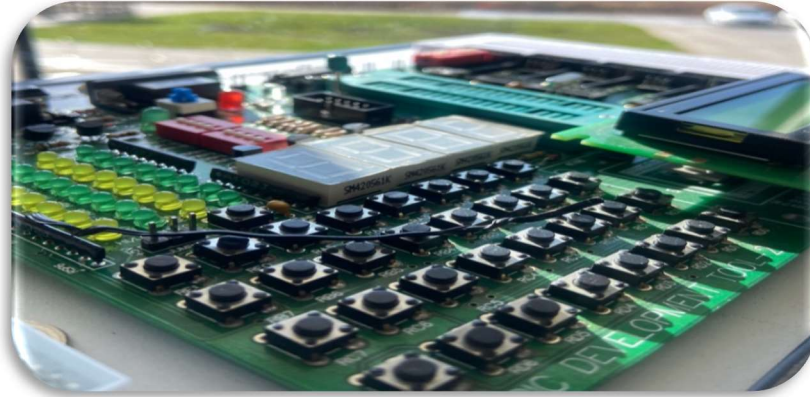




T.C.
MALATYA TURGUT ÖZAL ÜNİVERSİTESİ
MÜHENDİSLİK VE DOĞA BİLİMLERİ FAKÜLTESİ

ELEKTRİK - ELEKTRONİK MÜHENDİSLİĞİ
BÖLÜMÜ

EEM 304 MİKROİŞLEMCİLER LABORATUVARI
DENEY FÖYÜ



Doç. Dr. Ali YÜCE
Arş. Gör. Muhammed Buğracan ÖZKÜÇÜK

2025-2026 Bahar Dönemi

LABORATUVAR GÜVENLİK KLAVUZU

Laboratuvar ortamında çalışanların sağlık ve güvenliği ile yürütülen çalışmaların başarısı için temel güvenlik kurallarına uyulması büyük önem taşımaktadır. Bu sebeple aşağıda tanımlanan kurallara uyulması gerekmektedir.

- 13 mA 'den büyük akım veya 40 V 'dan büyük voltajlar insan sağlığı için tehlike arz etmektedir ve öldürücü etkisi vardır. Bu nedenle elektrik çarpmalarından korunmak için gerekli önlemleri alınız ve görevlilerin uyarılarına mutlaka uyunuz. Kaza ve yaralanmalar olduğu zaman görevliye derhal haber veriniz. Kazayı bildirmek için vakit geçirmeyiniz.
- Hasara uğramış veya çalışmayan alet ve cihazları derhal laboratuvar görevlisine bildirin.
- Herhangi bir nedenle hasar verdiğiniz tüm cihaz ve donanımlarının onarımı ya da yeniden alınma bedeli tarafınızdan karşılanacaktır. Cihazların üzerine kitap defter gibi ağır malzemeler yerleştirmeyiniz ve yerlerini değiştirmeyiniz.
- Multimetreleri ölçüm kademelerinin sınırı dışındaki akım veya gerilim kademelerinde çalıştırmayınız. Güç kaynaklarından düşük gerilim alınız.
- Laboratuvarların sessiz ve sakin ortamını bozacak yüksek sesle konuşmak, tartışma yapmak, başka grupların çalışmalarını engellemek, izin almadan laboratuvarı terk etmek, diğer gruplardan yardım almaya çalışmak ve laboratuvarda dolaşmak yasaktır.
- Laboratuvarlara yiyecek ve içecek sokmak yasaktır.
- Laboratuvarlarda cep telefonu kullanımı yasaktır.
- Çalışma esnasında saçlar uzun ise mutlaka toplanmalıdır.
- Çalışma bittikten sonra kullanılan cihazlar yerlerine konulmalıdır.
- Laboratuvarda çalıştığımız alanın temizliği sizin sorumluluğunuzdadır. Çalışmalar bittikten sonra gereken temizlik yapılmalıdır.
- Laboratuvardan çıkmadan önce masanın enerjisi kesilmelidir.

DİKKAT!

Laboratuvarda çalışan herkesin belirtilen kuralların tümüne uyması zorunludur. Bu kurallara uymayanlar laboratuvar sorumluları tarafından uyarılacak, gerekirse laboratuvardan süreli uzaklaştırma ile cezalandırılacaklardır. Laboratuvara kasıtlı olarak zarar verdiği tespit edilen kişiler laboratuvardan süresiz olarak uzaklaştırılacak ve verilen zarar tazmin ettirilecektir.

PIC MİKRODENETLEYİCİSİ VE UYGULAMALARI

GİRİŞ

Günümüzde gelişen teknolojik imkanların artması birçok uygulamada daha önce görülmemiş birçok yenilik insanların hizmetine sunulmaktadır. Kimi zaman bu hizmetler doğrudan insanlarla etkileşim halinde olan sistemlerle kimi zaman ise endüstriyel ortamlarda insanların görevini devir alan otomasyon sistemleri ile sağlanmaktadır. İlerleyen bilim, teknoloji düzeyi ve buna paralel olarak gelişen, değişen dünya pazarları; endüstriyel ürünlerde nitelik açısından ve de işlevsel olarak önemli değişimlere neden olmuştur. Hızla gelişen teknoloji ve sürekli değişen Pazar koşulları daha ekonomik ve kaliteli ürünler isterken, bunlara ek müşterilerde sürekli olarak daha esnek ve çok işlevli ürünler istemektedir. Hızla değişen müşteri istekleri ve yoğun rekabet sonucu ürünlerin teknolojik ömürleri çok kısalmıştır. Bu koşullar karşısında geleneksel tasarım ve imalat teknolojileri yetersiz kalmış, bu talepleri karşılayabilecek yeni kavramlar ve yöntemler gelişmiştir. Bunlardan birisi de Mekatronik Mühendisliği kavramıdır. Mekatronik çok disiplinli ve disiplinler arası konuları kapsayan bir mühendislik felsefesi ve mühendislik uygulamalarına tümleşik bir yaklaşımdır. Mekatronik Mühendisliği, Makine Mühendisliği, Elektrik/Elektronik Mühendisliği, Bilgisayar teknolojisinin eş amaçlı ve tümleşik bir yapıda uygulamasıdır. Çağdaş Mekatronik teknolojisi ürünleri bir veya birkaç mikroişlemci çerçevesinde yerleştirilen duyucular; eyleyiciler, tüm sistem veya makineyi merkezi veya dağıtık yapıda denetleyebilen bilgisayar programlarından oluşmaktadır. Bu tanıma uygun sistem ve makineler kendisine tanımlanan çevreyi gözlemlemekte, çevredeki değişimleri algılamakta ve algıladığı bilgileri yorumlayarak gerekli motor sistemler yardımıyla çevresini değiştirebilmektedir. Makineler mekanik işlevsellik ile tümleşik algoritmik denetimi beraberce içeren ürün ve sistemlerdir. Gelişmiş mekatronik ürünler basit makineler yerine çevrelerini değiştirebilen bilgisayar sistemlerine dönüşmüştür. Doğal olarak bu yapıdaki makine ve sistemler akıllı davranışlar göstermektedir. Mekatronik ürünlerde yer alan yazılımlar genellikle yapay zekâ tekniklerini kullanmakta ve böylece mekatronik tasarım ürünleri, basit işlevsel makineler yerine, çeşitli koşullara uyum sağlayan yetenekli sistemlere dönüşmektedir. Mekatronik sistemler Türkiye'nin uluslararası pazarlarda giderek daha etkin hale geldiği birçok teknoloji alanı ve sanayi sektörü için önemli bir girdi olmuştur. Otomotiv, beyaz eşya, elektrikli ev aletleri, akıllı bina ve otomasyon sistemleri, gıda, sağlık, sivil, askeri hizmet sektörleri, MEMS ve sensör teknolojileri bu alanlara örnek olarak gösterilebilir. Bu deneyler kapsamında anlatılacak olan PIC Mikrodenetleyicisi ile Elektrik elektronik mühendisliği disiplininin temel konularından biri olan ayrık zamanlı kontrol uygulamalarındaki kullanımına dair pratik bilgi eksikliği giderilmeye çalışılacaktır. Bu deney kapsamın da yapılacak olan uygulamalarda daha önceden teorik olarak verilen lojik ve programlama eğitimi vardır. Bu eğitimlerin uygulamaları geliştirilerek öğrencilerin daha sonraki meslek hayatlarında karşılaşacakları buna benzer problemlere belirli bakış için eğitimi sağlanacaktır.

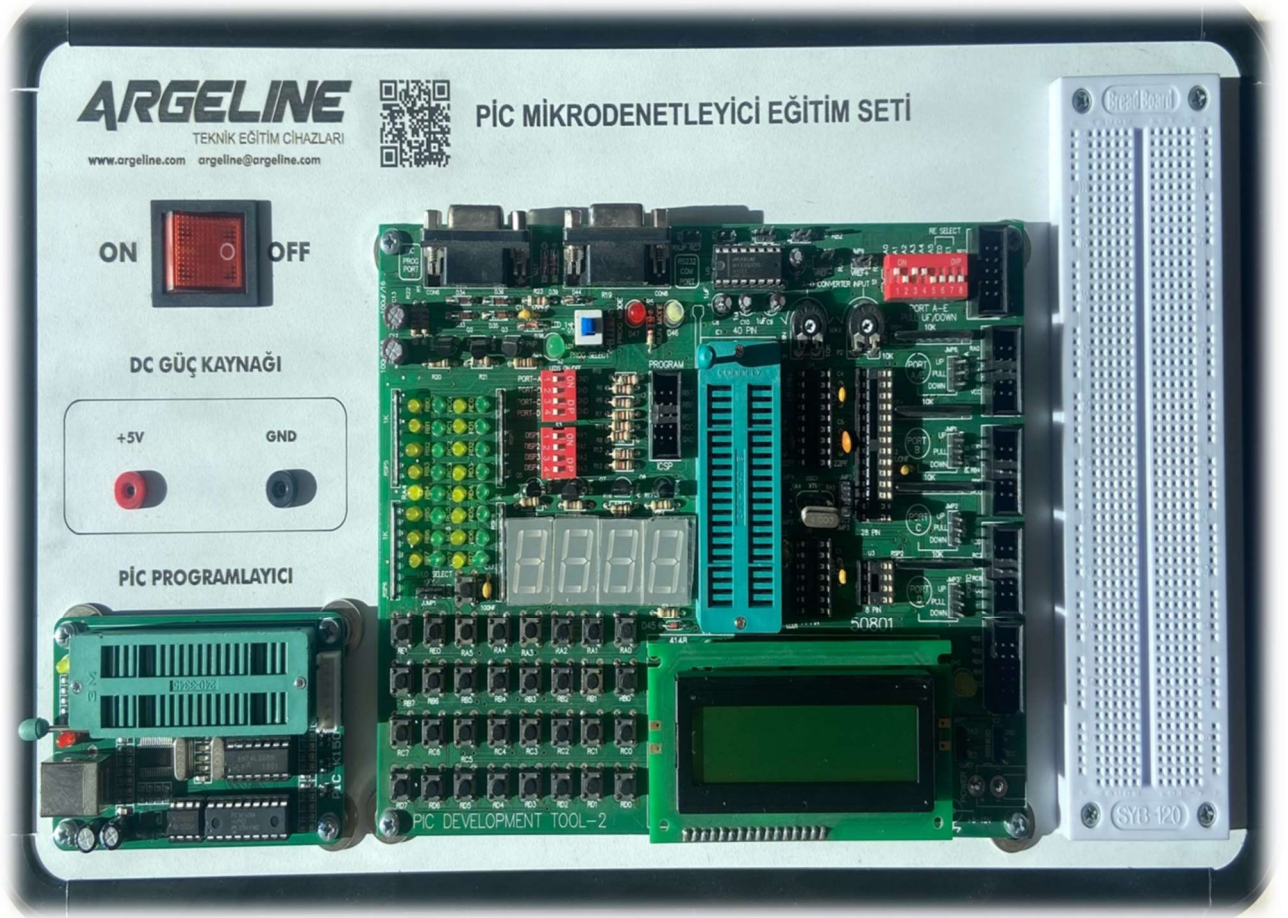
DENEYİN AMACI

Bu deneyde ucuz maliyeti ve kolay kullanımı sebebi ile tercih edilen PIC Mikrodenetleyicisinin tanıtılması gerçekleştirilecektir. Öncelikle bu elemanın sahip olduğu RISC (Reduced Instruction Set Computing) mimarisi tanıtılacak, daha sonra ise bu mimariyle gerçekleştirilebilecek birkaç temel uygulama örneği gösterilecektir. PIC Mikrodenetleyicisinin programlanması temelde sadece ASM dili kullanılarak gerçekleştirilebilmesine rağmen, daha karmaşık iş senaryolarının bir senaryo üzerine düzen içerisinde kolay bir şekilde yapılabilmesi için PICBASIC, PICC, PIC-CCS gibi birçok üst seviye dilde PIC programlarının yazılabilmesi mümkündür. Bizde bu deney kapsamında gerçekleştirilecek uygulamalar için eğitim açısından oldukça kolay ve basit bir kullanım sağlayan PIC-CCS derleyicisinden faydalanacağız.

ÖN BİLGİ

Devrelerin Kurulumu için Kullanılacak Deney Seti

Şekil 1’de genel amaçlı olarak tasarlanmış PIC Mikrodenetleyici eğitim seti üzerinde farklı amaçlarda kullanılabilen; kayan yazı, LCD ekran, AC Motor sürücüsü, enkoder sayacı, step motor sürücüsü, tuş paneli, proton board, ayarlı dirençler, çeşitli işlevlerde kullanılan ısı, ışık, manyetik sensörler gibi birçok eleman kolayca görüntülenebilmektedir.



Şekil 1. Devrelerin kurulumu için kullanılacak deney seti

Mikroişlemciler ve Mikrodenetleyiciler

Mikroişlemciler

Mikroişlemciler, sayısal bilgileri adres ve veri yolu ile alan, bu bilgileri bir hafıza biriminde saklanmış program komutlarına uygun olarak işleyen ve elde edilen sonuçları sayısal çıktıya dönüştüren mantık devreleridir. Bir mikroişlemcinin temel işlevi sistem register'lerinde bulunan verileri içerden dışarıya veya dışardan içeriye ve verileri bir durumdan diğerine çevirme işlemidir. Bir mikroişlemcinin iç yapısı 4 bölümden oluşur. Bu bölümler; Kontrol birimi, Genel Amaçlı Registerlar, Özel amaçlı registerlar, Aritmetik ve Lojik birimdir. Mikroişlemcinin kontrol birimi, mikroişlemcinin içinde ve dışında olan bütün veri aktarımlarını ve ALU işlemlerini kontrol eder ve çevre birimlere senkronize olmak için gerekli iletişimi sağlar. Mikroişlemcinin içinde işlem yaparken geçici olarak verilerin saklandığı yer Genel Amaçlı registerlardır (GPR) 8 bit mikroişlemcilerin çoğunda bu registerlar üzerinde veri saklama işleminden başka işlem yapılamaz. Kontrol birimi tarafından yürütülecek komuta işaret etmek amacıyla bütün mikroişlemcilerde özel amaçlı registerlar (Special Function Register) vardır. Bir mikroişlemcinin aritmetik ve lojik birimleri ALU adı verilen birimde yapılıdır.

Mikrodenetleyiciler

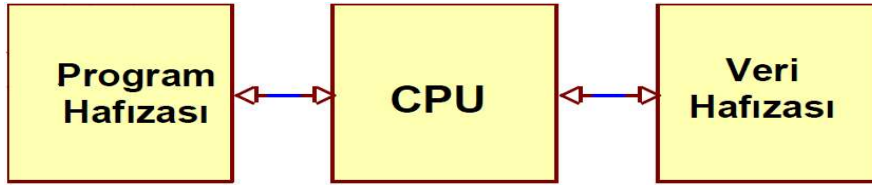
Mikrokontrolör, mikroişlemci ve mikroişlemciye bağlı bütün birimlerin üzerinde bulunduğu tek yongaya denir. Bir mikrokontrolör yongasında bulunan hafıza, giriş çıkış ve diğer donanım alt sistemleri bu işlemcinin birçok uygulama içinde gömülü (embedded) olarak ve tek başına, bir mikroişlemciye göre çok daha basit ve ucuz ara bilim teknikleriyle, kontrol amaçlı olarak kullanımını sağlar. Bir mikrokontrolör kullanıldığı sistemin birçok özelliğini aynı anda izleme (monitoring) ihtiyaç anında gerçek zaman da cevap verme, bu mikrokontrolörün hazır olduğu anda alıp ortamı bekletmeden işleyebilmesi demektir ve sistemi denetlemekten sorumludur. Mikrokontrolörün kullanım alanları çok çeşitlidir. Örneğin otomobillerde, motor kontrol, elektrik, iç panel kontrol, kameralarda, ışık ve odaklama kontrol, bilgisayar, telefon vb. her haberleşme cihazları, TV, teyp, özel kartlar vb. daha birçok alanda mikrokontrolörler kullanılırlar. Bu kadar geniş kullanım alanı mikrokontrolörler için 1980'lerden sonra gelişmiş özelliklere ve desteklere sahip 8-bit, 16-bit ve 32-bit mikrokontrolör kullanılmaktadır. İkili kod düzeyi ile uyumlu pek çok 8-bit ve 16-bit ürünleri piyasaya sürülmüştür. Mikrokontrolörün gelişimini yönlendiren talepler ise şunlardır:

- Karmaşık cihazlar için daha yüksek performans, daha geniş adres alanı,
- C gibi yüksek seviyede programlama desteği,
- Windows altında çalışan gelişmiş hata takibi özelliklerine sahip program geliştirme ortamları,
- Büyük geliştirme programları ve yazılım güvenliği açısından var olan programların kullanılması,
- Daha az güç tüketimi ve gürültü.

PIC Nedir?

PIC Mikrodenetleyicilerin tasarımı Harvard Üniversitesi Savunma Bölümünün araştırmalarına dayanmaktadır. Mimarisi de aynı isimle Harvard mimarisi olarak tasarlanmıştır. Bu tasarımlar General

Instrument'ın işlemcilerin giriş çıkış sayılarını arttırmak için çevresel bağlantı denetleyici (Peripheral Interface Controller – şimdilerde PIC olarak kısaca adlandırılıyor) tasarlandı. 1985'te Arizona'da Microchip Technology mikroelektronik bölümü ana üretimini PIC'e dönüştürdü. Mikrodenetleyici mimarisinde kullanılan geleneksel Von Neumann mimarisinde bir adet databus vardır. Dolayısıyla komut ve veri hattı aynı genişliktedir. Komutlar ve veriler bu hat üzerinden taşınır, genel bir hafızaya kaydedilir. Burada iki defa fetch olayı olduğundan mikrodenetleyicinin hızı düşer ancak geniş hafıza tasarımları için kolaylık sağlar. 8051, 8086 ve 68HC11'lerin mimarisinde Von Neumann bulunur. Şekil 2- Von Neumann mimarisinin basit blok şeması Harvard mimarisinde ise komut ve veri için ayrı hatlar vardır. Veri yolları genelde 8-bittir. Program hafızası ise 12, 14 ve 16 bitlik veri yoluna sahip flaş ya da ROM'lardan oluşmaktadır. Veri ve komut paralel olarak çalışabilir. Von Neumann mimarisinden bu yüzden hızlıdır. Ancak iki ayrı yol olması üretimde daha fazla silikon malzeme gerektirir.



Şekil 2. Harvard Mimarisinin Basit Blok Şeması

PIC Mikrodenetleyicisinin Faydaları Nelerdir?

PIC mikro MCU ve diğer mikrodenetleyiciler tek çip içerisinde MİB (CPU), hafıza, osilatör, watch dog, ve G/Ç birimlerini barındırırlar. Bunun sonucunda yerden tasarruf, tasarım zamanının azalması ve harici birimlerle ilgili uyumluluk sorunlarının azalması sağlanır. Fakat bazı durumlarda sabit hafıza boyutu ve sınırlı G/Ç kapasitesi nedeniyle tasarımın sınırlarını daraltır. PIC mikrodenetleyici ailesi geliştiricilere en çok ihtiyaç duyacakları geniş aralıklarda G/Ç, hafıza ve özel fonksiyonları sunar.

Kod verimliliği: PIC Harvard mimarisini esas alan 8-bit bir mikrodenetleyicidir. Bunun anlamı içerisinde hafıza ve veri için farklı yollar bulunur. Program ve veri hafızasının aynı anda erişebildiğinden dolayı hızı yüksektir. Geleneksel mikrodenetleyicilerde program ve veri hafızası aynı veri yolunu kullanır. Bu diğer yapı ile karşılaştırıldığında hızı en az 2 kat düşürür.

Emniyet: Tüm komutlar 12 veya 14-bit genişliğindeki program hafızasına sığar. Bu yüzden programın veri bölgesine atlayıp veriyi program komutları gibi çalıştırması gibi bir ihtimal yoktur. Bu Harvard mimarisinde olmayan 8-bit program adres yolları kullanan mikrodenetleyicilerde oluşabilir. Komut seti 16C5X ailesi ve 14 bit genişlikteki 16CXX ailesinde program yazabilmek için öğrenmeniz gereken sadece 33 komut vardır. CALL, GO TO ve bit test amaçlı komutlar haricindeki komutlar tek komut adımıyla işletilirler.

Hız: PIC osilatör ile dahili saat yolu arasında 4'e bölücü bir devre içerir. Bu özellikle 4 MHZ kristal kullanıldığında komut zamanının hesaplanmasını kolaylaştırır. Bu durumda her bir komut adımı 1 mikro saniye tutar. PIC 20 MHZ gibi bir kristal kullanıldığında saniyede 5 milyon komut işletebilen oldukça hızlı bir işlemcidir.

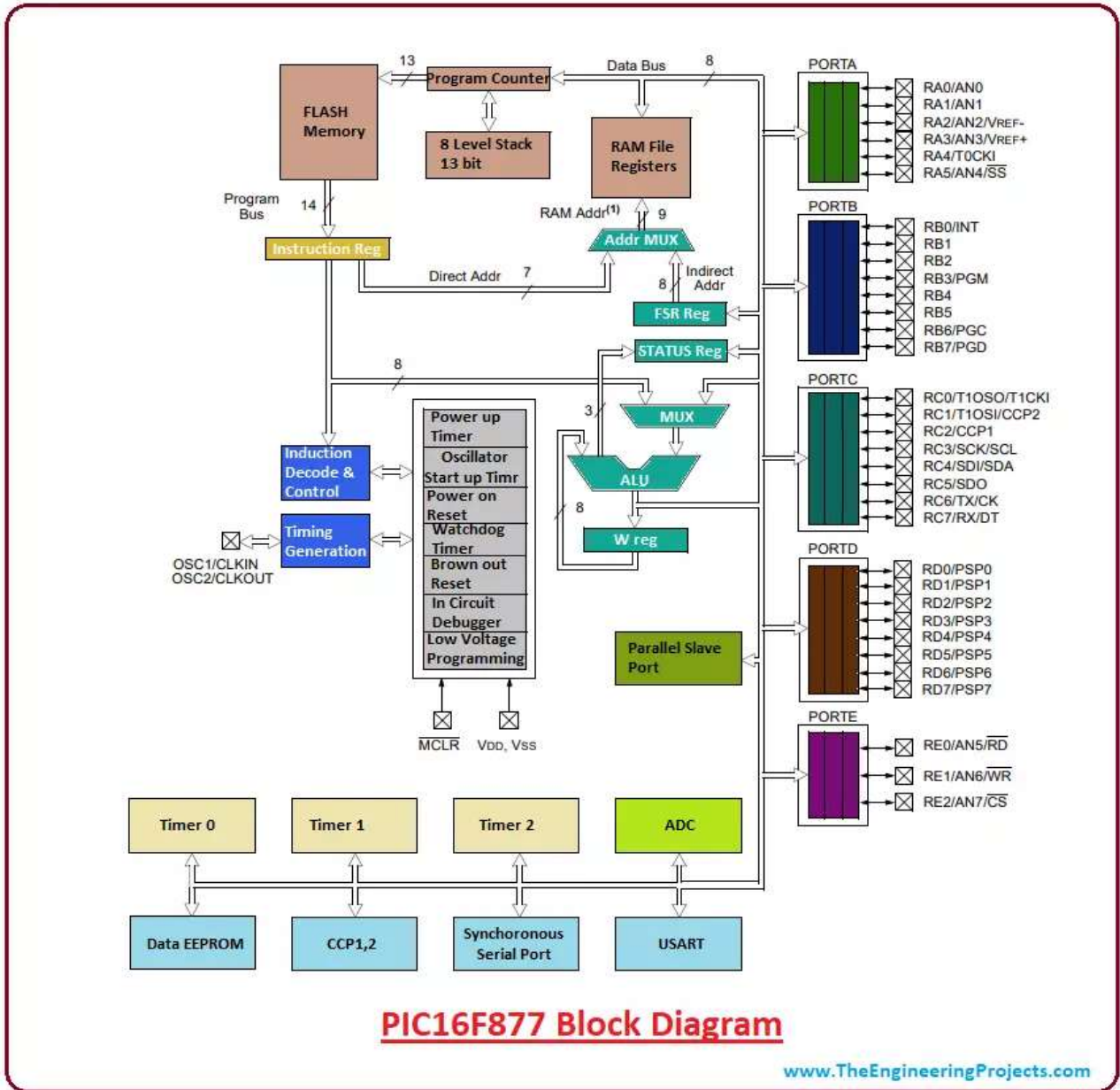
Statik İşlem: PIC tamamen statik bir mikrodenetleyicidir. Diğer bir deyişle saat frekansını durdurursanız tüm

yazmaç içerikleri olduğu gibi kalır. Pratikte tam olarak bunu yapmazsanız PIC'i uyku moduna geçirirsiniz. Bu saat frekansını durdurur PIC'i uyku modundan önce hangi durumda olduğunu bilebilmesi için bazı bayrakları ayarlar. Uyku modundayken PIC 1 mikro amperden az akım çeker.

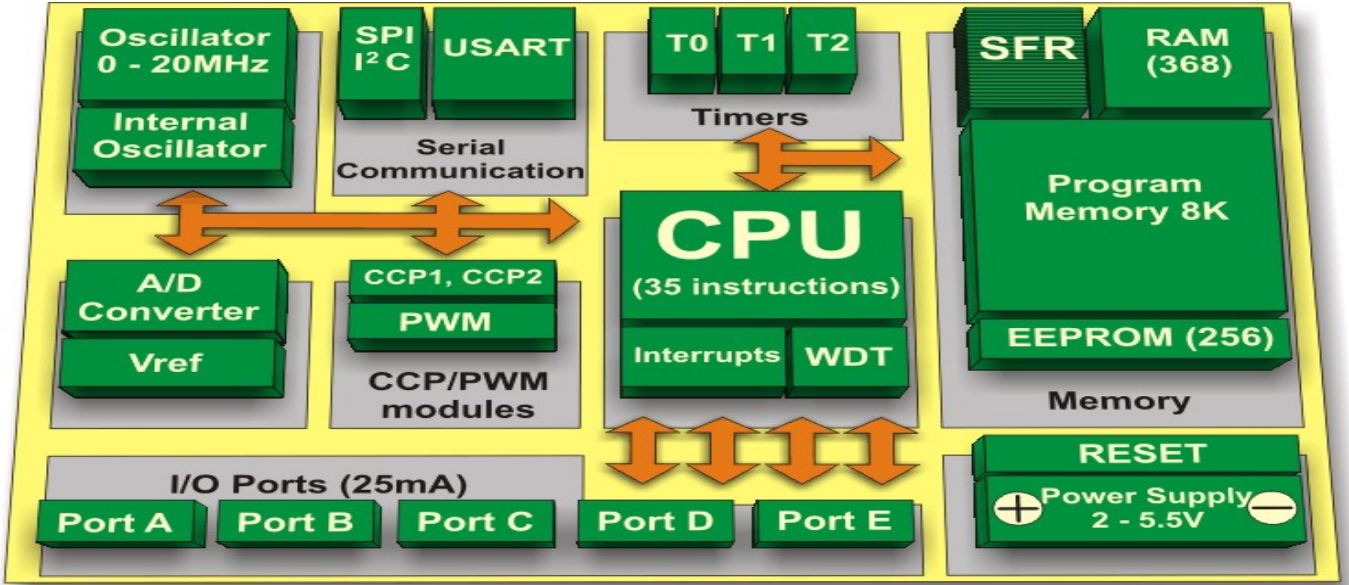
Çıkış Sürme Yeteneği: PIC yüksek çıkış sürme kapasitesine sahiptir ve LED triyak vs. direk sürülebilir. Herhangi bir g/ç pininden 25 mA'e kadar veya tüm çipten 100-200 mA akım çekilebilir. Seçenekler neredeyse tüm ihtiyaçlarınıza cevap verecek şekilde hız, sıcaklık, kılıf, G/Ç hatları ve hafıza seçenekleri mevcuttur.

Güvenlik: PIC mikro MCU endüstrideki en güvenli kod koruma sistemine sahiptir. Koruma biti bir kez programlandığında program hafızası okunamaz.

Geliştirme: PIC geliştirme için pencereleli veya flaş yapıda üretim için ise OTP yapıda bulunabilir. Geliştirme araçları ev kullanıcıları için kolayca ve uygun fiyatlı olarak temin edilmektedir. Hep birlikte tümünü elde etmeye çalıştığımız zaman elde etmiş oluruz.



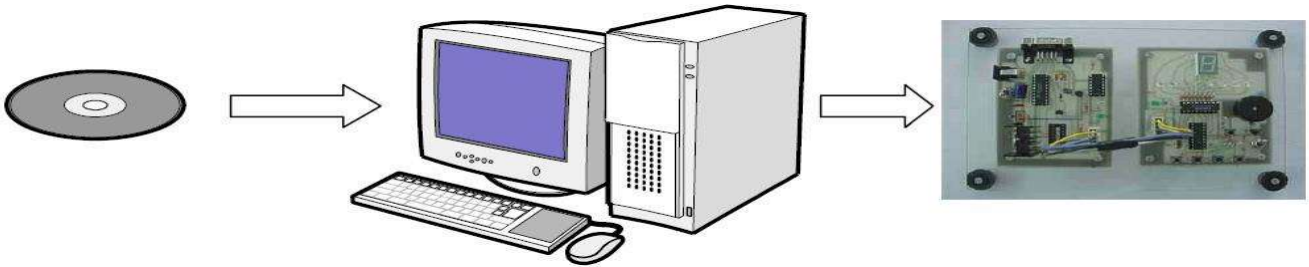
Şekil 3. PIC16F877A (14BIT) Block Diyagramı



Şekil 4. Bir Mikrodenetleyicinin İçyapısı

Mikrodenetleyici Programlama

Mikrodenetleyici programlama düşük seviyeli veya yüksek seviyeli programlama dilleri ile yapılabilir. Bunun için gerekli olan mikrodenetleyici devresi ve derleyici yazılımıdır. Mikrodenetleyici devresi programlayıcı ve uygulama kısımlarından oluşur. Bilgisayar programlama editöründe yazılan program çeşitli iletişim metotları ile mikrodenetleyiciye gönderilir. Devrenin uygulama kısmında ise program sonuçları gözlenir. Buradan da programın sonuçlarına ulaşabiliriz.



Şekil 5. Bir Mikrodenetleyicinin Programlanması

Mikrodenetleyici için yazılan basit bir ASM programının kodunu örnek alalım. Program uygulama devresindeki buton basılmasıyla PIC'in çıkış portuna bağlı bir ledin ışık vermesi işlemi yapmaktadır. Ve bunun gibi daha birden fazla ve kaliteli kullanışlı ve birbirinden farklı işlem sayabiliriz.

```

main()
{
    set_tris_b(0x00);           // port b yi çıkış olarak ata
    while(1)
    {
        if(input(PIN_A0))      // buton kapalı mı?
            output_high(PIN_B0); // eğer kapalı ise LED on
        else
            output_low(PIN_B0); // buton açık ise LED off
    }
}

```

Şekil 6. Bu Programın C Dili Eş Değeri

Derlenmiş olan C programı bellekte ASM programından daha fazla yer tutmaktadır. ASM dili mikrodenetleyici donanımının öğrenilmesinde temel teşkil eder. C dili ise kullanıcıya daha yakındır ve fonksiyon desteği sağlar. C dilinin sakıncası olan bellek kullanımı ancak etkin ve ileri düzey programlama metotları ile mümkündür. Derleyici kullanımının faydaları birbirleriyle zincir şeklinde bağlanmış metallere benzer hem sağlam hem devamlı hem de kullanışlıdır. Biz ders ve deneylerimizde C dilini kullanacağız. Yaptığımız ve yapacağımız tüm deneylerde c diline temel düzeyde hâkim olmanız beklenmektedir.

Derleyicinin Mikrodenetleyici Programlama Editörü ile Kullanımı

C dili derleyicisi kaynak kodları editör programı altında çalıştırılabilir. Bu işlemde editör programı içinde c derleyicisi tanımlanır ve bu şekilde editör programı derleyici desteği altında çalışır. Yazılan programı izlemek ve hata ayıklamak mümkündür. Hata ayıklama işlemi derleyicinin oluşturduğu makine kodları ile yapılır. Ve bizde bu kodları kullanırız.

Fonksiyon Desteği: Derleyici içindeki fonksiyonlar ile RS232 seri iletişim, A/D çevirici, I/O girişi çıkışı, bit -byte düzeyinde işlemler, LCD display gibi mikrodenetleyici uygulamaları kolaylıkla yapılabilir.

Veri Tanımlamaları: 1 bit, 8 bit, 16 bit ve 32 bit bilgi program içerisinde tanımlanabilir. Gerçek sayılar 32 bit olarak kullanılır ve bunların dışında herhangi bir şey kullanmak o verinin tanımına göreler.

ASM Komutları Kullanılabilir “C” dilinde yazılmış programın içinde ASM dili ile C dili arasındaki değişken dağılımı desteklenmektedir. Desteklendiğinden dolayı işlerimiz daha da kolaylaşmaktadır.

Standart Giriş Çıkış Fonksiyonları: RS-232 seri iletişim metodu ile bilgisayar bağlantısı yapılarak standart giriş-çıkış fonksiyonları kullanılabilir. Bu modelle standart olarak giriş çıkışlarımızı yapabiliriz.

Donanım Desteği: Derleyici içindeki başlık dosyaları ile mikrodenetleyici uygulamalarındaki çeşitli entegrelerin kullanım desteği sağlanmıştır. Bu sayede gerekli donanımı sağlamış oluruz.

Değişken Alanının Etkin Kullanımı: Komutlar program belleğinde değişkenler ise değişken alanında saklanır. Komutlar ve değişkenlerin geçici paylaşımını sağlamak için en az değişken alanı ayrılır.

Kolay ve okunabilir kod yapısı dili, Assembly’e göre çok daha anlaşılır bir sözdizimine sahiptir. Bu sayede kod yazmak, hata ayıklamak ve projeyi geliştirmek daha kolay hale gelir.

Zaman tasarrufu sağlar: Yüksek seviyeli bir dil olduğu için tek satırda çok işlem yapılabilir. Karmaşık donanım işlemleri kısa ve net ifadelerle gerçekleştirilebilir.

Taşınabilirlik (portability):C diliyle yazılan kodlar farklı mikrodenetleyicilere küçük değişikliklerle uyarlanabilir. Böylece projeler sadece PIC16F877A'ya değil, diğer PIC serilerine de kolayca aktarılabilir.

Hazır kütüphaneler sayesinde kolay donanım kontrolü: C dili için geliştirilen kütüphaneler sayesinde ADC, PWM, UART gibi modüller kolayca kullanılabilir. Geliştirici, düşük seviyeli donanım ayrıntılarıyla uğraşmadan hızlıca işlev ekleyebilir.

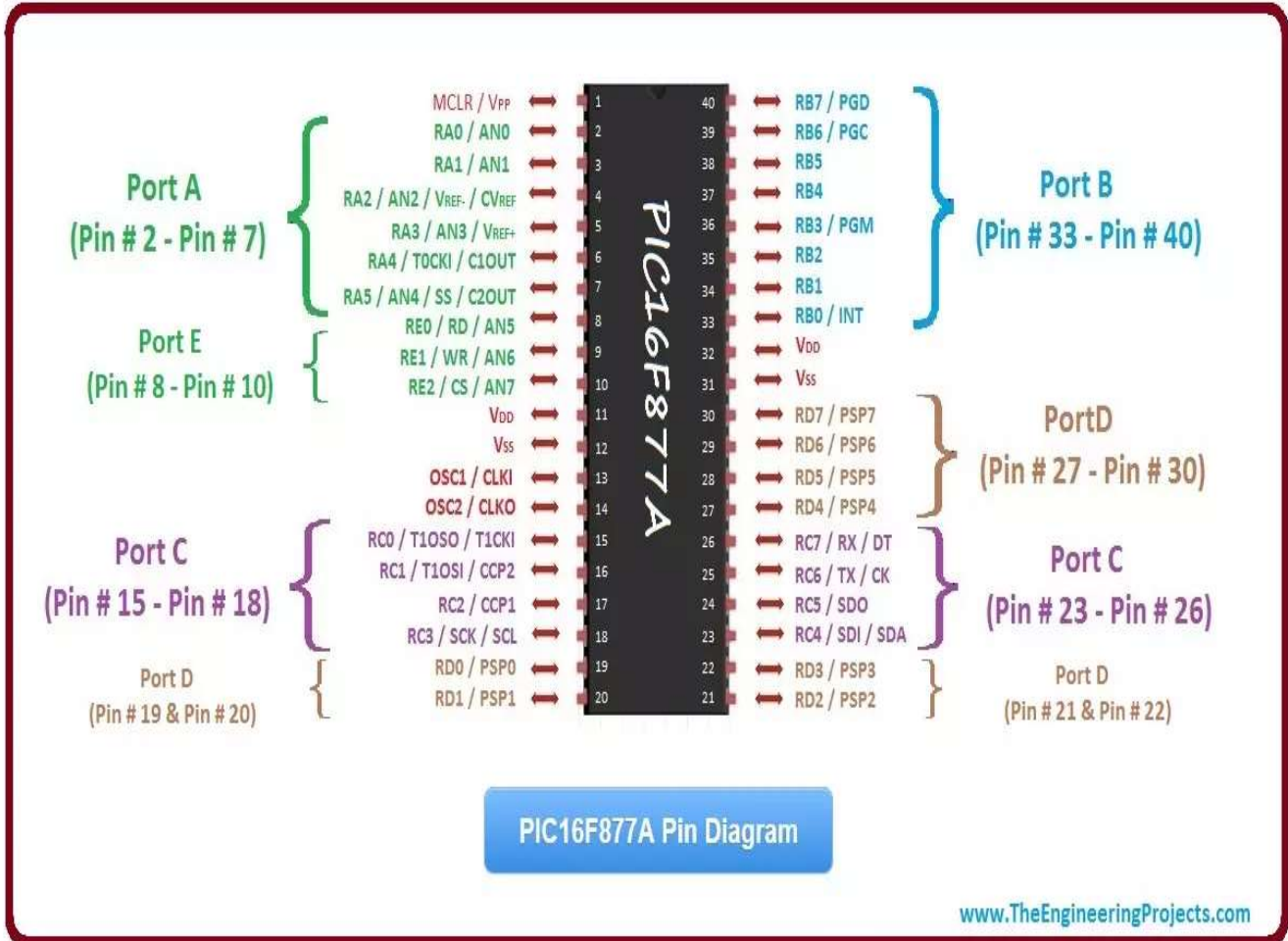
Bakım ve geliştirme kolaylığı: Kodun anlaşılır olması, ileride değişiklik veya hata düzeltmesi yapmayı kolaylaştırır. Ekip çalışmalarında kod paylaşımı ve sürüm güncellemeleri rahat olur.

Geniş topluluk ve destek: "C" dili, mikrodenetleyici programlamada en yaygın kullanılan dillerden biridir. İnternette çok sayıda örnek, forum ve kaynak bulunduğu için öğrenme süreci hızlanır.

Modüler programlama imkânı: C dili fonksiyonel yapısıyla projeleri parçalara bölmeye izin verir. Bu da büyük projelerde düzeni ve kod yönetimini kolaylaştırır.

DENEYLERİN YAPILIŞI

PIC 16F877A.h Elemanının Tanıtılması



Şekil 7. PIC16F877A.h Mikrodenetleyicisinin Bacak Bağlantıları

PIC 16F877'nin genel özelliklerine bakacak olursak çoğu uygulama için yeterli bir donanıma sahip olduğunu görürüz. Şekil 7 dikkatlice incelendiğinde denetleyicinin donanım mimarisini anlamak daha da kolaylaşacaktır. 40 pin PDIP kılıfı dışında PIC16F877 ve PIC16F877A denetleyicileri, 44 pin QFP, 44 pin PLCC, 44 pin MQFP, 44 pin TQFP, QFN kılıflarında üretilmektedir. QFP ve QFN kılıfları birbirine benzer olmakla beraber aralarında tek bir fark vardır. QFP kılıflarında pin uçları dışardayken, QFN kılıflarında pin uçları dışarıya çıkarılmamıştır. Uçlar kılıfın dışında metal yüzey şeklindedir.

Bellek Organizasyonu

Bu kısımda PIC16F877.h'nin bellek organizasyonu hakkında genel bilgiler verilecektir. CCS-C ile PIC programlarken bellek işlemlerini derleyici kendisi yaptığından dolayı bu kısımda ayrıntılı bilgi verilmekten kaçınılıp dersimiz ve ihtiyacımız olan bilgiler paralelinde bilgilerimizi vermeye devam edeceğiz. Ayrıntılı bilgi için ise denetleyicinin datasheet'ine bakabilirsiniz. PIC16F877 belleği üç kısımdan oluşmaktadır. Bunlar:

- 1.) Program Belleği yani Flaş Bellek,
- 2.) Data Belleği (veri belleği),
- 3.) EEPROM Data Belleği 'dir.

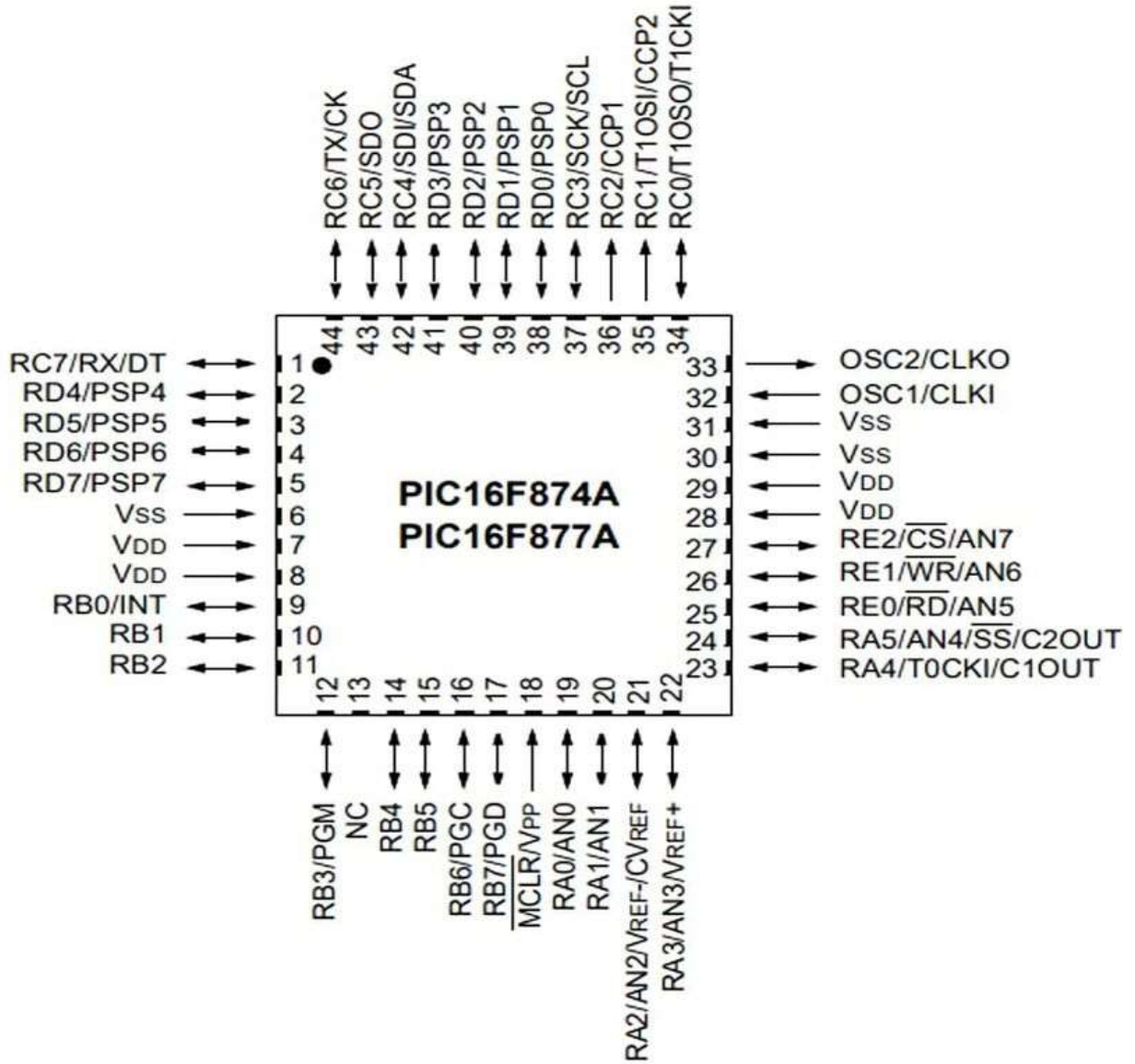
Program Belleği (Flaş Bellek)

Programın belleği her biri 2k'lık 4 sayfadan oluşur. Her sayfada sırasıyla 0h – 7FFh, 800h – FFFh, 1000h – 17FFh, 1800h – 1FFFh adreslerini içerir. Bu adreslere erişim ise program sayacı ile sağlanır. Program sayacı 13 bitlidir ve 8kx14 değerinde adresleme kapasitesine sahiptir. Burada 14 sayısı PIC denetleyicilerindeki komut uzunluğudur. Program belleği haritası incelendiğinde konu daha iyi anlaşılacaktır. Şekilden görüldüğü gibi PIC denetleyicilerde Reset vektörünün adresi ise 0004h'tir. Bu şekilde gerekli temel bilgileri edinmiş olacaksınız.

Data Belleği (Veri Belleği)

Veri belleği 4 adet banktan oluşan genel amaçlı kaydedicilerden ve özel amaçlı kaydedicilerden oluşmaktadır. Genel amaçlı kaydediciler, programcı tarafından program geliştirilirken kullanılan yazılım amaçlı kaydedicilerdir. Örneğin bir program yazarken tanımlanan bir değişken programcının tarafından belirlenen genel amaçlı veri belleğinde istenen bir adrese veya derleyici program tarafından otomatik olarak belirlenen bir adrese kaydedilir.

Bu kayıtlar geçicidir. Enerji kesildiğinde veya programda yapılan değişikliklerle adreslerdeki bilgiler silinir veya değişir. Özel amaçlı kaydediciler ise denetleyici donanımını kontrol etmek için kullanılan kaydedicilerdir. Bu kaydedicileri kullanırız. Örneğin CCPR1L özel amaçlı kaydedicisine CCP1 birimini kontrol etmek için gerekli 16 bitlik kontrol bilgisinin düşük seviyeli sadece 8 biti yazılır. Bunun gibi özel amaçlı kaydediciler ile denetleyici donanım kontrol edilmektedir. Şekil 8'de ram veri belleği adresleri belirgin şekilde verilmiştir, inceleyiniz.



Şekil 8. PIC16F877A RAM Veri Belleği Hafızası

Şekilden de görüleceği üzere 4 adet banktan oluşan RAM belleğinin her bir bankında bulunan adresler dosya adresi olarak isimlendirilir. Veri belleğindeki her bank 128 Byte kapasitesindedir. Yani her bankta 8 bitlik 128 adet kaydedici bulunmaktadır. Dolayısıyla toplamda $4 \times 128 = 512$ adet kaydedici mevcuttur. Şekil incelendiğinde özel amaçlı kaydedicilerden bazılarının birden fazla bankta bulunduğu görülmektedir. Bu şu anlama gelmektedir: Birden fazla bankta yer alan bir kaydedicinin değeri herhangi bir bankta değiştirildiğinde bu değişiklik diğer banklarda da etkili olmaktadır. Bir bakıma bir çeşit aynalama işlemi oluşmaktadır. Bu aynalama mantığının nedeni kaydedicilere hızlı erişmek ve programlamada daha az komut kullanmaktır. Çünkü 4 bankın hepsinde olan bir kaydediciye o anda bulunan banktan hemen erişilebilir. Diğer türlü örneğin i. Banktayken 3.bankta bulunan bir kaydediciye erişmek için 1.banktan 3.bankta geçmek gerekmektedir. Bu bize hem zaman hem daha komut kaybı anlamına gelmektedir ki bu değerler bizler için çok önemli ve değerlidir. Bankların son 16 byte'ları da birbiri ile

aynalamıştır. Bir bankın son 16 byte'ının herhangi birine yazılmış bir veri diğer bankların son 16 byte'lık kısmında aynı yerde görünür. Bu nedenle bu kısma gölgelenmiş belle adı da verilir. Veri belleği haritasında bazı adresler gri olarak boyanmıştır. Bunlar kullanılmayacak olan kaydedicilerdir bu adresler yazılım kısmında kullanılmamalıdır. Bu adresteki kaydedicilerin değeri her zaman "0" değerindedir. Bu değer her zaman koşul ve şartta bu değeri almaktadır.

Besleme Uçları ve Bağlantıları

PIC16F877 için besleme uçları; Vss, 12 veya 31 no'lu pinlerden birine, VDD 11 veya 32 no'lu pinlerden birine uygulanır. Besleme gerilimleri değişik durumlara göre 2.0V LE 5.5v arasında olabilir. Genelde laboratuvar çalışmalarında 5V idealdir. Devreye enerji verildiği anda meydana gelebilecek gerilim dalgalanmaları nedeniyle istenmeyen arızaları önlemek amacıyla VDD ile VSS arasına 0.1 F'lık bir dekaplaj kondansatörü bağlamak gerekir.

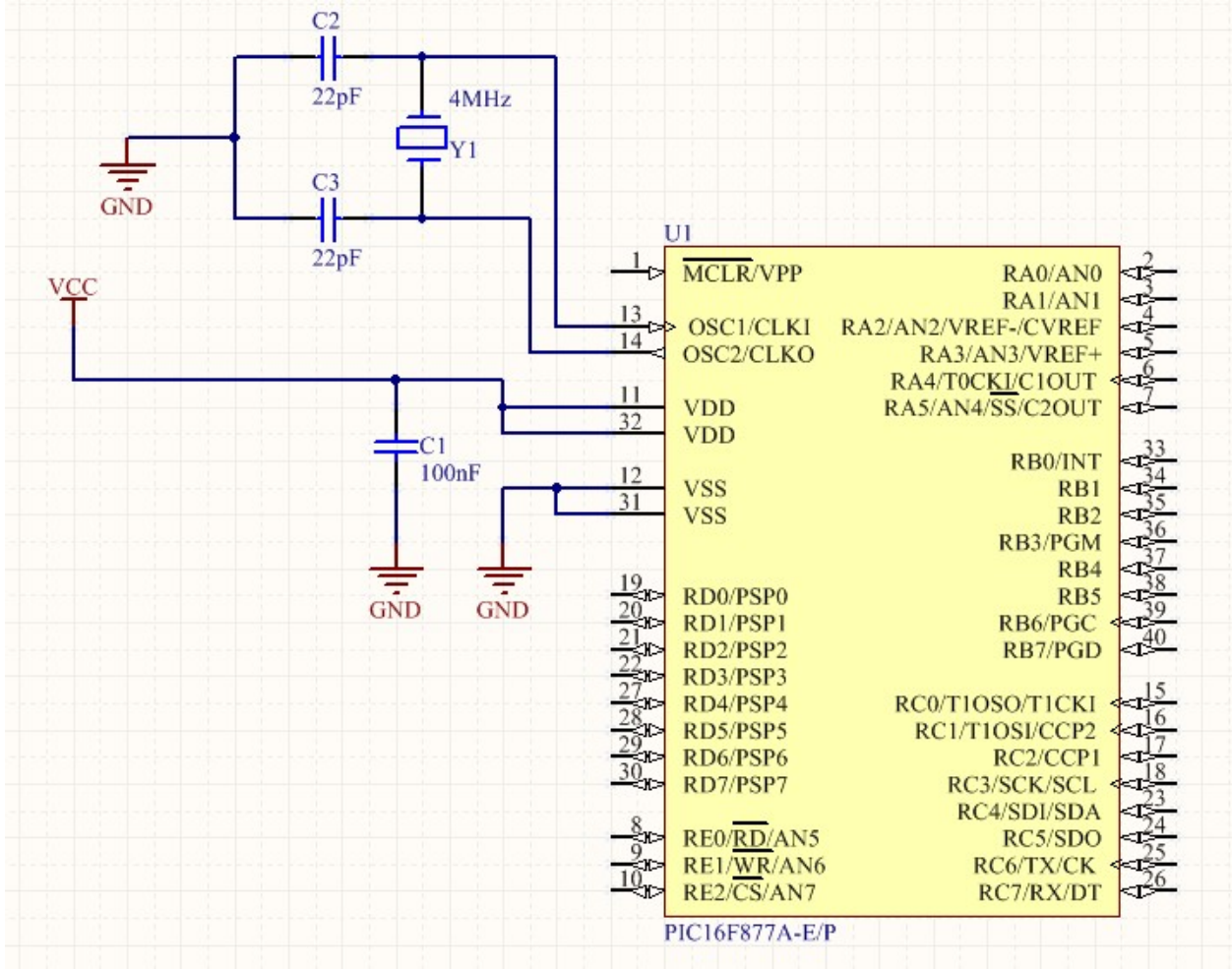
Osilatör Konfigürasyonları

PIC hafızasındaki komutları işlemek için bir sinyale ihtiyaç duyar. Bu sinyale saat sinyali denir. PIC bu clock sinyalini osilatör uçlarına bağlanan osilatör devresinden alır. PIC16F877'nin osilatör uçları 13 ve 14'tür. PIC girişinde bağlanan osilatör frekansını 4'e bölerek kullanır. Yani girişinde 20 MHZ'lik bir osilatör bağlanmışsa PIC bunu 4'e böler ve sonuçta 5 MHZ'lik bir uygulama frekansı ile çalışır. Bu da 0.2 sn'de bir komut işleneceği anlamına gelir. PIC denetleyicilerine bağlanabilen osilatör türleri şunlardır:

- LP – Düşük Güçlü Kristal Osilatör
- XT – Kristal / Resonatör Osilatör
- HS – Yüksek Hızlı Kristal
- RC – Direnç Kondansatör Osilatör

Denetleyiciye bağlanan değişik osilatör türlerinin hepsinde istenilen istenen osilatör frekansı, besleme geriliminin normal değere gelmesi anında elde edilir. Besleme gerilimi de hemen istenen derece ulaşamaz. Sonuçta devreye enerji verildiğinde osilatör frekansının istenen değerine ulaşmasını etkileyen faktörler şunlardır:

Kristal Frekansı, Kullanılan kapasitör değerleri, sistem sıcaklığı, seri direnç değeri, Kristal kalitesidir. Osilatör devresi baskı devresi çizim şekli tasarlanacak devreye göre istenen osilatör tipi seçilebilir.

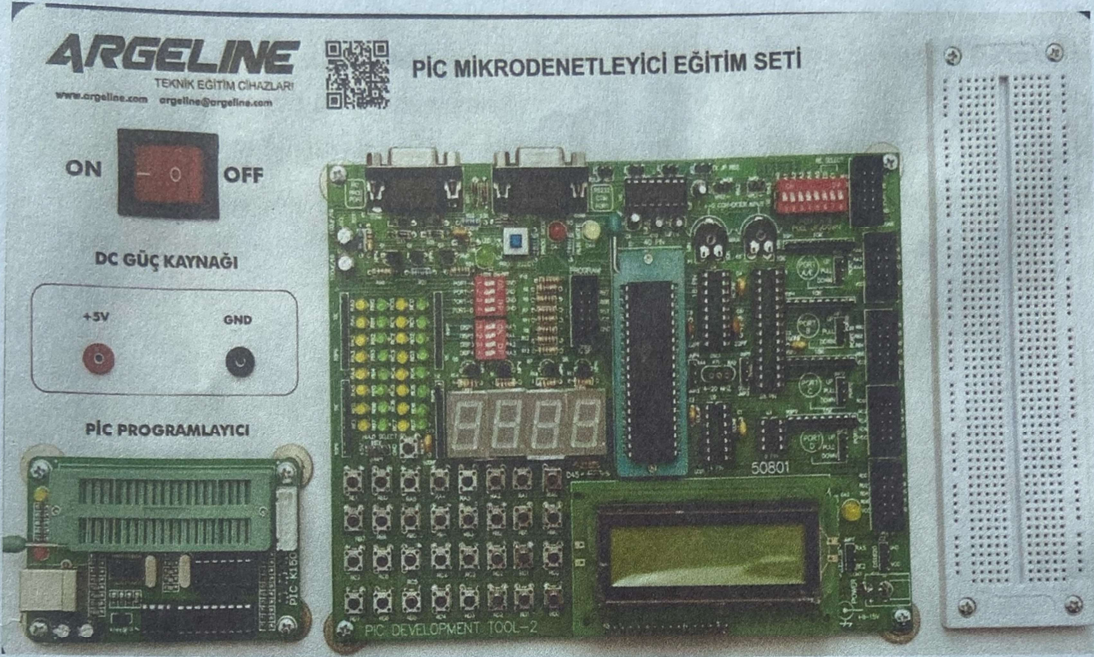


Şekil 9. HS, LP, XT Osilatörleri İçin Devre Bağlantı Şekli

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

Şekil 10. Kristal Frekansına göre Bağlanması Gereken Kondansatör Değerleri

PIC MİKRODENETLEYİCİ EĞİTİM SETİ DENEY FÖYÜ



Pic Mikrodenetleyici Eğitim Seti 4 ana bölümden oluşmaktadır. Bu bölümler;

- 1-DC Güç Kaynağı
- 2- Pic Programlayıcı
- 3- Pic Deney Kartı
- 4- Breadboard

1- DC GÜÇ KAYNAĞI

Eğitim seti üzerinde DC 5V besleme sağlayan 1 adet güç kaynağı bulunmaktadır. Bu besleme voltajı Breadboard üzerinde kurulacak devrelerde kullanım için ayrılmıştır. Pic deney kartı beslemesini kartın altından doğrudan almaktadır. Harici besleme yapmadan çalışır. Deney kartı ile güç kaynağının GND uçları ortaktır. Pic programlayıcı kartı ise enerjisini programlama yapılacak bilgisayar USB portundan almaktadır. Diğer kısımlarla bir bağlantısı yoktur.

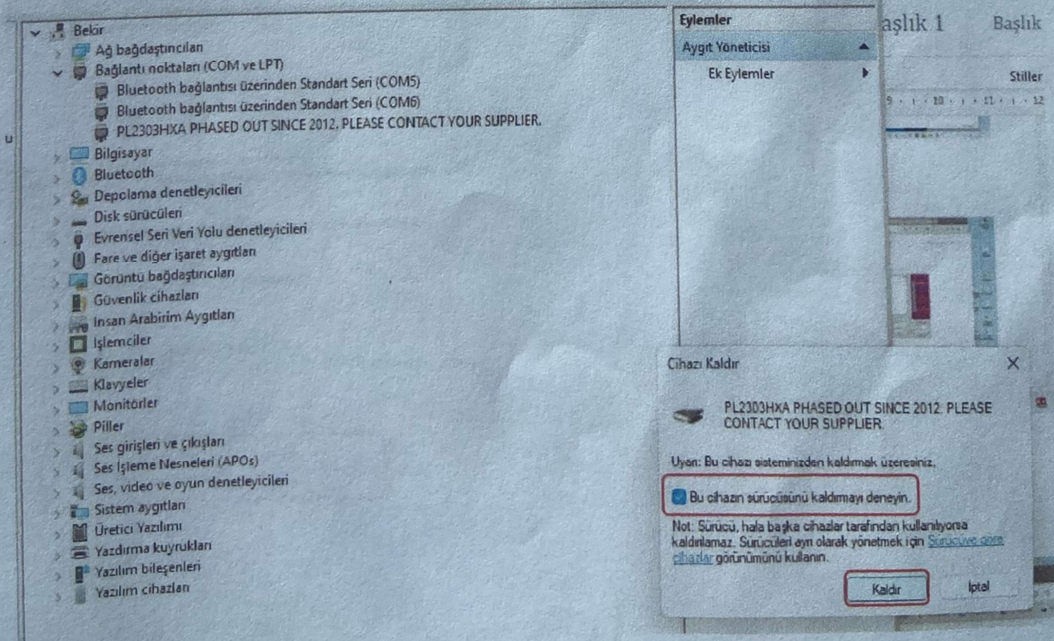
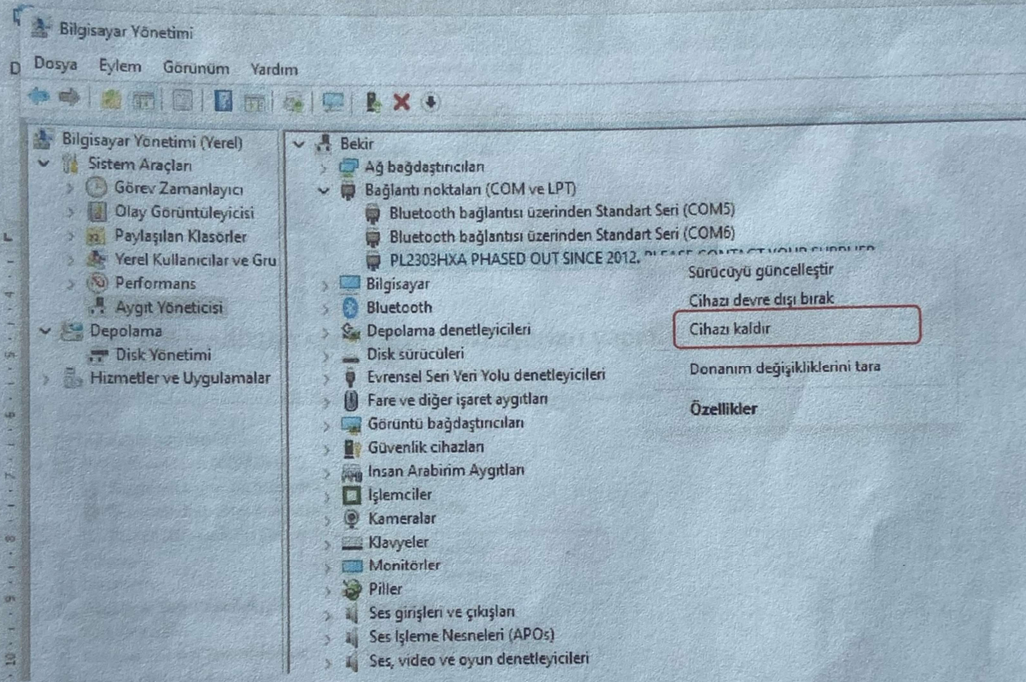
2- PİC PROGRAMLAYICI

a) Driver Kurulumu

Windows 10 ve sonrası için driver aşağıdaki anlatıldığı şekilde kurulmalıdır. Daha eski işletim sistemlerinde kurulum gerekmez.

Programlayıcı bilgisayara ilk bağlandığında aşağıdaki gibi hata verecektir.

- Görsellerdeki yolu takip ederek **Bilgisayar Yönetimi/ Aygıt Yöneticisi** bölümünden eski driver'ı kaldırın.



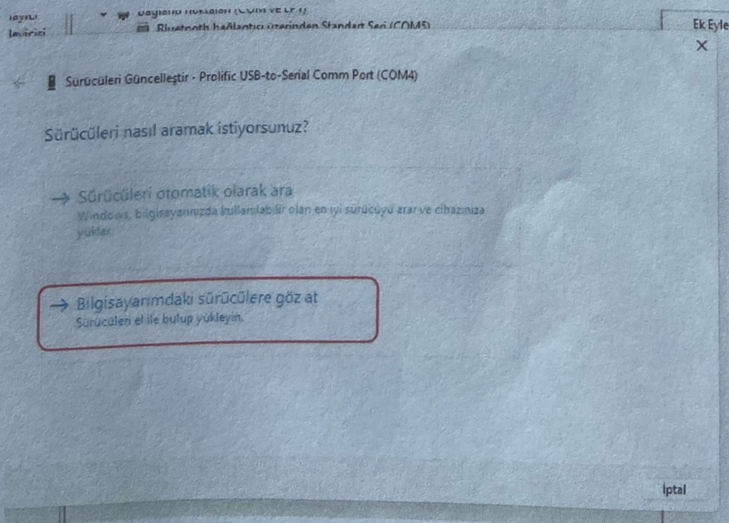
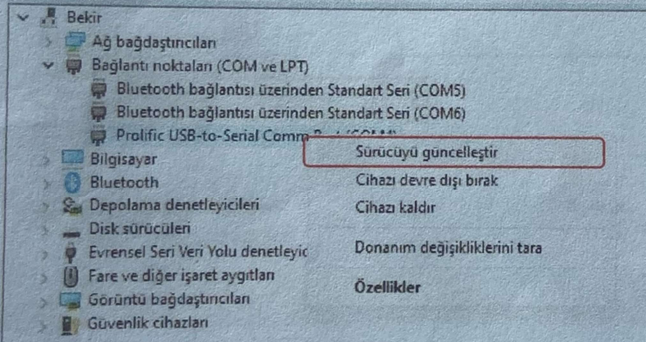
- Driver'i programını yönetici olarak kurun.

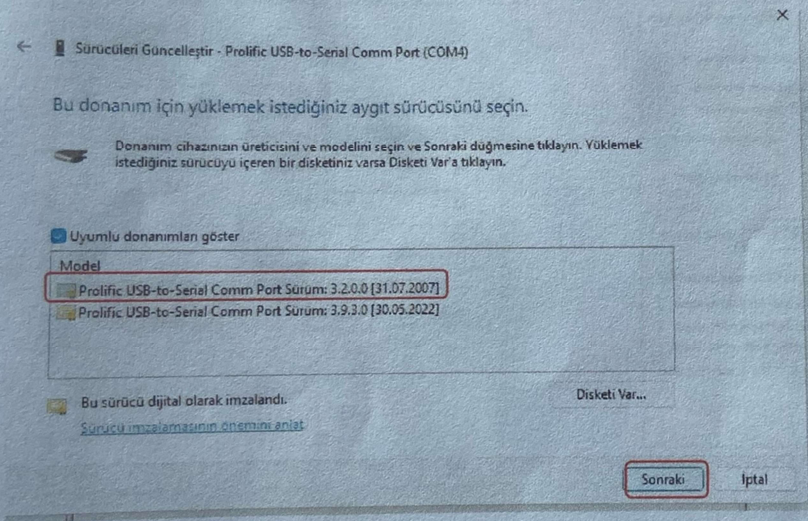
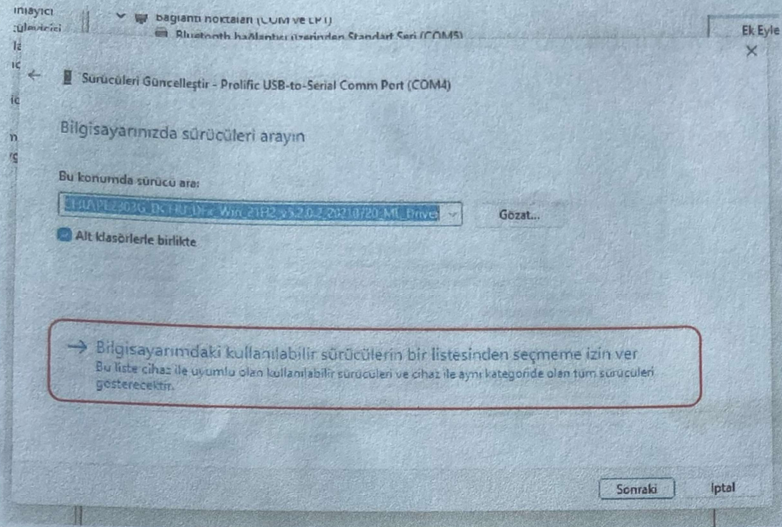
PL23XX_CheckChipVersion_ReadMe.txt		23.12.2023 00:46	Metin Belgesi	3 KB	
PL23XX_checkChipVersion_v1020.exe		23.12.2023 00:46	Uygulama	208 KB	
PL23XX_DriverInstallerv_4.0.8_ReleaseNote.txt		23.12.2023 00:46	Metin Belgesi	10 KB	
PL23XX-M_LogoDriver_Setup_406_2'		00:46	Uygulama	13,132 KB	
PL2303 Windows Driver Manual v1.2		00:46	Microsoft Edge P...	1,815 KB	
PL2303D_Win7_8_10_HWCertReport	Aç	Enter	00:46	Chrome HTML Do...	106 KB
PL2303D_Win11_HWCertReport_115	Paylaş		00:46	Chrome HTML Do...	105 KB
PL2303G_Win7_8_10_HWCertReport	Yönetici olarak çalıştır		00:46	Chrome HTML Do...	106 KB
PL2303G_Win11_HWCertReport_115	Başlangıç'a sabitle		00:46	Chrome HTML Do...	105 KB

Açık menü seçenekleri:

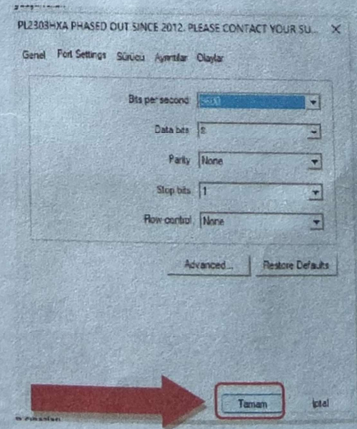
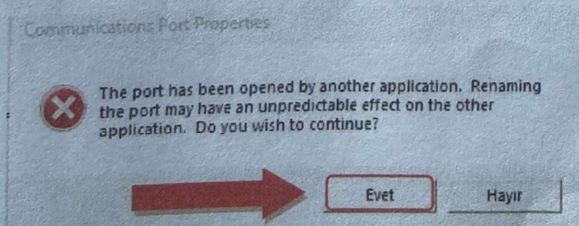
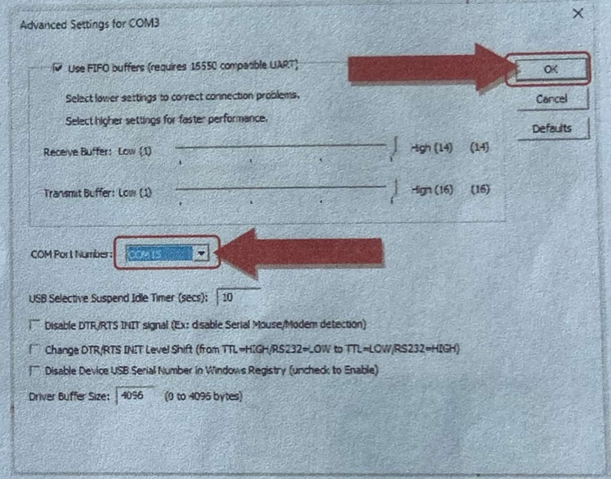
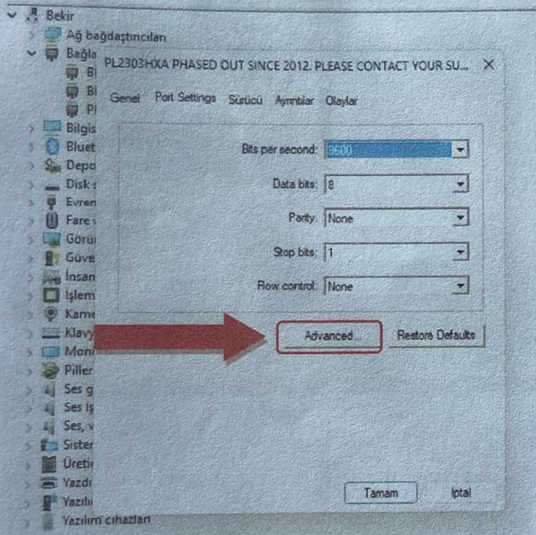
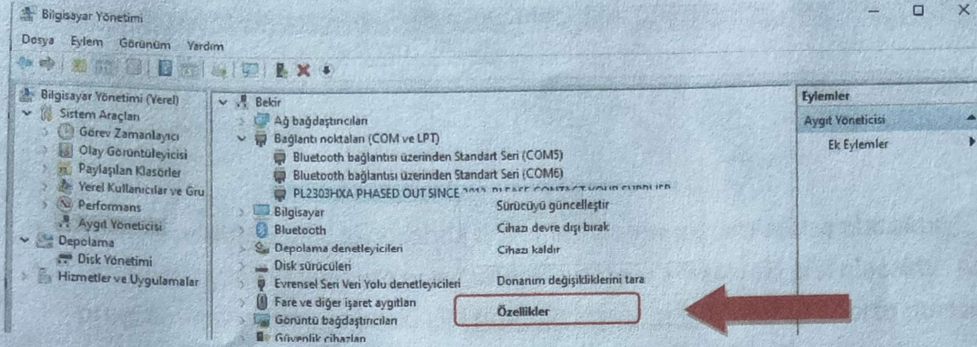
- Sık Kullanılanlara ekle
- ZIP dosyasına sıkıştır
- Yol olarak kopyala (Ctrl+Shift+P)
- Özellikler (Alt+Enter)
- Daha fazla seçenek göster

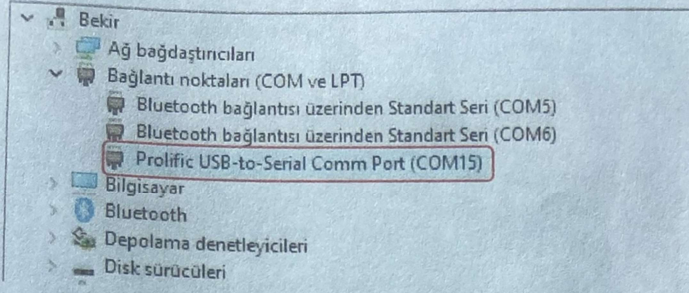
- Driver yüklendikten sonra aşağıdaki ayarları yapın.



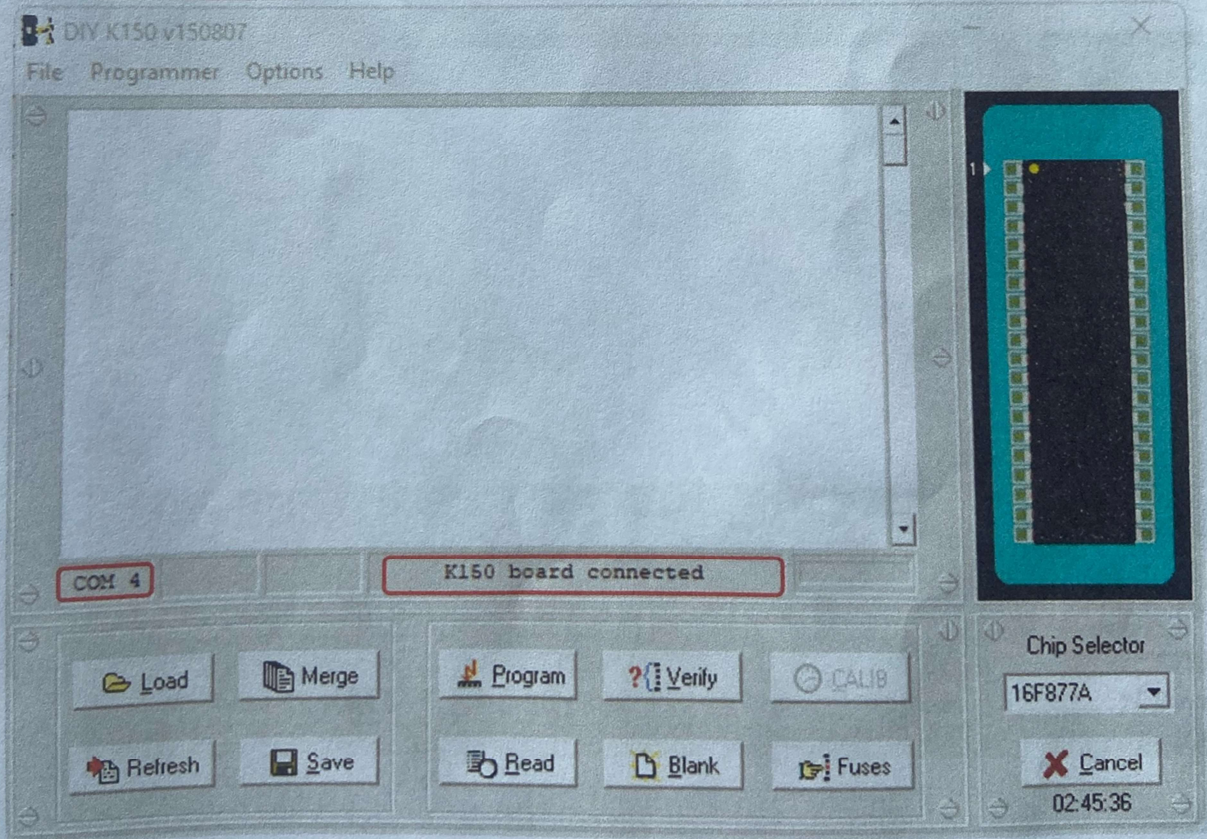
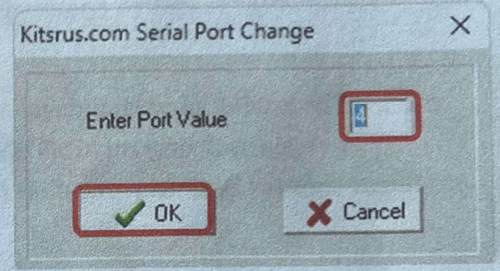
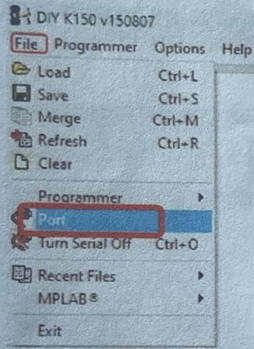


- Driver kurulduktan sonra aygıt yöneticisi üzerinden COM portun numarasını görsellerde anlatıldığı gibi değiştirin.

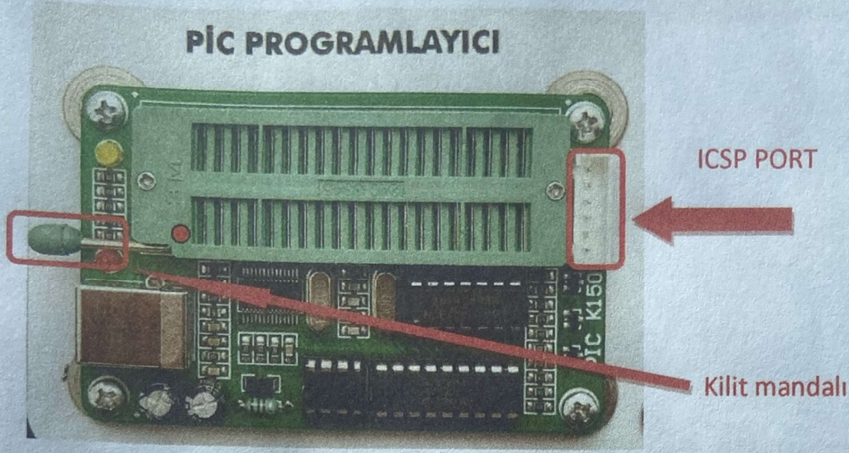




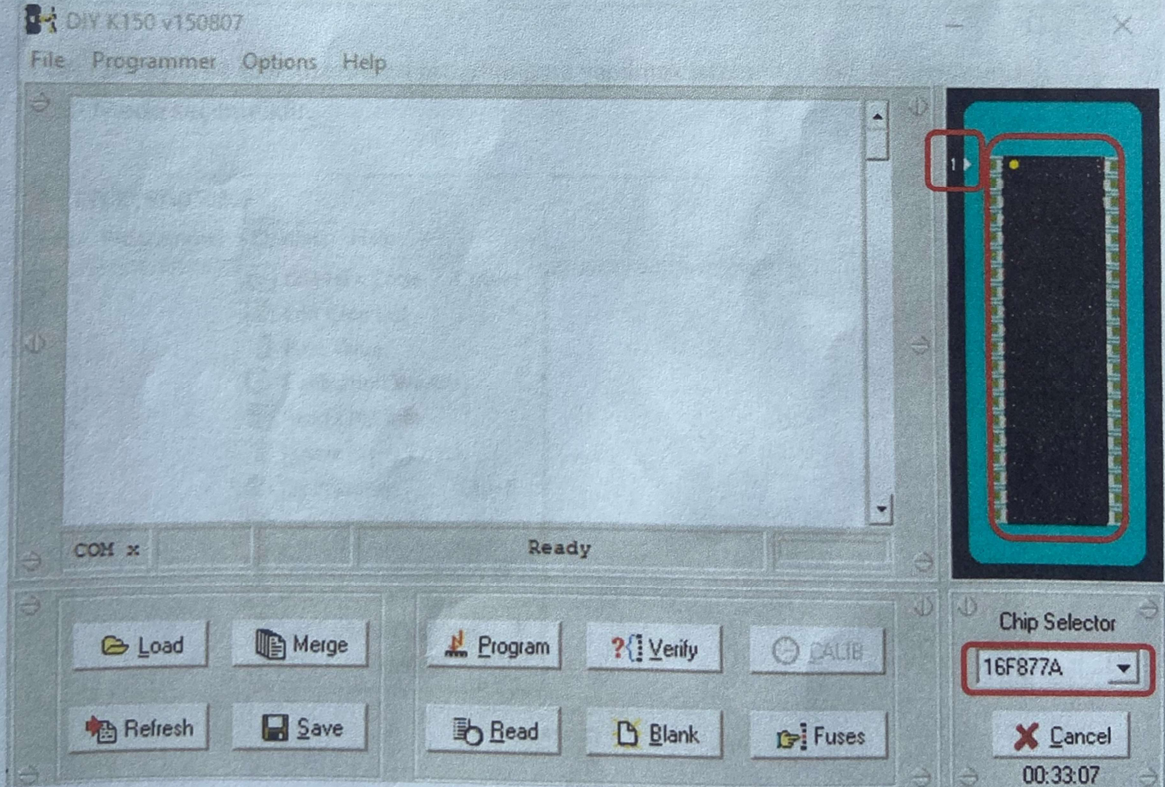
- Son görseldeki sonuç alındığında driver kurulumu tamamlanmış olacaktır.
- Driver düzgün yüklendiyse programdaki pencere görseldeki gibi olacaktır. Microbrn programını açın ve aşağıdaki gibi programlayıcının bağlı olduğu portu numarasını girin.

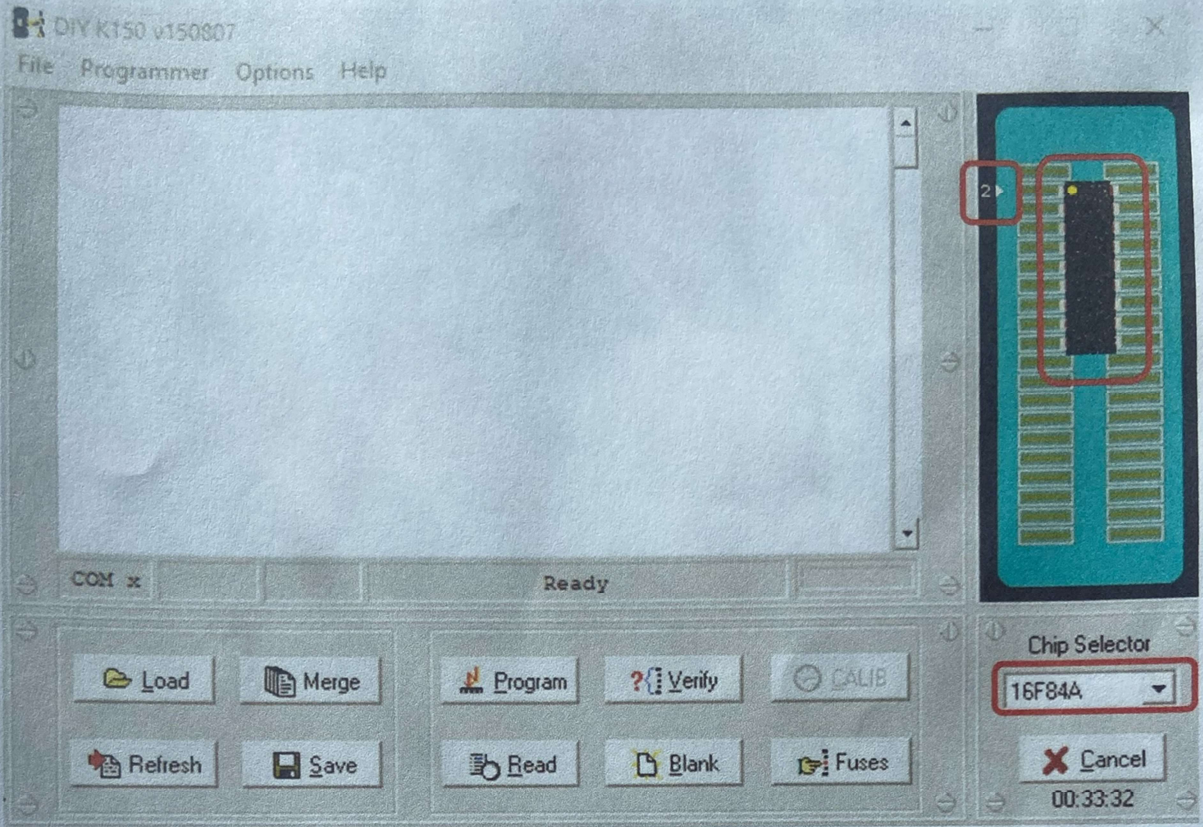


b) Pic'in Programlanması

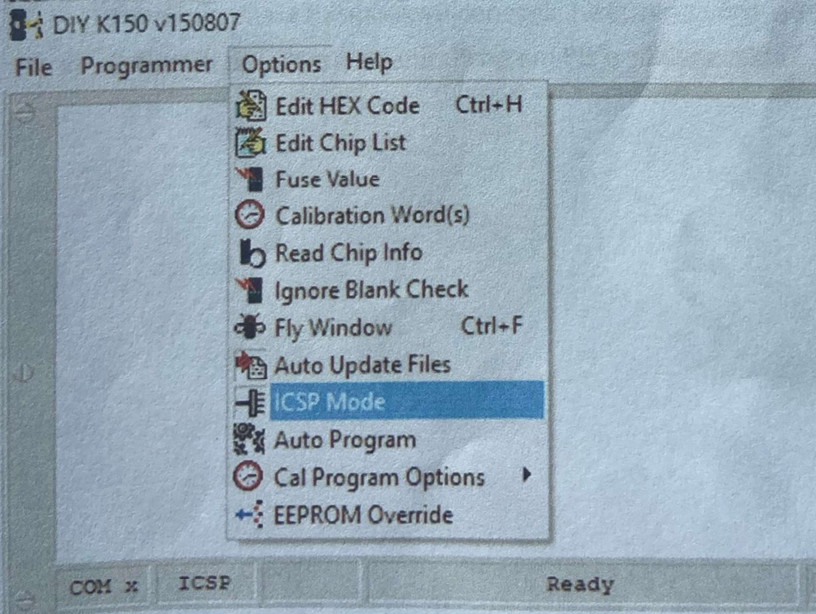


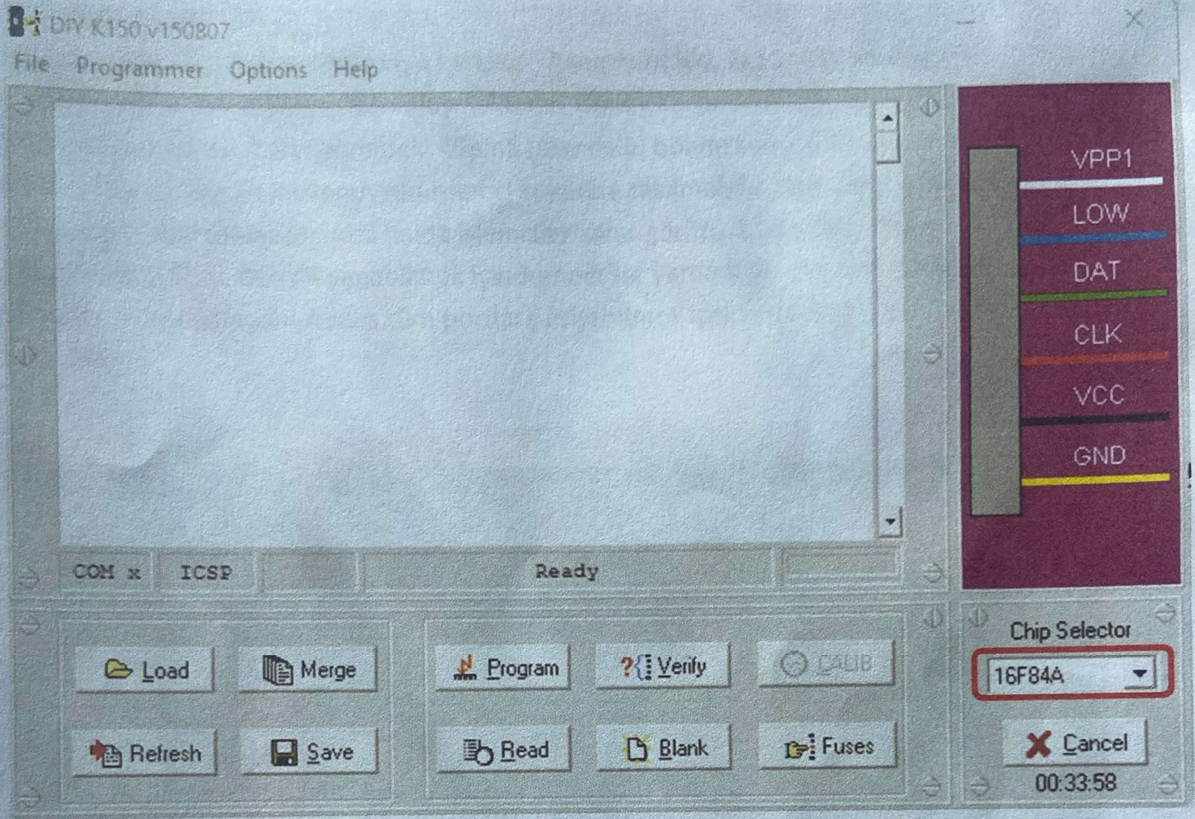
Pic programlayıcı kart birçok farklı tipte PIC'i programlayabilmektedir. Microbrn programı üzerinden programlanmak istenen pic tipi seçildiğinde Pic'in programlayıcıya nasıl yerleştirileceği program üzerinde gösterilmektedir. 1 nolu pin yukarıdaki resimde kırmızı ile işaretlenmiştir.





Programlayıcı ile ICSP üzerinden programlama yapılmak istenirse **Options** sekmesinden **ICSP Mode** seçilmelidir.



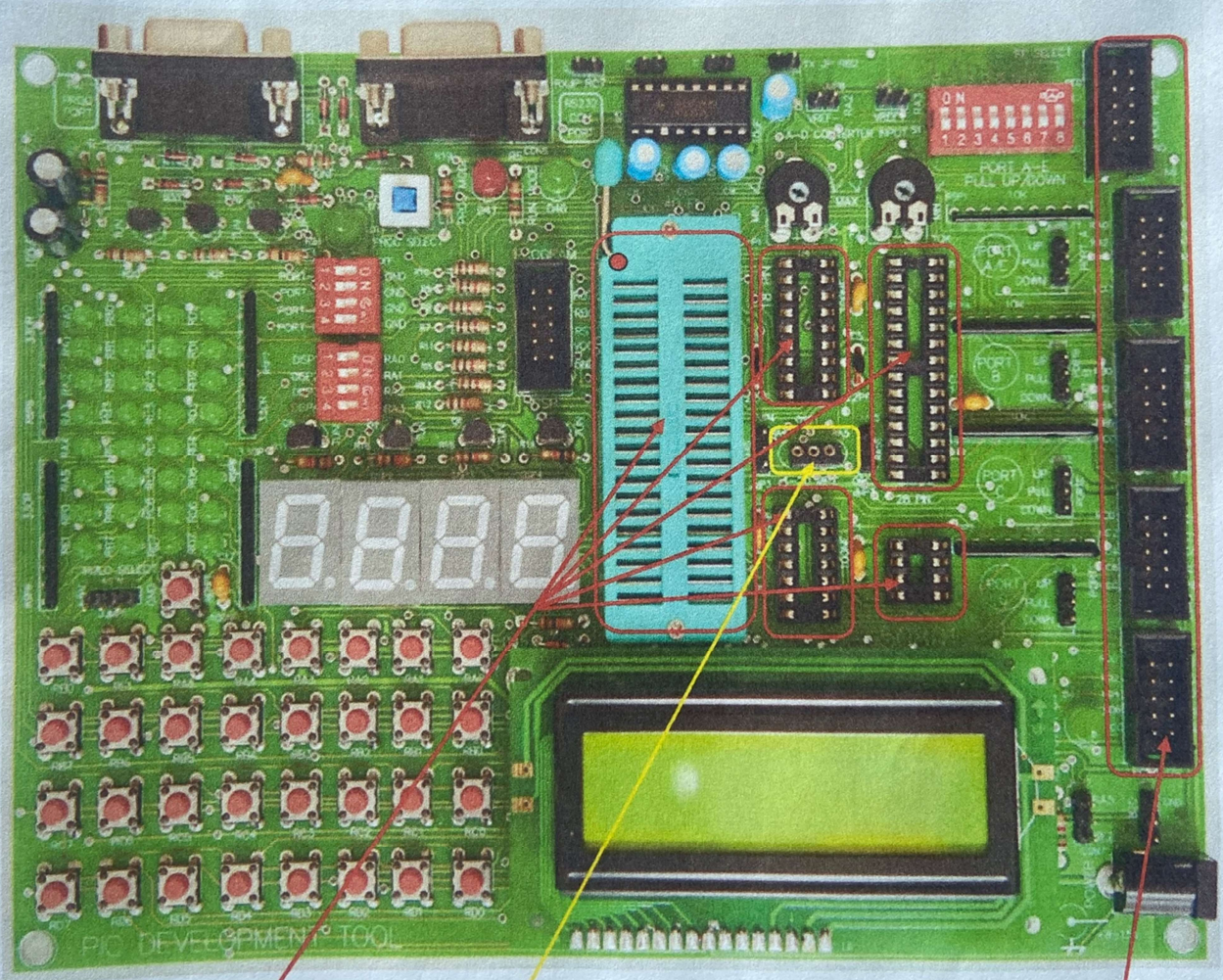


Programın işlemciye yüklenmesi

- Pic tipi seçildikten sonra mandal açılarak işlemci yuvasına yerleştirilir ve mandal kapatılır.
- Load butonuna basılarak yüklenecek .hex uzantılı dosya seçilir.
- Program butonuna tıklanır. Program Pic'e yüklenecektir.

3- PİC DENEY KARTI

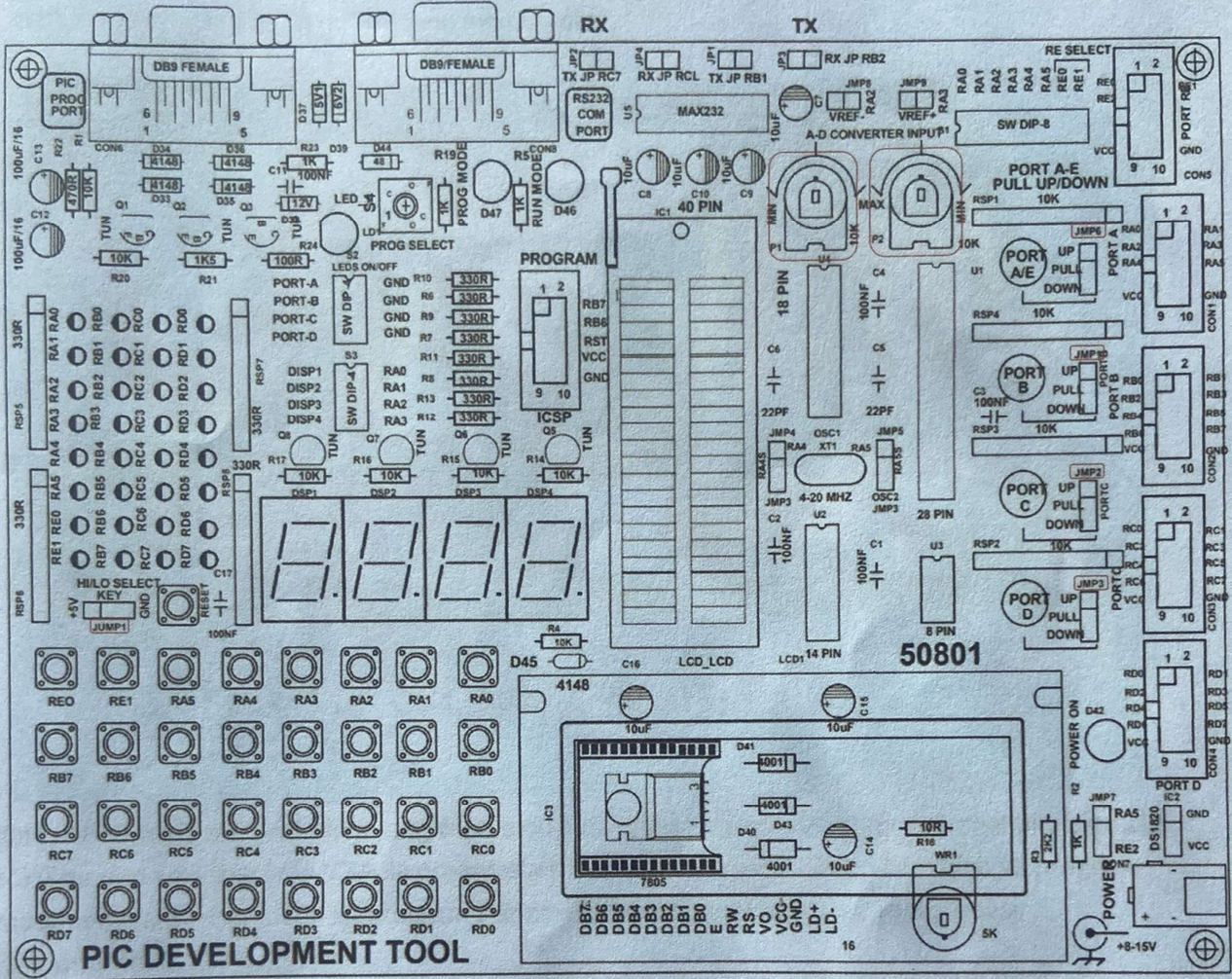
Pic deney kartı 8x4 led, 8x4 buton, 4 karakter 7 segment led, 2x16 LCD, analog porta bağlı 2 adet trimpot ve serial output'dan oluşmaktadır. Deney kartında 40,28,18,14,8pin işlemciler kullanılabilir. Kart üzerindeki 40pinli (mandallı) bölüm sadece 40 pin işlemciler ile kullanılabilir. Diğer işlemciler uygun işlemci soketine takılmalıdır. **Kart üzerinde aynı anda tek işlemci takılı olmalıdır.** Aksi halde işlemciler zarar görür. İşlemcinin tüm portlarını PULL UP yada PULL DOWN yapabilmek için Jumper'lar yardımı ile devreye alınabilen grup dirençler bulunmaktadır. Ayrıca tüm portlara erişebilmek için kartın sağ tarafında soketler mevcuttur.



İşlemci soketleri

Kristal yuvası

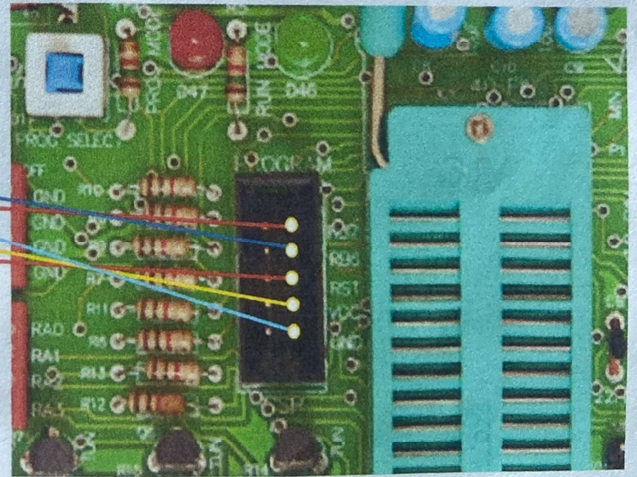
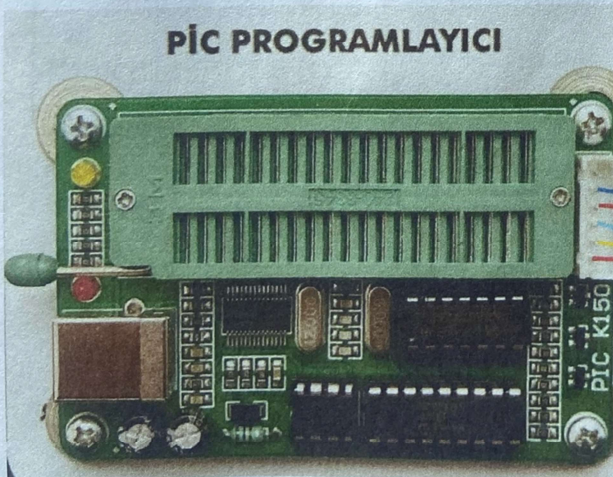
İşlemci Portları



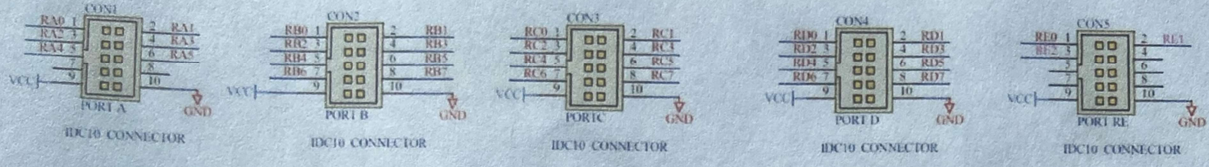
run mode program mode prog modu seçimi



Kart üzerinde deney yapılırken **prog mod** tuşu basılı olmalı ve **run mode** ledi yanmalıdır. İşlemciyi sökülmeden kart üzerinde ICSP ile programlama yapılacaksa buton yukarıda olmalı ve **program mode** ledi yanmalıdır. ICSP ile program yüklemek için aşağıdaki bağlantının yapılması gerekmektedir.



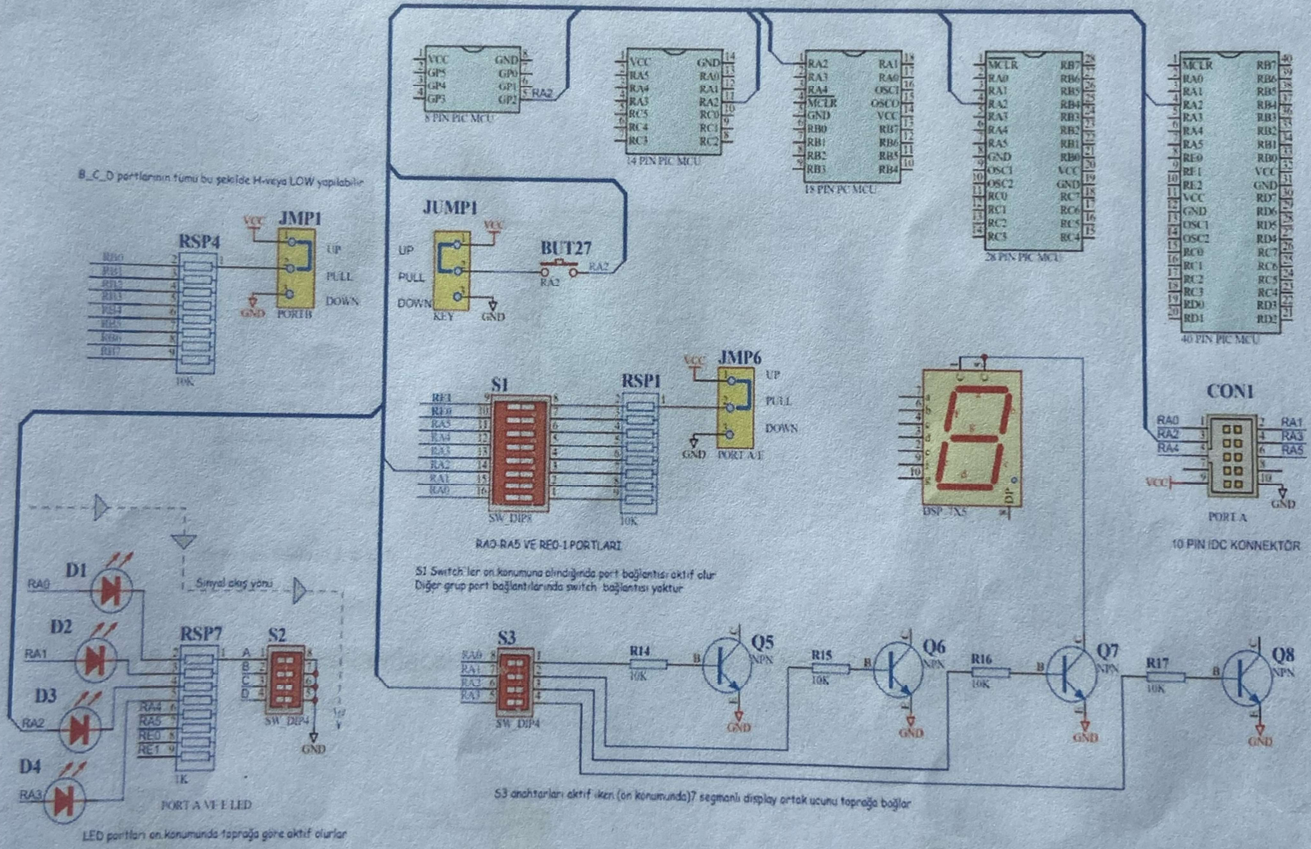
- İşlemci giriş-çıkış portlarının harici jak bağlantıları



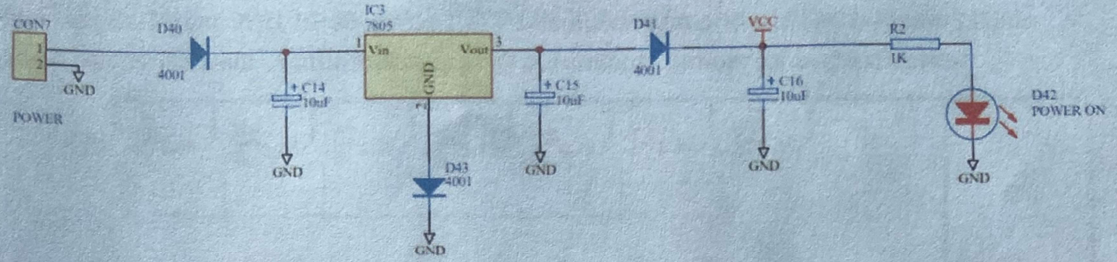
- İşlemci giriş-çıkış portlarının bağlantı şeması

Tüm işlemcilerin giriş-çıkış portları birbirlerine bağlı durumdadır.

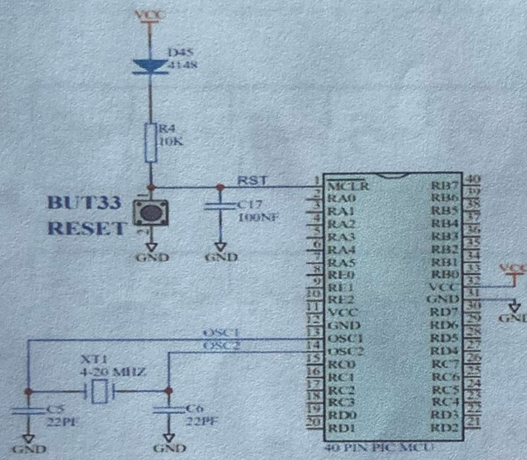
Sistem bağlantısında RA2 Portu'nun genel durumu örnek olarak gösterilmiştir



- **Deney Kartı Güç Katı**

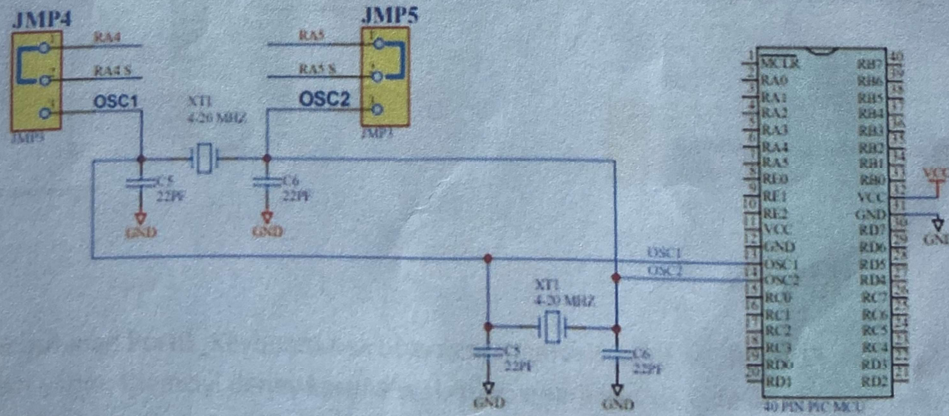


- **Reset butonu ve kristal bağlantısı**



Kristal Sokete bağlı

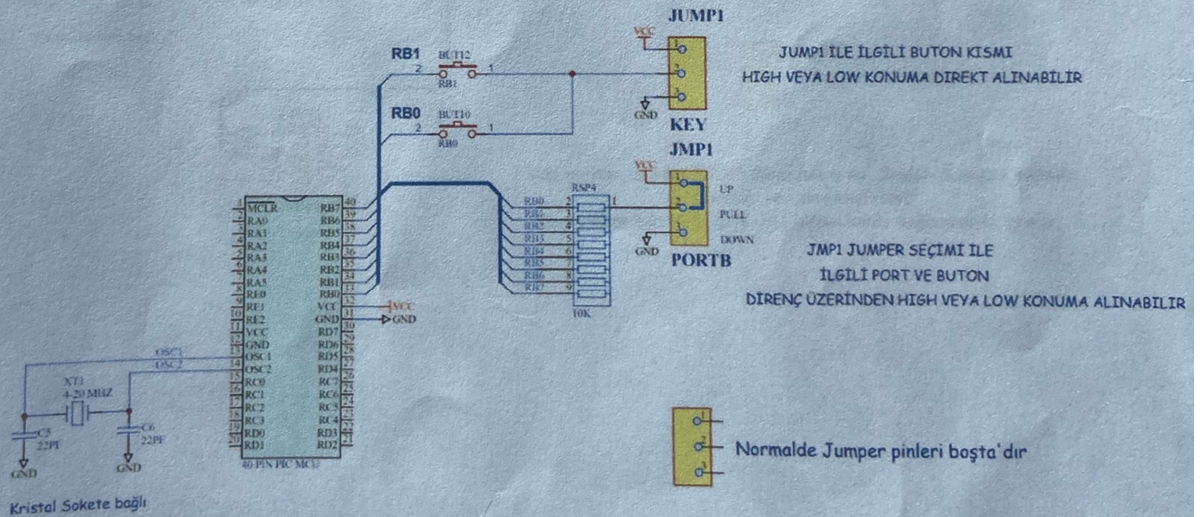
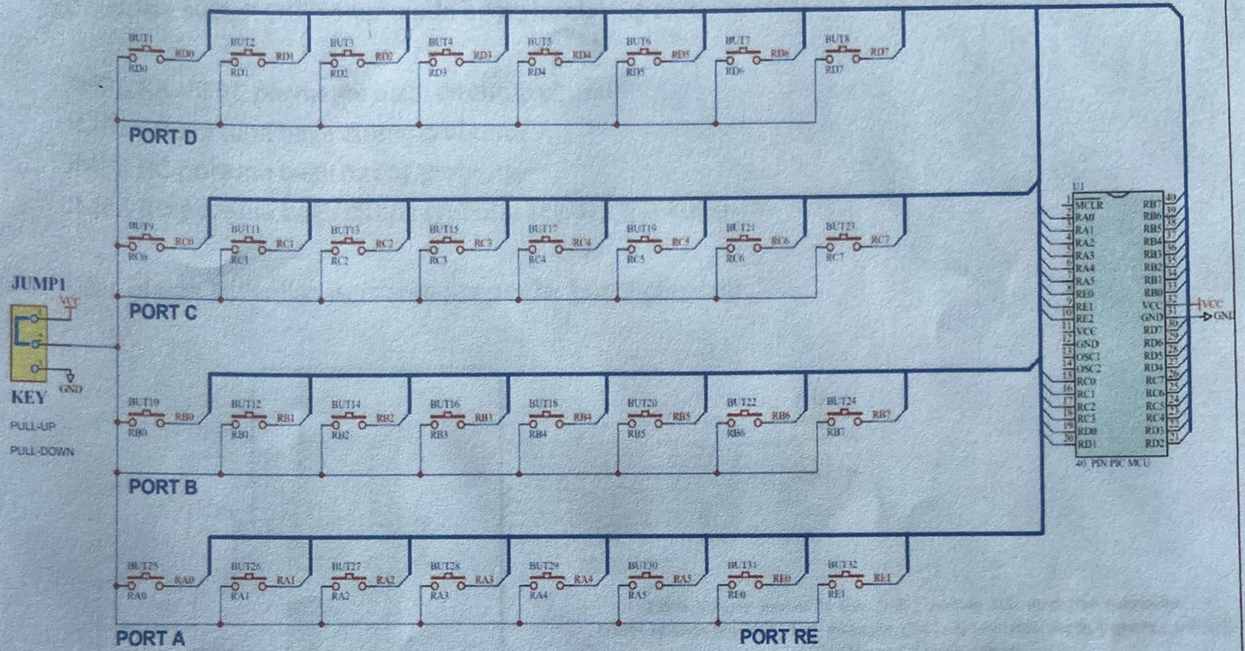
14 ve 8 pin işlemci kullanılacak ise JMP4 ve JMP5 takılı olmalıdır. Diğer işlemciler içi jumper uçları boşa olacaktır.



Kristal Sokete bağlı

- **Butonların kullanımı**

Bütün butonların bir bacağı işlemciye bağlı durumdadır. **JMP1** ile PULL UP/PULL DOWN direnci seçilir. Direnç PULL UP seçilir ise **JUMP1** ile butonların ortak ucu tersi olacak şekilde PULL DOWN seçilmelidir. Buton kullanılmadığı durumlarda jumper'lar boşta olacaktır.



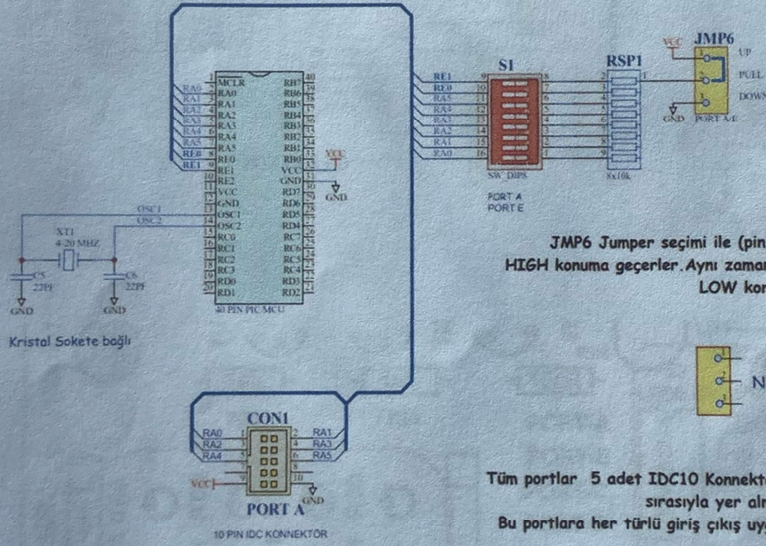
Dosyada bulunan PortD_Keyboard.hex dosyasını işlemciye yükleyin. JUMP1'i Pull-up JMP1'i Pull-down yapın. İşlemciyi deney kartına yerleştirip enerji verildiğinde örnek uygulama çalışmaya başlayacaktır. PortD'ye bağlı butonlara basarak sonucu gözlemleyin.

- İşlemci portlarının PULL-UP PULL-DOWN seçilmesi

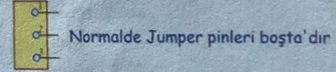
İşlemci giriş çıkış portlarının tamam jumper'lar vasıtası ile pull-up yada pull down olarak seçilebilir. Ancak RA ve RE portuna ayrıca bir DIP SWITCH eklenmiştir. Jumper takıldıktan sonra istenen pin dip switch ON konuma getirilerek pull-up yada pull-down seçilir. İstenen pinler ise switch OFF konumunda boşta bırakılmış olur.

JMP6 RA VE RE portlarına bağlı direnç grubunu,
 JMP1 RB portuna bağlı direnç grubunu,
 JMP2 RC portuna bağlı direnç grubunu,
 JMP3 RD portuna bağlı direnç grubunu seçmek için kullanılır.

Çıkış olarak kullanılan portlarda jumper'lar boşta olmalıdır.



JMP6 Jumper seçimi ile (pin 1-2) portlar 10k dirençler üzerinden HIGH konuma geçerler. Aynı zamanda (Pin2-3) arasında seçim yapılırsa portlar LOW konuma geçmiş olurlar.

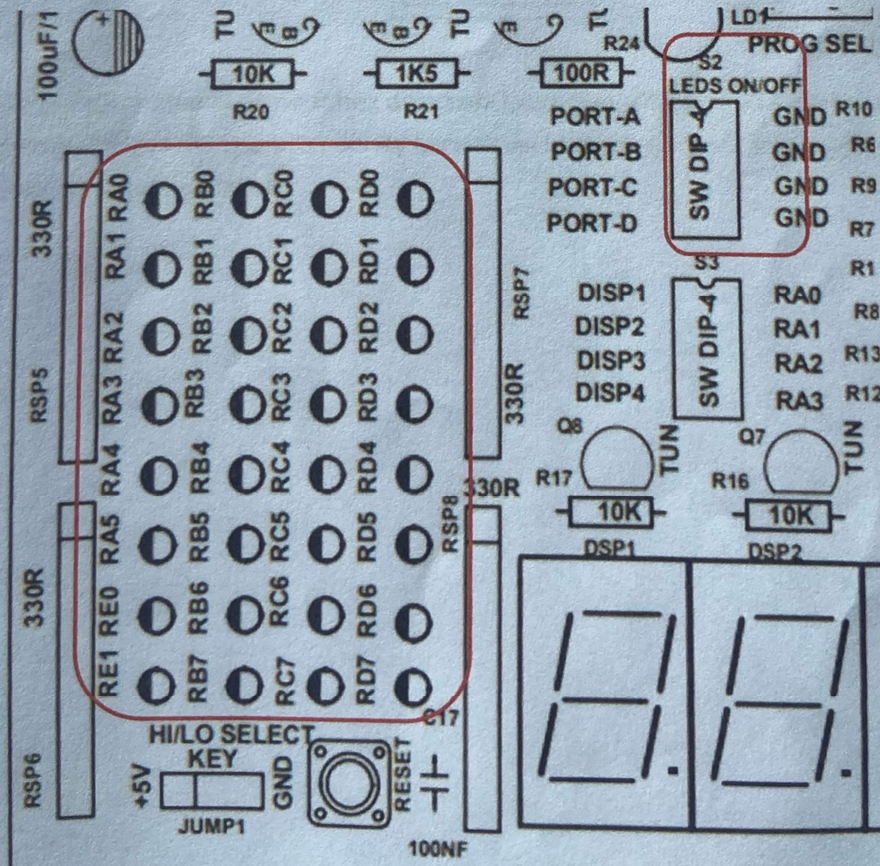
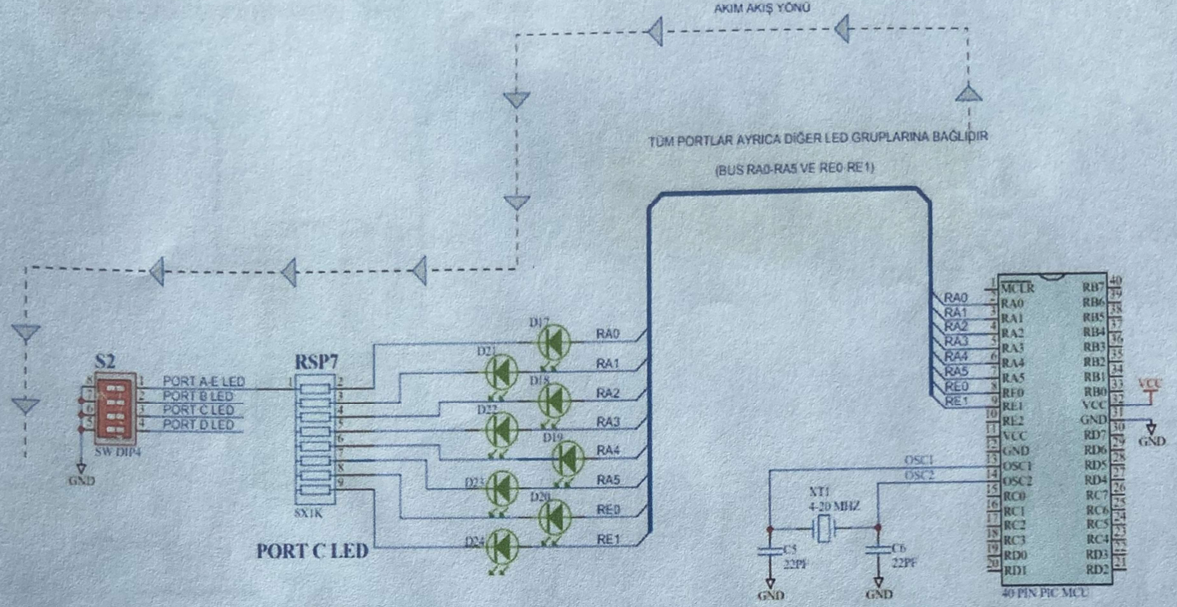


Normalde Jumper pinleri boşta'dır

Tüm portlar 5 adet IDC10 Konnektörü ile Baskılı devrenin sağında sırasıyla yer almaktadırlar. Bu portlara her türlü giriş çıkış uygun kablo bağlantısıyla yapılır.

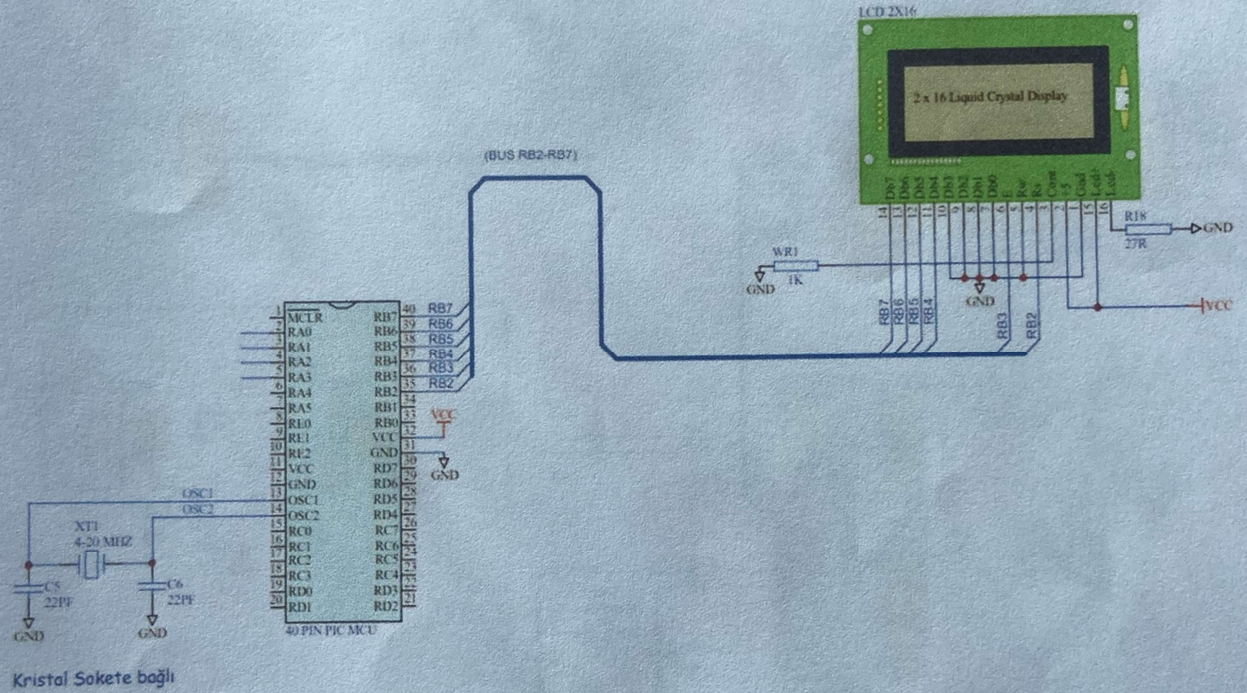
- **Ledlerin kullanımı**

Kart üzerinde işlemcinin her pinine bağlı 32 adet led bulunmaktadır. Bu ledler ile işlemci giriş-çıkışların durumu her an gözlemlenebilir. İstenilen gruptaki ledler S2 dip switch ile aktif edilir. Bağlantı şeması aşağıdaki gibidir.



- **LCD Display Kullanımı**

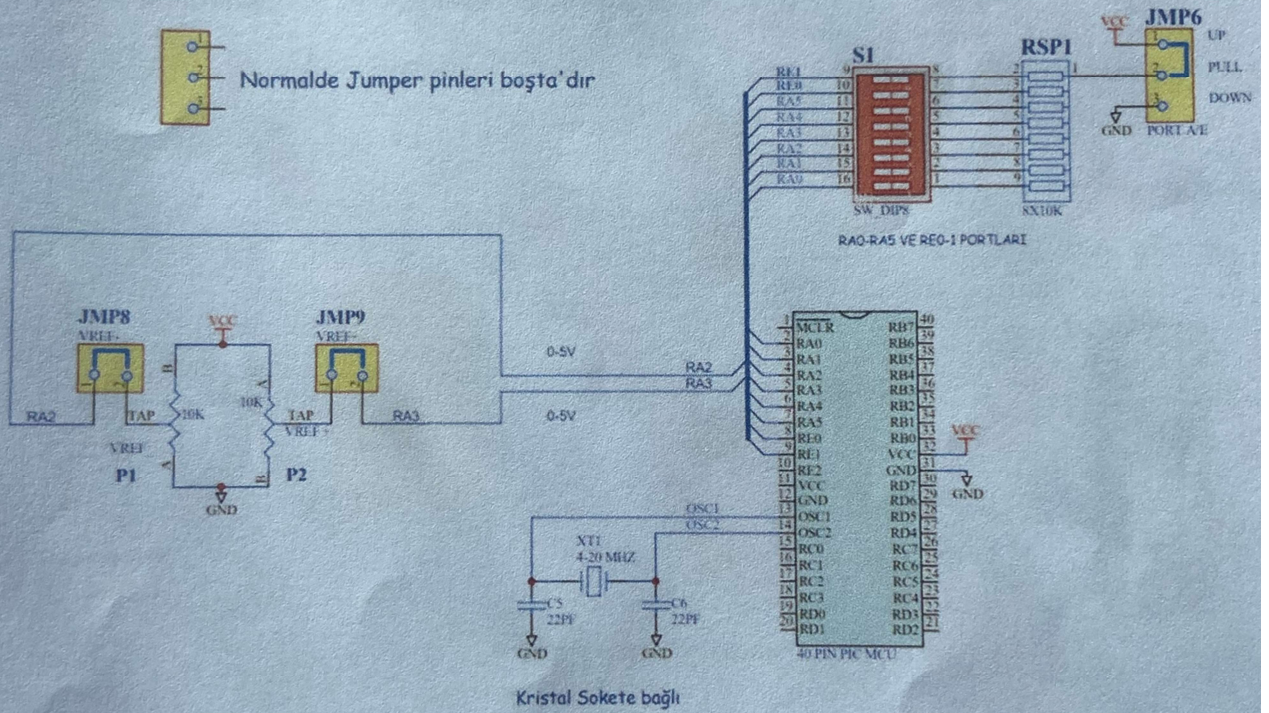
LCD display doğrudan RB portuna bağlı durumdadır. Başka deneyler yapılırken doğru kodlar gelmedikçe pinler bağlı olsa da çalışmayacaktır. Engel olacağı düşünüldüğünde yerinden çıkarılabilir. Bağlantı şeması aşağıdaki gibidir. LCD panel farklı port ile kullanılmak istenirse Breadboard üzerine takılarak gerekli bağlantılar yapılabilir.



Dosyada bulunan LCD_ornek.hex dosyasını işlemciye yükleyin. İşlemciyi deney kartına yerleştirip enerji verildiğinde örnek uygulama çalışmaya başlayacaktır.

- **ADC Kanalı ve Trimpotların Kullanımı**

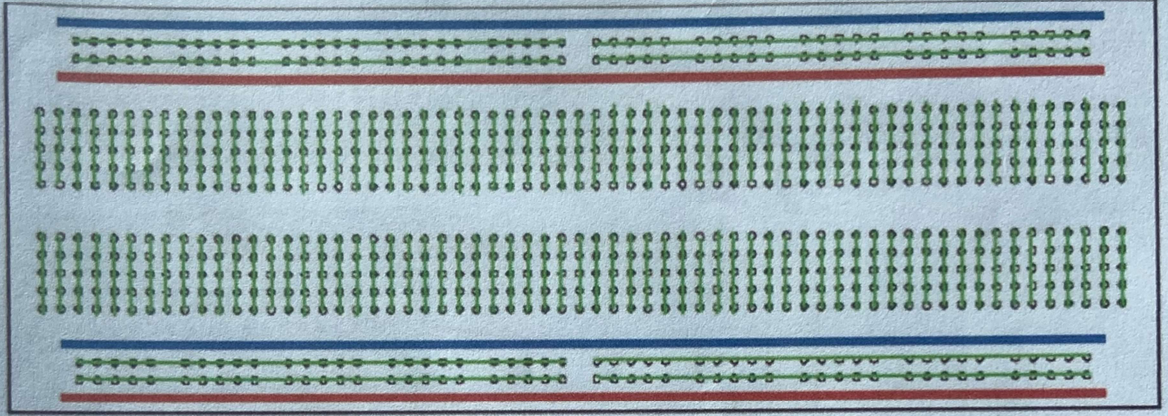
Kart üzerinde JMP8 ve JMP9 jumperları ile RA2 ve RA3 pinlerine bağlanabilen 2 adet trimpot bulunmaktadır. Trimpot kullanımında S1 switch üzerinde RA2 ve RA3'e bağlı anahtarlar OFF konumunda olmalıdır. Diğer ADC girişleri kullanılmak istenirse yan tarafta bulunan PORT A konnektöründen bağlantı yapılarak kullanılabilir. Bağlantı şeması aşağıdaki gibidir.



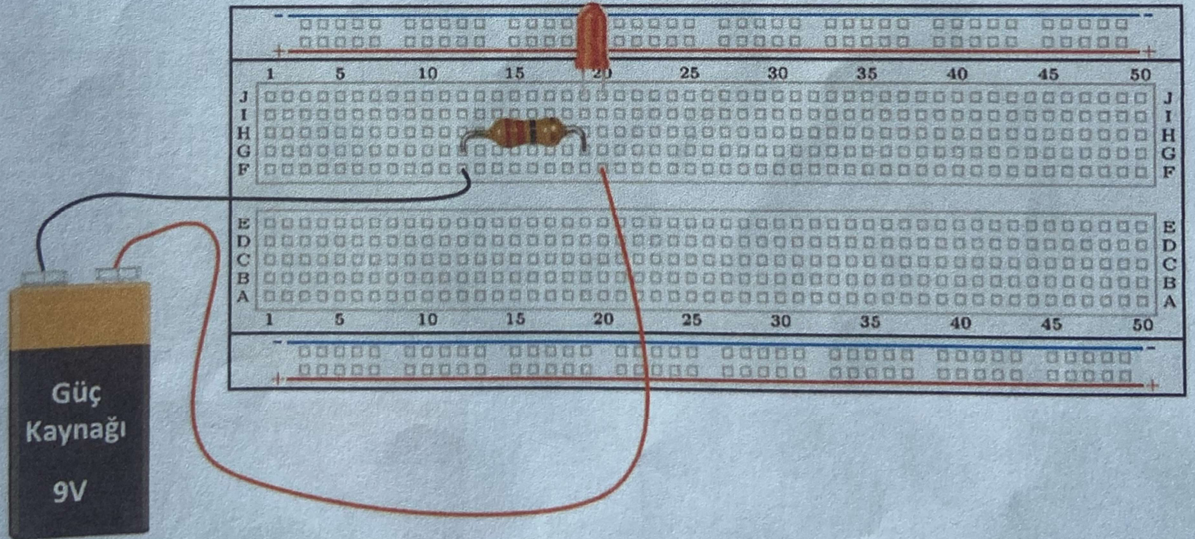
Dosyada bulunan ADC_Ornek.hex dosyasını işlemciye yükleyin. JMP8 ve JMP9 jumper'ları takın. İşlemciyi deney kartına yerleştirip enerji verildiğinde örnek uygulama çalışmaya başlayacaktır. P1 ve P2 Trimpot değerlerini değiştirerek sonucu gözlemleyin.

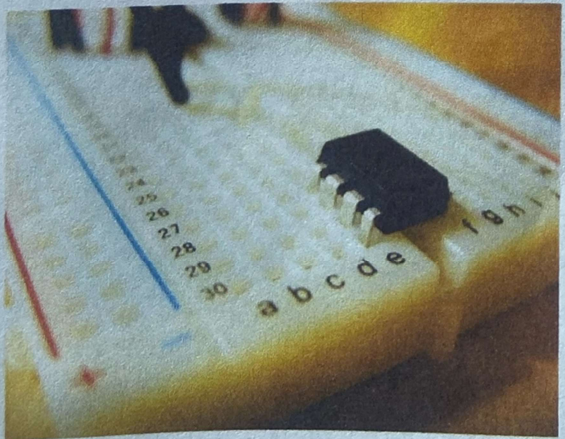
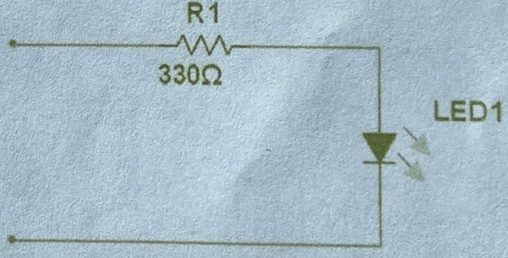
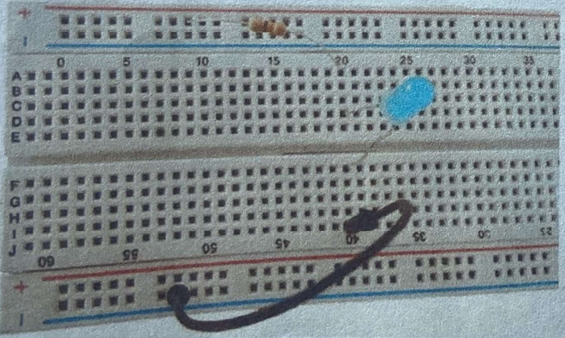
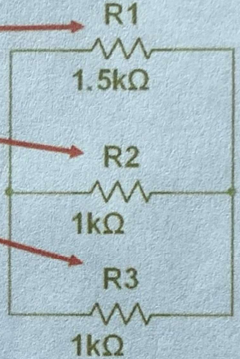
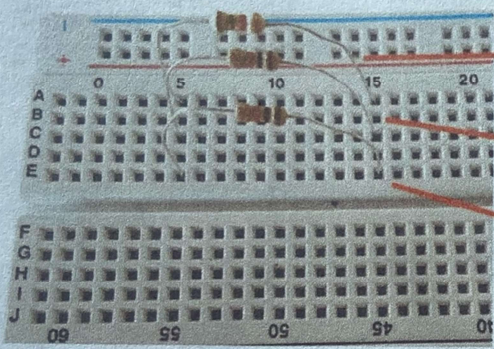
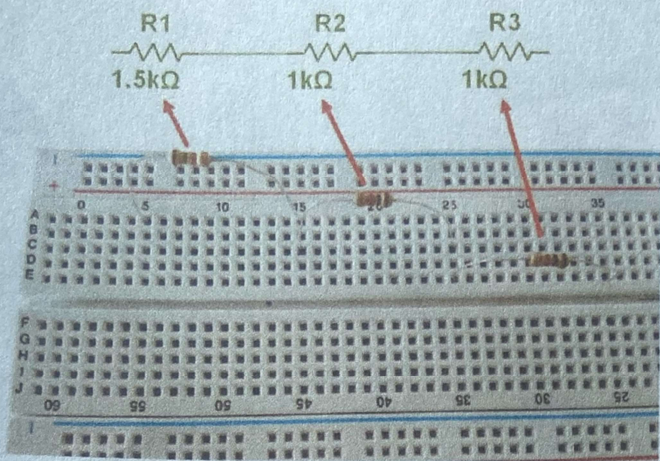
4- BREADBOARD

Deney seti üzerinde genel kullanım için ayrılmış bir adet breadboard bulunmaktadır. Breadboard'ın iç yapısı aşağıdaki gibidir. Üzeri çizgi ile birleştirilen pinler kendi arasında kısadevre durumundadır.



Breadboard üzerinde devre kurulurken aşağıdaki resimlerde görüldüğü gibi bağlantı yapılır.





DENEYLER

Deney No:	1
Deney Adı:	Led Yakma (Blink Deneyi)

Amaç

PIC mikrodnetleyici ile temel dijital çıkış işlemini öğrenmek. Belirli zaman aralıklarında bir LED'in yakılıp söndürülmesi sağlanacaktır.

Gerekli Malzemeler

- PIC16F877A veya benzeri PIC mikrodnetleyici
- 1 adet LED
- 220 ohm direnç
- Breadboard, jumper kablolar

Teorik Bilgi

Mikrodnetleyici içerisindeki PORT pinleri çıkış modunda kullanılarak bir LED'e lojik 1 (HIGH) verildiğinde LED yanar, lojik 0 (LOW) verildiğinde söner. Bu deneyde zaman gecikmesi sağlanarak LED'in belirli aralıklarla yanıp sönmeye sağlanır.

Deneyin Yapılışı

- 1.) ARGELINE setinde LED bağlı olan pinin TRIS register'ı çıkış olarak ayarlanır.
- 2.) LED pinine HIGH ve LOW değerleri belirli zaman aralıklarıyla verilir.
- 3.) Kod mikrodnetleyiciye yüklenir ve LED'in yanıp söndüğü gözlemlenir.
- 4.) İlk olarak uygulamayı açıyoruz ve c dilinde kodu "CCP C" uygulamasına yazıp seçeneklerden kaydediyoruz.
- 5.) Kodu "Compile" ediyoruz. Alt menüde "0 Warnings, 0 errors" yazısını görmemiz gerekiyor yoksa kodumuzda bir hata var demektir.
- 6.) Daha sonra kodu ".Hex" dosyası olarak kaydediyoruz.
- 7.) Ara kablo ile bilgisayar kasasından kodu; mikroşlemcimize atıyoruz
- 8.) Kodu "microbrn" uygulaması ile yükleyip kaydedip mikroşlemci çipine yüklemiş oluyoruz.
- 9.) Çipi yanındaki küçük kolu hareket ettirerek çıkarıyoruz ve mikroşlemcideki ana işlemcimize takıyoruz.
- 10.) Deney bu şekilde tamamlanmış oluyor.

C Dili Kod

```
#include <16f877A.h> // Mikrodenetleyici tanımlama dosyasını ekliyoruz.
#use delay(crystal=2000000) // Kristal osilatör frekansını giriyoruz

Void main () // ana fonksiyonumuz başlangıcı
While (True){//sonsuz döngünün başlangıcı
output_high(pin_b7); //b portu 7 nolu çıkışını 1 yapıyoruz
delay_ms (500); //500 milisaniye bekleme yaptırıyoruz
output_low(pin_b7); // b portu 7 numaralı pinini 0 yapıyoruz
delay_ms (500); // 500 milisaniye bekleme yaptırıyoruz
}
} // main fonksiyonumuzun sonu
```

Deney No:	2
Deney Adı:	Buton ile Led Kontrolü

Deney Malzemeleri

- PIC16F877A
- 1 adet LED
- 1 adet 330 ohm direnç (LED için)
- 1 adet buton
- 10K ohm direnç (pull-down için)
- Breadboard + jumper kabloları
- PICkit 3

Teorik Bilgi

Buton Nasıl Çalışır?

Buton basılmadığında, giriş pini LOW (0) durumundadır.

Basıldığında, giriş pini HIGH (1) olur.

Genellikle butonla birlikte bir pull-down ya da pull-up direnci kullanılır ki sinyal kararsız olmasın.

Deneyin Yapılışı

- 1.) ARGELINE setinde LED bağlı olan pinin TRIS register'ı çıkış olarak ayarlanır.
- 2.) LED pinine HIGH ve LOW değerleri belirli zaman aralıklarıyla verilir.
- 3.) İlk olarak uygulamayı açıyoruz ve c dilinde kodu “CCP C” uygulamasına yazıp seçeneklerden kaydediyoruz.
- 4.) Kodu “Compile” ediyoruz. Alt menüde “0 Warnings, 0 errors” yazısını görmemiz gerekiyor yoksa kodumuzda bir hata var demektir.
- 5.) Daha sonra kodu “. Hex” dosyası olarak kaydediyoruz.
- 6.) Ara kablo ile bilgisayar kasasından kodu; mikroişlemcimize atıyoruz.
- 7.) Kodu “microbrn” uygulaması ile yükleyip kaydedip mikroişlemci çipine yüklemiş oluyoruz.
- 8.) Çipi yanındaki küçük kolu hareket ettirerek çıkarıyoruz ve mikroişlemcideki ana işlemcimize takıyoruz.
- 9.) Deney bu şekilde tamamlanmış oluyor.

C Dili Kod

```
#include <16F877A.h>
```

```
#fuses HS, NOWDT, NOPUT, NOLVP
```

```
#use delay (clock = 4000000)
```

```
#define BUTON PIN_B0
```

```
#define LED PIN_B1
```

```
Void main () {
```

```
Set_tris_b(0x01);  
output_low(LED);
```

```
while(TRUE) {  
if (input(BUTON) == 0) {  
output_high(LED);  
}  
else {  
output_low(LED);  
}  
}  
}
```

Deney No:	3
Deney Adı:	7 segment Display İle Sayı Gösterimi

Deneyin Amacı

PIC mikrodenetleyici kullanarak 7 segment ekranda 0'dan 9'a kadar sayıları görüntülemek.

Teorik Bilgi

7 segment ekranlar, her bir segmentin aktif edilmesiyle rakamları oluşturur. Her rakam için belirli bir segment kombinasyonu gerekir. Kodda sayılara ait segment kodları dizi içinde saklanır.

Deneyin Yapılışı

- 1.) Segment pinleri çıkış olarak tanımlanır.
- 2.) Rakamların biner kod karşılıkları dizi olarak tanımlanır.
- 3.) Her 1 saniyede bir ekran bir sonraki sayıya geçecek şekilde kod yazılır.
- 4.) Her Bir Rakam İçin Aktif Edilmesi Gereken Segmentler (a-g) nelerdir ?
- 5.) Kullanılan segment tipi : Ortak Anot mu , Ortak Katot mu ?
- 4.) İlk olarak uygulamayı açıyoruz ve c dilinde kodu "CCP C" uygulamasına yazıp seçeneklerden kaydediyoruz.
- 5.) Kodu "Compile" ediyoruz. Alt menüde "0 Warnings, 0 errors" yazısını görmemiz gerekiyor yoksa kodumuzda bir hata var demektir.
- 6.) Daha sonra kodu ". Hex" dosyası olarak kaydediyoruz.
- 7.) Ara kablo ile bilgisayar kasasından kodu; mikroişlemcimize atıyoruz.
- 8.)Kodu "microbrn" uygulaması ile yükleyip kaydedip mikroişlemci çipine yüklemiş oluyoruz.
- 9.) Çipi yanındaki küçük kolu hareket ettirerek çıkarıyoruz ve mikroişlemcideki ana işlemcimize takıyoruz.
- 10.) Deney bu şekilde tamamlanmış oluyor.

C Dili Kodu

```
#include <16F877A.h>
#fuses XT, NOWDT, NOPUT, NOLVP
#use delay(clock=4000000) // 4 MHz kristal kullanıldığı varsayılıyor

// 0-9 arası sayıların 7 segment karşılıkları (a-g sırasına göre)
// RD0 -> a, RD1 -> b, ..., RD6 -> g

const int8 digit[10] = {
0b00111111, // 0
```

```
0b00000110, // 1
0b01011011, // 2
0b01001111, // 3
0b01100110, // 4
0b01101101, // 5
0b01111101, // 6
0b00000111, // 7
0b01111111, // 8
0b01101111 // 9
};

void main() {
    set_tris_d(0x00); // PORTD çıkış olarak ayarlanır
    output_d(0x00); // Başlangıçta ekran temizlenir

    while(TRUE) {
        for(int i = 0; i < 10; i++) {
            output_d(digit[i]); // Sayıya karşılık gelen segmentleri yak
            delay_ms(1000); // 1 saniye bekle
        }
    }
}
```

Deney No:	4
Deney Adı:	LCD ile Metin Gösterimi

Deney Malzemeleri

- PIC16F877A Mikrodenetleyici
- 16x2 LCD (HD44780 uyumlu)
- Potansiyometre (10K, kontrast için)
- 10K pull-up direnç (isteğe bağlı)
- Breadboard ve jumper kabloları
- PICkit 3 (ya da başka bir programlayıcı)
- Güç kaynağı (5V)

Teorik Bilgi

LCD (Liquid Crystal Display) Nedir?

16x2 LCD ekranlar, 16 sütun ve 2 satırdan oluşur. Her hücre, bir karakter gösterebilir. HD44780 LCD kontrolcüsü kullanılır (standarttır).

Çalışma Modları:

8-bit mod: Tüm veri hatları (D0-D7) kullanılır.

4-bit mod: Yalnızca D4-D7 kullanılır; pin tasarrufu sağlar (çoğunlukla tercih edilir).

Temel Pinler:

<u>Pin</u>	<u>Adı</u>	<u>Açıklama</u>
1	VSS	GND
2	VDD	+5V
3	V0	Kontrast ayarı (potansiyometre ile)
4	RS	Register Select (Komut/Veri seçimi)
5	RW	Read/Write (Yaz/Okuma seçimi – genelde GND'ye bağlanır)
6	E	Enable (Veri kabul sinyali)
11-14	D4-D7	Veri hatları (4-bit modda)

Deneyin Yapılışı

- 1.) ARGELINE setinde LED bağlı olan pinin TRIS register'ı çıkış olarak ayarlanır.
- 2.) İlk olarak uygulamayı açıyoruz ve c dilinde kodu “ CCP C ” uygulamasına yazıp seçeneklerden kaydediyoruz.
- 3.) Kodu “Compile” ediyoruz. Alt menüde “0 Warnings , 0 errors” yazısını görmemiz gerekiyor yoksa kodumuzda bir hata var demektir.
- 4.) Daha sonra kodu “.hex” dosyası olarak kaydediyoruz.
- 5.) Ara kablo ile bilgisayar kasasından kodu ; mikroişlemcimize atıyoruz.
- 6.) Kodu “microbrn” uygulaması ile yükleyip kaydedip mikroişlemci çipine yüklemiş oluyoruz.
- 7.) Çipi yanındaki küçük kolu hareket ettirerek çıkarıyoruz ve mikroişlemcideki ana işlemcimize takıp sabitliyoruz.
- 8.) Deney bu şekilde tamamlanmış oluyor.

C Dili Kodu

```
#include <16f877A.h>
#define _XTAL_FREQ 20000000

// LCD pin tanımlamaları
#define RS RC0
#define E RC1
#define D4 RC2
#define D5 RC3
#define D6 RC4
#define D7 RC5

void lcd_command(unsigned char cmd);
void lcd_char(unsigned char data);
void lcd_init();
void lcd_string(const char *str);
void lcd_enable();

void main(void) {
    TRISC = 0x00; // PORTC çıkış
    lcd_init();
    lcd_string("Merhaba");
    while(1);
}

void lcd_enable() {
    E = 1;
    __delay_ms(2);
    E = 0;
}

void lcd_command(unsigned char cmd) {
    RS = 0;
```

```
D4 = (cmd >> 4) & 1;
D5 = (cmd >> 5) & 1;
D6 = (cmd >> 6) & 1;
D7 = (cmd >> 7) & 1;
lcd_enable();
```

```
D4 = cmd & 1;
D5 = (cmd >> 1) & 1;
D6 = (cmd >> 2) & 1;
D7 = (cmd >> 3) & 1;
lcd_enable();
```

```
__delay_ms(2);
}
```

```
void lcd_char(unsigned char data) {
RS = 1;
D4 = (data >> 4) & 1;
D5 = (data >> 5) & 1;
D6 = (data >> 6) & 1;
D7 = (data >> 7) & 1;
lcd_enable();
```

```
D4 = data & 1;
D5 = (data >> 1) & 1;
D6 = (data >> 2) & 1;
D7 = (data >> 3) & 1;
lcd_enable();
```

```
__delay_ms(2);
}
```

```
void lcd_string(const char *str) {
while(*str) {
lcd_char(*str++);
}
}
```

```
void lcd_init() {
__delay_ms(20);
lcd_command(0x02); // 4-bit mod
lcd_command(0x28); // 2 satır, 5x8 font
lcd_command(0x0C); // Display açık, imleç kapalı
lcd_command(0x06); // Sağa doğru yazma
lcd_command(0x01); // Ekranı temizle
}
```

Deney No:	5
Deney Adı:	LDR (Işıık Sensörü) ile LED Parlaklığı Kontrolü

Deneyin Amacı

Ortam ışığına göre LED'in parlaklığını ayarlamak.

Deney Malzemeleri

- PIC16F877A
- 1x LDR
- 1x 10kΩ direnç
- 1x LED
- 1x 220Ω direnç
- Breadboard, jumper kablolar

Teorik Bilgi

LDR (ışığa duyarlı direnç), üzerine düşen ışığın şiddetine göre direnci değişen bir sensördür. Ortam aydınlığı arttıkça direnci azalır. Bu deneyde LDR'den okunan analog değerler ADC (Analog to Digital Converter) yardımıyla sayısal hale getirilir ve LED'in yanıp sönme durumu bu değere göre ayarlanır.

Deneyin Yapılışı

- 1.) LDR + 10k direnç ile voltaj bölücü kurulur; orta nokta RA0/AN0'a bağlanır ve RA0 ADC girişi olarak TRIS ayarlanır.
- 2.) LED çıkışı için ilgili port pininin TRIS register'ı çıkış yapılır (ör. RB0 çıkış).
- 3.) CCS C'de ADC ayarları (setup_adc, setup_adc_ports) yapılır ve LDR okumaları ile basit eşik veya PWM mapping kodu yazılır.
- 4.) Kod compile edilir; .hex oluşturulur.
- 5.) Program çipe yüklenir ve devreye takılır.
- 6.) Ortam ışığı değiştirildikçe LED parlaklığı/ON-OFF davranışı gözlenir; ADC ham değerleri not edilir.
- 7.) Eğer PWM ile parlaklık kontrolü yapılacaksa PWM modülü uygun şekilde kurulup test edilir.
- 8.) Deney raporunda ADC değerleri ile ışık-seviye ilişkisi grafiğı eklenir.

C Dili Kodu

```
#include <16F877A.h>
#fuses HS,NOWDT,NOPROTECT,NOLVP
#use delay(clock=20000000)
```

```
void main(){
    int16 adc_value;
    set_tris_a(0xFF);
    set_tris_b(0x00);
    setup_adc(ADC_CLOCK_DIV_32);
    setup_adc_ports(ALL_ANALOG);
    set_adc_channel(0);

    while(TRUE){
        adc_value = read_adc();
        output_d(adc_value >> 2); // opsiyonel: LED bar grafiđi
        if(adc_value < 300)
            output_high(PIN_B0);
        else
            output_low(PIN_B0);
        delay_ms(100);
    }
}
```

Deney No:	6
Deney Adı:	Sıcaklık Sensörü (LM35) ile Ölçüm

Deneyin Amacı

LM35 sensöründen sıcaklık değeri ölçüp LCD'de göstermek.

Deney Malzemeleri

- LM35
- 16x2 LCD
- PIC16F877A
- Breadboard, jumper

Teorik Bilgi

LM35 sıcaklık sensörü, sıcaklık başına 10 mV çıkış veren doğrusal bir sensördür. ADC ile ölçülen bu gerilim değeri sıcaklık olarak hesaplanır. Bu deneyde sıcaklık değeri LCD ekrana yansıtılarak ölçüm yapılır.

Deneyin Yapılışı

- 1.) LM35 çıkışı RA0/AN0'a bağlanır; ADC portları analog olarak konfigüre edilir (setup_adc_ports(ALL_ANALOG)).
- 2.) LCD bağlantısı yapılır, lcd_init() ile başlatılır; TRIS ayarları uygulanır.
- 3.) CCS C'de ADC okumadan °C dönüşüm formülü $((adc * Vref / 1023) * 100)$ kullanılarak hesaplama yazılır ve LCD'ye gönderilir.
- 4.) Kod compile edilir; hatasızsa .hex kaydedilip yüklenir.
- 5.) Çip ve devre çalıştırılır; ölçülen sıcaklık sabit referans (termometre) ile karşılaştırılır.
- 6.) Ölçümlerin stabil olması için birkaç okuma ortalaması alınabilir (ör. 5 okuma ortalaması).
- 7.) Sensör doğruluğu/kalibrasyonu gerekiyorsa referans değerlerle düzeltme katsayısı hesaplanır.
- 8.) Deney raporuna hesaplama adımları ve gözlem notları eklenir.

- LM35 çıkışı → RA0 (AN0)
- Vout (LM35) → RA0
- VCC → +5V, GND → GND

C Dili Kodu

```
#include <16F877A.h>
#device ADC=10
#fuses HS,NOWDT,NOLVP,NOPROTECT
#use delay(clock=2000000)
#include <lcd.c>

void main(){
    float sicaklik;
    int16 adc_deger;
    lcd_init();
    setup_adc(ADC_CLOCK_DIV_32);
    setup_adc_ports(ALL_ANALOG);
    set_adc_channel(0);
    lcd_putc("\fSicaklik Olcumu");

    while(TRUE){
        adc_deger = read_adc();
        sicaklik = (adc_deger * 5.0 / 1023.0) * 100; // 10mV/°C
        lcd_gotoxy(1,2);
        printf(lcd_putc,"T=%.1f C", sicaklik);
        delay_ms(500);
    }
}
```

Deney No:	7
Deney Adı:	Potansiyometre ile PWM Sinyali Kontrolü

Deneyin Amacı

Pot ile PWM darbe genişliğini ayarlamak.

Deney Malzemeleri

- Potansiyometre (10k Ω)
- LED
- 220 Ω direnç

Teorik Bilgi

PWM (Pulse Width Modulation), dijital sinyallerin görev döngüsünü değiştirerek analog etki oluşturan bir tekniktir. Potansiyometre aracılığıyla ADC'den okunan değer PWM çıkışına uygulanarak LED'in parlaklığı kontrol edilir.

Deneyin Yapılışı

- 1.) Pot orta ucu RA0/AN0'a bağlanır (ADC girişi). TRIS analog giriş ayarlanır.
- 2.) PWM çıkışı için CCP1 pini (RC2) kullanılacak şekilde TRIS ayarlanır (CCP modül ayarları otomatik yönlendirilir).
- 3.) CCS C'de setup_ccp1(CCP_PWM); setup_timer_2(...) ile PWM başlatılır; ADC'den okunan değere göre set_pwm1_duty() uygulanır.
- 4.) Kod compile edilip .hex oluşturulur; hata yoksa yüklenir.
- 5.) Devre çalıştırılır; pot çevrildikçe LED parlaklığı veya bağlı motorun hızı değişimi gözlenir.
- 6.) PWM frekansı/duty testleri yapılır; gözlemler kaydedilir.
- 7.) Eğer motor sürülecekse sürücü (transistör/H-bridge) kullanılması gerektiği not edilir.
- 8.) Deney raporunda PWM hesaplama ve ADC→duty eşlemesi gösterilir.

- Pot orta ucu → RA0 (AN0)
- LED → CCP1 çıkışı (RC)

C Dili Kodu

```
#include <16F877A.h>
#fuses HS,NOWDT,NOLVP
```

```
#use delay(clock=2000000)

void main(){
  int16 adc_val;
  setup_adc_ports(ALL_ANALOG);
  setup_adc(ADC_CLOCK_DIV_32);
  set_adc_channel(0);

  setup_ccp1(CCP_PWM);
  setup_timer_2(T2_DIV_BY_16,255,1);

  while(TRUE){
    adc_val = read_adc();
    set_pwm1_duty(adc_val);
    delay_ms(50);
  }
}
```

Deney No:	8
Deney Adı:	Ultrasonik Sensör (HC-SR04) ile Mesafe Ölçümü

Deneyin Amacı

Ses dalgası süresine göre mesafe ölçmek.

Deney Malzemeleri

- HC-SR04 modülü
- LCD (isteğe bağlı)
- PIC16F877A

Teorik Bilgi

HC-SR04 ultrasonik sensörü, ses dalgalarının yankılanma süresini ölçerek mesafe hesaplar. TRIG piniyle gönderilen ses dalgası, cisimden yansyarak ECHO piniyle geri döner ve bu süre mesafe hesaplamasında kullanılır.

Deneyin Yapılışı

- 1.) TRIG pini RB0, ECHO pini RB1 olarak bağlanır; TRIG çıkış, ECHO giriş olarak TRIS konfigüre edilir.
- 2.) LCD veya seri çıktı ile sonuç gösterimi yapılacaksa bağlantılar hazırlanır.
- 3.) CCS C'de TRIG için 10 µs darbe üretimi, sonra ECHO'nun HIGH süresi için timer veya pulse ölçüm kodu yazılır.
- 4.) Compile edilir; .hex dosyası oluşturulur ve çipe yüklenir.
- 5.) Devre çalıştırılır; nesne farklı mesafelerde konumlandırılarak ölçüm doğruluğu kontrol edilir.
- 6.) Ölçümlerde zaman aşımı (timeout) kontrolü olmalı; 0 okuma veya hatalı uzun okuma ele alınır.
- 7.) Sonuçlar cm cinsinden hesaplanıp LCD'ye yazdırılır; tutarlılık test edilir.
- 8.) Deney raporunda ölçüm aralığı (2–400 cm), doğruluk ve hata kaynakları belirtilir.

- TRIG → RB0
- ECHO → RB1

C Dili Kodu

```
#include <16F877A.h>
#fuses HS,NOWDT,NOLVP
```

```

#include <math>
#include <clock=20000000>
#include <lcd.c>

#define TRIG PIN_B0
#define ECHO PIN_B1

long zaman;
float mesafe;

void main(){
    lcd_init();
    lcd_putc("\fMesafe Olcumu");

    while(TRUE){
        output_low(TRIG);
        delay_us(2);
        output_high(TRIG);
        delay_us(10);
        output_low(TRIG);
        while(!input(ECHO));
        set_timer1(0);
        while(input(ECHO));
        zaman = get_timer1();
        mesafe = (zaman * 0.0343)/2;
        lcd_gotoxy(1,2);
        printf(lcd_putc,"%0.1f cm ",mesafe);
        delay_ms(500);
    }
}

```

Deney No:	9
Deney Adı:	DHT11 ile Sıcaklık ve Nem Ölçümü

Deneyin Amacı

Dijital sensörden sıcaklık ve nemi okumak.

Deney Malzemeleri

- DHT11 sensör
- LCD
- PIC16F877A

Deneyin Yapılışı

- 1.) DHT11 veri pini (ör. RB0) pull-up ile devreye bağlanır; başlangıçta pin çıkış yapılıp HIGH tutulur (idle).
- 2.) LCD bağlıysa lcd_init() çağrılır; TRIS ayarları kontrol edilir.
- 3.) CCS C projesine DHT okuma fonksiyonu (bit-bang) yazılır veya hazır kütüphane eklenir.
- 4.) Kod compile edilir; “0 Warnings, 0 Errors” doğrulanır ve .hex dosyası üretilir.
- 5.) Oluşan .hex dosyası programlayıcı ile mikroişlemciye yüklenir.
- 6.) Çip devreye takılıp DHT11 okuma başlatılır; LCD’de Nem ve Sıcaklık değerleri görüntülenir.
- 7.) Okumalar doğruluğu için birkaç saniyede bir yapılır (DHT11 max ~1 Hz), timeout durumları hata mesajı ile raporlanır.
- 8.) Deney raporundaki gözlemlerde checksum geçme/kalma durumları ve sensör stabilite notları yer alır.

Teorik Bilgi

DHT11 sıcaklık ve nem sensörü dijital çıkış verir. İçinde nem ve sıcaklık ölçüm elemanları bulunur. Mikrodenetleyiciye belirli zamanlama ile veri gönderir. Bu deneyde sensörden okunan sıcaklık ve nem bilgileri LCD ekranda gösterilir

C Dili Kodu

```
#include <16F877A.h>
#fuses HS, NOWDT, NOLVP
#use delay(clock=20000000)
#include <lcd.c>

#define DHT11_PIN PIN_B0
```

```
int8 rh_integral, rh_decimal, temp_integral, temp_decimal, checksum;
```

```
int1 DHT11_read() {  
    int8 i, j;  
    output_low(DHT11_PIN);  
    delay_ms(18);  
    output_high(DHT11_PIN);  
    delay_us(40);  
    output_float(DHT11_PIN);  
    if(!input(DHT11_PIN)) {  
        delay_us(80);  
        if(input(DHT11_PIN)) {  
            delay_us(80);  
            for(j=0;j<5;j++){  
                int8 result=0;  
                for(i=0;i<8;i++){  
                    while(!input(DHT11_PIN));  
                    delay_us(30);  
                    if(input(DHT11_PIN)) result |= (1 << (7 - i));  
                    while(input(DHT11_PIN));  
                }  
                if(j==0) rh_integral = result;  
                else if(j==1) rh_decimal = result;  
                else if(j==2) temp_integral = result;  
                else if(j==3) temp_decimal = result;  
                else checksum = result;  
            }  
            return 1;  
        }  
    }  
    return 0;  
}
```

```
void main() {  
    lcd_init();  
    lcd_putc("\fDHT11 Olcumu");  
}
```

```
while(TRUE) {
  if(DHT11_read()) {
    lcd_gotoxy(1,1);
    printf(lcd_putc,"Nem: %u.%u %%", rh_integral, rh_decimal);
    lcd_gotoxy(1,2);
    printf(lcd_putc,"Sic: %u.%u C ", temp_integral, temp_decimal);
  } else {
    lcd_putc("\fSensor Hatasi");
  }
  delay_ms(1000);
}
}
```

Deney No:	10
Deney Adı:	IR Sensörü ile Engel Algılama ve Buzzer Uyarısı

Deneyin Amacı

IR sensör ile cisim algılanınca buzzer ve LED uyarısı vermek.

Deney Malzemeleri

- IR sensör modülü
- LED
- Buzzer
- 220Ω direnç

Teorik Bilgi

IR sensörü, kızılötesi ışık yansımalarını algılayarak cisim tespiti yapar. Cisim algılandığında sinyal çıkışı verir. Bu deneyde engel algılandığında LED ve buzzer birlikte aktif olur.

Deneyin Yapılışı

- 1.) IR sensör çıkışı RB0'a, buzzer RB2'ye, LED RB1'e bağlanır; TRIS ayarları yapılır (RB0 giriş, RB1/RB2 çıkış).
- 2.) CCS C'de giriş okuma ve çıkış kontrol kodu yazılır; algılama toburnu için debouncing gerekebilir.
- 3.) Kod compile edilir; .hex oluşturulur ve çipe yüklenir.
- 4.) Devre çalıştırılır; engel yaklaştırıldığında LED ve buzzer aktifleşmesi gözlemlenir.
- 5.) Buzzer süresi ve LED yanma davranışı konfigüre edilir (sürekli vs kısa uyarı).
- 6.) Gerekirse algılama eşik değerleri sensörün potansiyometresi ile ayarlanır.
- 7.) Deney tekrarlanıp farklı yüzeyler/test mesafeleri ile güvenilirlik kontrolü yapılır.
- 8.) Deney raporunda alert tepki süresi ve yanlış pozitif/negatif oranları not edilir.

- IR OUT → RB0
- LED → RB1
- Buzzer → RB2

C Dili Kodu

```
#include <16F877A.h>
#fuses HS,NOWDT,NOLVP
```

```
#use delay(clock=2000000)

void main(){
  set_tris_b(0x01); // RB0 giriş, diğerleri çıkış
  while(TRUE){
    if(input(PIN_B0)){
      output_high(PIN_B1);
      output_high(PIN_B2);
    } else {
      output_low(PIN_B1);
      output_low(PIN_B2);
    }
    delay_ms(100);
  }
}
```

RAPOR DEĞERLENDİRME TABLOSU

DENEY NO									
1	2	3	4	5	6	7	8	9	10
ORTALAMA NOT									

AÇIKLAMA:

Hazırlayan Öğrenci

CELAL YAVUZ