                           00003FFF
_WDT_OFF                          00003FFB
_WDT_ON                           00003FFF
_XT_OSC                           00003FFD
__16F84                           00000001

MEMORY USAGE MAP ('X' = Used,  '-' = Unused)

0000 : XXXXX------- ---------------- ---------------- ----------------

All other memory blocks unused.

Program Memory Words Used:     6
```

Şekil 3.29. LED_YAK.LST dosyası

- **Hata (error) Dosyası**

MPASM assembler derleme sonunda hata çıktısı olarak OUTPUT penceresinde gösterdiği kısmı ERR uzantılı bir dosyada oluşturur. Bu dosya içerisinde programda hata varsa bunun hangi satırda olduğu ve açıklamasını buluruz. Hata yok ise bu dosyanın içeriği boştur.

Örneğin hata var ise aşağıdaki gibi bir dosya oluşmuştur. Burada hatanın bulunduğu satır numarası ve ne gibi hata olduğu açıkça görünmektedir.



Şekil 3.30: LED_YAK.ERR dosyası

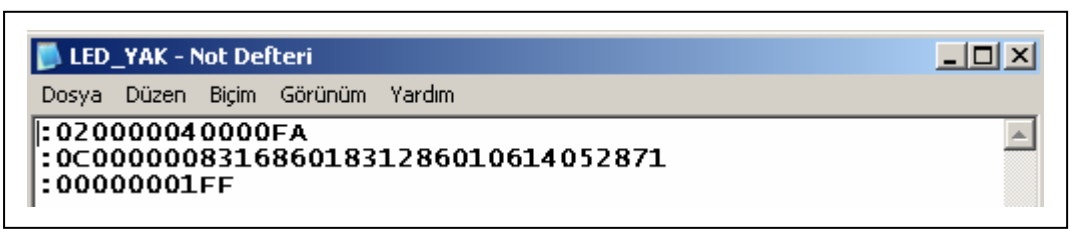
LİSTE dosyasında hata olan satırın hemen üstünde ne gibi hata olduğu açıkça yazdığından dolayı hatayı bulmak LİSTE dosyasında daha kolaydır.

- **HEX Formatında Dosya**

MPASM assembler derleme sonunda başarılı oldu ise kaynak dosya adında fakat uzantısı HEX olan bir dosya meydana getirir. Bu dosya bizim PIC'e kayıt edeceğimiz dosyadır. Değişik formatlarda derleme yapılabilmektedir.

Bunlar;

Intel Hex Format	INHX8M	.hex	8 bit
Intel Split Hex Format	INHX8S	.hxl .hxh	tek çift
Intel Hex 32 Format	INHX32	.hex	16 bit
Intel Hex Format	INHX8M		



Şekil 3.31: LED_YAK.HEX dosyası

Intel HEX Format (INHX8M) aşağıdaki gibidir.

:BBAAAATTHHHH……..HHCC

BB : İki heksedesimal sayı digiti ile bir satırdaki data sayısını gösterir.

AAAA: Dört heksedesimal sayı digiti ile ilk datanın kayıt edildiği adresi gösterir.

TT : Kayıt şeklidir 00 = Data kayıt 01 = Data kayıt sonu

HH : İki heksedesimal sayı digiti ile 1 bayt datayı gösterir.

CC : Toplam kodu kontrol eder (Bütün baytların toplamının ikili complementi)

3.3. Basit Bir Programın Açıklanması

Pic programının içersinde akış diyagramındaki görevleri yerine getirebilmesi için bazı tanımlamalarımn ve yöntemlerin belirlenmesi gerekir. Şimdi bir programa ait detayları inceleyelim.

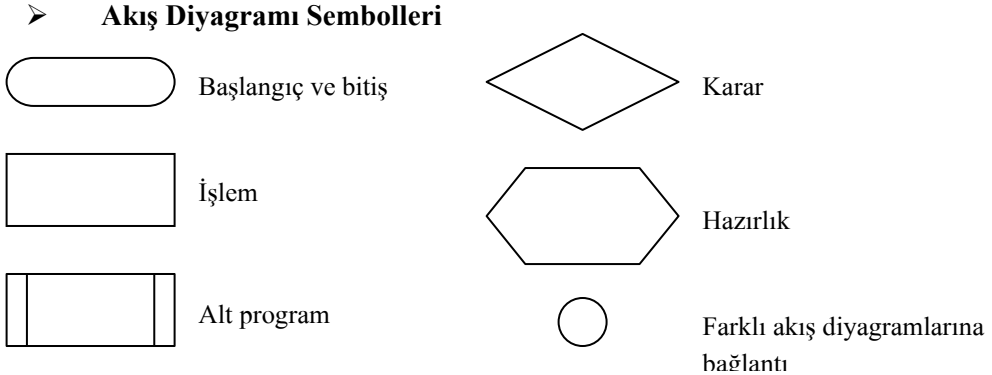
3.3.1. Akış Diyagramı ve Program Listesi

Programın yazılımına geçmeden önce akış diyagramının yapılması gereken işlemlere göre tasarımı gerçekleştirilir. Akış diyagramına göre program yazılır.

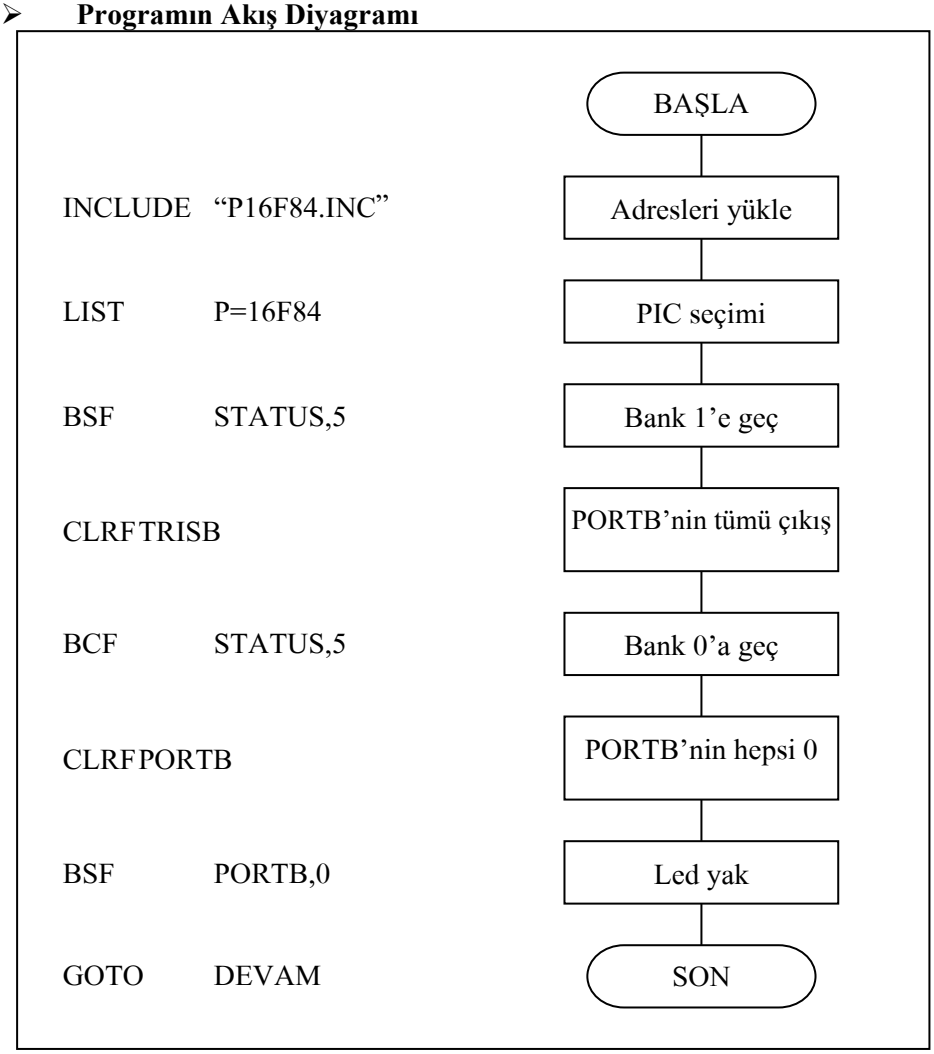
3.3.1.1. Akış Diyagramı

Genellikle program yazmadan önce akış diyagramı çizmekte büyük yarar vardır. Akış Diyagramları programcının ve daha sonra programı geliştirecek bir başkasının rahatça anlayabileceği tarzda olmalıdır. Asıl olan budur. Aşağıda akış diyagramı için kullanılan bazı standart semboller verilmiştir.

- **Akış Diyagramı Sembolleri**



➢ **Programın Akış Diyagramı**



Şekil 3.32: Akış diyagramı

3.3.1.2. Program Listesi

Akış diyagramına göre yazılmış program aşağıdadır.

```
INCLUDE   "P16F84.INC"    ;Adresleri belirten dosyayı yükle      (1)
LIST      P=16F84         ;16F84 ün tanıtımını yap               (2)
BSF       STATUS,5        ;Bank 1' e geç                         (3)
CLRF      TIRSB           ;PORTB nin hepsini çıkış yap           (4)
BCF       STATUS,5        ;Bank 0' a geç                         (5)
CLRF      PORTB           ;PORTB' nin hepsini 0 yap              (6)
BSF       PORTB,0         ;PortB' nin 0.bitini 1 yap ( LED yak ) (7)
DVM GOTO      DVM         ;Sonsuz döngü ile programı sonlandır   (8)
END                                                              (9)
```

3.3.2. Yöntem Şartnamesi

Anlatılan örnek ve uygulamalarda kullandığımız PIC tipi ya da kodu 16F84 olduğunu öncelikle hatırlatalım. Ayrıca program yazmaya başlarken LIST ifadesi ile başladığımızı birkez daha söyleyerek LIST ifadesi hakkında kısa bilgi verelim.

LIST bir talimat dilidir ve LIST talimatı ile derleyiciye hangi PIC çeşidini ve o PIC'in hangi özelliklerini kullanacağımızı bildiririz ki derleyicide bizim PIC'imize uygun hex dosyası oluşturabilsin. LIST talimatı kullanımı aşağıdaki biçimde olmalıdır.

List [<list_opsiyon>,…,<list_opsiyon>]

Örnek:

LIST P=16F84,F=INHX8M,R=DEC

3.3.2.1. List talimatları

Program yazılırken bazı standartların programımızın en başında List komnutu ile beirlenmesi gerekir. Bu standartların özellikleri aşağıdaki tabloda listelenmiştir.

Seçenek	Varsayım	Tanım
b=nnn	8	TAB boşluk sayısının tanımı
c=nnn	132	Bir satırdaki karakter sayısının tanımı
f=<format>	INHX8M	Derleme sonrasında elde edilecek HEX uzantılı dosya özellikleri
free	FIXED	Serbest form
fixed	FIXED	Sabit form
mm=ON|OFF	ON	Bellek haritasının listede belirlenmesi ON|OFF
n=nnn	60	Bir sayfadaki satır sayısının tanımı
p=<type>	None	İşlemci tanımı (Example: P=,16F84)
r=<radix>	HEX	Numerik değer tipi <HEX,DEC,OCT>
st=ON|OFF	ON	Sembol tablosunun listeye yazılması ON|OFF
t=ON|OFF	OFF	Satır değiştirme < satır taşması sonucu > ON|OFF
w=0|1|2	0	Assemblerin mesaj seviye tanımı
x=ON|OFF	ON	Makro kullanımı ON|OFF

Not: 'nnn' desimal sayılarla tanımlama eklenmesidir.

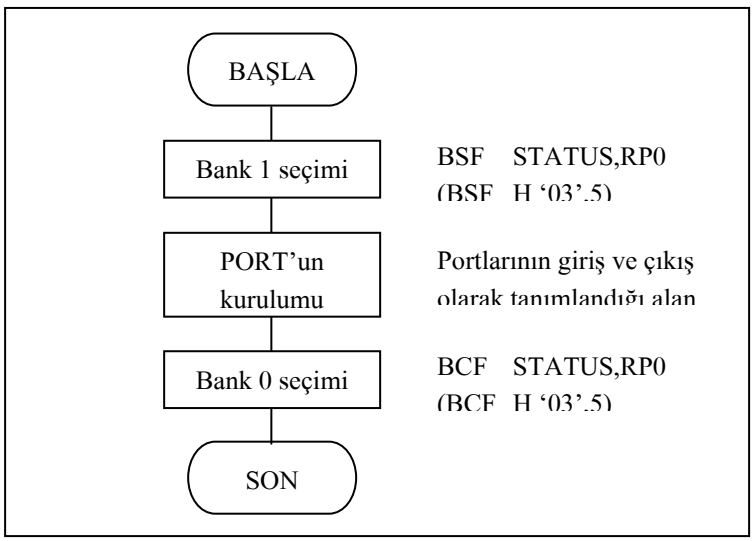
Tablo 3.2: List opsiyonları

3.3.3. Portun Kurulumu

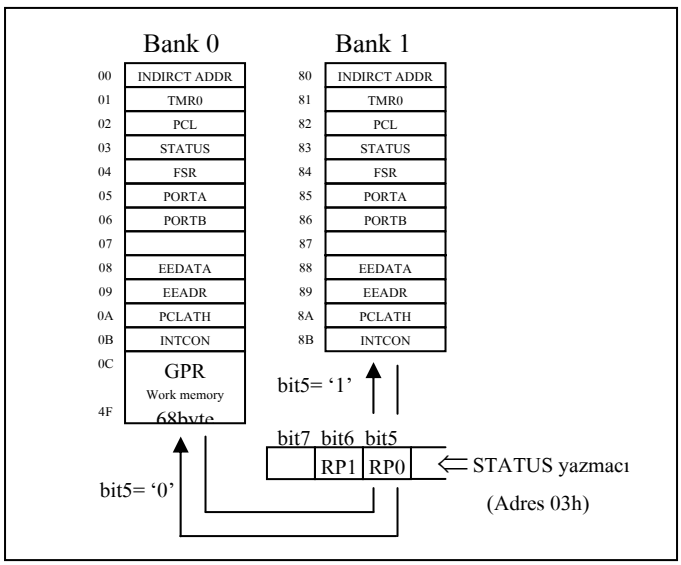
Mikro denetleyicide bulunan portlar bizim ihiyacımıza göre giriş yada çıkış olarak belirlebilir. Bu işlemlerin program içersinde belirtilmesi gerekir.

3.3.3.1. Bank seçimi

PIC'i kullanmaya başlamadan önce portların kurulması (I/O belirlenmesi) gerekmektedir.



Şekil 3.33. I/O Port'un Kurulumu



Şekil 3.34: Bank seçimi

PIC'in giriş/çıkış (I/O) portlarının belirlendiği yazmaç (**S**pecial **F**unction **R**egister) (Özel fonksiyon yazmacı) (TRISA(85h) ve TRISB (86h) olarak adlandırılır.

TRIS yazmacları Bank1'dedir. Bank1'e geçiş STATUS yardımı ile olur. Statusun adresi de (03h)' dir. (Şekil 3.34)

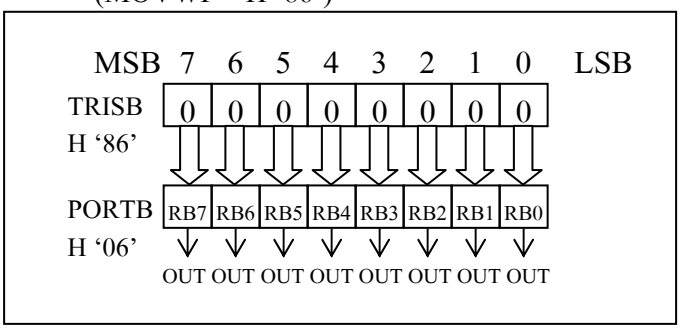
Bank 1 seçimini gösterdiği örnek program aşağıdaki gibidir:

```
BSF   STATUS,RP0
( BSFH'03',5 )
                                Bank1 seçimi

                    (PORT ların I/O olarak kurulumu. )
BCF  STATUS,RP0
( BCF        H'03',5 )
                                Bank0 seçimi
```

Portların kurulumu da aşağıdaki gibidir.

Örnek 3.6: PORTB nin hepsi çıkış olursa:

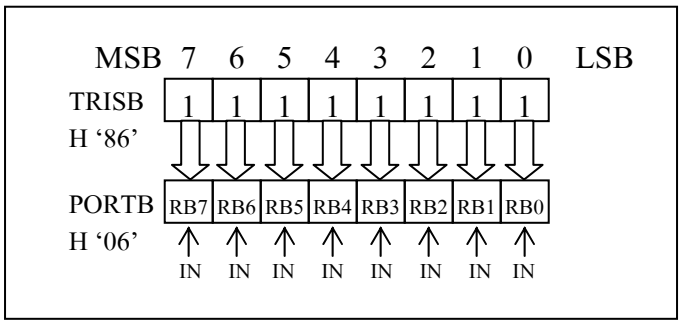
```
Form1       CLRF TRISB
            (CLRF      H '86'

Form2       MOVLW      H '00'      ; W yazmacına << 00h
            MOVWF      TRISB       ; TRISB << W yazmacı
            (MOVWF     H '86')
```



Şekil 3.35: PORTB' nin hepsi çıkış

Örnek 3.7: PORTB'nin hepsi giriş:

```
Form1       MOVLW      H 'FF'      ; W yazmacı << FFh
            MOVWF      TRISB       ; TRISB << W yazmacı
            (MOVWF     H '86')

Form2       MOVLW      B '11111111'     ; W yazmacı << FFh
            MOVWF      TRISB            ; TRISB << W yazmacı
            (MOVWF     H '86')
```

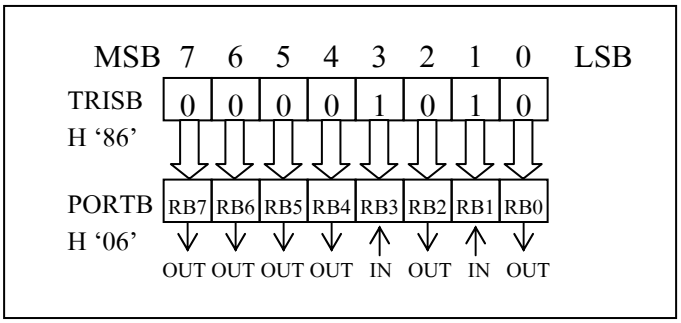


Şekil 3.36: PORTB' nin hepsi giriş

Örnek 3.8: PORTB bit0, bit2, bit4, bit5, bit6, bit7 >> ÇIKIŞ

Bit1, bit3 >> GİRİŞ

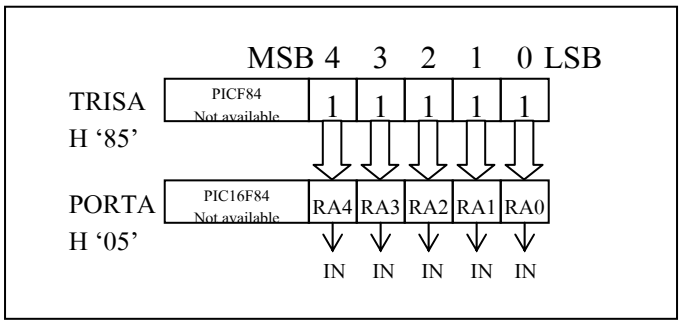
```
Form1       MOVLW   H '0A'          ; W yazmacı << 0Ah
            MOVWF   TRISB          ; TRISB << W yazmacı
            (MOVWF  H '86')

Form2       MOVLW   B '00001010'   ; W yazmacı << 0Ah
            MOVWF   TRISB          ; TRISB << W yazmacı
            (MOVWF  H '86')
```



Şekil 3.37: PORTB' de farklı giriş-çıkışlar

Örnek 3.9: PORTA'nın hepsi giriş

```
Form1       MOVLW   H '1F'          ; W yazmacı << 1Fh
      MOVWF   TRISA          ; TRISA << W yazmacı
      (MOVWF  H '85')

Form2       MOVLW   B '11111'      ; W yazmacı << 1Fh
      MOVWF   TRISB          ; TRISA << W yazmacı
      (MOVWF  H '85')
```



Şekil 3.38: PORTA' nın hepsi giriş

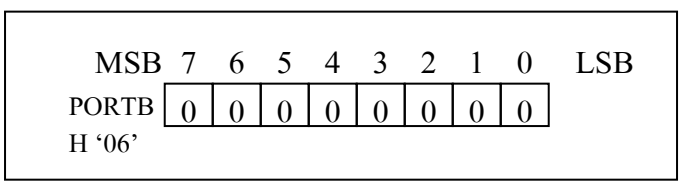
3.3.4: Ledlerin Yakılması

PORTB'deki herhangi bir led'in yakılmasına ait ana program aşağıda verilmiştir. Programın bu bölümüne ana program denir.

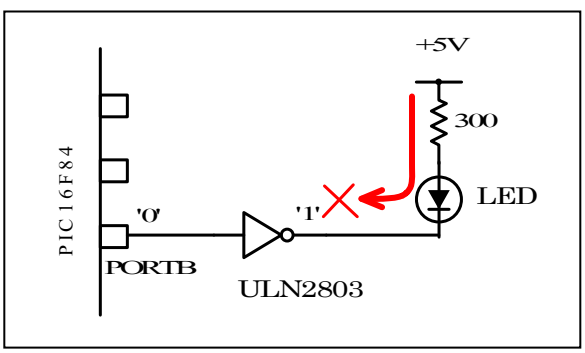
```
1        CLRF     PORTB     ; PORTB'nin tamamı 0 >> Bütün Led'ler sönük
2        BSF      PORTB,0   ; PORTB'nin 0.bitindeki Led yanar
3        (BSF     H '06',0)
4        STOP GOTO      STOP
5        END
```

3.3.4.1. CLRF PORTB

PORTB'nin adresi 06h dir. CLRF komutu ile bu adresteki bilgileri 0 hâline getiririz.



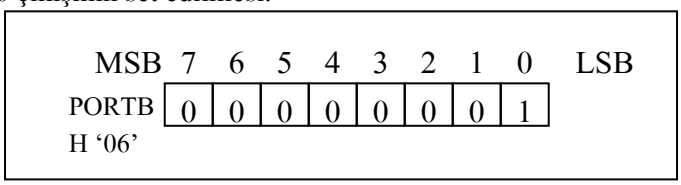
Şekil 3.39: PORTB' nin Temizlenmesi



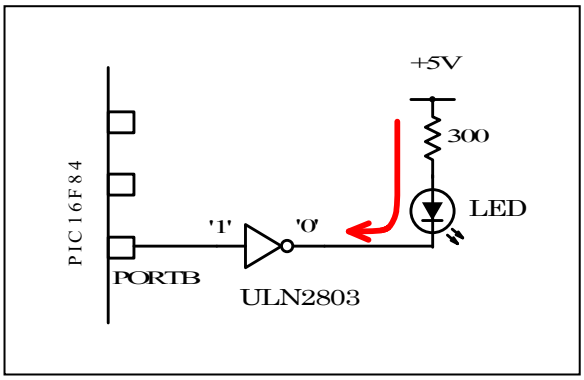
Şekil 3.40. PORTB >> 0 (Led sönük)

3.3.5. BSF PORTB,0 (BSF H ‘06’,0)

B portunun 0 çıkışının set edilmesi.



Şekil 3.41: Bit0’ın set(1) Hâle getirilmesi



Şekil 3.42: PORTB >> 1 (Led yanık)

3.4. Mikro Denetleyici Komutları

Mikro deneteliyicide asembler komutları aşaığıdaki tabloda listelenmiştir. Toplam 35 komutla bir çok işlemi gerçekleştirebiliriz.

1	ADDLW		19	IORLW
2	ADDWF		20	IORWF
3	ANDLW		21	MOVF
4	ANDWF		22	MOVLW
5	BCF		23	MOVWF
6	BSF		24	NOP
7	BTFSC		25	RETFIE
8	BTFSS		26	RETLW k
9	CALL		27	RETURN
10	CLRF		28	RLF
11	CLRW		29	RRF
12	CLRWDT		30	SLEEP
13	COMF		31	SUBLW
14	DECF		32	SUBWF
15	DECFSZ		33	SWAPF
16	GOTO		34	XORLW
17	INCF		35	XORWF
18	INCFSZ			

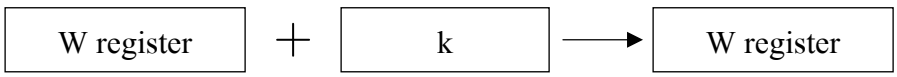
Tablo 4.1. PIC komutları

3.4.1. ADDLW (Add Literal ve W)

Komut ADDLW k

İşleçler 0 <= k <= 255

İşlem



Status Etkisi C,DC,Z flag(bayrak)

Açıklama W registerinin içeriğini k sabit sayısı ile toplayıp sonucu W registerine yazar.

Kelimeler 1

Saat çevrimi (Cycle) 1

Örnek

◎ (W) << (W)+30h (W) << (W)+1
ADDLW H'30' ADDLW 1

◎ (W) << (W)-1
ADDLW H'FF'

◎ MOVLW H'63' 01100011
ADDLW H'AF' + 10101111
100010010
(W) << H'12' C flag = 1

Kodlama

1 1	1 1 1 x	k k k k	k k k k

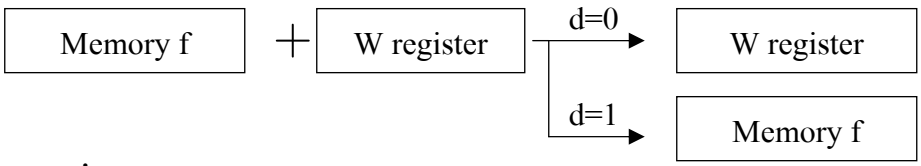
Q Çevrimi (Q Cycle Activity)

Q1	Q2	Q3	Q4
Kod çöz	'k' literalini oku	İşlemi yap	W'ye yaz

3.4.2. ADDWF (Add W ve f)

Komut ADDWF f,d

İşleçler 0 <= f <= 127
d ə {0,1}



İşlem

Status Etkisi C,DC,Z flag

Açıklama W registerinin içeriğini 'f' registeri ile toplar. Eğer 'd' 0 olursa sonucu W registerine 1 olursa f registerine yazar.

d = 0 ise W registerine
d = 1 ise f registerine

Kelimeler 1

Saat çevrimi (Cycle) 1

Örnek ◎ (W) << f + (W) FSR << f +(W)
ADDWF f,0 ADDWF f,1

◎ ADDWF f,0

İşlem öncesi W = 05h , f = F5h
İşlem sonrası W = FAh , f = F5h

Kodlama

0 0	0 1 1 1	d f f f	f f f f

Q Çevrimi (Q Cycle Activity)

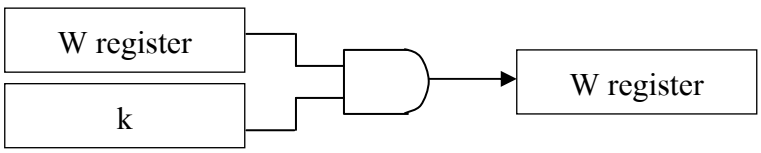
Q1	Q2	Q3	Q4
Kod çöz	f registerini oku	İşlemi yap	Gideceği yere yaz

3.4.3. ANDLW (AND literal with W)

Komut ANDLW k

İşleçler 0 <= k <= 255

İşlem



Status Etkisi Z flag

Açıklama W registerin içeriği ile k sabitine AND (ve) işlemi uygulanır ve sonuç W registerine yazılır.

Kelimeler 1

Saat çevrimi (Cycle) 1

Örnek

◎ ANDLW B'11110000' low 4 bits clear
ANDLW 0 = CLRW

◎ ANDLW H'FF' test W register

◎ bit test(0 test of low 3 bits of W register)
ANDLW B'00000111'
BTFSC STATUS,Z
GOTO NEXT

◎ bit test(0 test of low 3 bits of f)
MOVF f,W
ANDLW B'00000111'
BTFSC STATUS,Z
GOTO NEXT

Kodlama

1 1	1 0 0 1	k k k k	k k k k

Q Çevrimi (Q Cycle Activity)

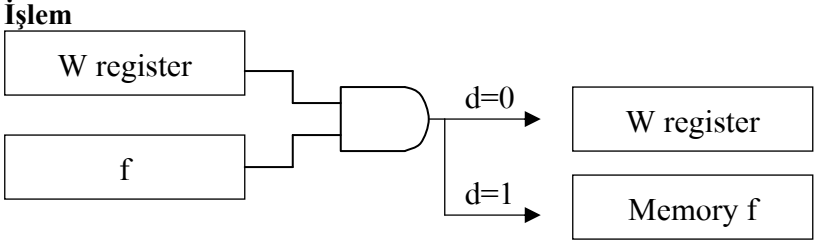
Q1	Q2	Q3	Q4
Kod çöz	'k' literalini oku	İşlemi yap	W'ye yaz

3.4.4. ANDWF (AND W with f)

Komut ANDWF f,d

İşleçler 0 <= f <= 127
d ə {0,1}

İşlem



Status Etkisi Z flag

Açıklama W registeri ile 'f' registerine AND işlemi uygulanır ve sonuç 'd' 0 ise W registerine, 'd' 1 ise 'f' registerine yazılır.
d = 0 ise W registerine
d = 1 ise f registerine

Kelimeler 1

Saat çevrimi (Cycle) 1

Örnek

◎ ANDWF f,0 (W) << f AND (W)
ANDWF f,1 FSR << f AND (W)

◎ Clear of PORTB (bit 0 – bit 3 : 4bit)
MOVLW B'11110000'
ANDWF PORTB,1

◎ Bit test of F (bit 0 – bit 2 : 3bit)
MOVLW B'00000111'
ANDWF f,0
BTFSC STATUS,Z
GOTO NEXT

◎ Bit test of F
MOVLW B'00000100'
ANDWF f,0
BTFSC STATUS,Z
GOTO NEXT

Kodlama

0 0	0 1 0 1	d f f f	f f f f

Q Çevrimi (Q Cycle Activity)

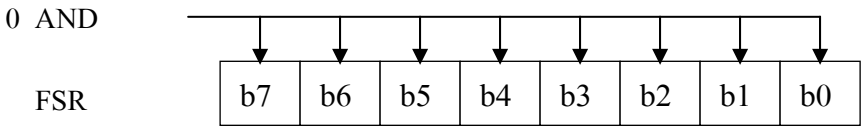
Q1	Q2	Q3	Q4
Kod çöz	f registerini oku	İşlemi yap	Gideceği vere vaz

3.4.5. BCF(Bit Clear f)

Komut BCF f,b

İşleçler 0 <= f <= 127 0 <= b <= 7

İşlem



Status Etkisi Yok

Açıklama 'f' registerinin 'b' ninci bitini temizle. (0 Sıfır yap)

Kelimeler 1

Saat çevrimi (Cycle) 1

Örnek

```
BCF  PORTB,1      PORTB in 1. bitini sıfırlar.
BCF  STATUS,0     Carry bayrağını sıfır yapar
BCF  STATUS,RP0   Bank0' a geçiş yapar
BCF  STATUS,2     Zero bayrağını sıfırlar
```

Kodlama

0 1	0 0 b b	b f f f	f f f f

Q Çevrimi (Q Cycle Activity)

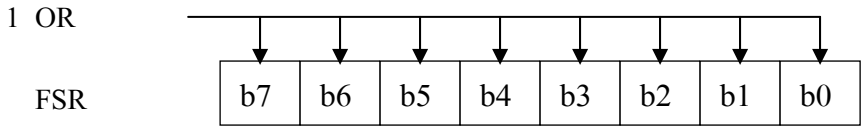
Q1	Q2	Q3	Q4
Kod çöz	f registerini oku	İşlemi yap	f registerine yaz

3.4.6. BSF (Bit Set f)

Komut BSF f,b

İşleçler 0 <= f <= 127 0 <= b <= 7

İşlem



Status Etkisi Yok

Açıklama 'f' registerinin içerinde bulunan sayının 'b' ninci bitini 1 yapar.

Kelimeler 1

Saat çevrimi (Cycle) 1

Örnek

◎
```
BSF  PORTB,3      PORTB'nin 3.bitini 1 yapar
BSF  STATUS,0     Carry bayrağını 1 yapar
BSF  STATUS,RP0   Bank1' e geçiş yapar
BSF  STATUS,2     Zero bayrağını 1 yapar
```

◎ PORTB'nin 0.bitinin kare dalga oluşması (yanıp sönmesi)
```
BSF  PORTB,0
CALL TIMER
BCF  PORTB,0
```

Kodlama

0 1	0 1 b b	b f f f	f f f f

Q Çevrimi (Q Cycle Activity)

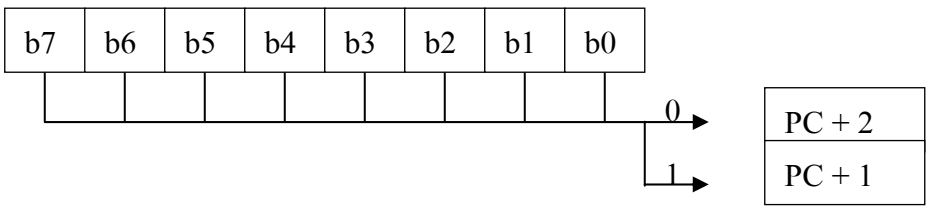
Q1	Q2	Q3	Q4
Decode	Read register f	Process data	Write register f

3.4.7. BTFSC (Bit Test f, Skip if Clear)

Komut BTFSC f,b

İşleçler 0 <= f <= 127 0 <= b <= 7

İşlem Skip if (f<b>) = 0



Status Etkisi Yok

Açıklama 'f' registerinin 'b' ninci biti 1 ise program akışı bir alt satırdan devam eder, eğer 0 ise bir alt satırı atlayarak sonraki komuttan devam eder.

Kelimeler 1

Saat çevrimi (Cycle) 1

Örnek

◎ BTFSC Memory,7 Eğer Memory'nin 7. biti 0 ise işlem bir satır atlayarak sonraki satırdan devam eder.

◎ BC (**B**ranch **C**arry)

```
BTFSC   STATUS,0    Carry flag= 0 >> 1 satır atlayarak devam eder
GOTO    LABEL       Carry flag=1 >> GOTO LABEL
```

◎ BZ (**B**ranch **Z**ero)

```
BTFSC   STATUS,2    Zero flag = 0 >> 1 satır atlayarak devam eder
GOTO    LABEL       Zero flag =1 >> GOTO LABEL
```

Kodlama

0 1	1 0 b b	b f f f	f f f f

Q Çevrimi (Q Cycle Activity)

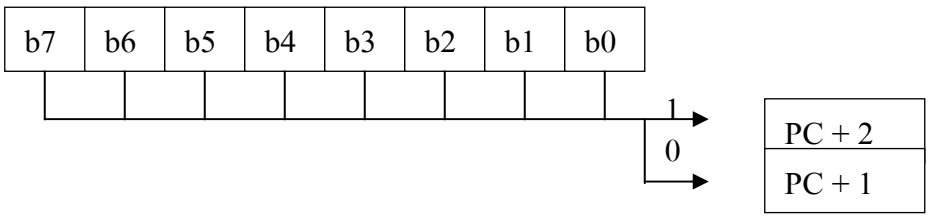
	Q1	Q2	Q3	Q4
Birincide	Kod çöz	f registeri oku	İşlemi yap	f registerine yaz
İkinci çevrimde (Atlama varsa)	İşlem yok	İşlem yok	İşlem yok	İşlem yok

3.4.8. BTFSS (Bit Test f, Skip if Set)

Komut BTFSS f,b

İşleçler 0 <= f <= 127 0 <= b <= 7

İşlem Skip if (f<b>) = 1



Status Etkisi Yok

Açıklama 'f' registerinin 'b' ninci biti 0 ise program akışı bir alt satırdan devam eder, eğer 1 ise bir alt satırı atlayarak sonraki komuttan devam eder.
NOP(No operation " işlem yok demektir ve bu sürede 2 saykıllık " 2T " zaman harcanır)

Kelimeler 1

Saat çevrimi (Cycle) 1(2)

Örnek

◎ BTFSS Memory,7 Eğer Memory'nin 7. biti 1 ise işlem bir satır atlayarak sonraki satırdan devam eder.

◎ BNC (**B**ranch **N**one **C**arry)
BTFSS STATUS,0 Carry flag= 1 >> 1 satır atlayarak devam eder
GOTO LABEL Carry flag=0 >> GOTO LABEL

◎ BNZ (Branch Yok Zero)
BTFSS STATUS,2 Zero flag = 1 >> 1 satır atlayarak devam eder
GOTO LABEL Zero flag =0 >> GOTO LABEL

Kodlama

0 1	1 1 b b	b f f f	f f f f

Q Çevrimi (Q Cycle Activity)

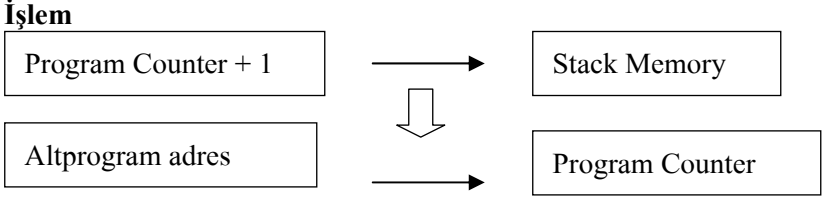
	Q1	Q2	Q3	Q4
Birincide	Kod çöz	f registerini oku	İşlemi yap	f registerine yaz
İkinci çevrimde (Atlama varsa)	İşlem yok	İşlem yok	İşlem yok	İşlem yok

3.4.9. CALL (Call Subroutine)

Komut CALL k

İşleçler 0 <= k <= 2047

İşlem



Status Etkisi Yok

Açıklama Altprogramı çağırır. Önce program counter'ı bir artırır (PC+1) ve bunu stack memory'e yükler sonra ait program adresi PC nin <10:0> bitlerine yüklenir. PC'nin üst bitleri PCLATH deki değerlerle yüklenir. CALL işlemi 2 saykıllık bir komuttur.

Kelimeler 1

Saat çevrimi (Cycle) 1(2)

Örnek

◎ CALL H'3AA' Bu komut programın 3AAh adresinden itibaren çalışmasını sağlar.

CALL SUB1 Bu komut SUB1 den itibaren programın çalışmasını sağlar.

◎ 1ms pulse programı

```
        BSF     PORTB,1   ;pulse on
        CALL    TIMER     ;yaklaşık 1ms Timer
        BCF     PORTB,1   ;pulse off
        ;1ms timer altprogram
TIMER   MOVLW   200       ; 1clock
        MOVWF   COUNT1    ; 1clock
DLY2    GOTO    $+1       ; 2clock
        DECFSZ  COUNT1,1  ; 1(2)clock
        GOTO    DLY1; 2clock
        RETURN            ; 2clock
```

Kodlama

1 0	0 k k k	k k k k	kkkk

Q Çevrimi (Q Cycle Activity)

	Q1	Q2	Q3	Q4
Birincide	Kod çöz	literal k sabiti oku PC yi stack'a yaz	F registerine yaz	İşlemi yap
İkinci çevrimde	NOP	NOP	NOP	NOP

3.4.10. CLRF (Clear f)

Komut CLRF f

İşleçler 0 <= f <= 127

İşlem Memory f | 00000000 |

Zero flag = 1 (Sıfır bayrağı)

Status Etkisi Z flag

Açıklama 'f' registerinin içeriğini temizler Z bayrağını 1 yapar. (set eder)

Kelimeler 1

Saat çevrimi (Cycle) 1

Örnek ◎ CLRF MEMORY MEMORY 'i temizler(sıfırlar)

Kodlama

0 0	0 0 0 1	1 f f f	f f f f

Q Çevrimi (Q Cycle Activity)

Q1	Q2	Q3	Q4
Kod çöz	f registerini oku	İşlemi yap	f registerine yaz

3.4.11. CLRW (Clear W)

Komut CLRW

İşleçler Yok

İşlem W register

Zero flag = 1 (Sıfır bayrağı) | 00000000 |

Status Etkisi Z flag

Açıklama W registerinin içeriğini temizler ve Sıfır (Zero) bayrağını 1 yapar. (set eder)

Kelimeler 1

Saat çevrimi (Cycle) 1

Örnek ◎ CLRW W register is cleared.

Kodlama

0 0	0 0 0 1	1 x x x	x x x

Q Çevrimi (Q Cycle Activity)

Q1	Q2	Q3	Q4
Kod çöz	İşlem yok	İşlemi yap	W registerine yaz

3.4.12. CLRWDT (Clear Watchdog Timer)

Komut CLRWDT

İşleçler Yok

İşlem 00h >> WDT
0 >> WDT prescaler (önbölücü)
1 >> TO
1 >> PD
Zero flag = 1

Status Etkisi Z flag (Z bayrağı)

Açıklama CLRWDT komutu Watchdog Timer içeriğini temizler (resetler) Status da bulunan sıfır bayrağıda 1 (set) olur. Aynı zamanda (TimeOut) TO ve (Power Down) PD bitleride 1 (set) olurlar.

Kelimeler 1

Saat çevrimi (Cycle) 1

Örnek ◎ CLRWDT

Kodlama

0 0	0 0 0 0	0 1 1 0	0 1 0

Q Çevrimi (Q Cycle Activity)

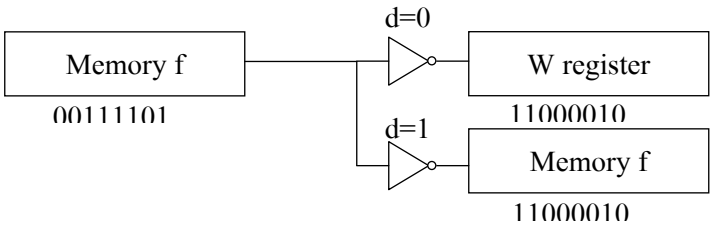
Q1	Q2	Q3	Q4
Kod çöz	İşlem yok	İşlemi yap	WDT counter'a yaz

3.4.13. COMF (Complement f)

Komut COMF f,d

İşleçler 0 <= f <= 127
d ə {0,1}

İşlem



Status Etkisi Z flag

Açıklama 'f' registerinin içeriği terslenir. Eğer 'd' 0 ise sonuç W registerine 'd' 1 ise 'f'. registerine yazılır.

Kelimeler 1

Saat çevrimi (Cycle) 1

Örnek

◎ COMF MEM,0 ;MEM datasını tersler ve sonucu W registerine yazar
COMF MEM,1 ;MEM datasını tersler ve sonucu MEM registerine yazar

◎ COMF PORTB,0 ;PORTB değerini tersler ve W registere yazar

◎ H'FF' datasının karşılaştırılması.
COMF MEM,0
BTFSC STATUS,Z
GOTO NEXT

Kodlama

0 0	1 0 0 1	d f f f	f f f f

Q Çevrimi (Q Cycle Activity)

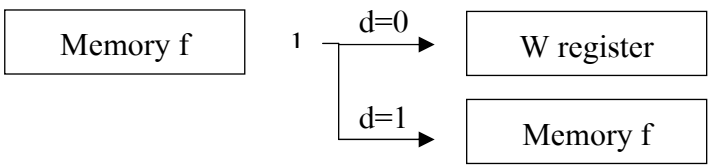
Q1	Q2	Q3	Q4
Kod çöz	F registeri oku	İşlem yap	Gideceği yere yaz

3.4.14. DECF (Decrement f)

Komut DECF f,d

İşleçler 0 <= f <= 127
d ə {0,1}

İşlem



Status Etkisi Z flag

Açıklama 'f' registerinin değeri bir azaltılır Eğer 'd' 0 ise sonuç W registere 1 ise f registere yüklenir.

Kelimeler 1

Saat çevrimi (Cycle) 1

Örnek

- DECF f,0 ;f registerinin değerini bir azaltır sonuç W registerine yüklenir.
- DECF f,1 ;f registerinin değerini bir azaltır sonuç f registerine yüklenir.
- DECF DATA,F
  BTFSC STATUS,Z
  GOTO NEXT

Burada DATA bilgisi bir eksilir eğer DATA 0 olursa sıfır bayrağı (Z flag) 1 olur ve program GOTO NEXT ten devam eder.

Kodlama

0 0	0 0 1 1	d f f f	f f f f

Q Çevrimi (Q Cycle Activity)

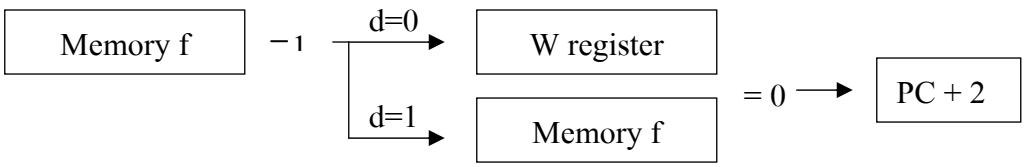
Q1	Q2	Q3	Q4
Kod çöz	F registeri oku	İşlem yap	Gideceği yere yaz

3.4.15. DECFSZ (Decrement f, Skip if 0)

Komut DECFSZ f,d

İşleçler 0 <= f <= 127
d ə {0,1}

İşlem



Status Etkisi Yok

Açıklama f registerinin değeri bir azaltılır. Eğer d=0 ise sonuç W registerine yüklenir, d=1 ise f registerine yüklenir.
İşlem sonucunda sonuç 1 ise bir sonraki satırdan program devam eder, 0 ise NOP (işlem yok) uygulayarak 1 satır atlar ve ikinci satırdan program devam eder.

Kelimeler 1

Saat çevrimi (Cycle) 1(2)

Örnek ◎

```
        MOVLW   8
        MOVWF   CNT
LOOP    INCF    f,1
        DECFSZ  CNT,1
        GOTO    LOOP
```

Kodlama

0 0	1 0 1 1	d f f f	f f f f

Q Çevrimi (Q Cycle Activity)

	Q1	Q2	Q3	Q4
Birincide	Kod çöz	F registeri oku	İşlem yap	Gideceği yere yaz
Eğer ikinciye atlarsa	İşlem yok	İşlem yok	İşlem yok	İşlem yok

3.4.16. GOTO (GOTO Unconditional Branch)

Komut GOTO k

İşleçler 0 <= k <= 2047

İşlem

Bölüm Adresi	⟶	Program Sayıcı

Status Etkisi Yok

Açıklama GOTO koşulsuz bir yönlendirme komutudur. Onbir bitlik bir adres hemen PC bitine <10:0>.yüklenir. PC nin yüksek bitleri de PCLATH<4:3> 'den yüklenir ve bu adrese program yönlenmiş olur. GOTO iki çevrimli (clock) bir komuttur.

Kelimeler 1

Saat çevrimi (Cycle) 1(2)

Örnek

◎ GOTO H'3AA' Bu komut 3Aah.bölüm adresine git demektir.

◎ GOTO LOOP Bu komut LOOP etiketinin adresine git demektir. Bu adres PC de oluşur.

```
                ;1ms gecikme alt programı
TIMER           MOVLW   200         ; 1clock
                MOVWF   COUNT1      ; 1clock
DLY2GOTO        $+1                 ; 2clock
                DECFSZ  COUNT1,1    ; 1(2)clock
                GOTO    DLY1; 2clock
                RETURN              ; 2clock
```

Kodlama

1 0	1 k k k	k k k k	kkkk

Q Çevrimi (Q Cycle Activity)

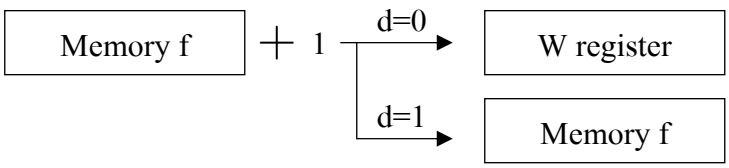
	Q1	Q2	Q3	Q4
Birincide	Kod çöz	literal k sabiti oku	F registerine yaz	İşlemi yap
İkinci çevrimde	NOP	NOP	NOP	NOP

3.4.17. INCF (Increment f)

Komut INCF f,d

İşleçler 0 <= f <= 127
d ə {0,1}

İşlem



Status Etkisi Z flag (Sıfır bayrağı)

Açıklama 'f' registerinin sahip olduğu değeri bir artırır. Eğer d=0 ise sonuç W registerine d=1 ise f registerine yüklenir.

Kelimeler 1

Saat çevrimi (Cycle) 1

Örnek

◎ INCF f,0 ; f registerinin değerini 1 artırarak sonucu W registerine yükler.

◎ INCF f,1 ; f registerinin değerini 1 artırarak sonucu f registerine yükler.

◎ taşıma bayrağı ile kullanılması

```
BTFSC   STATUS,C
INCF    f,1
```

◎ 8 bitlik D/A (dijital / Analog) çeviriciden çıkış dalgasını görmek için

```
LOOP  INCF      PORTB,F
      GOTO      LOOP
```

Kodlama

0 0	1 0 1 0	d f f f	f f f f

Q Çevrimi (Q Cycle Activity)

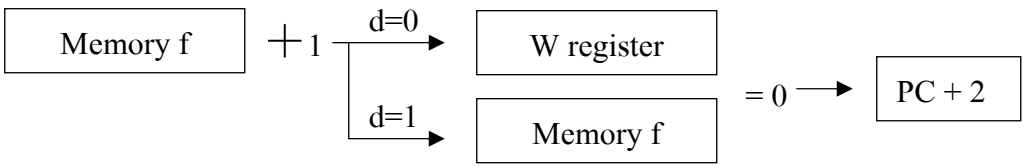
Q1	Q2	Q3	Q4
Kod çöz	F registeri oku	İşlem yap	Gideceği yere yaz

3.4.18. INCFSZ (Increment f, Skip if 0)

Komut INCFSZ f,d

İşleçler 0 <= f <= 127
d ə {0,1}

İşlem



Status Etkisi Yok

Açıklama 'f' registerinin değerini bir artırır Eğer d=0 ise sonucu W registerine d=1 ise f registerine yükler.
İşlem sonucunda sonuç 1 ise bir sonraki satırdan program devam eder, 0 ise NOP (işlem yok) uygulayarak 1 satır atlar ve ikinci satırdan program devam eder.

Kelimeler 1

Saat çevrimi (Cycle) 1(2)

Örnek

◎ INCFSZ f,0
INCFSZ f,1

Kodlama

0 0	1 1 1 1	d f f f	f f f f

Q Çevrimi (Q Cycle Activity)

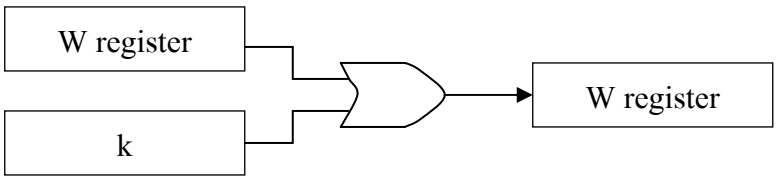
	Q1	Q2	Q3	Q4
Birincide	Kod çöz	F registeri oku	İşlem yap	Gideceği yere yaz
Eğer ikinciye atlarsa	İşlem yok	İşlem yok	İşlem yok	İşlem yok

3.4.19. IORLW (Inclusive OR literal with W)

Komut IORLW k

İşleçler 0 <= k <= 255

İşlem



Status Etkisi Z flag (Sıfır bayrağı)

Açıklama W registerinin içeriği 8 bitlik k literali ile OR lanır. Sonuç W registerine yüklenir. İşlem sonucu 0 ise statusun sıfır bayrağı (Zero flag) 1 olur.

Kelimeler 1

Saat çevrimi (Cycle) 1

Örnek

◎ IORLW B'11110000' W registeri 4 biti 1 olanla OR lanır.

◎ W registeri FFh. Ile OR lanır.
IORLW H'FF'

◎ bit test (W registerinin 0 testi)
IORLW 0

Kodlama

1 1	1 0 0 0	k k k k	k k k k

Q Çevrimi (Q Cycle Activity)

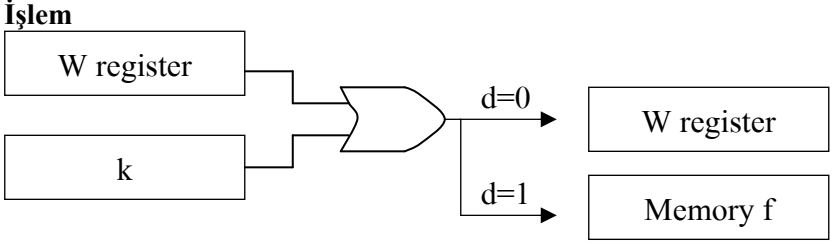
Q1	Q2	Q3	Q4
Kod çöz	literal k sabiti oku	İşlem yap	W registerine yaz

3.4.20. IORWF (Inclusive OR W with f)

Komut IORWF f,d

İşleçler 0 <= f <= 127
d ə {0,1}

İşlem



Status Etkisi Z flag

Açıklama W registeri f registeri ile OR mantıksal işlemin yapılır. Sonuç eğer d=0 ise W registerine d=1 ise f registerine yüklenir.

Kelimeler 1

Saat çevrimi (Cycle) 1

Örnek

◎ IORWFf,0 (W) << f OR (W)
IORWFf,1 f<< f OR (W)

◎ MOVLW B'00001111'
IORWFPORTB,F

◎ MOVLW B'00000111'
IORWFf,1

Kodlama

0 0	0 1 0 0	d f f f	f f f f

Q Çevrimi (Q Cycle Activity)

Q1	Q2	Q3	Q4
Kod çöz	F registeri oku	İşlem yap	Gideceği yere yaz

3.4.21. MOVLW (Move Literal to W)

Komut MOVLW k

İşleçler 0 <= k <= 255

İşlem

Status Etkisi Yok

Açıklama Sekiz bit literal ‘k’ bilgisini W registerine yükler.

Kelimeler 1

Saat çevrimi (Cycle) 1

Örnek

◎ (W) << 30h
ADDLW H’30’

◎ FSR << AFh
MOVLW H’AF’
MOVWF f

◎ PORTB << B’00001111’
MOVLW B’00001111’
MOVWF PORTB

Kodlama

1 1	0 0 x x	k k k k	k k k k

Q Çevrimi (Q Cycle Activity)

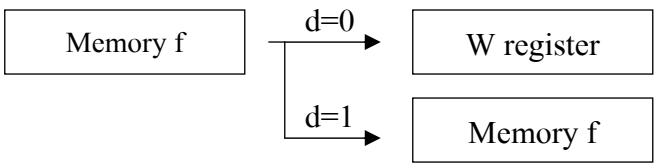
Q1	Q2	Q3	Q4
Kod çöz	literal k sabiti oku	İşlem yap	W registerine yaz

3.4.22. MOVF (Move f)

Komut MOVF f,d

İşleçler 0 <= f <= 127
d ə {0,1}

İşlem



Status Etkisi Z flag

Açıklama f registerinin içeriği d=0 ise W registerine d=1 ise f registerine (kendisine) yüklenir. d=1 olduğu durumlarda f registerinin içeriğini test etmemizde (0 olup olmadığına) fayda vardır. Çünkü bu durumda statustaki sıfır bayrağı etkilenir.

Kelimeler 1

Saat çevrimi (Cycle) 1

Örnek

◎ Memory Test
```
MOVF   f,F
```
F registerinin sıfır bayrağına etkisini test etmekte fayda vardır.

◎ f1 registerindeki bilgilerin f2 registerine iletilmesi.
```
MOVF    f1,W
MOVWF  f2
```

◎ F registerindeki bilgilerin PORT B den çıkış olarak alınması
```
MOVF    f,W
MOVWF  PORTB
```

Kodlama

0 0	1 0 0 0	d f f f	f f f f

Q Çevrimi (Q Cycle Activity)

Q1	Q2	Q3	Q4
Kod çöz	F registeri oku	İşlem yap	Gideceği yere yaz

3.4.23. MOVWF (Move W to f)

Komut MOVWF f

İşleçler 0 <= f <= 127

İşlem

Status Etkisi Yok

Açıklama W registerindeki bilgiler f registerine taşınır.

Kelimeler 1

Saat çevrimi (Cycle) 1

Örnek

◎ (0Ah) değeri f registerine iletilir.

```
MOVLW   H'0A'
MOVWF   f
```

Kodlama

0 0	0 0 0 0	1 f f f	f f f f

Q Çevrimi (Q Cycle Activity)

Q1	Q2	Q3	Q4
Kod çöz	f registerini oku	İşlemi yap	f registerine yaz

3.4.24. NOP (İşlem yok)

Komut NOP

İşleçler Yok

İşlem İşlem yok

Status Etkisi Yok

Açıklama İşlem yok

Kelimeler 1

Saat çevrimi (Cycle) 1

Örnek

```
                ◎Zamanlayıcı alt programı
TIMER   MOVLW   200         ; 1clock
        MOVWF   COUNT1      ; 1clock
DLY2    GOTO    $+1         ; 2clock
        NOP                 ;1clock
        NOP                 ;1clock
        DECFSZ  COUNT1,1    ; 1(2)clock
        GOTO    DLY1; 2clock
        RETURN              ; 2clock
```

Kodlama

0 0	0 0 0 0	0 x x o	0 0 0 0

Q Çevrimi (Q Cycle Activity)

Q1	Q2	Q3	Q4
Kod Çöz	İşlem yok	İşlem yok	İşlem yok

3.4.25. RETFIE (Return from interrupt)

Komut RETFIE

İşleçler Yok

İşlem

Stack Memory(TOS)	→	Program Sayıcı
1	→	GIE bit

Status Etkisi Yok

Açıklama Kesme alt programından ana programa dönemek için kullanılır. Saat çevrimi (Cycle) için POPed ve TOS (Top of Stack) ları PC ye yüklenir. (GIE)(INTCON<7>). (Global Interrupt Enable bit) 1 (set) yapılır. İki zamanlı bir komuttur.

Kelimeler 1

Saat çevrimi (Cycle) 2

Örnek

```
◎ORG   4
  BSF          PORTB,2
  RETFIE
```

Kodlama

0 0	0 0 0 0	0 0 0 0	1 0 0 1

Q Çevrimi (Q Cycle Activity)

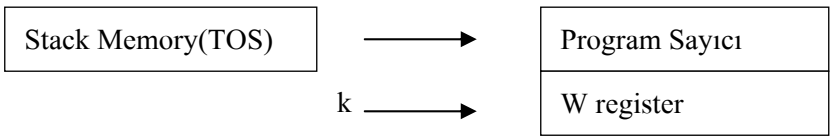
	Q1	Q2	Q3	Q4
Birincide	Kod çöz	İşlem yok	GIE bitini set et	Stacktan POP et
İkincide	İşlem yok	İşlem yok	İşlem yok	İşlem yok

3.4.26. RETLW (Return with Literal in W)

Komut RETLW k

İşleçler 0 <= k <= 255

İşlem



Status Etkisi Yok

Açıklama Alt programda W registerine sekiz bitlik 'k' literalini yükleme işini yapar. Program sayıcı döneceği adresi TOS dan PC ye yükleyerek gerçekleştirir. İki zamanlı bir komuttur.

Kelimeler 1

Saat çevrimi (Cycle) 2

Örnek

◎W << Z flag

```
  BTFSC   STATUS,Z
  RETLW   0
  RETLW   1
```

◎7 Segmen display bilgi tablosu

```
SEGDAT  ADDWF   PCL,F           ;PCL(program sayıcı) + W >>PCL
        RETLW   B'00111111'     ;segmen data 0 >> W (PCL+0)
        RETLW   B'00000110';segmen data 1 >> W (PCL+1)
        RETLW   B'01011011';segmen data 2 >> W (PCL+2)
        RETLW   B'01001111';segmen data 3 >> W (PCL+3)
        RETLW   B'01100110';segmen data 4 >> W (PCL+4)
        RETLW   B'01101101';segmen data 5 >> W (PCL+5)
```

Kodlama

1 1	0 1 x x	k k k k	k k k k

Q Çevrimi (Q Cycle Activity)

	Q1	Q2	Q3	Q4
Birincide	Kod çöz	literal k sabiti oku	GIE bitini set et	Stacktaki POP u W registere yaz
İkincide	İşlem yok	İşlem yok	İşlem yok	İşlem yok

3.4.27. RETURN (Return from Subroutine)

Komut RETURN

İşleçler Yok

İşlem

Stack Memory(TOS)	⟶	Program Counter

Status Etkisi Yok

Açıklama Alt programdan dönmek için kullanılır. Program sayıcıya POPed ve TOS yüklenir. Böylece program sayıcıdaki adrese geri dönülür. İki zamanlı bir komuttur.

Kelimeler 1

Saat çevrimi (Cycle) 2

Örnek ◎ Altprogramdan geri Saat çevrimi (Cycle)

```
        CALL    SUB1
SUB1    DECF    f,F
        RETURN
```

Kodlama

0 0	0 0 0 0	0 0 0 0	1 0 0 0

Q Çevrimi (Q Cycle Activity)

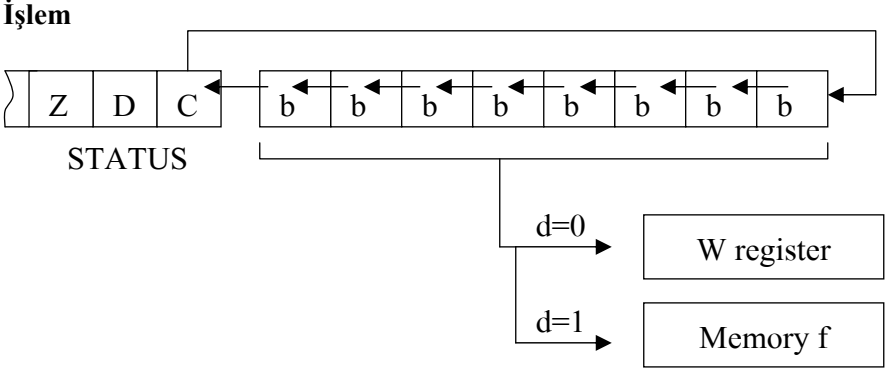
	Q1	Q2	Q3	Q4
Birincide	Kod çöz	İşlem yok	İşlem yok	Stacktan POP et.
İkincide	İşlem yok	İşlem yok	İşlem yok	İşlem yok

3.4.28. RLF (Rotate Left f through Carry)

Komut RLF f,d

İşleçler 0 <= f <= 127
d ə {0,1}

İşlem



Status Etkisi Taşıma bayrağı (C flag)

Açıklama f registerinin içeriğini bir bit SOLA doğru kaydırır. Eğer d=0 ise sonuç w registerine d=1 ise f registerinin içine yüklenir.

Kelimeler 1

Saat çevrimi (Cycle) 1

Örnek

◎ MEMORY multiplied by four.

```
BCF  STATUS,C
RLF  MEMORY,F
BCF  STATUS,C
RLF  MEMORY,F
```

◎ MEMORY multiplied by two and transfer to W register.

```
BCF  STATUS,C
RLF  MEMORY,W
```

Kodlama

0 0	1 1 0 1	d f f f	f f f f

Q Çevrimi (Q Cycle Activity)

Q1	Q2	Q3	Q4
Kod çöz	F registeri oku	İşlem yap	Gideceği yere yaz

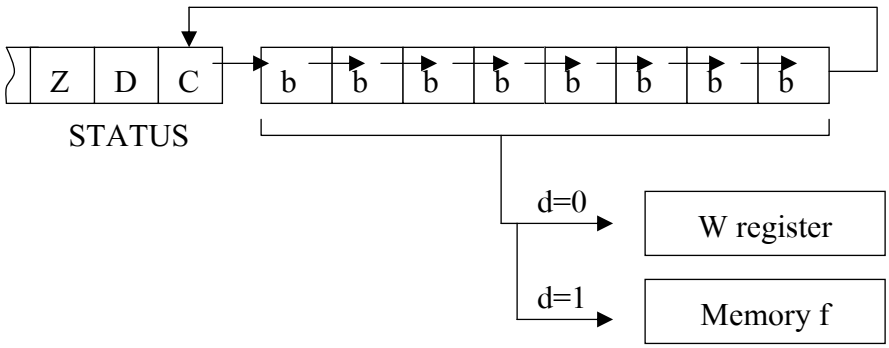
3.4.29. RRF(Rotate Right f through Carry)

Komut RRF f,d

İşleçler 0 <= f <= 127

d ə {0,1}

İşlem



Status Etkisi Taşıma bayrağı (C flag)

Açıklama f registerinin içeriğini bir bit SAĞA doğru kaydırır. Eğer d=0 ise sonuç w registerine d=1 ise f registerinin içine yüklenir.

Kelimeler 1

Saat çevrimi (Cycle) 1

Örnek

◎ MEMORY devided by four.

```
BCF  STATUS,C
RRF  MEMORY,F
BCF  STATUS,C
RRF  MEMORY,F
```

◎ MEMORY divided by two and transfer to W register.

```
BCF  STATUS,C
RRF  MEMORY,W
```

Kodlama

0 0	1 1 0 1	d f f f	f f f f

Q Çevrimi (Q Cycle Activity)

Q1	Q2	Q3	Q4
Kod çöz	F registeri oku	İşlem yap	Gideceği yere yaz

3.4.30. SLEEP (Sleep)

Komut SLEEP

İşleçler Yok

İşlem 00h >> WDT 0 >> WDT prescaler 1 >> TO 0 >> PD

Status Etkisi TO,PD

Açıklama Statustaki güç kesim (power-down PD) bitini temizler ve sure aşım (Time-out TO) bitini bir (set) yapar. Watchdog Timer (WDT) ve önbölücü (prescaler) da temizlenir. İşlemciyi uyku moduna geçirir ve daha az güç harcamasını sağlar.

Kelimeler 1

Saat çevrimi (Cycle) 1

Örnek

◎ WKesme ile uykudan uyandırma ve bir alt satırdan çalışmasına devam ettirme.

```
SLEEP
CALL    JOB
```

Kodlama

0 0	0 0 0 0	0 1 1 0	0 0 1 1

Q Çevrimi (Q Cycle Activity)

Q1	Q2	Q3	Q4
Kod çöz	İşlem yok	İşlem yok	Sleep'e git

3.4.31. SUBLW (Subtract W from Literal)

Komut SUBLW k

İşleçler 0 <= k <= 255

İşlem

Status Etkisi C,DC,Z flag

Açıklama k literalinden W registeri çıkartılır (2'ye tamamlama yöntemi ile) ve sonuç W registerine yüklenir.

Kelimeler 1
Saat çevrimi (Cycle) 1

Örnek

◎ W <= k

```
SUBLW   k
BTFSC   STATUS,C
GOTO    NEXT        ;W <= k
```

◎ W > k

```
SUBLW   k
BTFSS   STATUS,C
GOTO    NEXT        ;W > k
```

◎

k > W register Z=0,CY=1
k = W register Z=1,CY=1
k < W register Z=0,CY=0

◎ f <= k

```
MOVF    f,W
SUBLW   N           ; N – f >> W
BTFSC   STATUS,C
GOTO    NEXT        ; f <= k
```

◎ f > k

```
MOVF    f,W
SUBLW   k           ; N – f >> W
BTFSS   STATUS,C
GOTO    NEXT        ; f > k
```

Kodlama

1 1	1 1 0 x	k k k k	k k k k

Q Çevrimi (Q Cycle Activity)

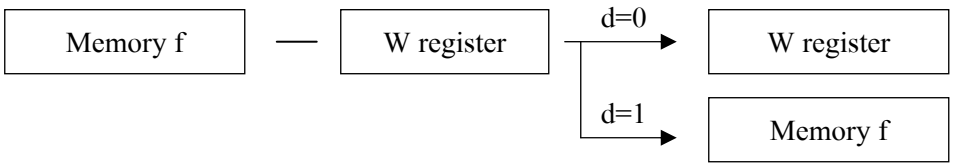
Q1	Q2	Q3	Q4
Kod çöz	literal k sabiti oku	İşlem yap	W registerine yaz

3.4.32. SUBWF (Subtract W from f)

Komut SUBWF f,d

İşleçler 0 <= f <= 127
d ə {0,1}

İşlem



Status Etkisi C,DC,Z flag

Açıklama f registerinden W registeri (ikiye tamamlama yöntemi ile) çıkartılır. Eğer d=0 ise sonuç W registerine d=1 ise f registerine yüklenir.

Kelimeler 1

Saat çevrimi (Cycle) 1

Örnek

◎ (W) << f - (W) f << f +(W)
SUBWF f,W SUBWF f,F

◎
f > W register Z=0,CY=1
f = W register Z=1,CY=1
f < W register Z=0,CY=0

Kodlama

0 0	0 0 1 0	d f f f	f f f f

Q Çevrimi (Q Cycle Activity)

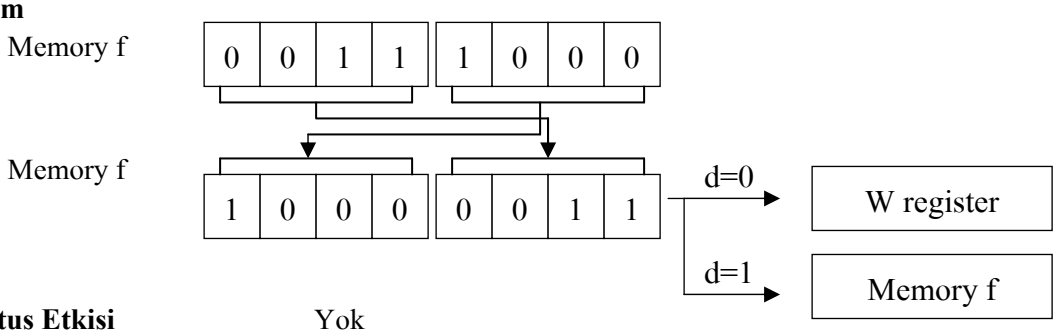
Q1	Q2	Q3	Q4
Kod çöz	F registeri oku	İşlem yap	Gideceği yere yaz

3.4.33. SWAPF (Swap Nibbles in f)

Komut SWAPF f,d

İşleçler 0 <= f <= 127
d ə {0,1}

İşlem



Status Etkisi Yok

Açıklama f registerinin alt dört biti ile üst dört biti yer değiştirir. Eğer d=0 ise sonuç W registerine d=1 ise f registerine yüklenir.

Kelimeler 1
Saat çevrimi (Cycle) 1
Örnek

◎ High 4 bit take out

```
SWAPF    DATA,W
ANDLW    B'00001111'
```

◎ Separate of high 4bit and low 4bit of W register.

```
MOVWF      MEM1
MOVWF      MEM2
SWAPF      MEM1,F
MOVLW      B'00001111'
ANDWF      MEM1,F     ; high 4bit
ANDWF      MEM2,f     ;low 4bit
```

Kodlama

0 0	1 1 1 0	d f f f	f f f f

Q Çevrimi (Q Cycle Activity)

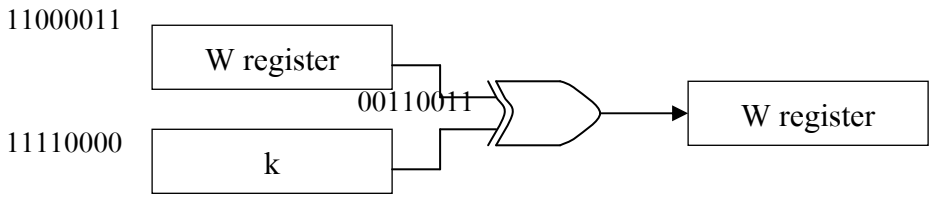
Q1	Q2	Q3	Q4
Kod çöz	F registeri oku	İşlem yap	Gideceği yere yaz

3.4.34. XORLW (Exclusive OR literal with W)

Komut IORLW k

İşleçler 0 <= k <= 255

İşlem



Status Etkisi Z flag

Açıklama W registerinin içeriği ile 8 bitlik k literaline XOR uygulanır. Sonuç W registerine yüklenir.

Kelimeler 1

Saat çevrimi (Cycle) 1

Örnek

◎ W registerinin yüksek 4 bitini tersler.

```
XORLW     B'11110000'
```

◎ W registerinin tam tersini alır.

```
XORLW     H'FF'
```

Kodlama

1 1	1 0 1 0	k k k k	k k k k

Q Çevrimi (Q Cycle Activity)

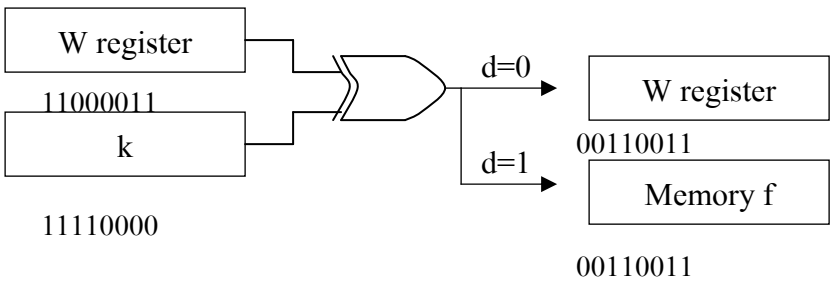
Q1	Q2	Q3	Q4
Kod çöz	literal k sabiti oku	İşlem yap	W registerine yaz

3.4.35. XORWF (Exclusive OR W with f)

Komut XORWF f,d

İşleçler 0 <= f <= 127
d ə {0,1}

İşlem



Status Etkisi Z flag

Açıklama W registeri ile f registerine XOR mantıksal ifadesi uygulanır. Eğer d=0 ise sonuç W registerine d=1 ise f registerine yüklenir.

Kelimeler 1

Saat çevrimi (Cycle) 1

Örnek

◎PORTB(bit2) yi tersler

```
MOVLW   B'00000100'
XORWF   PORTB,F
```

◎MEMORY(düşük 4bit) tersler ve sonucu MEMORY dosyasına yazar.

```
MOVLW   B'00001111'
XORWF   MEMORY,F
```

◎MEMORY dosyasının AA olduğunu kontrol eder. AA ise Zero = 1 dir.

```
MOVLW   H'AA'
XORWF   MEMORY,W
BTFSS   STATUS,Z
GOTO    ( AA değilse )
GOTO    ( AA ise )
```

Kodlama

0 0	0 1 1 0	d f f f	f f f f

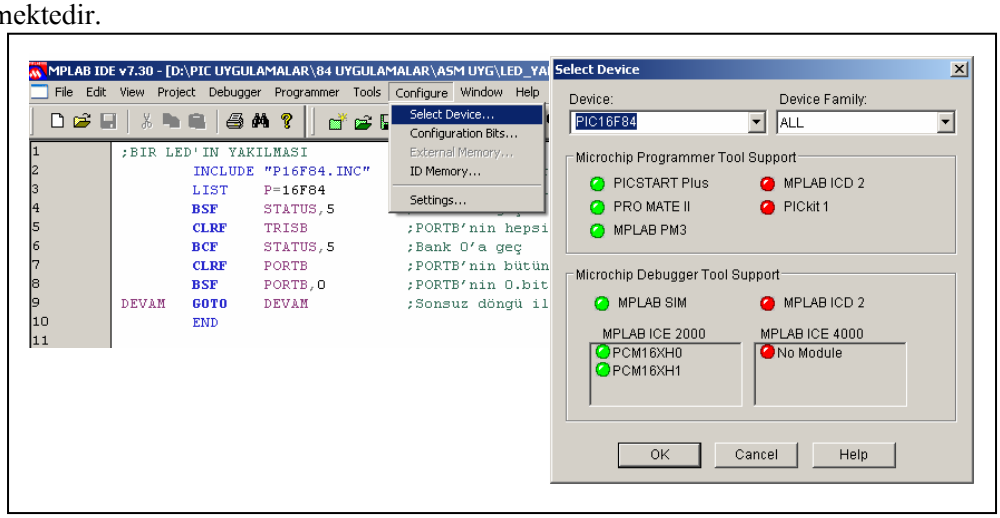
Q Çevrimi (Q Cycle Activity)

Q1	Q2	Q3	Q4
Kod çöz	F registeri oku	İşlem yap	Gideceği yere yaz

3.5. Hata Ayıklama Özelliği

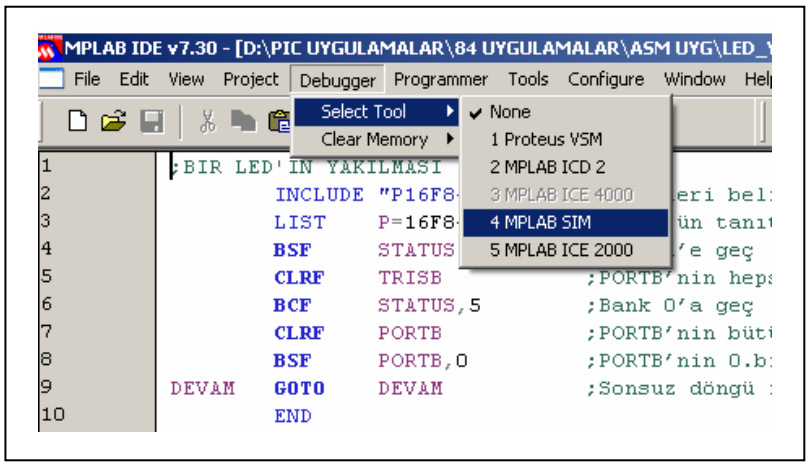
Yazılan programı elektronik ortamlarda denemek yerine yazılım olarak adım adım çalıştırarak kontrol de edebiliriz. Böylece mantık hatalarını daha kolay görebiliriz. Bu işleme simülasyon denilmektedir.

Sayfa 12'deki bir led'in yakılması ile ilgili programı bir önceki konuda MPLAB IDE programının editör sayfasına yazılıp derlemesi de yapılmıştı. Menü çubuğundan configure (Düzen) menüsünün Select Device(Eleman seçimi) alt menüsünü tıklanır. Açılan pencereden PIC seçimi gerçekleştirilir. Aynı zamanda seçilen PIC'e göre programın neleri desteklediğini görebiliriz. MPLAB SIM desteği varsa ki vardır. Simülatör yapabiliriz anlamına gelmektedir.

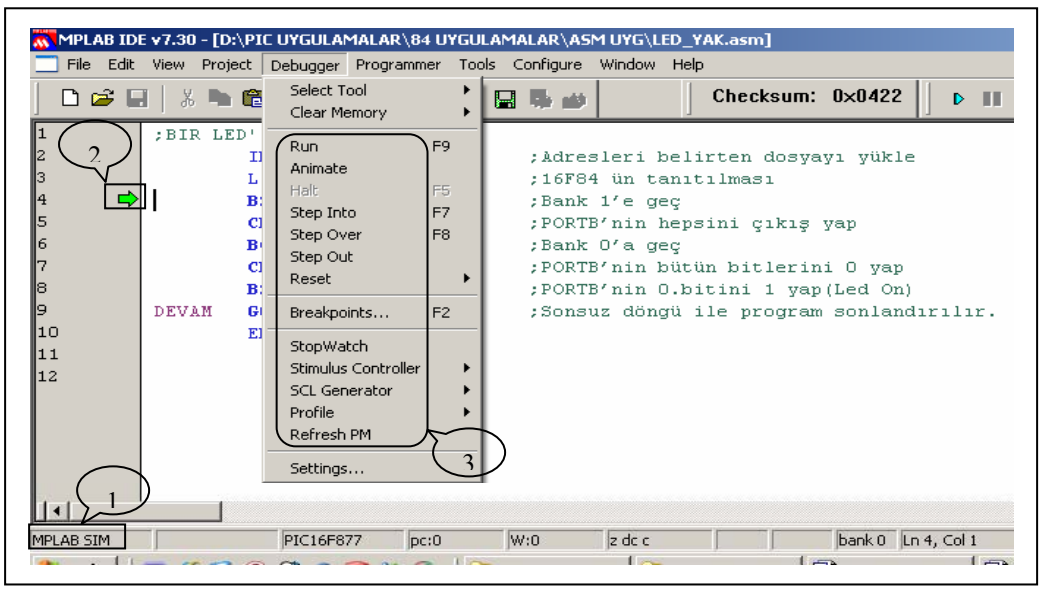


Şekil 3.32. Simülatör için eleman seçimi

Simülatörü aktif hâle getirebilmek için Debugger (Hata ayıklama) menüsünden Select Tool (Araç seçimi) alt menüsünü takip ederek MPLAB SIM seçeneği seçilir. Son olarak yazdığımız programı derlediğimizde simülatör kullanılmaya hazır hâle gelir. Derleme işlemi yapılmadan simülatör çalışmaz.



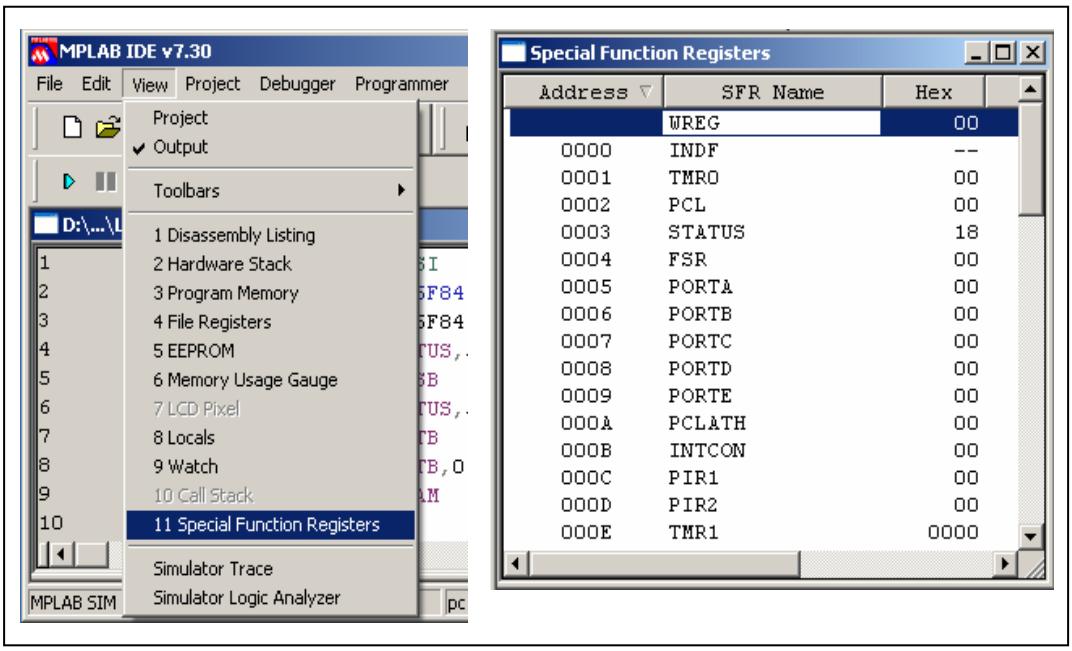
Şekil 3.33: Simülatörün aktif edilmesi



Şekil 3.34: Simülatör Doğrulama Penceresi

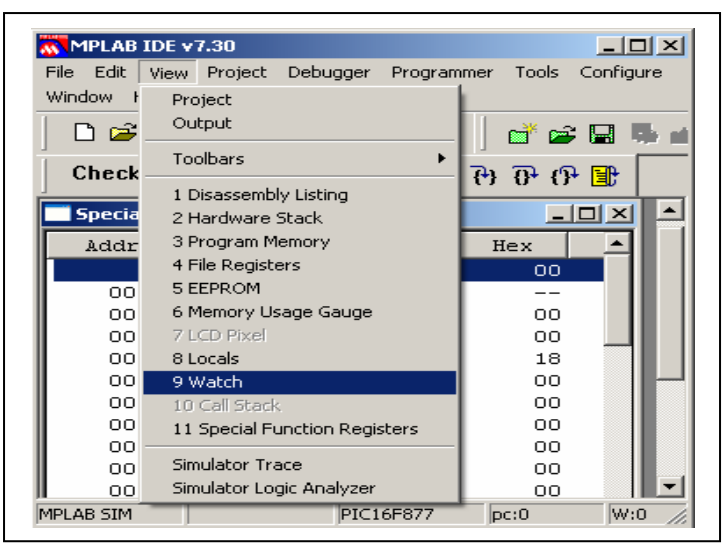
Simülatörün aktifleştiğini yukarıda görüldüğü gibi üç şekilde anlayabiliriz. Simülatör aktif olmadığında 3 numaralı kutucuk içerisindekiler görünmez.

Simülatörün çalışması esnasında bir takım değerleri görmemiz gerekmektedir. Bunlardan özel fonksiyon Register'leri görebilmek için View/ Special Function Registers seçmemiz yeterli olacaktır.

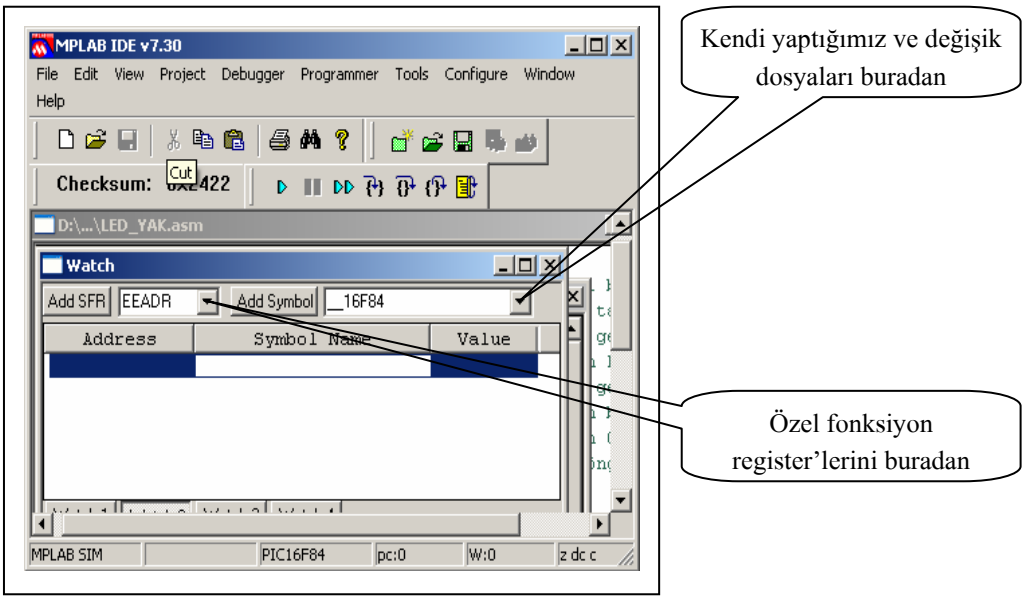


Şekil 3.35: Özel fonksiyon register'lerinin (sfr) görüntülenmesi

Eğer kendi oluşturduğumuz dosya içeriklerini veya özel fonksiyon register'lerinin bir kısmını görmek istediğimizde şekil 3.36'da görüldüğü gibi Watch penceresini açmamız gerekmektedir.

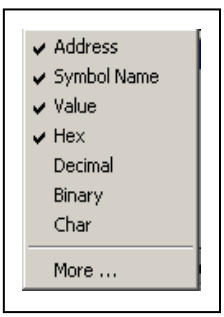


Şekil 3.36: Watch penceresinin açılması



Şekil 3.37: Görüntülemek istediğimiz dosyaların seçimi

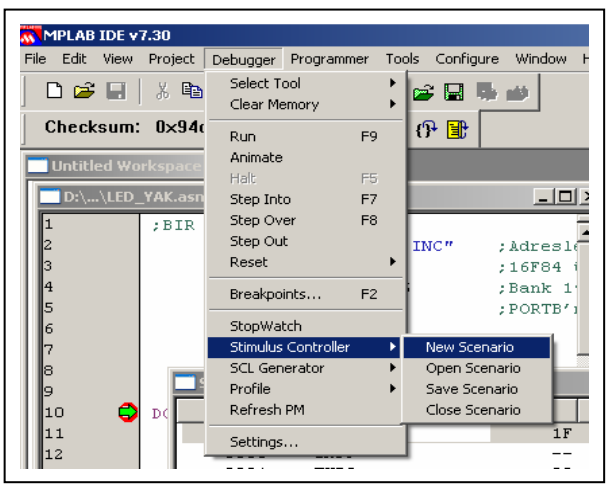
Watch penceresi açıldığında, Özel Fonksiyon Register'lerinin seçimi ok tuşu ile açılan menü listesinden yapılmaktadır. Daha sonra şekil 3.38'de gösterilen Add SFR veya Add Symbol ile bunu görüntü penceresine aktarmamız gerekmektedir. Gelen görüntü değeri hex biçiminde olup işaretçi, Value üzerinde iken Mouse sağ tuş tıklandığında açılan pencerede register içeriği hangi sayısal biçimde görüntüleneceğini seçmemiz yeterli olacaktır.



Şekil 3.38: Görüntülemek istediğimiz özel fonksiyon register'lerinin özellikleri

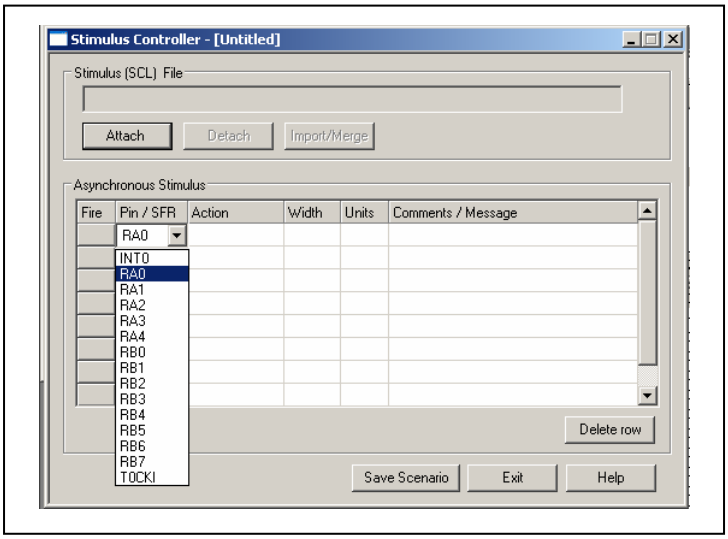
3.5.1. Buton Eklenmesi ve Kullanımı

Şekil 3.39'da Debugger menüsünden Stimulus Controller alt menüsünü takip eden New Scenario menüsü tıklanır.



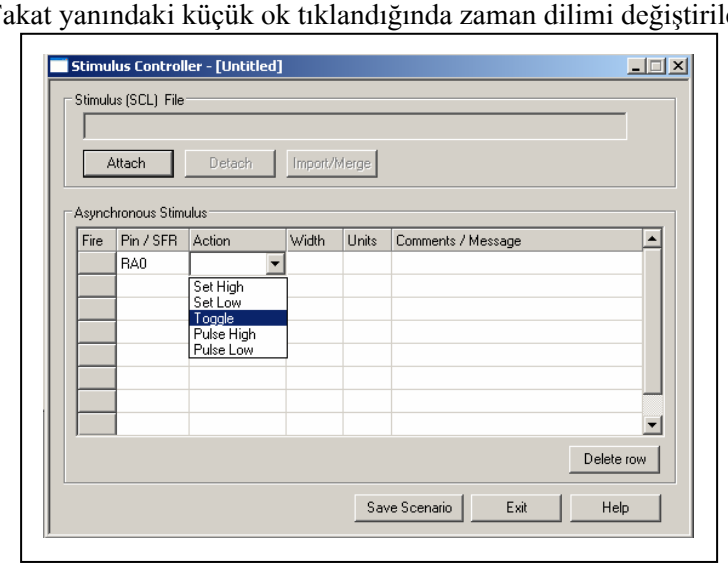
Şekil 3.39: Buton kullanımı penceresinin açılması

Açılan pencerede Pin/SFR butonu altındaki hücrelerden, kullanılmak istenen butonun hangi pin'e bağlanacağının belirlenmesi yapılır. Action butonunun altındaki hücrelerde ise buton için bir eylem çeşidi seçilir.



Şekil 3.40. Buton seçimi

Width butonu altındaki hücre içersine sadece puls high ve puls low seçeneklerinde 1 rakamı otomatik olarak yazılır. Bu rakamın anlamı 1 puls anlamındadır. Birimi ise bir yanındaki hücre olan Unit's (birim) butonu altındaki hücrelere yazılır. Otomatik olarak cyc (saykıl) gelir. Fakat yanındaki küçük ok tıklandığında zaman dilimi değiştirilebilir.

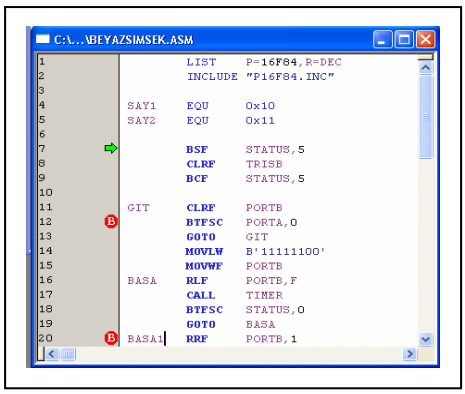


Şekil 3.41. Buton Eylem Seçimi

Buton eylem seçimi aşağıdaki şekillerde olmalıdır:

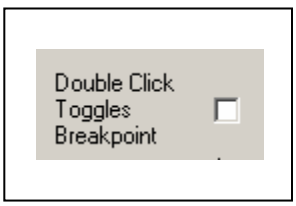
- Set high: pin girişi devamlı olarak lojik 1 seviyesindedir.
- Set low: pin girişi devamlı olarak lojik 0 seviyesindedir.
- Toggle: pin girişi en son aldığı durumun tersi lojik seviyeyi verir.

- Puls high: pin durumu devamlı lojik 0 dır. Fire butonu tıklandığında seçilen süre sonunda lojik 1 olur ve tekrar 0'a döner.
- Puls low: pin durumu devamlı lojik 1 dir. Fire butonu tıklandığında seçilen süre sonunda lojik 0 olur ve tekrar 1'e döner.



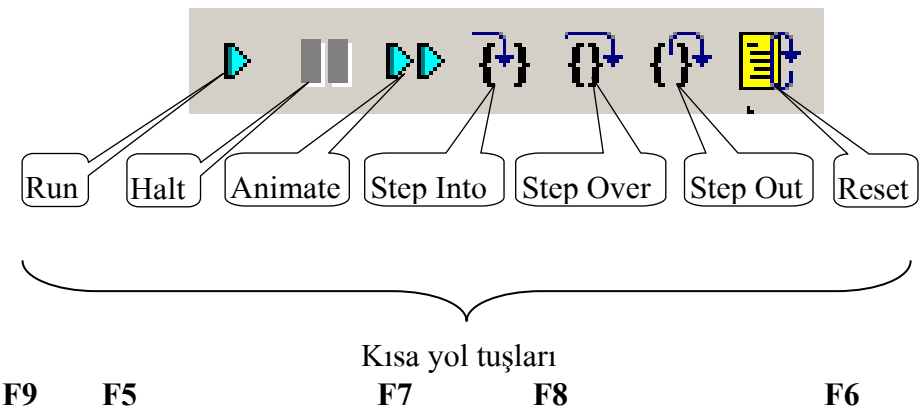
Şekil 3.42: Buton eylem seçimi

Program satırında similasyon esnasında durma yapmak istiyorsak (Break) hangi satırda duracaksa bu satır üzerinde Mouse sol tuşu iki defa klik yapılırsa sol tarafta işareti çıkacaktır. Run komutu çalıştırıldığında bu işarete kadar çalışır ve program beklemeye başlar. Eğer çift klik yapmada bu işaret çıkmıyorsa Editör ayarlarından (Şekil 4 – 12)



Şekil 3.43. Breakpoint Seçme Kutucuğu

3.5.2. Simulatör İkonlarının Açıklanması



Şekil 3.44. Similatör (Debug) İkonları

RUN: Similasyon çalışmaya başlar, halt tuşuna basılana kadar sürer ve durdurulduğunda değerleri yazar.

HALT: Program çalışmasını kesme tuşudur. Program olduğu yerde durur.

ANİMATE: Belirtilen zaman aralıklarında program satır satır otomatik olarak çalışır. Ne kadar sürede satır çalıştıracağımızı setting kısmının Debugger animation kısmından değiştirebiliriz.

STEP INTO: Bu ikonu her klik yapmada programda bir satır çalıştırılmış olur.

STEP OVER : Bu ikonu her klik yapmada programda bir satır çalıştırılmış olur. Ancak CALL komutu ile çalıştırılan Altprogramları göstermez RETURN komutuna kadar otomatik çalıştırır. Program CALL ile çağrılma satırının bir altından devam eder. Örneğin CALL TIMER dediğimizde timer alt programının nasıl çalıştığını adım adım göremeyiz.

STEP OUT : Eğer step into ile adım adım çalıştırılırken alt programın içerisine girilmiş ise bu ikona klik yapmayla alt programdan hemen çıkılır ve ana programda kaldığı yerden devam eder.

RESET : Programı reset eder ve tekrar programın en baştan başlaması için hazır hâle getirir.

Bütün bu tuşlar programda Debugger menüsünün içerisinde de mevcuttur. Bu menünün en altında bulunan setting kısmından ise kullanılan osilatör hızını seçerek gerçek zamanda program similasyonu da sağlanmış oluruz.

UYGULAMA FAALİYETİ

Aşağıdaki işlem basamaklarına göre uygulama faaliyetini yapınız.

- MPLAB programını bilgisayarınıza kurunuz.
- MPLAB programında yazdığınız hex kodunu derleyiniz.
- Derlediginiz kodu simulatör yardımı ile çalıştırarak yazmaçları kontrol ediniz.

İşlem Basamakları	Öneriler
➢ Bu öğrenme faaliyetindeki MPLAB v7.30b kurulumunu takip ederek programı bilgisayarınıza kurunuz.	➢ MPLAB programını kurarken yeni başlayanlar için complate (tamamı) seçeneği tercih edilmelidir.
➢ MPLAB sürüm 7.30'u bilgisayarınıza kurabilmek için sayfa 20 yi takip ediniz.	➢ Programınızı çalıştırırken uygulama donanımızı da kullanınız.
➢ MPLAB v7.30'u bilgisayarınıza kurulumunu tamamlayınca sayfa 12' deki programı editör sayfasına yazınız ve kaydediniz.	
➢ Yazdığınız programı derleyerek hex dosyanızı oluşturnuz.	
➢ MPLAB IDE programının simulatörünü kullanarak programı çalıştırarak kullandığınız yazmaçlardaki verileri kontrol ediniz.	

ÖLÇME VE DEĞERLENDİRME

Aşağıdaki soruları dikkatlice okuyunuz ve doğru seçeneği işaretleyiniz.

1. Program içerisinde kullanılan özel fonksiyon yazmaç isimlerinin tanıtıldığı dosyayı çağıran komut hangisidir?

A) List **B)** Status **C)** include **D)** debug

2. PIC16F84 mikro denetleyicisinde Bank0 dan Bank1 e geçmek için kullanılabilecek komut hangisidir.

A) bcf status,5 **B)** bsf status,5 **C)** bcf status,6 **D)** bsf trisa

3. Portb'nin tamamını çıkış yapmak için hangi komut uygulanmalıdır.

A) clrf status,5 **B)** clrf trisb **C)** clrf portb **D)** bsf trisa

4. İşlemci tanıtımı, nümerik değer tipi gibi tanımlamaların yapıldığı bildirim hangisidir.

A) list **B)** include **C)** equ **D)** org

5. Yazdığınız programını derledikten sonra derleyicide özel yazmaçların tanıtılmadığı belirtiyorsa ne tür bir hata yapılmıştır.

A) List talimatı yazılmamıştır
B) Giriş çıkış tanımlaması yapılmamıştır
C) Ana program yazılmamıştır
D) Include dosyası kullanılmamıştır.

6. Derleme işlemi başarısız ise aşağıdaki dosyalardan hangisi dizin içerisinde **yeralmaz.**

A) Eror **B)** Hex **C)** Cod **D)** Asm

7. Aşağıdaki sembollerden sonra yazılan yazı, rakam, sembol ve komutlar derleyici tarafından **derlenmez.**

A) ; **B)** : **C)**! **D)** "

8. Aşağıdakilerden hangisi bank değiştirmek için kullanılan yazmacın bitleridir.

A) Status 5,6 **B)** TrisA 5,6 **C)** Intcon 3,4 **D)** List 5,6

9. Sabit bir sayıyı ALI etiketli bir yazmaca kopyalamak için kullanılan komut seti hangisidir.

A) MOVF ALI,0
B) MOVLWD '…'
MOVWF ALI
GOTO ALI
C) MOVWF ALI
D) MOVLW D '…'

10. Kesme alt programlarının başlangıç adresini belirlemede kullanılan komut hangisidir.

A) call **B)** goto **C)** org **D)** clrf

DEĞERLENDİRME

Cevaplarınızı cevap anahtarı ile karşılaştırınız. Doğru cevap sayınızı belirleyerek kendinizi değerlendiriniz. Yanlış cevap verdiğiniz ya da cevap verirken tereddüt yaşadığınız sorularla ilgili konuları faaliyete geri dönerek tekrar inceleyiniz.

ÖĞRENME FAALİYETİ–4

AMAÇ

Seçilen IC-PROG programının bilgisayara yükleyip ayarlarını yapıp ve programı kullanarak PIC16F84 mikro denetleyicisini programlayacaksınız.

ARAŞTIRMA

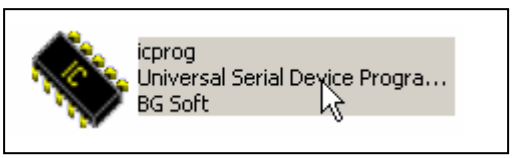
- Sevgili öğrenci, PIC'e program yüklemek için farklı yazılım ve donanımlar mevcuttur. Bizim kullandığımız yazılım ve donanımlardan başka en çok kullanılan yazılım ve donanımları araştırarak sınıfınıza sunum yapınız.

4.1. Mikro Denetleyiciye Veri Yükleme Programının Kurulumu

Akış diyagramını oluşturup programını yazdığımız ve derlediğimiz pic programımızı programlayıcı devresine göndererek entegremize yüklememiz gerekmektedir. Bu işlem için ilgisayarımızda özel bir programa ihtiyac vardır.

4.1.1. Ic-Prog Programının Kullanılması

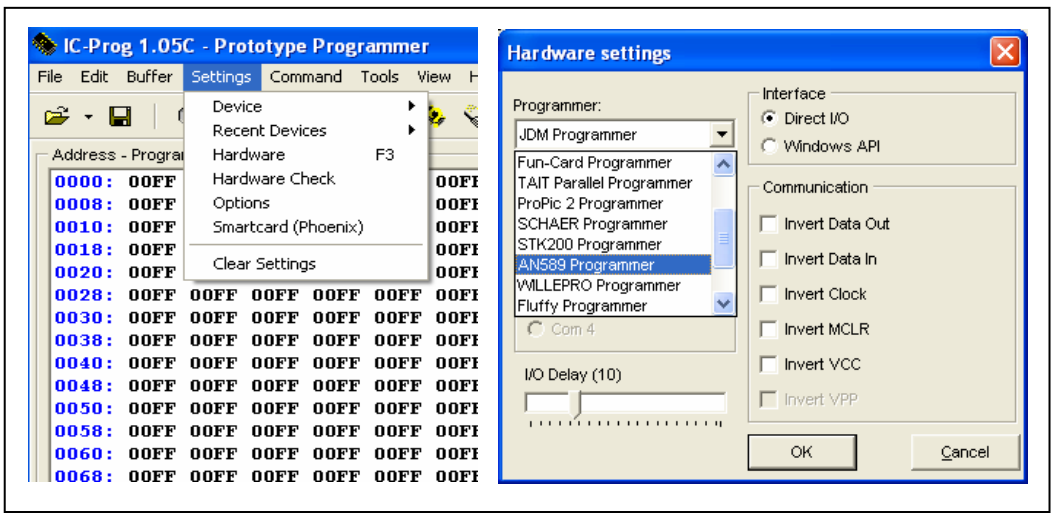
Bilgisayarda çeşitli dillerde yazılmış programlar derlenerek HEX dosya hâline getirilmektedir. Bu dosyaları da PIC mikro denetleyiciye yazmamız gerekmektedir. Bunun için çeşitli devreler ve yazılımlar mevcuttur. En yaygın olarak kullanılanlardan biri de IC-PROG yazılımıdır. IC-PROG yazılımı ile PIC çeşidinin büyük bir çoğunluğunu programlamamız mümkündür. Bunun yanı sıra birçok programlayıcı devreyi de desteklemektedir. Bu kitaptaki yazıcı devresi AN589 Programmer donanımına göre yapılmıştır. Programı çalıştırdığımızda donanım kısmından AN589 seçmemiz gerekmektedir. Bu yazılımı kurmamıza da gerek yoktur. Herhangi bir yerden kopyalamamız veya internetten indirmemiz yeterli olacaktır.



Şekil 4.1: IC-PROG programı ikonu

4.2. Program Ayarları

Yukarıdaki ikona çift tıkladığımızda ilk defa karşımıza İngilizce versiyonu açılır. İlk donanım seçimi ise JDM programmer'dir. Makinemize bu ayarları bir defa yapmamız yeterli olacaktır. Kapanıp açıldığında bizim son ayarlarımız gelmektedir. Öncelikle yazıcı tipimize göre donanım (donanım) AN589 Programmer' e seçmemiz gerekmektedir. Bunun için Setting ve Hardware menülerini takip etmemiz, F3 tuşuna basmamız veya ana sayfada tuşuna basmamız yeterli olacaktır. Her üç yolda aynı menüyü açar.

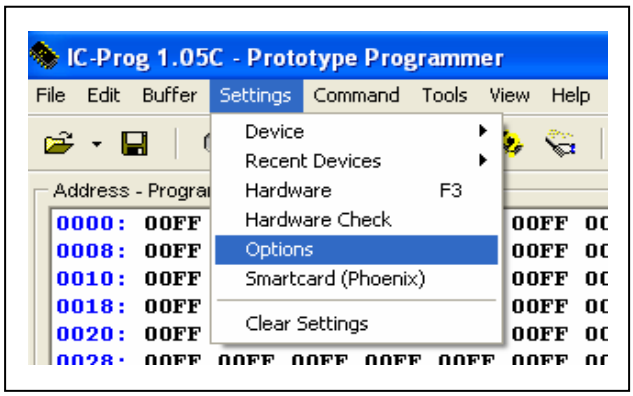


Şekil 4.2. AN589 seçim penceresi

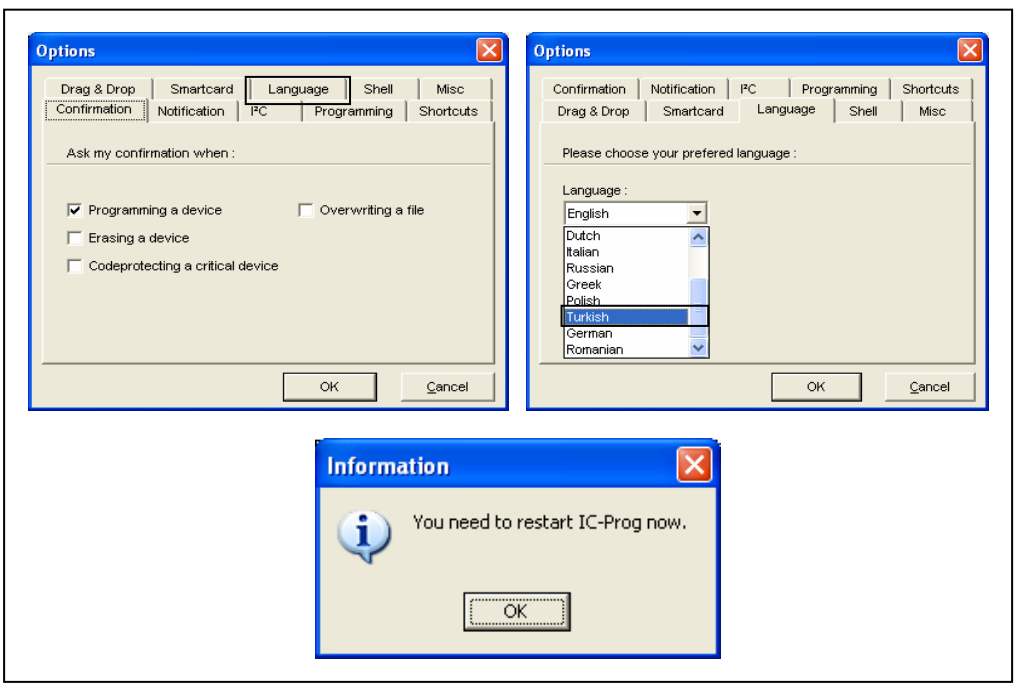
Setting / Hardware menüsünden Programmer için AN589 seçmemiz gerekir. Daha sonra OK ile menüdeki işlemimiz son bulmaktadır.

4.2.1. Yazılımın Türkçeleştirilmesi

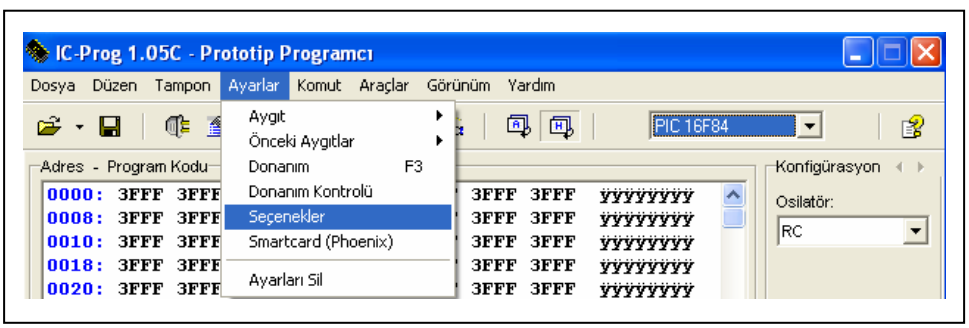
Kullandığımız yazılımın Türkçe olabilmesi içinde Setting Options menüsünü takip etmemiz gerekmektedir. Açılan Options menüsünden Language kısmıondan Turkish seçilip OK ile çıkıldığında program kendini bir defa açıp kapamak suretiyle programımız artık Türkçe olmuştur.



Şekil 4.3: Options ayarları



Şekil 4.4: Yazılımın Türkçeleştirilmesi

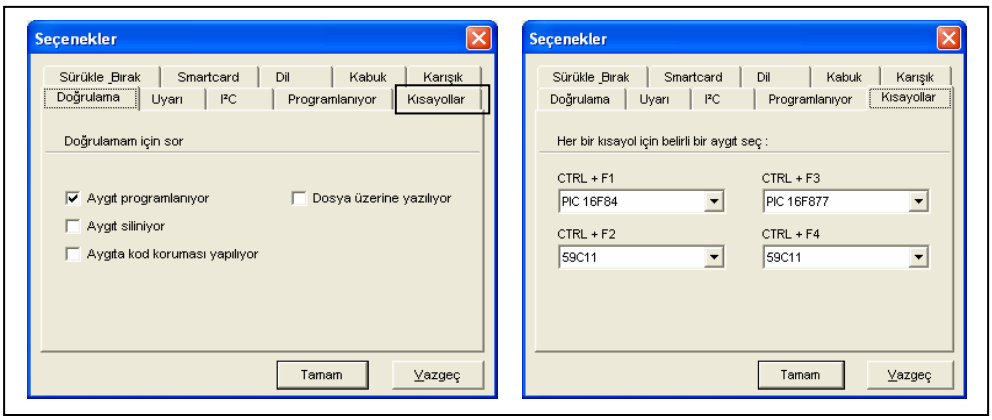


Şekil 4.5. Yazılımın Türkçe olmuş hâli

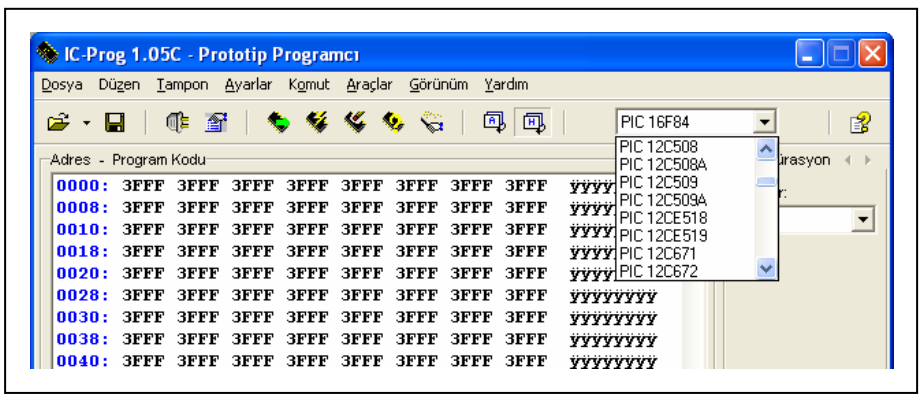
4.2.2. Pıc Türünün Seçilmesi

Bu yazılım ve yazıcı devremiz ile birçok PIC çeşidini rahatlıkla programlayabilmekteyiz. Tabi ki her defasında programlamak istediğimiz PIC çeşidini bulmak zaman almakta ve sıkıcı olmaktadır. Bunu için daha önceden belirleyeceğimiz 4 adet PIC çeşidine CTRL + F1 den CTRL + F4 e kadar kısa yol oluşturabilmek mümkündür.

Bunu gerçekleştirebilmek için Ayarlar Seçenekler menüsünü takip ettiğimizde karşımıza aşağıdaki menü çıkmaktadır. Burada Kısayollar menüsü seçilerek hangi kısa yol tuşuna hangi PIC çeşidi atanacaksa bunlar seçilir.



Şekil 4.6: Kısayol atamasının yapılması

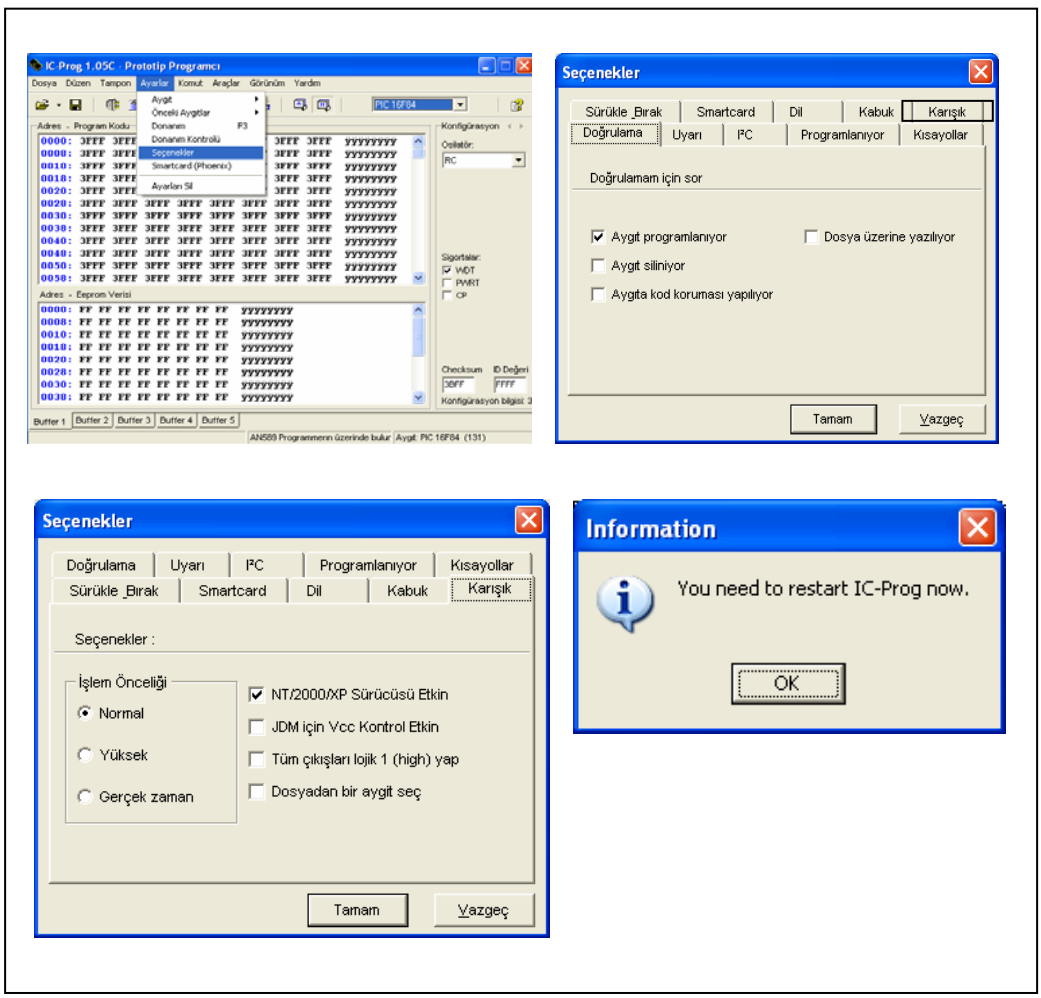


Şekil 4.7: PIC Çeşidinin seçilmesi

Tamam ile menüden çıkıldığında kısa yol tuşlarına seçilen PIC' ler atanmış olur ve artık sürekli onlar karşımıza çıkar. PIC çeşidi seçmenin bir başka yolu ise ana menüde aşağıdaki şekilde olduğu gibi açılan pencereden olmaktadır.

4.2.3. Kullanılan İşletim Sistemine Uyumu Sağlama

IC-PROG yazılımını kullanmaya başlamadan önce son yapacağımız ayar ise kullanacağımız bilgisayarda kurulu olan yazılıma uyum sağlamak olacaktır. Programı eğer Windows 95 – Windows 98 – Windows 98 me gibi yazılım yüklü makinelerde kullanacaksak bunun için ayarlarında herhangi bir değişiklik yapmamıza gerek yoktur. Ancak Windows 2000 - WindowsXP – Windows NT gibi yazılım yüklü makinelerde kullanacaksak ayarlarını buna uyumlu hâle getirmemiz gereklidir.

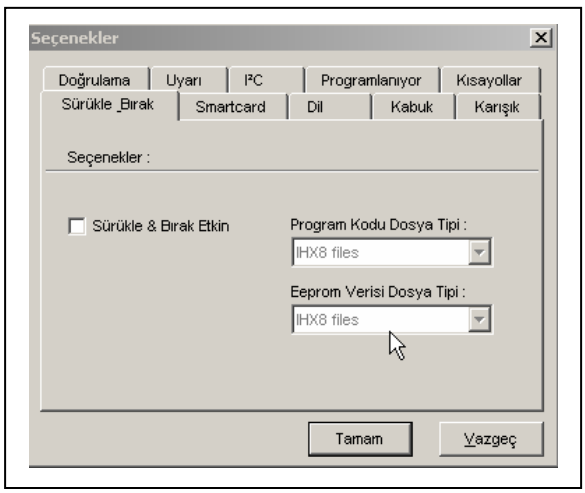


Şekil 4.8: İşletim sistemi uyumunu ayarlama

Bunun için Ayarlar Seçenekler menüsü takip ederek Karışık yazan kısmı seçmemiz daha sonra karşımıza çıkan pencerede ise NT/2000/XP kutucuğunun işaretli olması gerekmektedir. Bunu işaretleyip Tamam ile pencereyi kapattığımızda ayarların geçerli olabilmesi için program kendini bir kez açıp kapaması gerekmektedir. Tekrar açılan programda artık kullandığınız yazılım türüne uygun hâle gelmiş olmaktadır.

4.2.4. Sürükle Bırak Ayarı

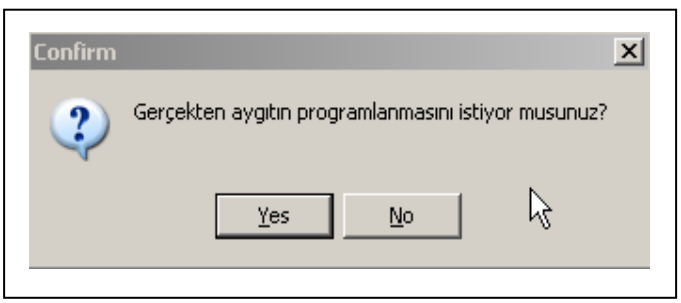
Kullanımı daha kolay hâle getirebilmek için değişik ayarlar da gerçekleştirebiliriz. Bunlardan HEX uzantılı bir dosyayı sürükleyerek programımızın üzerine bıraktığımızda açılmasını istiyorsak Ayarlar / Seçenekler / Sürükle Bırak kısmına ulaşıp buradaki kutucuğu işaretleyip Tamam ile çıkmamız yeterli olacaktır. Bu ayarı bir defa yapmamız yeterli olacaktır.



Şekil 4.9: Sürükle bırak ayarı

4.2.5. Doğrulama Ayarı

Ic-prog yazılımı ile programlama gerçekleştirirken veya PIC'teki bir bilgiyi silme esnasında bunun yanlışlıkla yapılabileceğini düşünenler için doğrulama bölümünü aktif hâle getirmemiz mümkündür. Hangi durumda onaylama istiyorsak bunun kutucuğunu işaretlememiz yeterli olacaktır. Örneğin Aygıt programlanıyor kutucuğu işaretli ise ve biz yazılıma programlama komutunu uyguladığımızda aşağıdaki soruyu mutlaka sorarak doğrulama gerçekleştirir. Eğer işaretli değilse direk programlama işlemini gerçekleştirir.



Şekil 4.10: Doğrulama penceresi

4.2.6. Ana Menüdeki İkonlar ve İşlevleri

 Dosya Açma

 Dosya kayıt etme

 Programlayıcı Seçimi

 Seçenekler menüsünü açar

 PIC'teki programı okuma

 PIC'e program yazma

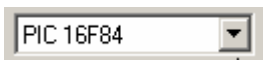
 PIC'teki programı silme

 Aygıt doğrulaması yapar

 Smartcard sihirbazı

 Yazılan veya okunan programın Assembler görünümü

 Yazılan veya okunan programın HEX görünümü

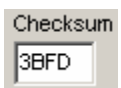


Programlanacak PIC türü

Osilatör seçim penceresi

Aktif olan Buffer katları. Bunlardan herhangi birini seçebiliriz.

PIC' in kimlik numarası seklindedir. Bunu değiştirmekte mümkündür.



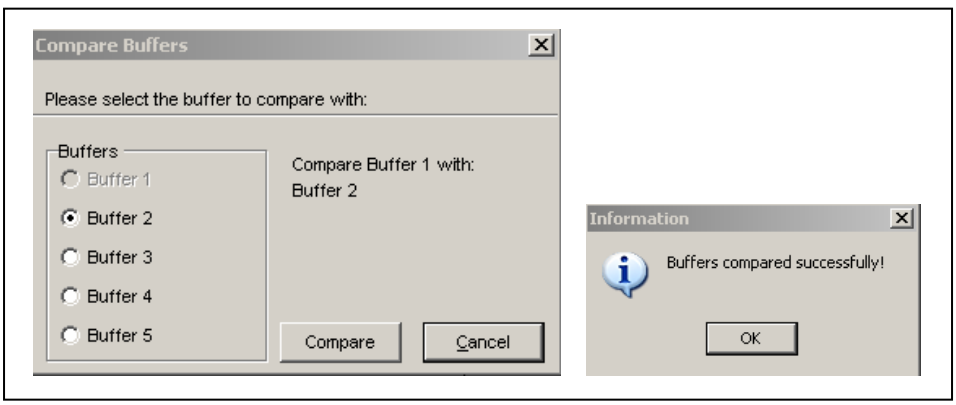
Yazılım programa göre bir sayı üretir. Her hangi bir değişiklik yapıldığında bu sayı da değişir. Böylece programda bir değişiklik yapıldığında yenisinin açılıp açılmadığını bu sayı ile kontrol edebiliriz veya Buffer 2'ye yazılan program okutulursa bu sayı aynı ise yazım kesin olarak doğru sonucuna varılır.

4.2.7. Karşılaştırma Yapılması

Şekil 4.11: Karşılaştırma yapılması ile ilgili alt menü

İki program arasında fark var ise ve bu farkın nerelerde olduğunu anlamak istiyorsak bunu da yazılım ile karşılaştırabiliriz. Bunun için karşılaştırılacak iki programı değişik Buffer'lara açmamız gerekir.

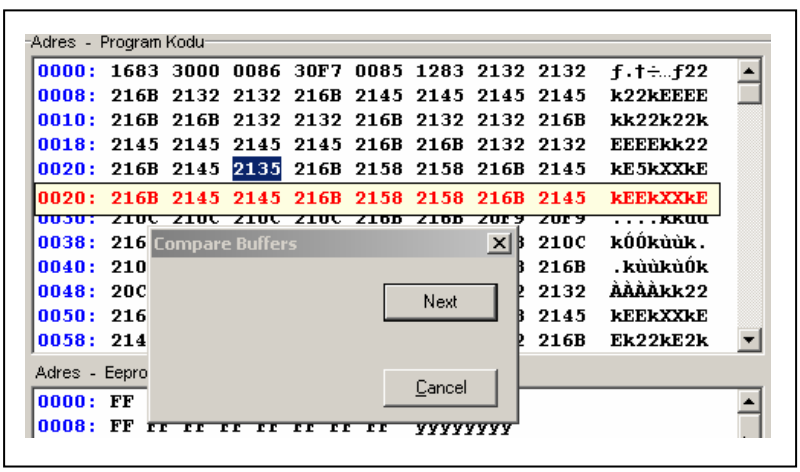
Örneğin birini BUFFER 1 'e diğerini ise BUFFER 2 ' ye açabiliriz. Daha sonra Tampon menüsünden karşılaştır seçilir.



Şekil 4.11: Karşılaştırma penceresi

Karşılaştır seçildikten sonra Buffer 1 aktif olduğundan karşılaştırılacak diğer Buffer işaretlenir. Daha sonra Compare (Karşılaştır) ikonu seçilerek karşılaştırma yapılır. Her ikisi de aynı ise yandaki bilgi mesajı gözükür.

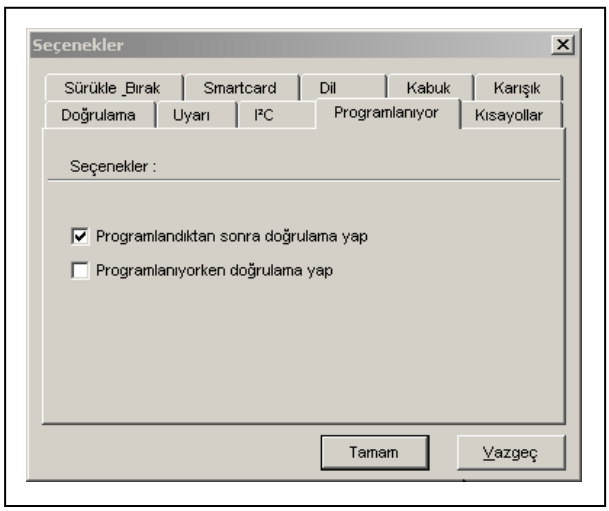
Eğer programda değişiklik varsa bunu da aşağıdaki şekildeki gibi adım adım gösterebilmektedir. Örnekte 2135 ile 2145 arasındaki farkı göstermektedir. Başka değişikliler olup olmadığını anlayabilmek için ise Next tuşuna basmamız yeterli olacaktır.



Şekil 4.12: Farklı programların karşılaştırılması

4.2.8. Yazım Doğrulaması

IC-Prog yazılımı PIC' e program yazarken iki çeşit kontrol yapmaktadır. Birincisi önce bir adresi yazar, sonra onu okur ve doğrular bu işlem program yazımı bitene kadar gerçekleşir. Hata olursa hemen uyarı verir. İkinci bir seçenek ise tümünü PIC'e yazar ve hepsini okuyup doğrulama yapar. Bu iki seçenek programın PIC'e doğru yazılıp yazılmadığını büyük ölçüde bize bildirir. Tabi ki bu işlem bize program yazma esnasında zaman harcamamıza sebep olur ve işlem süresi uzar. Ancak okuyup doğrulama yapmasını istemediğimizde daha kısa sürede işlem gerçekleşir fakat sağlıklı yazıp yazmadığı kuşkuludur.

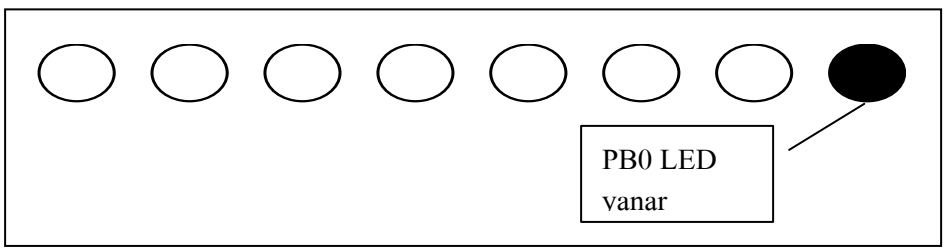


Şekil 4.12: Programlama doğrulama penceresi

Bu işlemleri yazılımda Ayarlar / Seçenekler kısmına girdiğimizde karşımıza aşağıdaki gibi çıkan pencereden ayarlayabiliriz.

4.3. Yazılan Programın Sonucu

Öğrenme faaliyeti 3 sayfa 12'deki basit programı yine aynı modülde yazıp ve derlemiştiniz. Elde edilen hex dosyasını IC-PROG programı yardımıyla PIC eğitim setindeki PIC' e göndererek çalıştırdığımızda şekil 4.13'teki sonuç elde edilecektir.



Şekil 4.13: Yazılan programın sonuç ifadesi

UYGULAMA FAALİYETİ

Aşağıdaki işlem basamaklarına göre uygulama faaliyetini yapınız.

- ICPROG programını kurup gerekli ayaraları yapınız.
- Önceki uygulamadaki derlediginiz hex programını ICPROG programı yardımı ile PIC e yazınız.
- Programın çalışmasını kontrol ediniz.

İşlem Basamakları	Öneriler
➢ Bu öğrenme faaliyetindeki IC-PROG programının kurulumunu takip ederek programı bilgisayarınıza kurunuz.	➢ IC-PROG programını kurduktan sonra kısa yolunu masa üstüne atabilirsiniz.
➢ Kurulan programın ayarlarını bu öğrenme faaliyetindeki işlemleri takip ederek uygulayınız.	
➢ Öğrenme faaliyeti 3 deki uygulamada elde edilen hex dosyasını eğitim setinizdeki PIC'e yazdırmak için gerekli donanımı hazırlayınız.	➢ Eğitim setinin besleme kaynağını kullanmayı unutmayınız.
➢ Hex dosyanızı IC-PROG programında açarak yapılandırma ayarlarını yaparak PIC'e gönderiniz.	➢ Program çalışmıyorsa yazılım ayarlarını ve donanım kontrollerini yapınız.
➢ Elde edilen sonucu ledler üzerinde görünüz.	➢ Hatanızı bulamadınız ise öğretmeninize danışınız.

ÖLÇME VE DEĞERLENDİRME

Aşağıdaki soruları dikkatlice okuyunuz ve doğru seçeneği işaretleyiniz.

1. Kullandığımız eğitim setine uygun programlama çeşidi hangisi seçilmelidir?

A) JMD **B)** AN589 **C)** Pro Pic 2 **D)** Seri

2. IC-PROG programında dil seçimi hangi menüden yapılır?

A) Setting/Options/Language **C)** Edit/Options/Language

B) Setting/Hardware/Language **D)** View/Configure/Language

3. IC-PROG programında açılan hex dosyası hangi ikon tıklanarak PIC'e gönderilir.

A)  **B)**  **C)**  **D)** 

4. Hangi ikon tıklanarak PIC'e yazılmış olan program aktif buffer'a kopyalanır.

A)  **B)**  **C)**  **D)** 

5. Eğitim setine program yüklendikten sonra ilk çalıştırmada otomatik resetleme yapması için yapılandırma ayarlarından hangi kutucuk işaretlenmelidir.

A) WDT **B)** PWRT **C)** CP **D)** LVP

DEĞERLENDİRME

Cevaplarınızı cevap anahtarı ile karşılaştırınız. Doğru cevap sayınızı belirleyerek kendinizi değerlendiriniz. Yanlış cevap verdiğiniz ya da cevap verirken tereddüt yaşadığınız sorularla ilgili konuları faaliyete geri dönerek tekrar inceleyiniz.

MODÜL DEĞERLENDİRME

Aşağıdaki soruları dikkatlice okuyunuz ve doğru seçeneği işaretleyiniz.

1. PIC16F84 mikro denetleyicisinin kaç adet I/O portu vardır?

A) 8 **B)** 5 **C)** 13 **D)** 18

2. PIC16F84 mikro denetleyicisinin program bellek kapasitesi ne kadardır?

A) 1kB **B)** 1MB **C)** 64KB **D)** 64MB

3. PIC16F84 mikro denetleyicisinde osilatör frekansı 4 [MHz] ise 1 clock süresi ne kadardır?

A) $4\,\mu s$ **B)** $1\ \mu s$ **C)** 1 ms **D)** Hiçbiri

4. PIC16F84 mikro denetleyicisinde, kaç adet bank vardır?

A) Yok **B)** 1 **C)** 2 **D)** 4

5. Uygulamalarımızda kullanacağımız pic programlayıcı hangisidir?

A) AN589 **B)** JDM **C)** Fun_Card **D)** ProPic 2

6. Programlayıcıda data(veri) ucu pic'in hangi pininine bağlıdır?

A) RA2 **B)** RB3 **C)** RB7 **D)** RB6

7. Kullandığımız pic mikro denetleyicisini programlarken kaç volt gerilime ihtiyaç vardır?

A) 5V **B)** 18V **C)** 12,5V **D)** 3V

8. PIC16F84 mikro denetleyicisinde giriş veya çıkış olarak ayarlanabilecek toplam kaç pin vardır?

A) 7 **B)** 8 **C)** 5 **D)** 13

9. Sabit bir sayıyı ALI etiketli bir yazmaca kopyalamak için kullanılan komut seti hangisidir.

A) MOVF ALI,0 **C)** MOVWF ALI

B) MOVLW D '…'
MOVWF ALI **D)** MOVLW D '…'
GOTO ALI

10. Kesme alt programlarının başlangıç adresini belirlemede kullanılan komut hangisidir.

A) call **B)** goto **C)** org **D)** clrf

KONTROL LİSTESİ

Bu faaliyet kapsamında aşağıda listelenen davranışlardan kazandığınız becerileri "**Evet"** ve "**Hayır"** kutucuklarına (X) işareti koyarak kontrol ediniz.

DEĞERLENDİRME ÖLÇÜTLERİ		Evet	Hayır
1.	Mikroişlemci ile mikrodenetleyici arasındaki farklılıkları kavrayabildiniz mi?		
2.	PIC16F84 mikrodenetleyicisinin çalışma mantığını kavrayabildiniz mi?		
3.	Mikrodenetleyici seçiminin nasıl yapılması gerektiğini anlayabildiniz mi?		
4.	PIC16F84 mikrodenetleyicisinin yapısı öğrenilebildi mi?		
5.	PIC16F84 mikrodenetleyicisinin programlanabilmesi için hangi koşulların oluşturulması gerektiği öğrenilebildi mi?		
6.	MPLAB IDE ve IC-PROG programlarını bilgisayarınıza yükleyebilirmisiniz?		
7.	MPLAB IDE ve IC-PROG programlarını kullanma becerisine sahip oldunuz mu?		
8.	PIC16F84 mikrodenetleyici eğitim setini oluşturabilmek için gerekli elektronik elemanların özelliklerini ve kullanımını kavradınız mı?		
9.	PIC16F84 mikrodenetleyicisine program yazarken meydana gelebilecek yazılımsal hataları çözebilecek beceriye sahip oldunuz mu?		
10.	PIC16F84 mikrodenetleyicisine program yazarken meydana gelebilecek donanımsal arızaları çözebilecek beceriye sahip oldunuz mu?		

DEĞERLENDİRME

Değerlendirme sonunda **"Hayır"** şeklindeki cevaplarınızı bir daha gözden geçiriniz. Kendinizi yeterli görmüyorsanız öğrenme faaliyetini tekrar ediniz. Bütün cevaplarınız **"Evet"** ise bir sonraki modüle geçebilirsiniz.

CEVAP ANAHTARLARI

ÖĞRENME FAALİYETİ–1'İN CEVAP ANAHTARI

1.	**C**
2.	**A**
3.	**B**
4.	**C**
5.	**D**
6.	**A**
7.	**C**
8.	**C**

9. Bir mikro denetleyici ile bir mikro işlemci arasındaki en büyük fark, mikro denetleyicide program hafızası ve giriş çıkış özelliklerini kendi içinde barındıran üniteleri tek bir yonga içersinde yer almaktadır.

10. PIC16F84 8 bitlik bir mikro denetleyicidir.

ÖĞRENME FAALİYETİ–2'NİN CEVAP ANAHTARI

1.	**A**
2.	**C**
3.	**C**
4.	**D**
5.	**A**
6.	**A**
7.	**D**
8.	**B**

9. Program çalışırken, programı durdurmak ve başlangıç durumuna gelmek için reset pininin lojik seviyesi 0 yapılmalıdır.

10. Pic'e yazılan program çalışırken reset pinin gerilim seviyesi 5V olmalıdır.

ÖĞRENME FAALİYETİ–3'ÜN CEVAP ANAHTARI

1.	**C**
2.	**B**
3.	**B**
4.	**A**
5.	**D**
6.	**B**
7.	**A**
8.	**A**
9.	**B**
10.	**C**

ÖĞRENME FAALİYETİ–4'ÜN CEVAP ANAHTARI

1.	**B**
2.	**A**
3.	**C**
4.	**A**
5.	**B**

MODÜL DEĞERLENDİRMENİN CEVAP ANAHTARI

1.	**C**
2.	**A**
3.	**B**
4.	**C**
5.	**A**
6.	**C**
7.	**C**
8.	**D**
9.	**B**
10.	**C**

KAYNAKÇA

- Teramoto Koshi, İşbilen Turgay, Güneş Mustafa, **PIC 16F84 Mikro denetleyici Temel Bilgileri, Programlanması ve Uygulamaları** JICA 2.Baskı, İzmir, 2004.

- Altınbaşak Orhan, **Mikro denetleyiciler ve PIC programlama** Eylül İstanbul, 2000.