

İSTANBUL TEKNİK ÜNİVERSİTESİ ★ FEN BİLİMLERİ ENSTİTÜSÜ

**PIC VE BİLGİSAYAR DESTEKLİ DİJİTAL ÖLÇME SİSTEMİ
TASARIMI VE UYGULAMASI**

**YÜKSEK LİSANS TEZİ
İbrahim SÖYLER**

Anabilim Dalı : Elektrik Mühendisliği

Programı : Elektrik Mühendisliği

Tez Danışmanı: Doç. Dr. Özcan KALENDERLİ

HAZİRAN 2009

**PIC VE BİLGİSAYAR DESTEKLİ DİJİTAL ÖLÇME SİSTEMİ
TASARIMI VE UYGULAMASI**

**YÜKSEK LİSANS TEZİ
İbrahim SÖYLER
504061032**

**Tezin Enstitüye Verildiği Tarih : 4 Mayıs 2009
Tezin Savunulduğu Tarih : 4 Haziran 2009**

Tez Danışmanı : Doç. Dr. Özcan KALENDERLİ

Diğer Jüri Üyeleri: Y. Doç. Dr. Levent OVACIK (İTÜ)

Y. Doç. Dr. Serhat İKİZOĞLU (İTÜ)

HAZİRAN 2009

ÖNSÖZ

Bu tezin konusu belirlenirken, pratik bir uygulama içermesi ve endüstride uygulanabilir olmasına özen gösterilmiştir. Uzunca bir araştırma sürecinden sonra USB ile bilgisayara bağlanabilen bir enerji analizörü ve aynı zamanda bir osiloskop görevi görecektir bir cihaz tasarımı ve uygulaması yapılmasına karar verilmiştir. USB-osiloskop veya USB-enerji analizörü tasarımının hem elektronik donanım hem de yazılım gerektirmesi, bu konunun seçilmesinde en büyük etken olmuştur.

Tasarlanan cihazdaki her bir özellik farklı bir uygulama niteliğinde olup gereksinime göre uyarlanabilir olması amaçlanmıştır. Bu nedenle, kullanılan malzemeler ayrıntılı bir şekilde tanıtılmıştır. Böylece bu tezden yararlanacak kişiler, ilgilendikleri özellikleri geliştirip daha kapsamlı tasarım yapabileceklerdir.

Lisans ve yüksek lisans öğrenimim boyunca bana yol gösteren her zaman olduğu gibi tezimi hazırlarken de yardımlarını esirgemeyen değerli hocam Doç. Dr. Özcan Kalenderli 'ye sonsuz teşekkürlerimi sunarım. Ayrıca yetişmemde emeği bulunan aileme ve hocalarıma teşekkürü bir borç bilirim.

Haziran 2009

İbrahim SÖYLER

İÇİNDEKİLER

	<u>Sayfa</u>
KISALTMALAR	vii
ÇİZELGE LİSTESİ.....	ix
ŞEKİL LİSTESİ.....	xi
ÖZET.....	xiii
SUMMARY	xv
1. GİRİŞ	1
1.1 Amaç	2
1.2 Osiloskop Nedir?.....	3
1.2.1 Analog Osiloskobun Çalışması	3
1.2.2 Dijital Osiloskobun Çalışması.....	5
2. ÖLÇME SİSTEMİ TASARIMI	7
2.1 Evrensel Seri Yolu, USB	7
2.1.1 USB'nin Tarihçesi.....	8
2.1.2 Sürümleri.....	9
2.1.3 Terimler.....	11
2.1.4 USB'nin Sunduğu Özellikler	12
2.1.4.1 Cihaz Tespiti	12
2.1.4.2 Veri Akışının Yönetimi.....	12
2.1.4.3 Hata Denetimi	13
2.1.4.4 Güç Sağlama	13
2.1.4.5 Çevrebirimleriyle Veri Alışverişi.....	13
2.1.5 Çevrebiriminin Görevleri	14
2.1.6 Yongayı Hedefleyen İletişimin Saptanması.....	14
2.1.7 Standart İsteklerin Yanıtlanması	14
2.1.8 Hata Denetimi	15
2.1.9 Güç Yönetimi	15
2.1.10 Karargâhla Veri Alışverişi	15
2.1.11 Hız	16
2.1.12 USB Protokolü	17
2.1.13 Veri Aktarım Yöntemleri	18
2.1.13.1 Kontrol Transferi.....	18
2.1.13.2 Yığın Transferi	19
2.1.13.3 Kesme Transferi	19
2.1.13.4 Asenkron Transfer.....	19
2.1.14 USB'nin Sağladığı Avantajlar	20
2.1.14.1 Kullanıcıya Sağladığı Avantajlar	20
2.1.14.2 Tasarımcıya Sağladığı Avantajlar	22
2.1.15 USB Kullanımının Dezavantajları	24
2.1.15.1 Kullanıcıya Yönelik Sorunlar	24
2.1.15.2 Tasarımcıya Yönelik Sorunlar	26
2.1.16 Bedeller	26
2.2 PIC Mikrodenetleyiciler ve Kontrol Sistemi.....	27
2.2.1 PIC Mikrodenetleyiciler.....	27

2.2.1.1 PIC Mikrodenetleyicilerin Tercih Sebepleri	27
2.2.1.2 PIC Mikrodenetleyicilerin Kullanımı İçin Gerekli Aşamalar	27
2.2.1.3 PIC Mikrodenetleyicilerinin İç Yapısı	29
2.2.2 PIC 18F2550	29
2.2.2.1 PIC 18F2550'nin Özellikleri	30
2.2.2.2 Bellek organizasyonu	34
2.2.2.3 Saat ölçüm şeması/komut süreci	36
2.2.2.4 Kayıt Dosyaları	36
2.2.2.5 Timer Modülü	39
2.2.2.6 Prescaler (Bölücü)	40
2.2.2.7 EEPROM Veri Belleği	40
2.2.2.8 Osilatör Türleri	42
2.2.2.9 USB	43
2.2.2.10 Elektriksel Özellikleri	45
2.3 Bilgisayar Yazılımı	46
3. DEVRENİN OLUŞTURULMASI	49
3.1 Gerilim Bölücü ve Aşırı Gerilim Koruması	49
3.2 PIC Mikrokontrolör ve Program Yükleme	50
3.3 PIC18F2550 İçin Yazılan Kaynak Dosyası	51
3.4 PIC-PC İletişim Arayüzü Yazılımı	51
3.5 USB Bağlantısı	52
4. ÜRETİM MALİYETİ.....	53
5. SONUÇLAR	55
5.1 Gerçekleştirilen USB Osiloskobun Özellikleri	59
KAYNAKLAR.....	61
EKLER.....	63

KISALTMALAR

PC	: Personal Computer (Bilgisayar)
PIC	: Peripheral Interface Controller (Giriş-Çıkış Mikroişlemcisi)
USB	: Universal Serial Bus (Evrensel Seri Yolu)
GUI	: Graphical User Interface (Grafik Kullanıcı Arayüzü)
ADC	: Analog to Digital Converter (Analog/Dijital Dönüştürücü)
CRT	: Cathode Ray Tube (Katot Işınlı Tüp)
GPB	: General Purpose Interface Bus (Genel Amaçlı Arabirim)
ALU	: Arithmetic Logic Unit (Aritmetik Mantık Birimi)
RAM	: Read Access Memory
EEPROM	: Electronic Erasable Programmable Read Only Memory
PLL	: Phase Locked Loop
ICSP™	: In-Circuit Serial Programming™
MSSP	: Main Synchronous Serial Port
SFR	: Specific Function Register
CCP	: Capture/Compare/PWM
USART	: Universal Synchronous Asynchronous Receiver Transmitter
NRZI	: Non-Return-to-Zero Inverted

ÇİZELGE LİSTESİ

Çizelge 2.1 : Donanımların karşılaştırılması	10
Çizelge 2.2 : Veri akışı yöntemlerinin özellikleri.....	20
Çizelge 2.3 : PIC 18F2550'nin özellikleri	33
Çizelge 2.4 : PortA Kaydedicileri.....	39
Çizelge 2.5 : Kristal osilatör için kapasite seçimi.....	42
Çizelge 2.6 : Seramik osilatör için kapasite seçimi.	43
Çizelge 3.1 : Örnekleme frekansı ve hata oranları.	51
Çizelge 3.2 : 20 MHz osilatör frekansı ve PLL÷5 ile CPU frekansı.	51
Çizelge 4.1 : Yapılan harcamalar.....	53

ŞEKİL LİSTESİ

Şekil 1.1	: Enerji analizörü (fiyatı \$6600).	1
Şekil 1.2	: 20 MHz çift kanallı CRT osiloskop (fiyatı \$800).....	2
Şekil 1.3	: 150 MHz analog/dijital CRT osiloskop (fiyatı \$3150).....	2
Şekil 1.4	: Analog osiloskobun yapısı.	4
Şekil 1.5	: Katot ışınlu tüp örnekleri.	4
Şekil 1.6	: Dijital osiloskobun çalışması.....	5
Şekil 2.1	: Verinin paketlere ayrılıp işlenmesi (Insam, 2002).	18
Şekil 2.2	: Temel PIC blok diyagramı.	29
Şekil 2.3	: PIC18F2550'nin Pin diyagramı.....	30
Şekil 2.4	: PIC18F2550'nin özellik blok diyagramı.	30
Şekil 2.5	: PIC 18F2550'nin blok diyagramı.	34
Şekil 2.6	: PIC 18F2550'nin program belleği.	35
Şekil 2.7	: Komut yürütme süreci.	36
Şekil 2.8	: Genel amaçlı kayıt dosyası temel yapısı.....	36
Şekil 2.9	: Özel fonksiyon kayıtları.	37
Şekil 2.10	: Genel giriş çıkış uygulaması.	38
Şekil 2.11	: 10 Bitlik A/D girişleri.....	38
Şekil 2.12	: 8-Bitlik sayaç durumu.	39
Şekil 2.13	: 16-Bitlik sayaç durumu.	40
Şekil 2.14	: Verilen saat değişimlerinde prescaler'ın değişimi.....	40
Şekil 2.15	: Osilatörün devre montaj şeması.	42
Şekil 2.16	: USB bağlantısı temel yapısı.	43
Şekil 2.17	: USB RAM.	44
Şekil 2.18	: Yalnızca BUS gücü modu.	44
Şekil 2.19	: Yalnızca kendi gücü modu.	45
Şekil 2.20	: İkili güç modu.....	45
Şekil 2.21	: PIC18F2550'nin çalışma aralığı.....	46
Şekil 3.1	: Ölçüm cihazının temel devre yapısı.	49
Şekil 3.2	: PIC18F2550.....	50
Şekil 3.3	: PICKit2 PIC Programlayıcı.	50
Şekil 3.4	: Geliştirilen programın Visual Basic proje penceresi.....	51
Şekil 5.1	: Devrenin gerçekleşmiş hali.	55
Şekil 5.2	: Baskı devre boyutları.....	55
Şekil 5.3	: Baskı devre ve elemanları.....	56
Şekil 5.4	: Programın masaüstü kısayol ikonu.....	56
Şekil 5.5	: Cihazın bağlı olmadığı konumda yazılım görünümü.	57
Şekil 5.6	: Cihaz bağlandığı konumda yazılım görünümü.....	58
Şekil 5.7	: 1,2 V değerinde 1 adet AAA pil ile deney.	58
Şekil 5.8	: 20 kHz 5 V kare dalga.	59
Şekil A.1.1	: 1.2V değerinde 2 adet AAA pil ile deney.....	65
Şekil A.1.2	: 23 kHz 5 V kare dalga.	65
Şekil A.1.3	: 25 kHz 5 V kare dalga.	66
Şekil A.1.4	: ADC verisinin alınabilecek en hızlı durumda alınması.....	66
Şekil A.1.5	: Sürekli okuma durumu.	67

PIC VE BİLGİSAYAR DESTEKLİ DİJİTAL ÖLÇME SİSTEMİ TASARIMI VE UYGULAMASI

ÖZET

Bu çalışmada hem geleneksel osiloskop işlevi görecekt hem de enerji analizörü işlevini yerine getirebilecek çok işlevli bir cihaz üretmek ve aynı zamanda bu cihazın ucuz, daha hızlı ve kullanıcıya daha fazla kontrol olanağı sağlıyor olması amaçlanmıştır. Cihazın fiyatının emsallerine göre çok düşük olmasının sebebi işin büyük bir kısmını (kontrol ve görüntü) bilgisayarın yapmasından kaynaklanmaktadır. Üretilen cihazın en temel kısıtlaması ise kullanılan veri edinme birimi PIC18F2550'nin en çok 48 MHz'e kadar veri aktarımına izin vermesidir. Ancak bilgisayara USB yolu ile bağlandığı için yine de emsallerine göre yeterince hızlıdır. Ayrıca geleneksel osiloskoplarda olmayan yazdırma ve kaydetme özelliklerinin yanı sıra donanımı değiştirmeden yalnızca yazılım kısmını değiştirerek farklı kontroller ve işlevler eklenebilir. Microsoft Visual Studio program grubu içindeki Visual Basic programı ile oluşturulan arayüz, gereksinime göre uyarlanarak istenilen herhangi bir elektriksel büyüklük 0-5V aralığına getirilerek ölçülebilir.

Cihaz, yazılan arayüz ile her bilgisayar tarafından hiçbir sürücü program gerektirmeden kolayca tanınır ve çalıştırılabilir. Tüm enerji analizörlerinin standart olarak ölçebildiği elektriksel büyüklükleri (akım, gerilim, güç, reaktif güç, $\cos\phi$ ve harmonik... vb. değerleri gibi) ölçer ve yazılan program yardımıyla görsel hale getirir. Yalnızca kibrit kutusu büyüklüğünde bir devre, bir bilgisayar ve 0,5 metre ölçüm kablosu ile her yerde akım, gerilim dalga şekilleri görülebilir ve enerji kalitesi ölçümü yapılabilir.

DESIGN AND APPLICATION OF DIGITAL MEASUREMENT SYSTEM BASED ON PIC AND COMPUTER

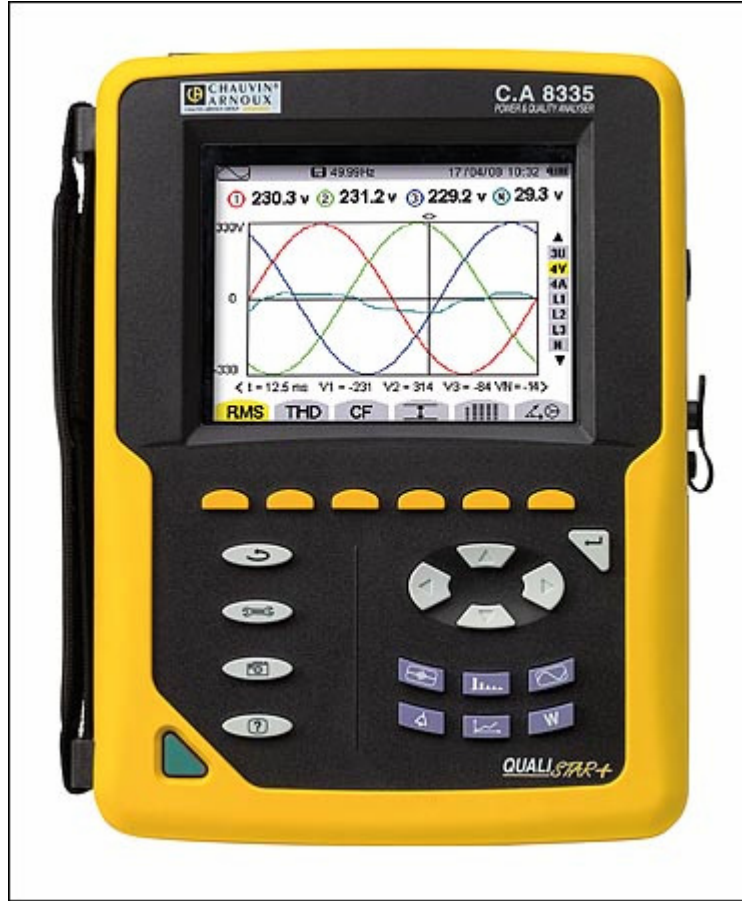
SUMMARY

This project attempts to achieve a multi-functional device having the same functionality as a traditional oscilloscope and also the same functionality as an energy analyser. The objective was to design and build a low cost, high performance and provide more control functions to users. The device cost was cheaper than others because of utilising a personal computer to provide both the display functions and the majority of the control functions. One of the major limitations of USB Oscilloscope is PIC18F2550, which is the data acquisition unit of the device. The maximum value of PIC18F2550 frequency is 48 MHz. It is also as fast as other PC based oscilloscope. Moreover with the application software have additional features not present on a traditional oscilloscope (e.g. printing / saving waveforms). With greater flexibility as additional features can be added as they developed without the need for new hardware. The software of the device was written in Visual Basic Program, which is a part of the Microsoft Visual Studio, can be changed by the user through the needs and then all electrical signals between 0-5 V can be measured by the device.

The device can work on all computers and do not want any driver program to load on the PC. All energy analysers measurements (like current, voltage, power, reactive power, $\cos\phi$, harmonics etc.) can be measured by this device and can be seen on the PC. Everywhere on the world if you have only like a matchbox dimensions circuit, one PC and 0.5 meter measurement cable, you can see current, voltage wave and measure energy quality.

1. GİRİŞ

Enerji analizörleri kullanılan şebekeye ait akım, gerilim, güç, güç faktörü, aktif ve reaktif enerjilerin izlenmesi, kompanzasyon ihtiyacının ve aşırı kompanzasyonun saptanması, endüktif ve kapasitif yüklenmelerin izlenmesi gibi ölçümler yapabilmektedir (Şekil 1.1). Ancak pahalı olmaları sebebiyle profesyonel kullanıcılar dışındaki kullanıcılar bu tür analizörleri edinememektedirler.



Şekil 1.1 : Enerji analizörü (fiyatı \$6600).

Osiloskoplar ise genellikle katot ışınlu tüpler (CRT, cathode ray tube) olup gerilim değişimlerini göstermek üzere tasarlanmıştır (Şekil 1.2). Ancak çok büyük boyutlu olmaları, pahalı olmaları ve düşük frekanslardaki dalga şekillerini göstermedeki zorlukları sebebiyle kullanıcılar yeni arayışlar içerisine girmişlerdir (Şekil 1.3).



Şekil 1.2 : 20 MHz çift kanallı CRT osiloskop (fiyatı \$800).



Şekil 1.3 : 150 MHz analog/dijital CRT osiloskop (fiyatı \$3150).

1.1 Amaç

Bu tezde, bilgisayarın özelliklerinden alınabilecek en çok yararı sağlayarak dijital bir ölçüm cihazı geliştirmek amaçlanmaktadır.

Bilgisayarın;

- Daha geniş ekranı sayesinde dalga şeklini görmekte rahatlık sağlaması,
- Tüm Windows ve grafik arayüzünden sağlanan özelliklerden yararlanarak kopyala, kes, yapıştır gibi basit komutlarla rahatlık sağlaması (GUI, Grafical User Interface),
- Veri kaydı yapılabilmesi,
- Uzak masaüstü bağlantısı gibi bir özellik ile dünyanın her yerinden erişilebilmesi,
- Yazılımın arayüzünün değiştirilebilir ve güncellenebilir olması gibi özellikleri sayesinde kullanıcıya hem uygun fiyat hem de büyük kullanım kolaylığı sağlamaktadır.

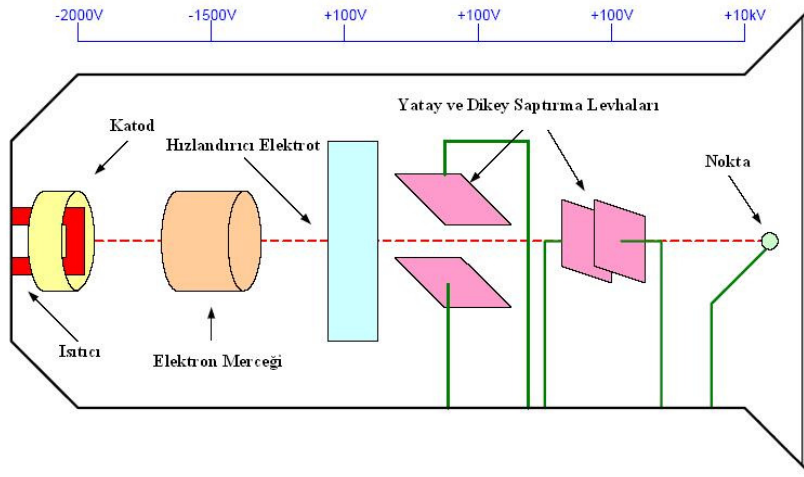
PC osiloskop farklı şekillerde yapılabilir. Örneğin, ses kartı kullanılarak elektriksel işaret, ses dalgaları şeklinde ölçülür ve kaydedilen ses dalgaları bir arayüz yardımıyla grafik şeklinde görüntülenir. Ancak bu sistem düşük frekanslarda ölçüm yapabilmektedir. En yüksek 88 kHz frekanslı işaret ölçümüne izin vermektedir. Frekansı sınırlayan birim analog bilgiyi dijitale çevirecek olan arabirimdir. Bu yüzden kullanılan arabirimin frekansı ne kadar yüksekse üretilen PC osiloskop da o kadar yüksek frekanslarda ölçüm yapabilecektir.

1.2 Osiloskop Nedir?

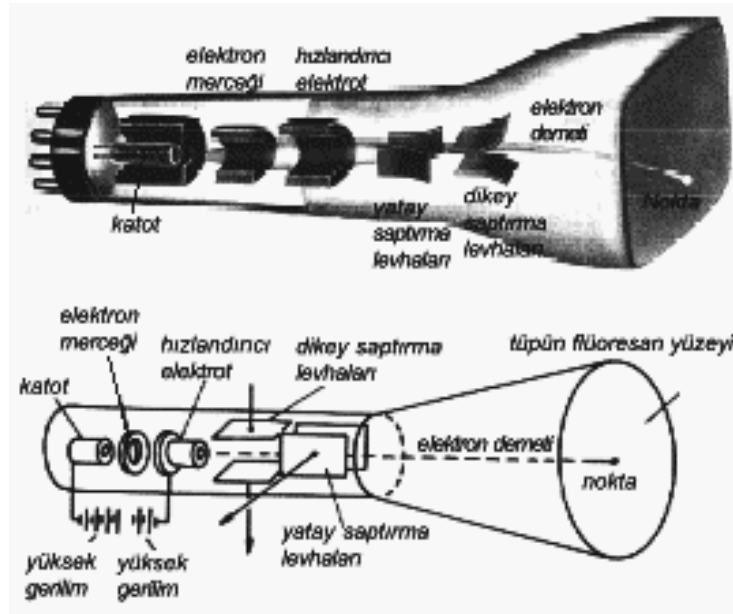
Elektriksel işaretlerin ölçülüp değerlendirilmesinde kullanılan cihazlar içinde en geniş ölçüm olanaklarına sahip olan cihazdır. İşaretin dalga şeklinin, frekansının ve genliğinin aynı anda belirlenebilmesini sağlar. Genel hatlarıyla küçük bir televizyona benzer fakat ekranında çizgiler vardır ve televizyondan daha fazla kontrol mekanizması vardır. Ön paneli yatay, dikey, tetikleme bölümlerinin yanı sıra ekran kontrolü ve giriş bağlantı noktaları gibi bölümlere ayrılmıştır. Osiloskoplar günümüzde en basit televizyondan en karmaşık endüstriyel elektronik devrelerine kadar çok geniş bir alanda kullanılmaktadır. Özellikle araştırma geliştirme ve arızanın tespiti gibi konularda yarar sağlamaktadır. Osiloskoplar yalnızca gerilim ölçmezler, dönüştürücüler (transdüserler) yardımıyla diğer elektriksel büyüklüklerin ölçülmesine de olanak tanırırlar. Günümüzde osiloskoplar analog ve dijital olmak üzere iki çeşittir.

1.2.1 Analog Osiloskobun Çalışması

Hareket halindeki elektronların yörüngelerinin bir elektrik alan içerisinde geçerken sapmaları temel ilkesine dayanır. Osiloskobun temel yapısı Şekil 1.4'te görülmektedir. Katot ışın tüpündeki saptırma plakaları adı verilen düzlemsel levhalara uygun potansiyel farkı yaratacak gerilimler uygulanarak oluşturulan elektrik alanlar, plakalar arasından geçen elektronları (elektron demetini) saptırarak fosfor ekrana çarptığı noktanın yerini değiştirir. Bu noktanın konumu saptırma plakalarına uygulanan gerilimin ani değeri ve dalga şekline bağlı olarak değişecek ve ekranda ışıklı bir çizgi oluşacaktır. Böylelikle dalga şeklinin ölçümü Şekil 1.5'te de gösterildiği gibi gerçekleşecektir.



Şekil 1.4 : Analog osiloskobun yapısı.

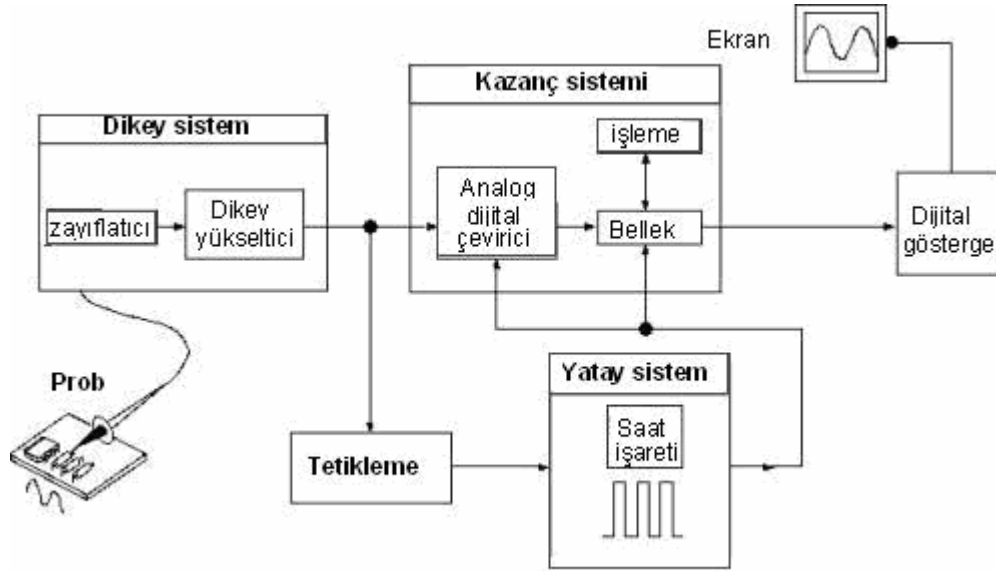


Şekil 1.5 : Katot ışınlı tüp örnekleri.

Osiloskop devreye daima paralel bağlanır. Çok yüksek olan iç direnci nedeniyle seri bağlanması halinde ölçüm yapılmak istenen devreden akım akmasını engelleyecektir. Akım dalga şekillerini incelemek için akımın aktığı devreye küçük değerli bir direnç (ölçüm direnci, şönt direnç) seri bağlanarak uçlarında düşen gerilimin dalga şekli incelenir. Bir omik dirençte içinden akan akım ve uçlarında düşen gerilimin dalga şekilleri, fazlarının aynı oldukları ve ohm yasası gereği $V = I.R$ bağıntısı da göz önünde tutularak incelenir. Dikkat edilmesi gereken nokta, kullanılan direncin değerinin devre akımını çok fazla sınırlamayacak kadar küçük seçilmesi (genellikle akıma bağlı olarak 10 ile 200 miliohm arası) ve gücünün bu akıma dayanabilecek kadar büyük olmasıdır.

1.2.2 Dijital Osiloskobun Çalışması

Dijital osiloskop da tıpkı analog osiloskop gibi analog dalga şeklini alır. Ancak daha sonra bir analog dijital çevirici (ADC Analog to Digital Converter) yardımıyla dalga şeklini dijital olarak ölçer ve gösterir. Dijital osiloskopların temel yapısı Şekil 1.6’da görülmektedir.



Şekil 1.6 : Dijital osiloskobun çalışması.

Dijital osiloskoplar görülmesi güç ve sadece bir defa oluşabilecek dalga şekillerini kaydedebilme özelliği sayesinde görebilme yeteneği sağlar. USB, RS232, LAN... gibi bağlantı yollarıyla bilgisayara bağlanarak yüksek frekanslardaki ölçümlere izin vermektedirler. Dijital osiloskoplar bilgisayar ortamında işleme tabi tutulabilme, daha sonra tekrar görülebilme, yazdırılabilme ve boyut olarak daha küçük olmaları gibi özellikleri sebebiyle analog osiloskopların yerini almaya başlamışlardır.

2. ÖLÇME SİSTEMİ TASARIMI

2.1 Evrensel Seri Yolu, USB

Bilgisayar donanımı söz konusu olduğunda işe sıfırdan başlamak pek mümkün değildir. Yeni bir şeyin kendinden öncekilerle uyumlu olması gerekir. Bu durum hem bilgisayarlar, hem de onlara bağlı çevre birimleri açısından böyledir. Devrim sayılabilecek nitelikteki bir cihaz bile bağlanacağı bilgisayarın desteklediği bir arabirime gereksinim duyar. Bir arabirimi en baştan tasarlama şansına sahip olunsaydı, böyle bir cihazda bulunmasını istenebilecek nitelikler aşağıdaki gibi olurdu:

- Yapılandırmayı ve kurulumu sorun olmaktan çıkaracak ölçüde kolay kullanılabilir;
- İletişimde darboğaza yol açmayacak kadar hızlı;
- Hataları azaltıp giderilmelerini otomatikleştirecek kadar güvenilir;
- Farklı çevrebirimlerine hizmet verebilecek düzeyde esnek;
- Kullanıcıyı ve cihazı ürünlerine takmayı planlayan üretici için yeterli düzeyde ucuz;
- Taşınabilir bilgisayarların pilini koruyacak şekilde enerji tasarruflu;
- Çevrebirimlerine ilişkin alt düzey sürücülerin yazımında ek zorlukları ortadan kaldıran işletim sistemi desteğine sahip olmalıdır.

Evrensel seri yolunun (Universal Serial Bus, USB) geliştirilmesiyle, kimsenin sıfırdan böyle bir cihazı geliştirme çabasına girmesine gerek kalmamıştır. USB, çok çeşitli çevrebirimleriyle en verimli haberleşebilecek şekilde tasarlanmıştır. Mevcut arabirimlerin sınırlamalarından arınmıştır. Yeni üretilen PC'lerde birden çok USB kapısı (portu) yer alır. Klavye, fare, tarayıcı, harici disk, yazıcı vb standart ve belli başlı tüm donanımlar bu kapı üzerinden PC'ye bağlanabilir. Ancak asıl sorun USB'li çevrebirimlerinin ve USB üzerinden çalışacak programların tasarımında ortaya çıkmaktadır. USB'nin kullanım kolaylığı sağlayan arabirimi, diğer tüm arabirimlerle kıyaslandığında oldukça karmaşık bir yapıya sahiptir. Yeni olması ise kullanıcılar

açısından bir diğer olumsuzluktur, örneğin, yaygın kullanımı olan tüm çevrebirimleri için gereken cihaz sürücüler henüz Windows'ta mevcut değildir. Standartları belirlenmediği sürece, protokol analizörleri ile diğer geliştirme araçlarının kullanılmasına olanak yoktur.

2.1.1 USB'nin Tarihçesi

Mevcut arabirimlerle çalışmanın getirdiği rahatlık, yeni arabirimlerin devreye girmesinin önündeki engellerin başında gelir. Harcanacak zaman ve para ise hemen arkasında yer alır. Ne var ki, bilgisayarların gelişmesi ve çevrebirimlerin çeşitlenmesi, mevcut arabirimleri, gelişimin önünde duran engellere çevirmeye başlamıştır. Evrensel seri yolu (Universal Serial Bus, USB), dış donanımların bilgisayar ile bağlantı kurabilmesini sağlayan zaman paylaşımli seri yapıli veri akışı biçimidir. Tak ve çalıştır (plug and play) özelliğinden ve yalnızca dört telli olan basit yapısı sebebiyle birçok cihazın bağlantısında kullanılmaktadır. Bu dört telin ikisi besleme ve topraklama diğer ikisi ise veri akışı için kullanılmaktadır. Aslında USB iki telli seri bağlantı şeklinde olup her kablo saniyede 1,5 ila 12 Mega bit veri aktarabilecek kapasitededir. Hem yüksek hızlı veri hem de düşük hızlı veri aktarımı aynı tel üzerinden yapılabilir. Tek bir USB çıkışından çoklayıcılar yardımıyla 127 adet çevre birimi bağlantısı sağlanabilir. Çevre birimleri takıldıkları anda sistem tarafından otomatik olarak algılanır ve düzenlenirler.

Değişimin masrafı ve uyumsuzluklar gölgelenmeye başlayınca, USB'ye yönelik arayışlar giderek güçlendi. Sonuçta her bilgisayarın standart cihazlarla iletişimini sağlayacak incelikli bir arabirim ortaya çıktı. Önceleri bu işle sadece HP (Hewlett Packard) uğraşıyordu. Laboratuvar teçhizatına yönelik, genel amaçlı arabirim veriyolu (GPIB, General Purpose Interface Bus) olarak bilinen, HP arabirim veriyolu geliştirildi. Daha sonra Centronics adıyla tanınan popüler yazıcı arabirimi devreye girdi. Ancak, tek bir firmanın kontrolünde bulunan bir arabirimin ideal olmadığı açıktır. Lisans bedelleri, erişim engelleri istenmeyen durumlardır. Üreticiler böylesi sorunlardan korunmak için ortak çıkarlar üzerinde işbirliğine giderek sonuca varmaya çalışırlar. Örneğin, IEEE veya TIA gibi örgütler temelinde kurulan komitelerde standartlar hazırlanır ve sonuçlar düzenli olarak yayınlanır. Nitekim IEEE-1284 standardı Centronics arabiriminden devşirilmiştir. IEEE-488'in kökeni ise GPIB'dir.

USB açısından durum ufak bir farklılık gösterir. Standardı belirlemek üzere bir örgüt kurulmuştur. USB 2.0 standardının hakları yedi işletmeye aittir: Compaq, HP, Intel, Lucent, Microsoft, NEC ve Philips. Tümü de bu özelliğin bedelsiz sunumu konusunda uzlaşmışlardır.

İlk spesifikasyon I²C senkron seri yoldan türetilen ACCESS veri yoluydu. Bu elektrik arabirimi farklı olmakla birlikte bir çok yönden USB'ye benzemekteydi. Tasarımındaki amaç klavye işaretçi ve 100 kilobit/saniye hızlı cihazlar için arabirim oluşturmaktı. Veriyolu 125 cihaza ve 125 metre mesafeye olanak veriyordu. Kolay takılabilme, sınıf tanımlama gibi olanaklara sahipti. USB'den başlıca farkı bir veri bir de saat hattı olan açık-kollektör sürücülerıyla çalışmasıydı. Ancak, PC'lerle çalışma fırsatı olmadı.

2.1.2 Sürümleri

Ocak 1996'da USB 1.0'ın, Eylül 1998'de USB 1.1'in sürümleri yapıldı. USB 1.1'de Kesme OUT transfer tipi eklenmiştir. 1.x ifadesi 1.0 ve 1.1'i anlatır. Nisan 2000'de de USB 2.0'ın sürümü yapıldı. Yüksek hız bu sürümde eklenmiştir. Aralık 2000'de bazı düzeltmelere ek olarak bir mini-B konnektörü tanımlanmıştır. Spesifikasyonun yayınlanması ürünlere kıyasla bir parça gecikmeli gerçekleşmiştir. Fakat USB'nin işlek sürümleri son kullanıcıyla buluşmamıştır. Windows 95'in kurulu olduğu PC'lerde yer alıyordu. Yaygınlaşması ise Windows 98 ile birlikte. Windows 98'in ilk sürümü Gold, ikinci sürümü ise SE olarak bilinir. Windows NT'de ise USB desteği yoktur. Ancak Windows 2000'de vardır, Windows ME 'de USB'yi destekler. Genel olarak, daha dengeli ve ticari kullanıma uygun sürümünün Windows 2000 olduğu söylenebilir. Windows 98 ve Me ise evde kullanıma yöneliktirler. Arkadan gelen Windows XP ise, 2000'nin ev kullanımına yönelik bir türevidir. 98'den itibaren tüm Windows'ları ikame etmeyi amaçlamaktadır.

USB 2.0'ın büyük bir adım olduğunu söylenebilir. Burada eklenen başlıca özellik, daha hızlı transferlere yönelik destektir. USB 1.1 ile uyumludur ve 480 megabit/s hıza kadar desteklemektedir. USB 2.0'ın kablo ve konnektörleri 1.x çevre birimlerinininkilerle aynıdır. Çevre birimlerinin yüksek hızı kullanmaları için USB 2.0 uyumlu karargâh ve hub'a bağlanmaları şarttır. USB 2.0 karargâh ve hub'lar 1.x cihazlarla haberleşebilirler.

Yeni çevre birimleri için bir diğer seçenek IEEE-1394'tür. Arabirimin Apple Computer kökenli adı Firewire'dir. USB'den daha hızlı ve daha esnektir. Ancak daha pahalıdır. Video gibi hızın gerekli olduğu ve karargâh PC'nin olmadığı durumlarda en uygun cihazdır. USB haberleşmesinde bir karargâh ile çeşitli çevrebirimleri bulunur. İşin büyük kısmını karargâh yürütür. Çevre birimlerinin elektronik aksamı nispeten daha ucuz ve basittir. IEEE-1394 cihazdan cihaza doğrudan haberleşmeyi sağlar. Çizelge 2.1 donanımları kıyaslama için yeterli bilgiyi vermektedir.

Çizelge 2.1 : Donanımların karşılaştırılması

Arabirim	Format	Cihaz Sayısı	Uzunluk-feet	Hız-bit/saniye	Kullanım
USB	Asenkron seri	127	16 (5 hubla 96)	1,5M, 12M, 480M	Fare, klavye, disk sürgüsü, modem, audio
RS-232 (EWTIA-232)	Asenkron seri	2	50-100	20k (donanıma göre 115k olabilir)	Modem, fare, ensirumantasyon
RS-485 (EIA/TIA-485)	Asenkron seri	32 (bazı donanımlarda 256'ya kadar)	4000	10M	Veri toplama ve kontrol sistemi
IrDA	Asenkron seri kızılaltı	2	6	115k	Yazıcı, dizüstü
Microwire	Senkron seri	8	10	2M	Mikrokontrolör haberleşmesi
SPI	Senkron seri	8	10	2.1M	Mikrokontrolör haberleşmesi
I ² C	Senkron seri	40	18	3.4M	Mikrokontrolör haberleşmesi
IEEE-1394 (FireWire)	Seri	64	15	400M (IEEE1394 bile 3.2G olabilir)	Video, toplu kayıt
IEEE-488 (GPIB)	Paralel	15	60	6 M	Enstrümantasyon
Ethernet	Seri	1024	1600	10M, 100M, 1G	PC network
MIDI	Seri akım döngüsü	2 (akış modunda daha çok)	50	31.5k	Müzik, görüntü kontrolü
Paralel Yazıcı Portu	Paralel	2 (papatya zinciri ile 8)	10-30	8M	Yazıcı, tarayıcı, disk sürgüleri

2.1.3 Terimler

USB çerçevesinde, bazı günlük sözcüklerin özel anlamları vardır.

Karargâh, arabirimi kontrol eden bilgisayarı anlatır. Yine, *fonksiyon*, *hub* ve *cihaz* bu kapsamda özel anlam taşıyan diğer üç sözcüktür.

Fonksiyon, cihaz olarak tanımlanır ve karargâha belli bir yetenek sağlar. Fare, hoparlör seti ya da veri-alım ünitesi vb gibi çevrebirimlerinin her biri tek ve belli bir işi görür. Bununla birlikte çevrebirimlerinin birden fazla fonksiyon üstlenmesi de mümkündür. Örneğin üzerinde yerleşik bir top (trackball) bulunan klavye, yalnızca USB konnektör üzerinden, hem klavyenin hem de topun fonksiyonlarını yerine getirir.

Hub, USB cihazlarına yönelik bir veya daha fazla sayıda konnektörü ya da iç bağlantıları olan bir cihazdır. Donanım üzerinden diğer birimlerle iletişimi sağlar. Her konnektör USB'nin bir portudur. Hub, gelen USB trafiğini yönlendirir (Buna güç yönetimi de bilgisi dahildir). Status (durum) ve Control (kontrol) mesajlarının alışverişini yürütür. Yavaş cihazlara tam-hızla veri iletilmesini engeller.

Cihaz, bir fonksiyon ya da bir hub olabilir. Her cihazın veriyolu adresi bir tanedir. Ancak bileşik cihazlar bu kapsamda değildir. Çok sayıda fonksiyonu olan çevrebirimleri çok-yönlü (composite) cihazlardır.

USB kapısı, bazı yönleriyle, PC üzerindeki seri ve paralel kapılardan farklıdır. Bilgisayar kapıları, adreslenebilir bağlantı noktalarıdır. Kapı devresinin sonunda bir konnektör yer alır. Bu konnektör, diğer ucunda klavye vb gibi bir çevrebirimi bulunan bir kabloya bağlanır. Bazen çevrebirimi devreleri porta sabitlenmiş halde de olabilmektedir. Port devrelerinin izleme ve kontrolü, port adresi üzerinden yapılan okuma/yazma işlemleriyle, bilgisayardaki yazılım üzerinden gerçekleşir. Bellekler de adreslenebilir yerleşimler olmakla birlikte, bunlara erişimde daha farklı makine dili talimatları kullanılır. PC'lerde, bellek adreslerinin tamamı yalnızca sistemin veriyoluna bağlıdır, çevrebirim devreleriyle bağlantıları yoktur.

USB'deki başlıca fark, veriyolu üzerindeki tüm portların karargâha giden tek bir yolu paylaşmalarıdır. RS-232'nin portları bağımsızdır. Örneğin iki portu varsa, her birinin kendine ait bir veri yolu vardır ve sadece kendi verisini taşır. Aynı anda her iki portta veri alışverişi yapmak olanaklıdır.

USB'de durum farklıdır. Normalde bir PC'de en az iki adet USB konnektörü yer alır. Bunlar veri yolunu ana denetleyici ile paylaşırlar. Her konnektör, bir USB portunu temsil eder. Ancak, RS-232'den farklı olarak, zaman cihazlar tarafından paylaşılır. Ana hub'a bağlı ek hublar ve başka cihazlar varsa, bunlar da aynı zamanı paylaşmak durumundadırlar. Dolayısıyla her birinin kendi konnektörü ve kablosu bulunan çok sayıda port olmakla birlikte yalnızca bir tane veri yolu mevcuttur ve belli bir anda yalnızca bir cihaz iletim yapabilir. Veri yolunu paylaşan başka cihazlar da bulunmaktadır. Bunlar Firewire ve SCSI cihazlarıdır.

2.1.4 USB'nin Sunduğu Özellikler

Karargâh, yani merkez bilgisayar, veriyolundan sorumludur. Veriyolunda hangi cihazın olduğunu ve bu cihazın kapasitesini bilmek karargâhın işidir. Dahası, veriyolundaki cihazların yapmaları gereken veri alışverişini de güvenceye almak zorundadır. Veriyolunda her biri farklı ihtiyaçları olan çok çeşitli cihazlar yer alır ve bunların hepsi aynı anda veri aktarmak isteyebilir. Yani karargâhın işi hafife alınacak türden değildir. Veriyolu, Windows'un USB'ye ve ana denetleyici donanımına yönelik desteği sayesinde idare edilmektedir. Karargâha bağlanan her USB cihaz için bir sürücü (uygulamaların cihazla iletişimini sağlayan bir kod) vardır. Bazı çevrebirimlerinin sürücüleri Windows'ta mevcut iken, bazı cihazların sürücüleri özeldir. Ana denetleyiciyle cihaz arasındaki haberleşme ise sistem düzeyindeki diğer yazılım elemanlarınca yönetilirler. Karargâhın desteğe gerek duymadan ya da çok az destekle gördüğü işler de vardır. Bu görevler şunlardır:

2.1.4.1 Cihaz Tespiti

Bilgisayarın açılışında hub'lar takılı cihazları bildirirler. Karargâh, sırayla adlandırma (enumeration) denilen bir işlemle cihazlara bir adres tayin ederek her birinden ek bilgi isteğinde bulunur. Bilgisayar açıkken bir cihazın takılması (veya çıkartılması) halinde karargâh durumu öğrenir ve cihazı adlar sıralamasına alır (veya adlar sıralamasından çıkartır).

2.1.4.2 Veri Akışının Yönetimi

Veriyolundaki veri akışını karargâh yönetir. Birkaç cihazın aynı anda veri transferine girişmesi durumunda karargâh veri yolunu 1 milisaniyelik çerçevelere böler ve her etime bundan bir pay verir. Mutlaka belli bir hızda gerçekleşmesi gereken

transferlerin, her çerçevede belli bir zaman ihtiyaçları vardır. Sırayla adlandırma yapılırken, zamanlama garantisi gerektiren transferler için ihtiyaç duyulan bant genişliği cihazın sürücüsü tarafından istenilir. İstenilen bant genişliği mevcut değilse, karargâh o cihazla iletişime izin vermeyecektir. Bu durumda sürücü ya daha küçük bir bant genişliği isteyecek ya da istemiş olduğu bant genişliği tahsis edilene kadar bekleyecektir. Garantili bir hız gerektirmeyen transferler çerçevede kalanla yetinirler, dolayısıyla beklemek durumunda kalabilirler.

2.1.4.3 Hata Denetimi

Karargâh, gönderdiği veriye hata-denetim bitleri ekler. Veriyi alan cihaz, veri üzerinde hesaplamalarını yapar ve sonuçları hata-denetim bitleriyle karşılaştırır. Sonuçların uymaması halinde alım onayı göndermez, böylece karargâh yeniden iletim gerektiğini anlar. USB, yeniden iletme izin vermeyen bir transfer türünü de desteklemektedir. Burada, sabit transfer hızının korunması gerekmektedir. Karargâh, cihazdan alım ya da gönderim yapamadığına ilişkin başka hata göstergeleri alması halinde cihazın sürücüsünü bilgilendirir. Sürücü de gereğini yapabilmesi için ilgili uygulamayı uyarır.

2.1.4.4 Güç Sağlama

Bir USB kablosunda iki sinyal hattına ek olarak bir +5 V hattıyla toprak hatları bulunur. Bazı çevre birimleri gereksindikleri gücü bu hatlardan alırlar. Karargâh, tüm cihazlara -açılıştaki veya yeni takılması halinde- güç sağlar. Öte yandan cihazlarla çalışırken mümkün olduğu zaman güç tasarrufuna gider. Veriyolundan beslenen tam güçlü her cihaz 500 miliampere kadar akım çeker. Pille çalışan bir kısım hub ve PC'lerdeki portlar yalnızca -100 miliampere kadar akım çeken- düşük güçlü cihazları desteklerler. Bazı cihazların kendi güç kaynakları vardır. Bunlar bus gücünü sadece açılıştaki iletişimde kullanırlar.

2.1.4.5 Çevrebirimleriyle Veri Alışverişi

Karargâhın asli görevi veri alışverişidir. Buraya kadar sözü edilen işlerse destek görevlerdir. Bazen bir sürücü karargâhtan belli bir cihazı daha önce belirtilen aralıkla (rate) sürekli olarak yoklamasını isteyebilir. Bazen de karargâh bir uygulamanın ya da bir yazılım bileşeninin isteğine göre iletişime geçebilir. İlgili uygulamaya yönelik sorunlar sürücü tarafından bildirilir.

2.1.5 Çevrebiriminin Görevleri

Çevrebiriminin ya da cihazın görevleri birçok yönden karargâhın ayna görüntüsüdür. Karargâh iletişimi başlattığında çevrebirimi yanıtlamak zorundadır. Bununla birlikte kendine has görevleri de vardır.

Cihaz USB iletişimini kendi kendine başlatamaz. Karargâhın iletişimi başlatmasını bekler ve yanıt verir. Uzaktan uyandırma -remote wakeup- özelliği bir istisnadır. Bu özellik cihazın karargâhtan iletişim isteğinde bulunmasına olanak tanır. Cihazdaki USB denetleyicisi USB'nin görevlerinin büyük kısmını otomatik olarak yapar. Cihazın programındaki gerekli destek yongaya bağlı olarak değişir.

2.1.6 Yongayı Hedefleyen İletişimin Saptanması

Veriyolundaki her iletişimde cihazın adresi cihaz tarafından izlenir. Eğer adres cihazın kayıtlı adresine uymuyorsa iletişim isteği görmezden gelinir. Adres kayıtlı olanla aynıysa, cihaz veriyi alıcı tamponuna kaydettikten sonra verinin geldiğini bildirmek üzere bir kesme (interrupt) gönderir. Bu işlem hemen hemen bütün yongalarda otomatiktir, donanımda mevcuttur. Dolayısıyla yonga kendi adresini içeren bir iletişim belirlemediği sürece cihazda yüklü program çalışmaya başlamayacaktır.

2.1.7 Standart İsteklerin Yanıtlanması

Cihaz, bilgisayar açıldığında ya da bir sisteme takıldığında, karargâhın sırayla adlandırma sürecindeki isteklerine yanıt vermek durumundadır. USB cihazlarının yanıtlamaları gereken onbir adet standart istek kodu vardır. Bu kodlar cihazın yetenekleriyle durumunu sorgulamaya ve konfigürasyon seçimine yöneliktir. Cihaz bir istek aldığında, karşılık gelen bilgiyi verici tamponuna yerleştirir. Adres ya da konfigürasyon belirlemesi gibi durumlarda ek bilgiler içeren başka eylemlere de girişebilir. Çevrebirimi her istenileni yapmak zorunda değildir. Ancak, isteğe karşılık gelen anlaşılır bir yanıt vermek zorundadır. Örneğin, karargâh, cihazda bulunmayan bir konfigürasyon isteğini iletmiş olabilir. Burada cihaza düşen, isteğin desteklenmediğine ilişkin yanıtı göndermesidir.

2.1.8 Hata Denetimi

Cihazlar da, karargâh gibi, gönderdikleri veriye hata-denetim bitleri eklerler. Cihaz, hata-denetimli bitlerle gelen veriyi aldıktan sonra hesaplamalarını yapar ve hata söz konusuysa (bazı istisnalarla) yeniden gönderim isteğinde bulunur. Bu fonksiyonların programlanması gerekmez. Çünkü donanımda mevcuttur. Cihaz uygun durumlarda karargâhın -aldığı veriye yanıt olarak- yolladığı onayları da saptamaktadır.

2.1.9 Güç Yönetimi

Enerjisini veriyolundan almayan bir cihaz kendi güç kaynağını kullanmak durumundadır. Veriyolunda bir işlem yoksa cihaz düşük-güçlü askı (suspend) durumuna geçer, Bu durumdayken veriyolunu izlemeye devam eder ve işlem başladığında askı durumundan çıkar.

Karargâhın askı durumuna geçtiği hallerde veriyolu üzerindeki iletişim duraklar. Bu duraklama karargâhın her milisanide yolladığı zamanlama işaretlerini de kapsar. Bağlı cihazlar üç milisaniye boyunca veriyolunda faaliyet algılamazlarsa askıya geçerler ve veriyolundan çektikleri akımı sınırlarlar. Bazen karargâh da belli bir cihazdan askıya geçmesini isteyebilir. Veriyolunda faaliyet başladığında cihazın da askıdan çıkması gerekir.

Uzaktan uyandırma özelliği olmayan cihazlar askı durumundaki veriyolundan 500 mikroamperden fazla akım çekemezler. Bu özellik varsa ve karargâh tarafından da mümkün kılmıyorsa limit 2,5 miliamper olur. Ancak bunlar bir saniye için ortalama değerlerdir, akımın tepe değeri (peak current) daha yüksek olabilir.

2.1.10 Karargâhla Veri Alışverişi

Buraya kadar sözü edilen işler cihazın USB portunun yan görevleridir. Esas görev karargâh ile veri alışverişidir. Konfigürasyon bittikten sonra cihazın veri alım ve gönderimi için tepki vermesi beklenir. Karargâh, cihazı ya düzenli aralıklarla ya da bir uygulamadan gelen iletişim isteğine bağlı olarak yoklar. Karargâhın hangi istekleri hangi sıklıkta yapacağını, cihazı kullanan uygulama, cihazın sürücüsü ve konfigürasyonu birlikte belirlerler. Transferler türlerinin pek çoğunda, cihaz ya veriyi aldığını gösteren bir onay kodu (ACK) ya da veriyle ilgilenemeyecek kadar yoğun olduğunu gösteren bir olumsuz-onay kodu (NAK) göndermek durumundadır. Uygun

yanıt donanım tarafından otomatik olarak gönderilir. Bazı transferlerde ise onay kodları gerekmez. Karargâh böyle durumlarda cihazın veriyi aldığını varsayar. Veriyolundaki verinin formatlanmasına ilişkin ayrıntılar denetleyici yonganın donanımıyla halledilir. Hata-denetimi bitlerinin ilavesi, alınan verideki hataların kontrolü, veriyolundan tek bitlik alım ve gönderimler bu kapsamdadırlar.

Cihaz üstüne düşen her işi yapmak zorundadır. Burada "iş", örneğin bir fare için, hareketi ya da tıklamayı fark etmeye hazır olmak; bir veri alım ünitesi için sensörlerden (algılayıcılardan) gelen veriyi okumak; ya da bir yazıcı için gelen veriyi kâğıt üzerindeki imgelere çevirmek anlamındadır.

2.1.11 Hız

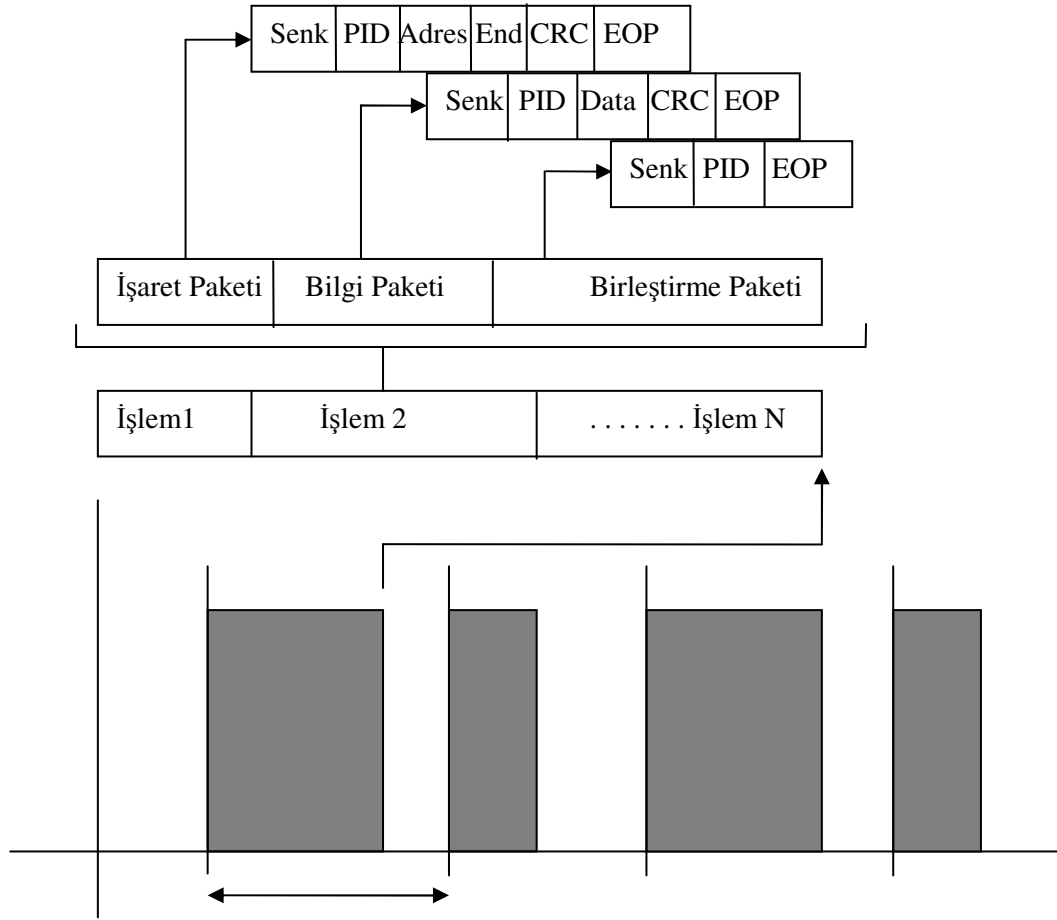
Bir denetleyici yongasının hızı düşük ya da yüksek belirlenebilir. Her iki durumda da devre birimleri her hangi bir USB hub'a bağlanabilmektedir. Kullanıcıların cihazın hızına yönelik hiç bir ayarlamayla uğraşmalarına gerek yoktur. Karargâh ile birim arasındaki fiili veri transfer hızın veriyolunun hızından daha düşüktür. Ancak, hız şu kadardır demek, her zaman mümkün değildir. Çünkü aktarılan bitler bazen veriden çok tanımlamaya, senkronizasyona ve hata-denetimine yöneliktirler, Öte yandan, veri hızının belirleyicileri arasında transferin tipi ve veriyolunun başka transferlerle meşgul olup olmadığı da bulunmaktadır. Zamana-duyarlı veriler söz konusuysen, hızı ya da zaman sınırı (maximum latency) belirli transfer türlerini de destekler. Asenkron (veri gönderim sınırı garantili olan) transferler hızı belirli transferlerdir. Karargâh, alım yahut gönderim için, birimden her 1 milisaniyelik çerçevede 1023'e kadar bir byte değeri isteyebilir. Kesme transferleri ise zaman sınırı belirli transferlerdir. Burada kesin bir hız söz konusu olmamakla birlikte istenilen transferler arasındaki zaman belirli bir değeri aşamaz. Yüksek (tam) hızlı cihazlar için bu süre 1-255 milisaniye arasında, düşük hızlı birimler içinse 10-255 milisaniye arasındadır. Veriyolunun paylaşıyor olmasından dolayı cihaza tahsis edilecek belirli bir hız ya da zaman sınırı garantisi yoktur. Eğer veriyolu istenilen hızı ya da sınırı sağlayamıyorsa, karargâh transferin gerçekleşmesini sağlayacak yapılandırma tamamlanmasına izin vermez. Boş bir veriyolundaki en hızlı transferler yığın transferleridir. Buda teorik olarak saniye 1.216 megabayt aktarıma karşılık gelmektedir. Asenkron transferler için garantili en yüksek hız saniyede 1.023 megabayt'tır. Veriyolunun düşük düzey hızı saniyede 1.5 megabit olmakla birlikte,

tek bir transfer için garantili aktarım 10 milisaniyede 8 bayt ya da saniye 800 bayt'tır. Düşük hızlar genelde ucuz kablolarla ya da basit şemalı devre kartlarına başvurulduğunda söz konusu olurlar. Düşük hızlı yongalar daha ucuz olabilmektedirler, ancak böyle bir kural yoktur.

2.1.12 USB Protokolü

Veri akışı 1 ms boyutundaki zaman aralıklarına bölünmüş pencereler halinde gerçekleşmektedir. Düşük hızda 1,5 Mbit/s uzunluk olarak ise 667 ns'de, yüksek hızda ise 12 Mbit/s ve 83,3 ns uzunlukta çalışmaktadır. USB protokolü her bir pencere başladığında çıkışı eşzamanlı yapar. Bu geçici belleğe sabit veri akışı için gereklidir. Veri akışı ise NRZI (Non-Return-to-Zero Inverted) formatında yani sinyaldeki herhangi bir değişime karşı 0 bit depolanması buna karşın sinyalde bir değişim olmaması ve sabit olması durumunda ise 1 bit depolanıp belleğe kaydedilmesidir. Şekil 2.1'de verinin paketlere ayrılıp zamanı bölerek farklı kaynakların nasıl işleme tabi tutulduğu gösterilmektedir.

Tipik USB örneklerinde birden fazla sınır noktası (endpoint) bulunabilir. Farklı sınır noktaları aynı adresi paylaşan farklı kayıt alanları için bir şey ifade etmez. Fakat farklı sınır noktaları cihaz belirleme, sıradan veri akışı ve kontrolü amacıyla kullanılabilir. USB'deki boruyla taşıma terimi özel bir sınır noktasından veri aktarmak anlamına gelmektedir. Bu veri akış hızını arttırmak için cihazın aynı anda birden fazla boru işgal etmesi gerekir.



Şekil 2.1 : Verinin paketlenip işlenmesi (Insam, 2002).

2.1.13 Veri Aktarım Yöntemleri

Bilgisayara USB yoluyla veri aktarmak için 4 yöntem vardır.

2.1.13.1 Kontrol Transferi

Kontrol transferleri cihazların yeteneklerine ve yapılanmalarına ilişkin veri ve isteklerin iletimini yürütür. Bu işaretler yüksek önceliklidir, hata koruması sağlarlar ve blok bilgi transferine de uygundur. Cihaz bilgisini belirler fakat genellikle düşük hızlarda kullanılırlar.

Kontrol transferinin iki ya da üç basamaklı bir yapısı vardır.

- Setup
- Veri (seçimlik)
- Durum (statü)

Bir basamak, bir veya daha fazla işlemten oluşabilir. Bir kontrol transferinde setup ve durum basamaklarının bulunması zorunludur. Veri basamağı ise, belli bir istek

gerektirmekle birlikte, seçime bağlıdır. Kontrol transferinde bazı bilgilerin her iki yönde de aktarılması gerektiği için bu transferin mesaj borusu, uç noktanın hem IN hem de OUT adreslerini kullanır. Tüm USB ile çalışan cihazlar kontrol-transferi desteklemek zorundadır.

2.1.13.2 Yığın Transferi

Sürenin kritik olmadığı veri transferinde kullanılır. Yığın transferde zamandan bağımsız olarak büyük boyutlardaki bilgileri veriyolunda yığılmaya yol açmadan aktarılmasını sağlar. Bu aktarım türü kontrol transferinde olduğu gibi yüksek öncelikli değil düşük önceliklidir. Yani ikinci plandadır ve zaman uygun oluncaya kadar bekleyebilir. Bu transfer türü yalnızca tam hız cihazlara özgüdür. Her cihaz bu tür transfere destek vermese de, belli grup cihazlar için böyle bir zorunluluk söz konusu olabilir.

2.1.13.3 Kesme Transferi

Belli zamanda aktarılması gereken orta büyüklükteki veri miktarı için kesme transferi kullanılır. Genellikle klavye, fare ve oyun çubukları gibi çevre birimlerinin durum raporlarını aktarmak için kullanılır. Kullanıcılar tuşa basma veya fareyi hareket ettirme gibi bir eylemle eylemin ekranda görünmesi arasında bir gecikme istemezler. Kesme transfer, cihazın PC'den gelen ve hızlı bir yanıtla sonuçlanan donanım kesmesi anlamına gelmektedir. Ancak kesme transferler diğer USB transferler gibi yalnızca karargâhın cihaz yoklaması sırasında ortaya çıkarlar.

2.1.13.4 Asenkron Transfer

Verinin belirli bir sürede ya da sabit bir hızda hareket etmesinin gerektiği durumlarda yarar sağlayan gerçek zamanlı bir transfer, bir akış durumu transferidir. Ancak veriyi hata koruması yapmadan aktardığı için sistem az da olsa veri kaybını kabul etmelidir. Bu transfere örnek olarak gerçek zamanda çalışan kodlanmış ses ve müziği verebiliriz. Burada akılda tutulması gereken şey ise sabit hızda kullanılacak veri için mutlaka asenkron transfer gerekmez. Örneğin karargâh bir müzik dosyasını gönderirken yığın transferine başvurabilir ve daha sonra alıcı cihaz tarafından sabit hızla çalıştırılabilir.

Şu da bilinmelidir ki asenkron transfer için mutlaka sabit hız olması gerekmez. Asenkron transfer büyük çaplı bir veriyi meşgul bir veriyoluna hızlı bir şekilde

yerleřtirmekte kullanılan bir yöntemdir. Hatta bu durumda transferin gerçek zamanlı olması bile şart değildir. Yığın transferden farkı, verinin gönderileceđi sabit hızın güvenceye alınması ve buna bađlı olarak da transferin ne zaman biteceđinin kestirilebilir olmasıdır. Sonuç olarak söz edilenler ařađıdaki çizelgede özetlenmiřtir (Gerçek, 2002).

Çizelge 2.2 : Veri akışı yöntemlerinin özellikleri

Transfer Tipi	Kontrol	Yığın	Kesme	Asenkron
Her transfer için veri aktarımı bayt/milisanıye, her borudaki en fazla akış hızı (yüksek hızda)	832, 30 adet 64-baytlık işlemde	1216, 90 adet 64-baytlık işlemde	64	1023
Her transfer için veri aktarımı bayt/milisanıye, her borudaki en fazla akış hızı (düşük hızda)	24, 3adet 8-baytlık işlemde	İzin verilmiyor	Her 8 baytlık 10 ms’de 0.8	İzin verilmiyor
Tüm transferdeki sahip olunan bant genişliđi % olarak	10%	Hiç	%90 kesme & asenkron birleşik	%90 kesme & asenkron birleşik
Hata ayıklama ve düzeltme var mı?	Evet	Evet	Evet	Evet
Verinin tümü kayıpsız olarak ulaşıyor mu?	Hayır	Hayır	Hayır	Evet
Transferler arasında gecikme var mı?	Hayır	Hayır	Evet	Evet

2.1.14 USB’nin Sağladığı Avantajlar

2.1.14.1 Kullanıcıya Sağladığı Avantajlar

Kullanıcıya sağladığı avantajlar řunlardır; Kullanım kolaylığı, hızlı ve güvenilir veri transferi, esneklik, düşük maliyet ve güç tasarrufu sağlamasıdır.

2.1.14.1.1 Kullanım Kolaylığı

USB’nin tasarımındaki başlıca nedendir ve sonuç da gayet başarılıdır. Çeřitli cihazlarla kullanım için bir tane arabirim kullanılmaktadır. Böylece donanım her cihaz için ayrı destekler içermemekte, ayrı bir konektöre gereksinim duymamaktadır. Kullanıcı bir USB çevreirimini sisteme bađladığı (taktığı) zaman, Windows, cihazı otomatik olarak tanır ve uygun sürücüyü yükler. Cihazın ilk takılışı sırasında Windows kullanıcıdan sürücü disketini isteyebilir. Yükleme işlemi otomatik olarak

yürür. Özel bir kurulum bilgisi gerektirmez. USB cihazlarda, kullanıcı ayar yapmaz. Örneğin port adresleri, kesme isteği kanalları (IRQ) ile uğraşmaz. Sadece bu olanak bile USB'yi tercih etmek için yeterli bir sebeptir. USB kullanımıyla, cihazların ihtiyaç duymadığı IRQ'lar boşaltılır, yani aktif olmayan donanım kaynakları serbest kalır. PC'de, USB arabirimine bir IRQ hattı ile bir dizi port adresi ayrılır. Çevre birimleri, tek tek, ek kaynağa gereksinim duymazlar. USB-olmayan cihazlar ise özel port adresleri, çoğu kez bir IRQ hattı ve kimi zaman da genişleme yuvası (expansion slot) isterler. PC'de normal olarak en az iki USB portu vardır. Cihazı bir USB portuna takmak yeterlidir. Portların sayısı, bağlanacak bir hub ile artırılabilir. Kablo konnektörleri hatalı takmaya olanak vermezler. Kablolar basittir ve en fazla 5 metre uzunlukta olabilirler. Hub kullanımıyla uzaklık 30 metreye çıkarılabilir.

2.1.14.1.2 Hız

USB üç hızı destekler. Bunlar, yüksek hız (saniyede 480 Megabit), tam hız (saniyede 12 Megabit) ve düşük hız (saniyede 1.5 Megabit). USB seçeneği olan her PC düşük ve tam hızı destekler. Yüksek hız hakkında bilgi, sürümleri başlığı altında verilmiştir. Bunun için anakartta USB 2.0'ı destekleyen donanım veya genişleme kartı gerekir. Söz konusu hızlar sinyalizasyon -bus tarafından desteklenen bit- hızlarıdır. Bir cihaz için beklenen transfer hızı bunlardan daha düşüktür. Veri dışında veriyolunda durum, kontrol ve hata-kontrol sinyalleri de taşınır. Bir transferin teorik maksimum hızı yüksek ve tam hızlarda, sırasıyla, 53 ve 1.2 megabayt/saniye; düşük hızda 800 bayt/saniyedir.

Düşük hız iki nedenle desteklenmektedir. Birincisi bu tür cihazlar ucuzdur. İkincisi ise fare gibi esnek kablo isteyen cihazlar için düşük hız kabloları daha uygundur. Büyük çaplı kılıflama gerektirmezler. Mevcut seri ve paralel portlarla ulaşılan hızlar, tam hız düzeyinde veya daha düşüktür. Dolayısıyla bunların yerini tutabilir.

2.1.14.1.3 Güvenilirlik

USB'nin güvenilirliği donanım tasarımından ve transfer protokollerinden kaynaklanır. USB düşürücüleri, atıcıları ve kablolarına yönelik donanım spesifikasyonları veri hatalarına yol açan gürültünün büyük kısmını gidermektedir. USB protokolü veri hatalarının tespitini olanaklı kılmakta ve vericiyi yeniden gönderim yapabileceği konusunda bilgilendirmektedir. Tespit, bildirim ve yeniden

iletim donanımında çözülmektedir. Böylelikle programlama ya da kullanıcı müdahalesine gerek bırakmaz.

2.1.14.1.4 Maliyet Avantajı

Önceki arabirimlerden karmaşık olmakla birlikte, USB'nin devre elemanları pahalı sayılmazlar. USB arabirimli bir cihazın maliyeti en fazla eskilerin maliyeti kadardır. Çok ucuz çevrebirimleri düşünülürse, düşük hız seçeneğine başvurulmak kaydıyla son derece düşük maliyetlerle çalışmak mümkündür.

2.1.14.1.5 Düşük Güç Harcaması

Tasarruflu devreler ve kodu sayesinde, kullanılmayan USB cihazlarının gücü kesilir. Ancak bu durumda bile cihazlar, gerektiğinde yanıt vermeye hazırdırlar. Bu özelliğin asıl yararı, genel tasarruf avantajına ek olarak, bir miliamperin bile önemli olduğu pilli bilgisayarlarda ortaya çıkar.

2.1.14.2 Tasarımcıya Sağladığı Avantajlar

Kullanıcıya sağladığı avantajlar nedeniyle başka türlü cihazlar geliştirmeye uğraşmak zaman kaybı anlamına gelmektedir. Bu durumun tasarımcıya sağladığı ek kolaylıklar da söz konusudur. Örneğin kablo standartları veya bir diğer örnek, otomatik hata-denetimi, tasarımcı ve yazılımcıları kablo karakteristikleriyle ya da yazılımda hata denetimiyle uğraşmaktan kurtarmaktadır. Gerek donanım gerekse yazılım ve yonga kodu açısından, tasarımcıya sağladığı çok önemli bir avantajdan bahsetmek gerekir. Bu avantaj USB protokolünün yapısal esnekliğinden, kontrolörlerden ve işletim sisteminden gelen destekten ve arabirimin üretici sayısının çokluğundan kaynaklanır.

2.1.14.2.1 Esneklik

Dört transfer tipi ve üç hızı ile USB, birçok cihaz için uygulanabilirlik kazanmaktadır. Gerek transfer edilen verinin boyutları, gerek zaman sınırlamaları açısından çeşitli transfer tipleri söz konusudur. Gecikme toleransı olmayan transferlerde, hız veya ardışık transferler arasında geçen zaman USB tarafından garanti edilmektedir. Bunların en büyük faydası Windows ortamında görülür. Çünkü burada cihazlara gerçek zamanlı erişim ciddi bir sorundur. İşletim sistemi, cihaz sürücüler ve uygulama yazılımları nedeniyle gecikmelerin ortaya çıkması

kaçınılmazdır. USB sayesinde transferler gerçek zamanlara yakın düzeyde gerçekleşmektedir. USB, diğer arabirimlerden farklı olarak, sinyallere belli fonksiyonlar yüklemeyiz. Arabirimin kullanım şekline ilişkin ön belirlemelere gerek bırakmaz. Örneğin paralel porttaki durum ve kontrol hatları yazıcılarla haberleşmeye yönelik belirlemelerdir. Sadece bunlar için tahsis edilen beş bit, portun tarayıcıyla haberleşmede kullanılması halinde bu durum doğrudan bir engel olarak ortaya çıkar. USB’de böyle sınırlamalar yoktur. Yazıcı ve modem gibi genel cihazlar için belirlenmiş USB sınıfları sayesinde tasarımcıların bütün aşamaları yeniden elden geçirmeleri gerekmez.

2.1.14.2.2 İşletim Sistem Desteği

USB’ye güvenilir bir şekilde destek veren ilk sistem Windows 98’dir. Sonrakiler bu desteği devam ettirmişlerdir. İşletim sistemi desteği son derece ciddi bir katkıdır. Ancak desteğin düzeyinde değişimler her zaman mümkündür. USB’ye destek veren bir işletim sistemi en az isteği karşılamalıdır:

- Sisteme takılan ya da sökülen cihazı algılamak,
- Takılan cihazlarla veri alışveriş şeklini belirlemek üzere haberleşme yapmak,
- Yazılım sürücülerinin karargâhın USB donanımıyla haberleşmesini ve uygulamaların USB cihazlara ulaşmasını sağlayacak bir mekanizma olmaktır.

İşletim sistemi desteği, programlama düzeyinde, uygulama programcılarının sistemce desteklenen fonksiyonlar üzerinden cihazlara erişimini mümkün kılan sürücülerini anlatır. Bir cihaza ilişkin sürücünün işletim sisteminde bulunmaması halinde, sürücünün üretici tarafından karşılanması gerekir. Windows sürümlerinde Microsoft’un eklediği sınıf sürücülerini bulunur. Sürücülerini bulunan cihazlar şunlardır; İnsan arabirim cihazları (klavye, fare, joystick), ses cihazları, modemler, durgun görüntü kameraları, tarayıcılar, yazıcılar. Uygulamalar, ilerle haberleşirken API fonksiyonlarından ve diğer sistem elemanlarından yararlanırlar.

Windows’un cihaz sınıflarının sayısı artma eğilimindedir. Öte yandan, üreticiler tarafından sunulan kimi sürücüler ufak tefek değişikliklerle kendi yongalarıyla anılabilmektedir. USB cihaz sürücülerinde yeni Win32 sürücü modeli (WDM) kullanılır. Bu model, Windows 98 ve sonrasında çalışan sürücülerin mimarisini tanımlamaktadır. Amaç, tasarımcıların tüm işletim sistemi desteğini tek bir sürücüyle sağlamalarıdır. Oysa bazı cihazlar için hâlâ iki sürücü (biri Windows 98/Me, diğeri

Windows 2000 için) mevcuttur. Windows'un USB donanımıyla haberleşmesi düşük düzeyli sürücülerle gerçekleşir. Bu nedenle, USB sürücülerinin yazımı başka arabirimler için sürücü yazımına kıyasla daha kolaydır.

2.1.14.2.3 Çevrebirim Desteği

USB'li cihazın donanımında bir kontrolör bulunur. Haberleşmenin ayrıntılarını yürüten bu yonga bazen bir CPU ve cihaz özgü kodu içeren bir bellek de içerebilir. Diğer yongalar veriyoluna bağlı ve USB dışı fonksiyonlar yanında gerektiğinde USB kontrolörüyle haberleşmeyi yürüten bir diğer mikrokontrolörle çalışır ve sadece USB'ye özgü işleri görürler, istekleri yanıtlamak çevrebirimine düşer. USB kontrolörlerinin çoğunun mimarisi çoğu kez Intel 8051 gibi yaygın yongaları esas alır. USB'yi destekleyen ek devreler ve makine kodları içerirler. Cihaz üreticilerinin büyük çoğunluğu yongayla birlikte örnek kod da sunarlar. Bunu bir başlangıç noktası olarak kullanmakta yarar vardır.

USB Implementers Forum (USB-IF): USB-IF kâr amaçlı olmayan bir kuruluştur. USB spesifikasyonunu hazırlayan firmaların katkılarıyla kurulmuştur. Temel görevi USB teknolojisinin kullanımını ve yaygınlaşmasını sağlamaktır. Forum, USB konusunda bilgi, yazılım ve donanım aletleri ile test olanakları sunmaktadır. Web sitesinden (www.usb.org) ayrıntılı bilgiye ulaşmak ve bağlantıya geçmek mümkündür.

2.1.15 USB Kullanımının Dezavantajları

Bir arabirimle her şeyi en iyi ölçüde yapmak mümkün değildir. Bir takım eksiklikleri söz konusudur. Bunlardan bazıları aşağıda sıralanmıştır.

2.1.15.1 Kullanıcıya Yönelik Sorunlar

Eski donanımlar ve işletim sistemleriyle uyumlu olmaması en temel handikaptır. Bundan başka, hız ve mesafe sınırlamaları nedeniyle USB kullanımı bazı durumları pratik olmamaktadır.

2.1.15.1.1 Eski Donanımlarla Uyumsuzluk

Eski donanımlarda USB portu yoktur. USB olmayan bir cihazı USB porta bağlamanın yollarından biri konvertör kullanmaktır. Ancak bu çözüm sadece konvertör sürücüsünün tanıdığı konvansiyonel protokollerin kullanıldığı çevre

birimlerinde işe yarar. Örneğin paralel port çeviricisi yazıcıları destekleyecek fakat diğer birimleri tanımayacaktır. USB cihazını USB portu olmayan bir PC ile kullanırken, PC'ye USB kapasitesi eklemek şeklinde bir çözüm düşünülebilir. Bunun için iki şey gerekir: bir PC kontrolör donanımı ve bir de işletim sistemi. Donanım PCI yuvasına takılacak bir genişleme kartı ile sağlanabilir. İletim sistemi ise Windows 98 veya daha ileri sürümlerinden biri olmalıdır. Windows 98 desteği olmayan PC donanımıyla çalışmak hemen hemen olanaksızdır. PC'nin USB'yi tanıyacak şekilde yükseltilmesi makul görünmüyorsa, cihazın USB arabirimini PC'nin RS-232 veya bir diğer arabirimine gösterecek bir konvertör akla gelebilir. Ancak bunun da iyi bir seçenek olduğunu söylemek kolay değildir.

2.1.15.1.2 Hız sınırları

USB ile her şeyi halletmeye çalışmak doğru değildir. Görünen tek rakibi Firewire'dir. IEEE-1394 ve IEEE-1394b ile yarışabilecek bir seçenek sunmaktadır.

2.1.15.1.3 Mesafe Sınırlamaları

USB masaüstü veriyolu olarak tasarlanmıştır. Yani çevre birimlerinin yakınlarda bir yerde olacağı kabul edilmiştir. Kablo boyu 5 metreye kadar çıkabilir. Oysa RS-232, RS-485 veya Ethernet daha büyük mesafelere izin verir. Bununla birlikte beş adet hub kullanarak mesafeyi 30 metreye çıkarmak olanaklıdır. Bir diğer seçenek ise 485 çeviriciyle mesafeyi kat edip PC üzerinde USB kullanmak olabilir.

2.1.15.1.4 Cihazlar Arası Haberleşme

Masaüstü tasarımın bir sonucu, haberleşmenin karargâh PC tarafından denetlenmesidir. Yani cihazlar doğrudan haberleşemezler. Spesifikasyonuna 2001'de yapılan bir ekleme ile USB On-The-Go şeklinde kısmi bir çözüm olanağı söz konusu olmuştur. Burada, kapasitesi daraltılmış bir karargâh P tanımlanmaktadır. Bu PC, tek bir USB cihazla bağlanması gereken gömülü cihazlarda kullanılmaya uygundur.

2.1.15.1.5 Sorunlu Ürünler

USB çalışıyorsa her şey harika demektir. Ancak çalışmadığı zaman, haberleşme bozukluğundan sistemin çakılmasına kadar her türlü sorunla karşılaşmak olasıdır. Ancak, işletim sistemindeki gelişmelere ve tasarımcıların ustalaşmasına bağlı olarak bu tür sorunların azalma eğiliminde olduğunu söylemek mümkündür.

2.1.15.2 Tasarımcıya Yönelik Sorunlar

Programlamanın karmaşıklığı başlıca engeldir. Gerek PC'nin ve gerekse cihazın donanımında gözden kaçan noktalar (bug'lar) ciddi sorunlar yaratır.

2.1.15.2.1 Protokolün Karmaşıklığı

USB çevrebiriminin programlanması, protokoller hakkında ayrıntılı bilgi sahibi olmayı gerektirir. Haberleşmenin büyük kısmı yonga tarafından otomatik olarak halledilir. Ancak, iş bununla bitmez. Sürücü yazarlar, protokolleri ve sürücünün işlevini iyice anlamış olmak durumundadırlar. Eski ara birimlerdeki devreler ve protokoller son derece basittir. Temel giriş ve çıkışların bağlantıları ayrıntılarda boğulmadan kolayca halledilebilir.

2.1.15.2.2 İşletim Sistemi Desteğinin Artırılması

Windows sınıf sürücüleri, uygulamaların cihazlarla haberleşmesini sağlar. Cihaz bu sürücülerle çalışmak üzere tasarlanabiliyorsa ciddi bir sorun çıkmaz. Aksi durumda bir sürücü yazma yükümlülüğüyle karşılaşmak işten değildir.

2.1.15.2.3 Donanım Sorunları (Bug'lar)

İlk karargâh denetleyicilerinin donanımları sorunlu olabiliyordu. Aynı şekilde çevrebirimlerinde de sorunlarla karşılaşabiliyordu. Bundan kaçınmanın en iyi yolu donanım seçiminde gereken dikkati göstermek ve son bilgileri elde etmektir.

2.1.16 Bedeller

USB-IF tarafından sunulan sürüm, web sitesinden ücretsiz olarak alınabilir. USB yazılımı geliştirmek için lisans bedeli ödemek gerekmez. USB arabirimi satışlarında durum farklıdır. Çünkü belli bir Üretici Kimliği'ne sahip olmayı gerektirir. Forum'dan alınacak bu kimliğin bedeli 1500 dolardır. 2500 dolar yıllık katılım bedeli ödeyerek, bu kimliği ücretsiz alma yolunu tercih etmek de mümkündür. Üretici Kimliği (Vendor ID) ile Ürün Kimliği (Product ID), cihazın işletim sistemine tanıtımı açısından zorunluluk arz ederler (Gerçek, 2002).

2.2 PIC Mikrodenetleyiciler ve Kontrol Sistemi

2.2.1 PIC Mikrodenetleyiciler

PIC'in kelime anlamı PERIPHERAL INTERFACE CONTROLLER giriş-çıkış işlemcisidir. İlk olarak 1994 yılında 16 bitlik ve 32 bitlik büyük işlemcilerin, giriş ve çıkışlarındaki yükü azaltmak ve denetlemek amacıyla çok hızlı ve ucuz bir çözüme gereksinim duyulduğu için geliştirilmiştir.

2.2.1.1 PIC Mikrodenetleyicilerin Tercih Sebepleri

- 1) Lojik uygulamalarının hızlı olması
- 2) Fiyatının oldukça düşük olması
- 3) Veri ve belleğe hızlı olarak erişimin sağlanması
- 4) PIC'e göre diğer mikrodenetleyicilerin veri ve programı taşıyan tek bir veriyolunun bulunması
- 5) Yüksek frekanslarda çalışabilme özelliği
- 6) Bekleme konumunda çok düşük akım çekmesi
- 7) Kod sıkıştırma özelliği ile aynı anda birçok işlem gerçekleştirebilmesi
- 8) Herhangi bir ek bellek veya giriş/çıkış elemanı gerektirmeden sadece iki kondansatör ve bir direnç ile çalışabilmeleri.

2.2.1.2 PIC Mikrodenetleyicilerin Kullanımı İçin Gerekli Aşamalar

2.2.1.2.1 I/O (Giriş / Çıkış)

Mikrodenetleyicinin dış dünya ile ilişkisini sağlayan, girdi ve çıktı şeklinde yararlanılabilen bir bağlantı bacağıdır. I/O çoğunlukla mikrodenetleyicinin iletişim kurmasına, kontrol etmesine veya bilgi okumasına izin verir.

2.2.1.2.2 Yazılım

Mikrodenetleyicinin çalışmasını ve işletilmesini sağlayan bilgidir. Başarılı bir uygulama için yazılım hatasız (bug) olmalıdır. Yazılım C, Pascal veya Assembler gibi çeşitli dillerde veya ikilik (binary) olarak yazılabilir.

2.2.1.2.3 Donanım

Mikrodenetleyici, bellek, arabirim bileşenleri, güç kaynakları, sinyal düzenleyici devreler ve bunları çalıştırmak ve arabirim görevini üstlenmek için bu cihazlara bağlanan tüm bileşenlerdir.

2.2.1.2.4 Simülatör

Bilgisayar üzerinde çalışan ve mikrodnetleyicinin içindeki işlemleri simüle eden yazılım paketidir. Hangi olayların ne zaman meydana geldiği biliniyorsa bir simülatör kullanmak, tasarımları sınamak için kolay bir yol olacaktır. Öte yandan simülatör, programları tümüyle veya adım adım izleyerek arıza ve hatalardan arındırma fırsatı sunar. Şu anda en gelişmiş simülatör programı Microchip firmasının geliştirdiği MPLAB programıdır.

2.2.1.2.5 ICE (In- Circuit Emulator / İç devre takipçisi)

Bilgisayarda çalışırken bir programa girilmesini, mikro içinde neler olduğunu ve dış dünyayla nasıl iletişim kurulduğunun izlenilmesini mümkün kılar.

2.2.1.2.6 Programcı

Yazılımın mikrodnetleyici belleğinde programlamasını ve böylece iç devre takipçisinin yardımı olmadan çalışmasını sağlayan bir birimdir. Çoğunlukla seri port'a (örneğin PICSTART, PROMASTER) bağlanan bu birimler çok çeşitli biçim, ebat ve fiyatlara sahiptir.

2.2.1.2.7 Kaynak Dosyası

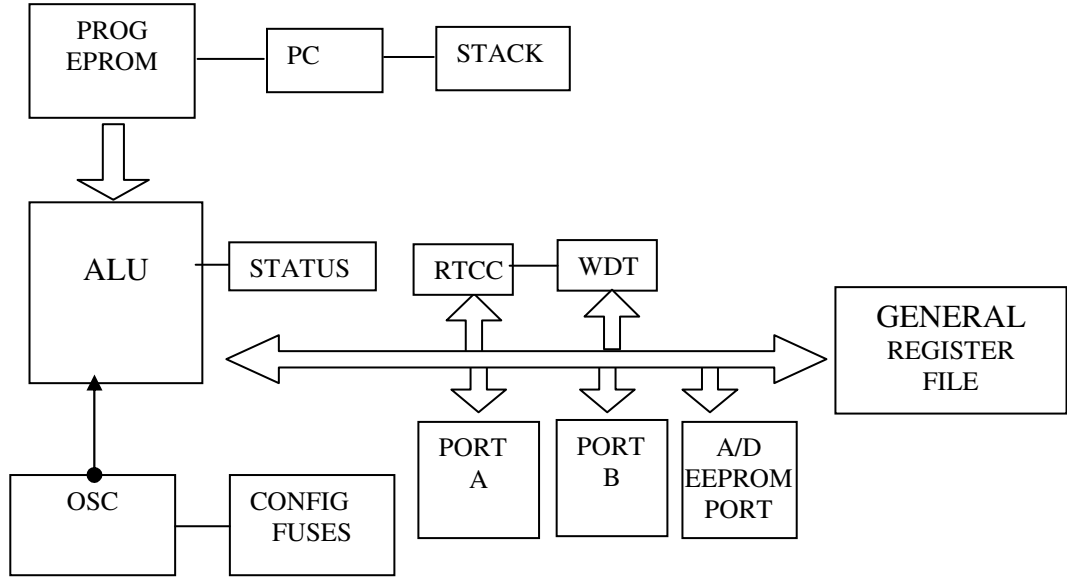
Hem derleyicinin (assembler) hem de tasarımcının anlayabileceği dilde yazılmış bir programdır. Kaynak dosya mikrodnetleyicinin anlayabilmesi için önceden derlenmiş olmalıdır.

2.2.1.2.8 Derleyici (Assembler)

Kaynak dosyayı bir nesne dosyaya dönüştüren yazılım paketidir. Hata araştırma bu paketin yerleşik bir özelliğidir. Bu özellik derlenme sürecinde hatalar çıktıkça programı hatalardan arındırırken kullanılır.

2.2.1.2.9 Nesne Dosyası (Object File)

Derleyici tarafından üretilen bu dosya; programcı, simülatör veya iç devre takipçisinin anlayabilecekleri ve böylelikle dosyanın işlevlerinin çalışmasını sağlayabilecekleri bir dosyadır. Dosya uzantısı derleyicinin emirlerine bağlı olarak, .OBJ veya .HEX olur.



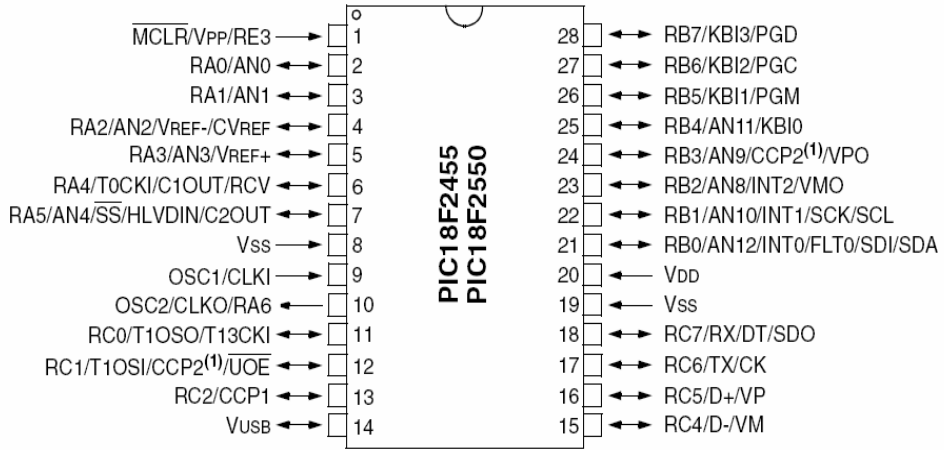
Şekil 2.2 : Temel PIC blok diyagramı.

2.2.1.3 PIC Mikrodenetleyicilerinin İç Yapısı

CPU bölgesinin kalbi ALU'dur (Aritmetic Logic Unit-Aritmetik mantık birimi). ALU, W (Working-Çalışan) adında bir yazmaç içerir. PIC, diğer mikroişlemcilerden, aritmetik ve mantık işlemleri için bir tek ana yazmaca sahip oluşuyla farklılaşır. W yazmacı 10 bit genişliğindedir ve merkez işlem birimindeki (CPU) herhangi bir veriyi transfer etmek üzere kullanılır. CPU alanında ayrıca iki kategoriye ayırabileceğimiz veri yazmaç dosyaları (Data Register Files) bulunur. Bu veri yazmaç dosyalarından biri, I/O ve kontrol işlemlerinde kullanılırken, diğeri RAM olarak kullanılır. PIC'lerde Harward mimarisi kullanılır. Harward mimarisi mikrodnetleyicilerde veri akış miktarını hızlandırmak ve yazılım güvenliğini arttırmak amacıyla kullanılır. Ayrı veriyollarının kullanımıyla veri ve program belleğinde hızlı bir şekilde erişim sağlanır.

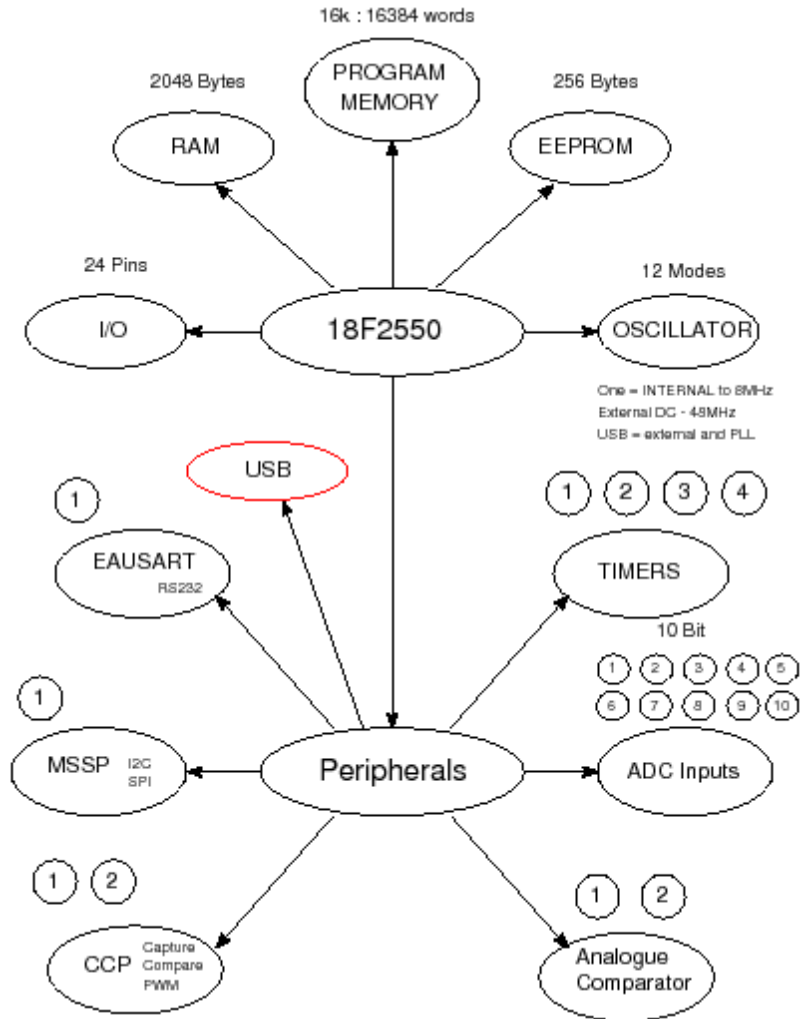
2.2.2 PIC 18F2550

18 serisi mikrodnetleyiciler en ekonomik yoldan yüksek işlem performansı sağlarlar. Yüksek dayanımlı flaş program belleğine sahiptirler ve nano watt teknolojisiyle çok az güç tüketmektedirler. PIC18F2550'nin pin diyagramı ise Şekil 2.3'te gösterilmiştir.



Şekil 2.3 : PIC18F2550'nin Pin diyagramı.

2.2.2.1 PIC 18F2550'nin Özellikleri



Şekil 2.4 : PIC18F2550'nin özellik blok diyagramı.

2.2.2.1.1 Evrensel Seri Yolu Özellikleri

- USB 2.0 sürümüyle uyumludur.
- Düşük hızda (1.5 Mb/s) ve yüksek hızda (12 Mb/s) çalışabilir.
- Kontrol, kesme, asenkron ve büyük hacim transferini desteklemektedir.
- En çok 32 adet işlem bitiş noktasını (endpoint) (ya da 16 çift yönlü işlem) desteklemektedir.
- 1-Kilobaytlık çift erişimli RAM(Read Access Memory) kullanarak USB'ye ulaşır.
- Gerilim regülasyonu ve USB'ye çevrimini çipin kendisi yapılmaktadır.
- USB'ye çevrimde arayüz oluşturmaktadır.

2.2.2.1.2 Güç kullanımı

- Çalışma durumu: CPU çalışıyor, cihazlar çalışıyor.
- Boşta durumu: CPU çalışmıyor, cihazlar çalışıyor.
- Uyuma durumu: CPU çalışmıyor, cihazlar çalışmıyor.
- Boşta durumundayken çektiği tipik akım 5.8 μ A'den azdır.
- Uyku durumunda çekilen tipik akım 0.1 μ A'dir.
- Zamanlayıcı1(Timer1) Osilatörü: Tipik akım 1.1 μ A, frekans 32 kHz ve gerilim 2V'tur.
- Koruma zamanlayıcısı: çektiği tipik akım 2.1 μ A'dir.
- Başlangıçta çift hızlı osilatör kullanılmaktadır.

2.2.2.1.3 Esnek osilatör yapısı

- 4 kristal modu, USB için yüksek doğrulukta Faz Kilitlemeli Çevrim arabirimi vardır (PLL, Phase Locked Loop).
- 2 adet harici zamanlayıcı arabirimi sayesinde çalışma frekansı 48 MHz'e kadar çıkabilmektedir.
- İç osilatör: 8 adet kullanıcı seçimli frekans, 31 kHz'den 8 MHz'e kadar ve frekans kayması kullanıcı tarafından ayar yapılarak telafi edilebilir.
- İkincil osilatör Zamanlayıcı1'dir ve 32 kHz de çalışır
- İkili osilatör yapısı mikrodenetleyici ve USB arabiriminin farklı saat hızlarında çalışmasını sağlar.
- Arıza güvenli (Fail-Safe) saat görüntüleme: Bir parçası çalışmadığında arızayı giderir ve eğer saatlerden biri bozulduğunda ya da durduğunda güvenli olarak kapatılma olanağı sağlar.

2.2.2.1.4 Cihazın göze çarpan özellikleri

- Yüksek akım 25 mA,
- 3 adet harici kesme,
- 4 adet zamanlayıcı arabirimi (Timer0'dan Timer3'e kadar),
- 2 adede kadar tutucu karşılaştırıcı ve darbe genişlik modülasyonu (CCP, Capture/Compare/PWM):
 - Tutucu 16-bit boyutunda ve en çok 5,2 ns'de çözömlenmektedir;
 - Karşılaştırıcı 16-bit boyutunda ve en çok 83,3 ns'de çözömlenmektedir;
 - Darbe genişlik modülasyonu çıkışı: Çözömlemesi 1 to 10-bit boyutundadır;
- Geliştirilmiş tutucu karşılaştırıcı ve darbe genişlik modülasyonu (ECCP, Enhanced Capture/Compare/PWM):
 - Çoklu çıkış arabirimi
 - Kutup seçebilme
 - Programlanabilir ölü zaman
 - Otomatik kapama ve tekrar başlatma özelliği
- Gelişmiş evrensel senkron asenkron alıcı verici (USART, Universal Synchronous Asynchronous Receiver Transmitter) arabirimi:
 - LIN bus'ını destekler
- Ana senkron seri port (MSSP, Main Synchronous Serial Port) arabirimi:
- Programlanabilir zaman kazancı ile 10-bit büyüklüğünde en fazla 13 kanalda analog sinyal dijitalle çevrilebilir.
- Giriş çoklayıcılarıyla birlikte iki adet analog karşılaştırıcı bulundurulur.

2.2.2.1.5 Cihaza özel özellikler

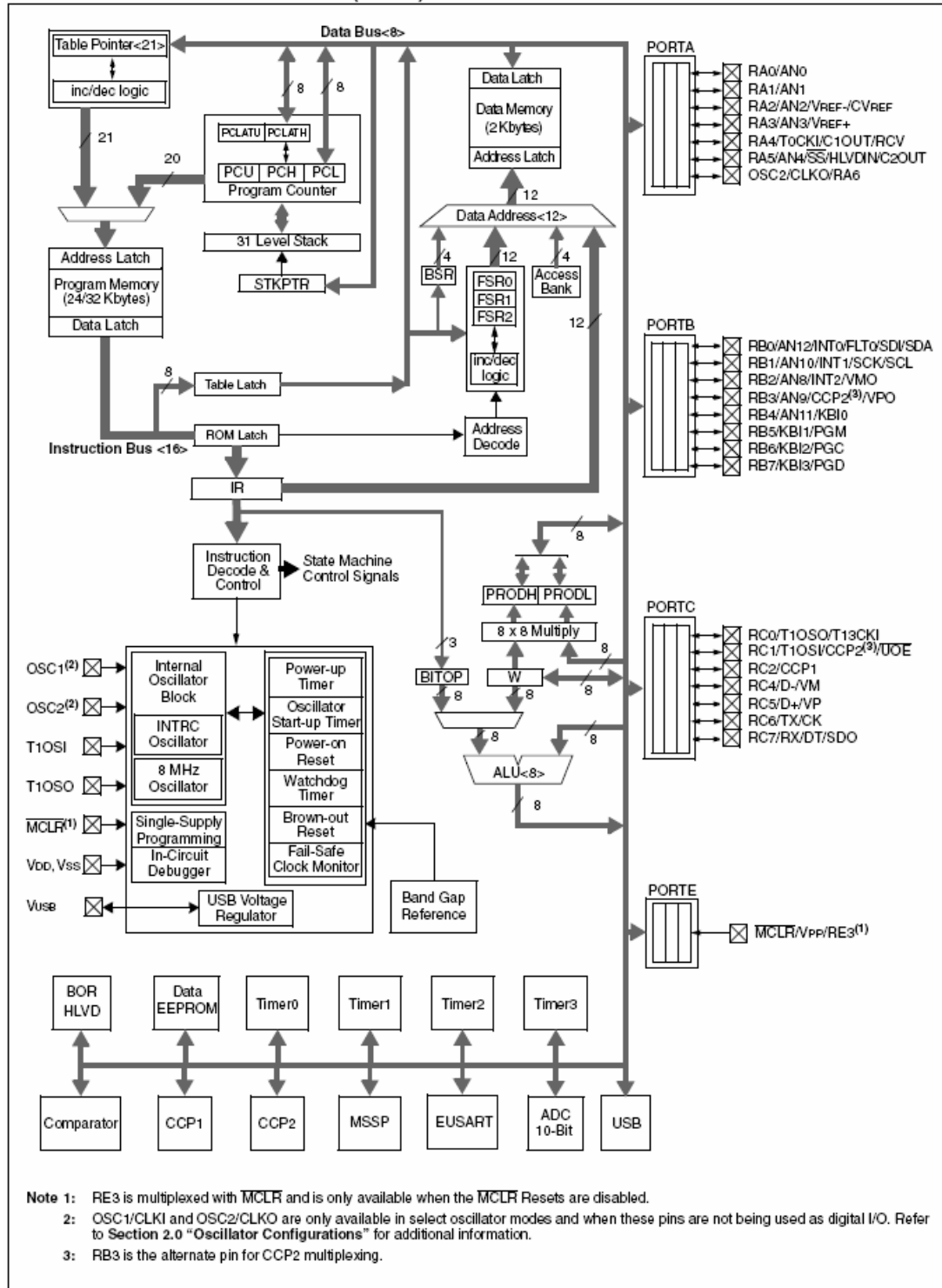
- C derleyiciyi mimarisini seçilen genişletilmiş komut setiyle birlikte en iyi şekilde kullanabilir.
- 100,000 sil/yaz döngüsüyle geliştirilmiş flash program belleği vardır.
- 1,000,000 sil/yaz döngüsüyle oluşturulmuş silinebilir ve programlanabilir elektronik okunur belleği vardır. (EEPROM, Electronic Erasable Programmable Read Only Memory) belleği vardır.
- Flash program belleği EEPROM'a göre veriyi 40 yıl daha fazla saklayabilir.
- Yazılım kontrolü ile kullanıcı tarafından programlanabilir
- Kesme için yüksek öncelik tanır.

- 8 x 8 basit çevrim donanım çarpımı söz konusudur (Hardware Multiplier)
- Genişletilmiş koruma zamanlayıcısı (WDT): 41 ms'den 131s'ye kadar periyot programla ayarlanabilir.
- Programlanabilir kod korumasına sahiptir.
- 5V'luk bir tek kaynak ve 2 pin yoluyla devrede seri programlama olanağı sağlar. (ICSP™, In-Circuit Serial Programming™)
- 2 pin yoluyla devrede hata ayıklaması yapılabilir.
- İsteğe bağlı olarak ICD/ICSP portu atanabilir. (sadece 44-pinli cihazlar için)
- Geniş gerilim aralığında çalışabilir. (2.0V ile 5.5V arası)

Çizelge 2.3 : PIC 18F2550'nin özellikleri

Cihaz	Program Belleği		Veri Belleği		I/O	10-Bit A/D (ch)	CCP/ECCP (PWM)	SPP	MSSP		EAUSART	Karşılaştırmacı	Zamanlayıcı 8/16-Bit
	Flash (b ayt)	# Tek Kelime Tanımı	SRAM (b ayt)	EEPROM (b ayt)					SPI	Master I ² C™			
PIC18F2550	32K	16384	2048	256	24	10	2/0	No	Y	Y	1	2	1/3

PIC 18F2550'nin temel yapısı ise Şekil 2.5'te basitçe gösterilmektedir.

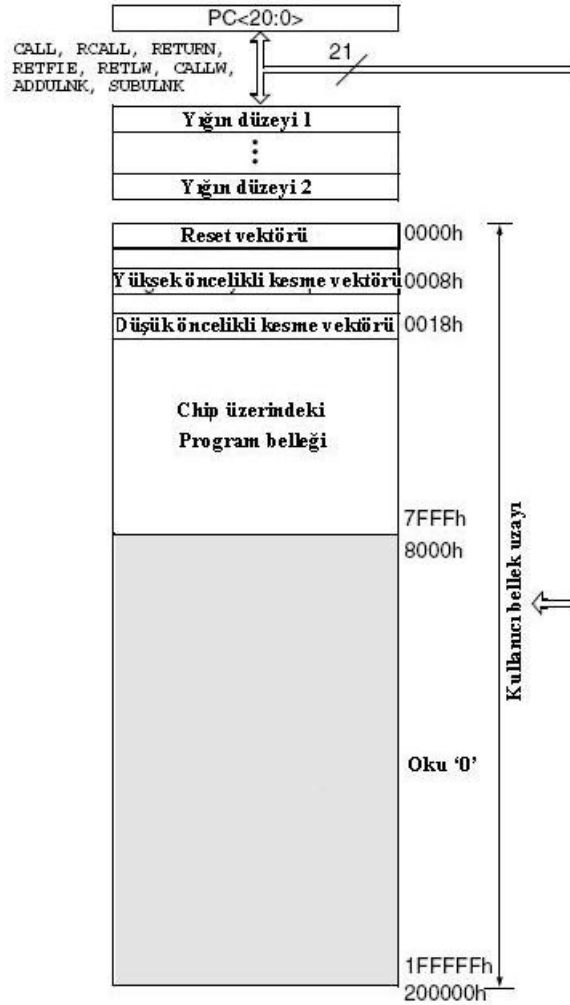


Şekil 2.5 : PIC 18F2550'nin blok diyagramı.

2.2.2.2 Bellek organizasyonu

Veri belleği ikiye ayrılır. Birincisi özel fonksiyon kayıt alanı (SFR), diğeri ise genel amaçlı kayıt alanıdır. SFR'ler aygıtın işlemini kontrol eder. Veri belleğinin bölümleri kümelenmiştir. Bu kümeler BANK adını alırlar. Bu hem SFR alanı hem de GPR alanı içinde geçerlidir. GPR alanı genel amaçlı RAM'in 16 baytıdan daha fazlasına

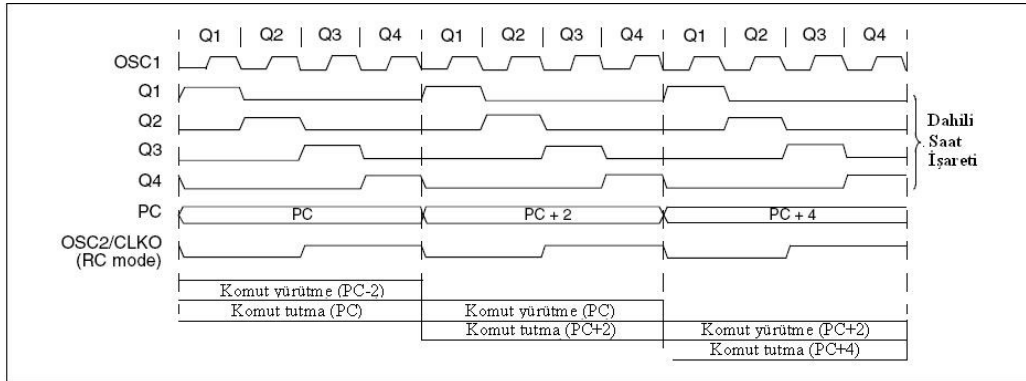
olarak sağlanabilmesi için kümelenmiştir. SFR'nin kümelenmiş alanı özel fonksiyonları kontrol eden kayıtlara aittir. Kümeleme küme seçimi için kontrol bitleri gerektirmektedir. Bu kontrol bitleri STATUS kaydında yer almaktadır. Şekil 2.6 veri belleği haritası organizasyonunu göstermektedir. Veri belleğin tümüne ya direkt her kayıt dosyasının mutlak adreslerini kullanarak, ya da, dolaylı yoldan dosya seçim kaydı (FSR) üzerinden erişilebilir. Dolaylı adresleme, veri belleğinin kümelenmiş alanına erişmek için RP1: RPO` un şimdiki değerlerini kullanmaktadır. Veri belleği genel amaçlı kayıt ve özel fonksiyon kaydını içeren iki kümeye bölünmektedir. RPO bitinin (STATUS <5>) (Yani 5. Bit RPO bitidir.) silinmesiyle BANK 0 seçilir. RPO` in kurulması BANK 1'i seçer. Her bir BANK (küme) 7Fh (128 bytes) kadar uzanır (genişler). Her bir kümenin ilk on iki yerleşimi özel fonksiyon kaydı için rezerve edilmiştir. Kalanı ise statik RAM olarak genel amaçlı kayıt yürütebilmektedir.



Şekil 2.6 : PIC 18F2550'nin program belleği.

2.2.2.3 Saat ölçüm şeması/komut süreci

Saat girişi (OSC1'den) içten dörde bölünmüştür ve Q1, Q2, Q3 ve Q4 olarak gelmeyen 4 kare dalga sinyali ortaya çıkar. İçten olarak, program sayacı (PC) her Q1 de bir arttırılmakta ve komutlar program belliğinde sürece sokularak Q4 sürecinde komut kaydına katılmaktadır. Komutlar Q1 ve Q4 aralığı boyunca yürütülür. Saat işareti ve komut yürütme akımı şekilde görülmektedir.

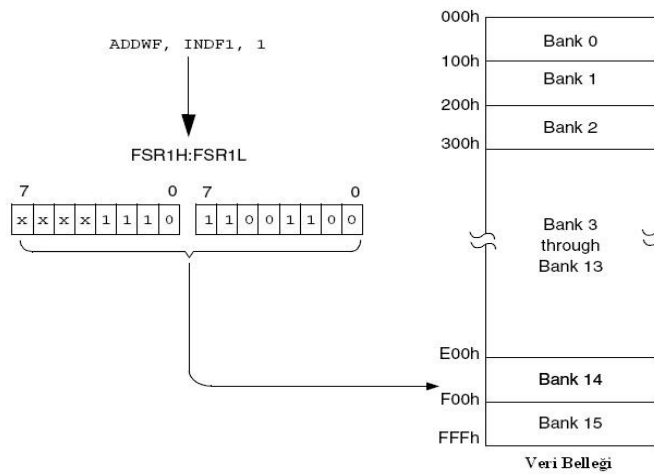


Şekil 2.7 : Komut yürütme süreci.

2.2.2.4 Kayıt Dosyaları

2.2.2.4.1 Genel amaçlı kayıt dosyası

Bütün aygıtlar belirli bir miktarda genel amaçlı kayıt (GPR, General Purpose Register) alanına sahiptir. Her bir GPR 8 bit enindedir ve her GPR'ye dolaylı ya da doğrudan dosya seçimli kaydı (FSR, File Select Register) üzerinden erişilmektedir. BANK 1`deki GPR adresleri BANK 0`daki adreslere planlanır. Örnek olarak, 0Ch veya 8Ch adresleme yerleşimi aynı GPR`ye erişecektir.



Şekil 2.8 : Genel amaçlı kayıt dosyası temel yapısı.

2.2.2.4.2 Özel fonksiyon kayıtları

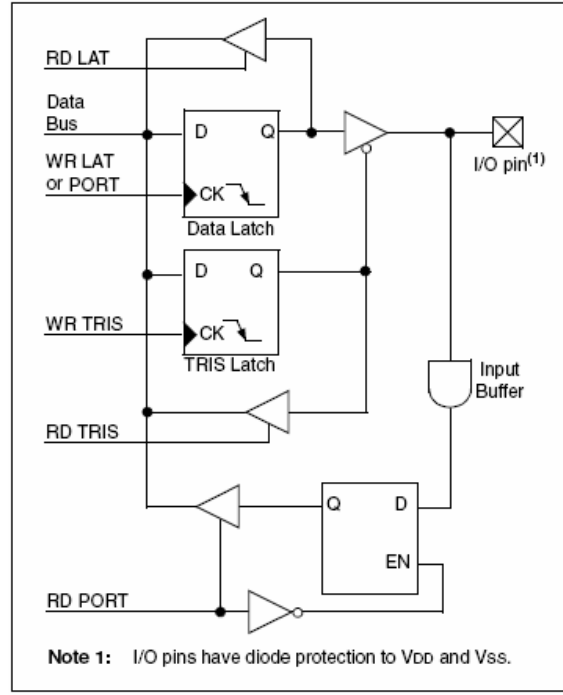
Özel fonksiyon kayıtları, aygıtın işlemini kontrol etmek için CPU ve özel fonksiyonlar tarafından kullanılmaktadır. Bu kayıtlar statik RAM'lerdir.

Adres	İsim	Adres	İsim	Adres	İsim	Adres	İsim	Adres	İsim
FFFh	TOSU	FDFh	INDF2 ⁽¹⁾	FBFh	CCPR1H	F9Fh	IPR1	F7Fh	UEP15
FFEh	TOSH	FDEh	POSTINC2 ⁽¹⁾	FBEh	CCPR1L	F9Eh	PIR1	F7Eh	UEP14
FFDh	TOSL	FDDh	POSTDEC2 ⁽¹⁾	FBDh	CCP1CON	F9Dh	PIE1	F7Dh	UEP13
FFCh	STKPTR	FDCh	PREINC2 ⁽¹⁾	FBCh	CCPR2H	F9Ch	___ ⁽²⁾	F7Ch	UEP12
FFBh	PCLATU	FD8h	PLUSW2 ⁽¹⁾	FBBh	CCPR2L	F9Bh	OSCTUNE	F7Bh	UEP11
FFAh	PCLATH	FDAh	FSR2H	FBAh	CCP2CON	F9Ah	___ ⁽²⁾	F7Ah	UEP10
FF9h	PCL	FD9h	FSR2L	F89h	___ ⁽²⁾	F99h	___ ⁽²⁾	F79h	UEP9
FF8h	TBLPTRU	FD8h	STATUS	F88h	BAUDCON	F98h	___ ⁽²⁾	F78h	UEP8
FF7h	TBLPTRH	FD7h	TMR0H	F87h	ECCP1DEL	F97h	___ ⁽²⁾	F77h	UEP7
FF6h	TBLPTRL	FD6h	TMR0L	F86h	ECCP1AS	F96h	TRISE ⁽³⁾	F76h	UEP6
FF5h	TABLAT	FD5h	T0CON	F85h	CVRCON	F95h	TRISD ⁽³⁾	F75h	UEP5
FF4h	PRODH	FD4h	___ ⁽²⁾	F84h	CMCON	F94h	TRISC	F74h	UEP4
FF3h	PRODL	FD3h	OSCCON	F83h	TMR3H	F93h	TRISB	F73h	UEP3
FF2h	INTCON	FD2h	HLVDCON	F82h	TMR3L	F92h	TRISA	F72h	UEP2
FF1h	INTCON2	FD1h	WDTCON	F81h	T3CON	F91h	___ ⁽²⁾	F71h	UEP1
FF0h	INTCON3	FD0h	RCON	F80h	SPBRGH	F90h	___ ⁽²⁾	F70h	UEP0
FEFh	INDF ⁽¹⁾	FCFh	TMR1H	FAFh	SPBRG	F8Fh	___ ⁽²⁾	F6Fh	UCFG
FEeh	POSTINC0 ⁽¹⁾	FCEh	TMR1L	FAeh	RCREG	F8eh	___ ⁽²⁾	F6eh	UADDR
FEDh	POSTDEC0 ⁽¹⁾	FCDh	T1CON	FADh	TXREG	F8dh	LATE ⁽³⁾	F6dh	UCON
FECh	PREINC0 ⁽¹⁾	FCCh	TMR2	FACH	TXSTA	F8Ch	LATD ⁽³⁾	F6Ch	USTAT
FE8h	PLUSW0 ⁽¹⁾	FC8h	PR2	FABh	RCSTA	F8Bh	LATC	F6Bh	UEIE
FEAh	FSR0H	FCAh	T2CON	FAAh	___ ⁽²⁾	F8Ah	LATB	F6Ah	UEIR
FE9h	FSR0L	FC9h	SSPBUF	FA9h	EEADR	F89h	LATA	F69h	UIE
FE8h	WREG	FC8h	SSPAD	FA8h	EEDATA	F88h	___ ⁽²⁾	F68h	UIR
FE7h	INDF1 ⁽¹⁾	FC7h	SSPSTAT	FA7h	EECON2 ⁽¹⁾	F87h	___ ⁽²⁾	F67h	UFRMH
FE6h	POSTINC1 ⁽¹⁾	FC6h	SSPCON1	FA6h	EECON1	F86h	___ ⁽²⁾	F66h	UFRML
FE5h	POSTDEC1 ⁽¹⁾	FC5h	SSPCON2	FA5h	___ ⁽²⁾	F85h	___ ⁽²⁾	F65h	SPPCON ⁽³⁾
FE4h	PREINC1 ⁽¹⁾	FC4h	ADRESH	FA4h	___ ⁽²⁾	F84h	PORTE	F64h	SPPEPS ⁽³⁾
FE3h	PLUSW1 ⁽¹⁾	FC3h	ADRESL	FA3h	___ ⁽²⁾	F83h	PORTD ⁽³⁾	F63h	SPPCFG ⁽³⁾
FE2h	FSR1H	FC2h	ADCON0	FA2h	IPR2	F82h	PORTC	F62h	SPPDATA ⁽³⁾
FE1h	FSR1L	FC1h	ADCON1	FA1h	PIR2	F81h	PORTB	F61h	___ ⁽²⁾
FE0h	BSR	FC0h	ADCON2	FA0h	PIE2	F80h	PORTA	F60h	___ ⁽²⁾

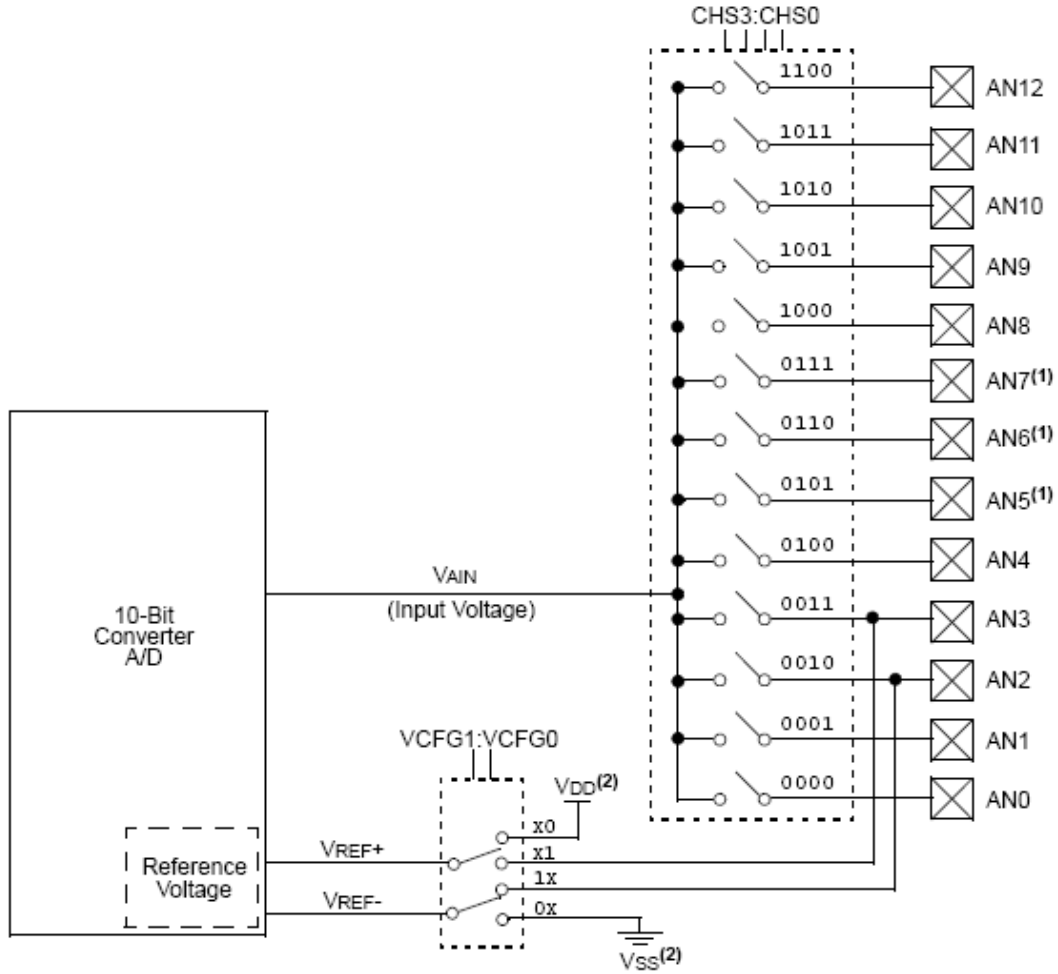
Şekil 2.9 : Özel fonksiyon kayıtları.

2.2.2.4.3 PortA ve TRISA Kayıtları

PortA 8 bit eninde iki yönlü porttur. Buna uygun veri yönlendirici kaydı TRISA'dır. TRISA kaydındaki herhangi bir bit 1 ise, buna uygun çıkış sürücüsü yüksek direnç moduna getirilecektir. TRISA kaydındaki herhangi bir bitin 0 olması, çıkış mandalının içeriğini seçilen pinin üzerine getirir.



Şekil 2.10 : Genel giriş çıkış uygulaması.



Şekil 2.11 : 10 Bitlik A/D girişleri.

Çizelge 2.4 : PortA Kaydedicileri

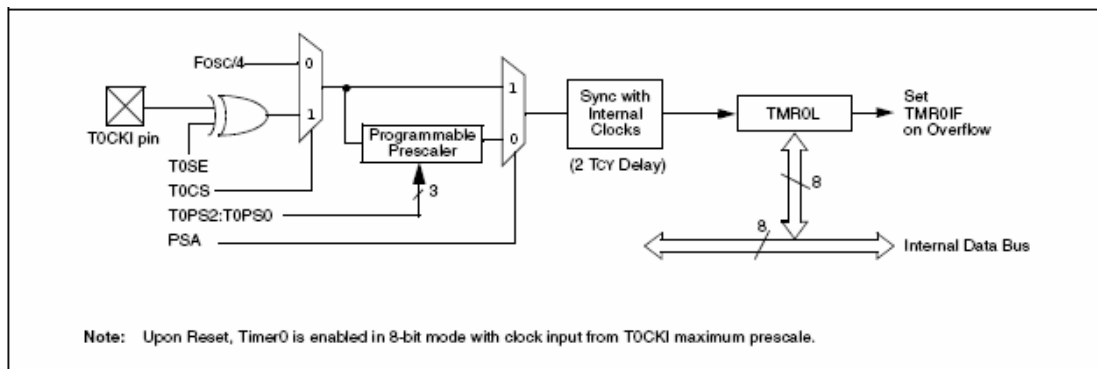
Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Reset Values on page
PORTA	—	RA6 ⁽¹⁾	RA5	RA4	RA3	RA2	RA1	RA0	54
LATA	—	LATA6 ⁽¹⁾	LATA5	LATA4	LATA3	LATA2	LATA1	LATA0	54
TRISA	—	TRISA6 ⁽¹⁾	TRISA5	TRISA4	TRISA3	TRISA2	TRISA1	TRISA0	54
ADCON1	—	—	VCFG1	VCFG0	PCFG3	PCFG2	PCFG1	PCFG0	52
CMCON	C2OUT	C1OUT	C2INV	C1INV	CIS	CM2	CM1	CM0	53
CVRCON	CVREN	CVROE	CVRR	CVRSS	CVR3	CVR2	CVR1	CVR0	53
UCON	—	PPBRST	SE0	PKTDIS	USBEN	RESUME	SUSPND	—	55

2.2.2.5 Timer Modülü

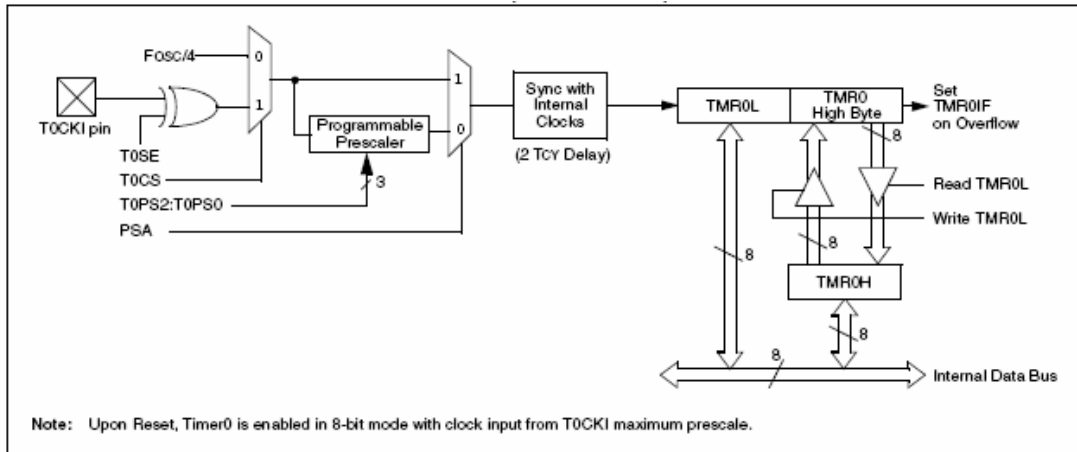
Timer modülü aşağıdaki özelliklere sahiptir:

- Hem 8 hem de 16 bitlik timer/sayaç
- Okunabilir ve yazılabilir
- 8 bitlik programlanabilir prescaler.
- İçten veya dıştan saat ayarı
- Taşma üzeri kesme
- Dış saatin sınır seçimi

Timer modu, TOCS bitinin (OPTION<5>) temizlenmesiyle seçilir. Timer modunda Timer0 modülü her bir komut sürecini uzatır (Prescaler olmaksızın). Eğer TMR0 kaydı yazılıysa, uzama takip eden 2 süreci engeller. Kullanıcı ayarlanan değeri TMR0 kaydına yazarak, bunun etrafından çalışabilir. Sayaç modu TOCS bitinin (OPTION<5>) ayarlanmasıyla seçilir. Bu modda, TMR0, RA4/TOCK1 pininin sınırlarının her bir artışında ya da düşüşünde artacaktır. Genişleyen sınır, TO kaynak sınır seçim biti tarafından, TOSE (OPTION<4>) tarafından belirlenmektedir. TOSE bitinin temizlenmesi artan sınırları seçecektir.

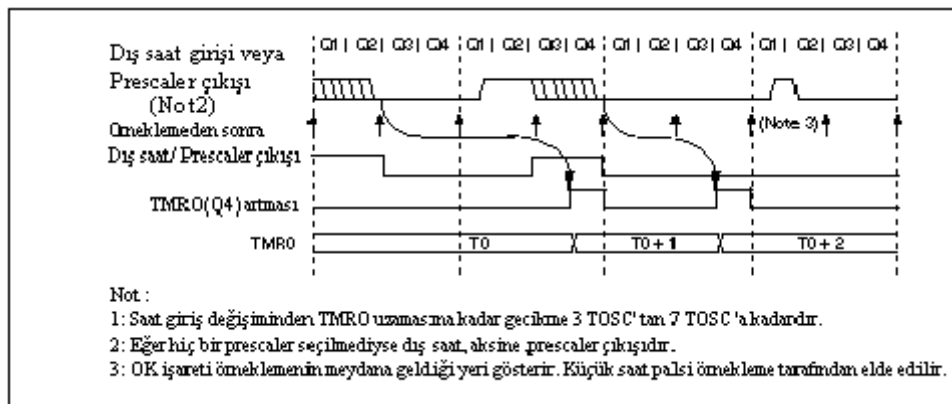


Şekil 2.12 : 8-Bitlik sayaç durumu.



2.2.2.6 Prescaler (Bölücü)

8 Bitlik sayaç Timer0 modülünde veya Watchdog sayacında bulunur. Prescaler dışarıdan verilen sinyali 256 ya kadar bölmeye yarar. Timer0 modülü ile Watchdog sayacı arasında karşılıklı istisna tutulan yalnızca bir tek prescaler mevcuttur. Böylece Timer0 modülüne prescaler ataması, watchdog sayacının prescaler'ı olmadığı anlamına gelmektedir.



2.2.2.7 EEPROM Veri Belleği

EEPROM veri belleği normal işlem boyunca okunabilir ve yazılabilir. Bu bellek direkt olarak kayıt dosya boşluğuna planlanmamıştır. Bunun yerine bu bellek, özel fonksiyon kaydı üzerinden dolaylı olarak adreslenir. Burada bu belleği okuyan ve yazan 4 özel kaydedici (SFR) mevcuttur. Bu kayıtlar:

- EECON1
- EECON2

- EEDATA
- EEADR

EEDATA yazma/okuma için 8 bitlik veri tutar ve EEADR erişilen EEPROM adreslerini saklar. PIC16C84 aygıtı 0H ile 3FH genişliğindeki adresli EEPROM belleğinin 64 bitine sahiptir. EEPROM veri belleği baytları okuma ve yazmaya olanak verir. Baytlar otomatik olarak veri siler ve yeni veri yazar (yazmadan önce siler). EEPROM veri belleği yüksek silme/yazma süreçlerine oranlanmıştır. Yazma zamanı chip üzeri sayaç tarafından denetlenmektedir. Yazma zamanı chipten chipe göre değiştiği gibi, gerilim ve ısı değerlerine göre de değişebilir. Aygıt kod korumalı olduğu zaman, CPU EEPROM belleğini okumaya ve yazmaya devam edebilir ve PIC programlayıcısı artık bu belleğe erişemeyebilir.

2.2.2.7.1 EEPROM Veri Belleğinin Okunması

Veri bellek yerleşimini okumak için, kullanıcı, adresi EEADR kaydına yazmalıdır ve RD kontrol bitini kurmalıdır. (EECON1<0>). Veri sıradaki devirde, EEDATA kaydında mevcuttur, bunun için bu sıradaki komutta okunabilmektedir. EEDATA bu değerleri diğerleri okununcaya kadar veya kullanıcı tarafından yazılınca kadar tutmaktadır. Örnek olarak,

```
MOVLW DATA_EE_ADDR;
MOVWF EEADR          ; Veri belleğinin adresindeki düşük bitlerini oku
BCF EECON1, EEPGD     ; Veri belleğine git
BCF EECON1, CFGS      ; EEPROM'a git
BSF EECON1, RD        ; EEPROM'u oku
MOVF EEDATA, W        ; W = EEDATA
```

2.2.2.7.2 EEPROM Veri Belleğine Yazma

EEPROM veri yerleşimini yazmak için kullanıcı ilkin adresleri EEADR kaydına, verileri EEDATA kaydına yazmalıdır. Daha sonra kullanıcı her bite yazımın başlatması için özel diziye takip etmelidir. Örneğin,

```
MOVLW DATA_EE_ADDR;
MOVWF EEADR          ; Veri belleğinin adresindeki düşük bitlerine yaz
MOVLW DATA_EE_DATA ;
MOVWF EEDATA         ; Veri belleğine yaz
```

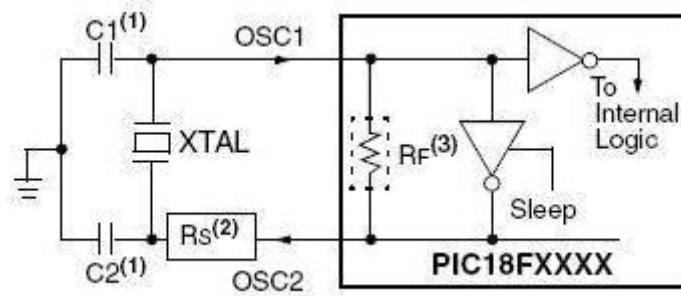
```

BCF EECON1, EPGD      ; Veri belleğine git
BCF EECON1, CFGS      ; EEPROM'a git
BSF EECON1, WREN       ; Yazmayı etkinleştir
BCF INTCON, GIE        ; Kesmeyi etkisizleştir
    MOVLW 55h          ;
Dizi    MOVWF EECON2 ; 55h'yi yaz
Gerekli MOVLW 0AAh    ;
    MOVWF EECON2 ; 0AAh'yi yaz
    BSF EECON1, WR      ; WR bitini kur
BSF INTCON, GIE        ; Kesmeyi etkinleştir
                        ; Yazmayı başlat
BCF EECON1, WREN       ; Yazım tamamlandığında yazımı bitir

```

2.2.2.8 Osilatör Türleri

XT, XTPLL, HS, HSPLL modları, kristal veya seramik rezonatörlerin, OSC1/CLKIN ve OSC2/CLKOUT pinlerine bağlanmalarıyla kurulur.



Şekil 2.15 : Osilatörün devre montaj şeması.

Kristal osilatör için kapasite değeri seçimi Çizelge 2.5'e göre yapılmaktadır.

Çizelge 2.5 : Kristal osilatör için kapasite seçimi

Osilatör Tipi	Kristal Frekansı	Tipik Kapasite Değerleri	
		C1	C2
XT	4 MHZ	27 pF	27 pF
HS	4 MHz	27 pF	27 pF
	8 MHz	22 pF	22 pF
	20 MHz	15 pF	15 pF

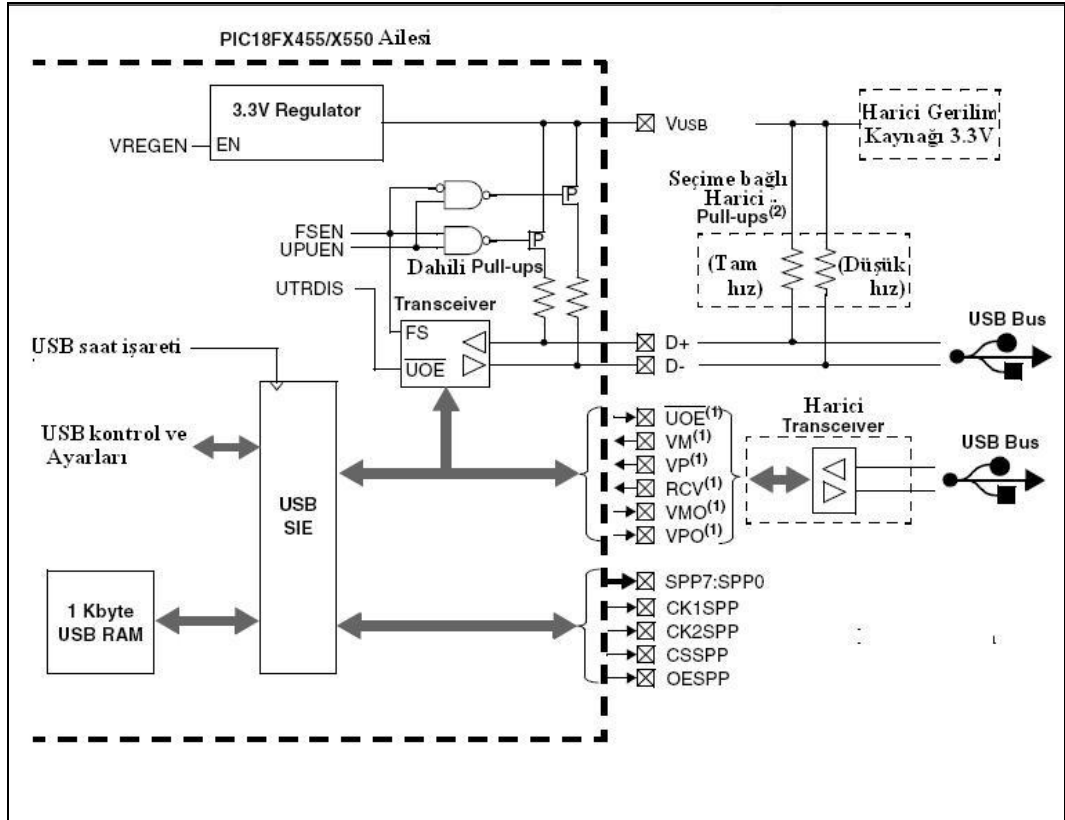
Seramik osilatör için kapasite değeri seçimi Çizelge 2.6'ya göre yapılmaktadır.

Çizelge 2.6 : Seramik osilatör için kapasite seçimi.

Osilatör Tipi	Kristal Frekansı	Tipik Kapasite Değerleri	
		C1	C2
XT	4 MHz	33 pF	33 pF
HS	8 MHz	27 pF	27 pF
	16 MHz	22 pF	22 pF

2.2.2.9 USB

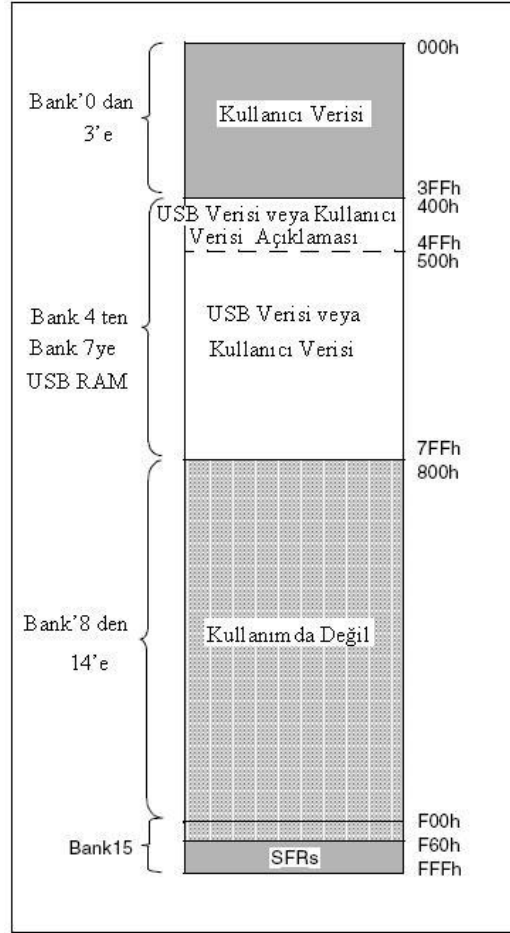
PIC18FX455/X550 serisi cihazlar tam hızda ve düşük hızda USB seri arayüzünü destekler Serial Interface Engine (SIE) PIC ve USB arasında çok hızlı iletişim kurmayı sağlar. İçerisindeki 3.3V'luk regülatör 5V uygulamaları içinde kullanılabilmektedir. Şekil 2.16'da genel hatlarıyla özellikleri gösterilmiştir.



Şekil 2.16 : USB bağlantısı temel yapısı.

2.2.2.9.1 USB RAM

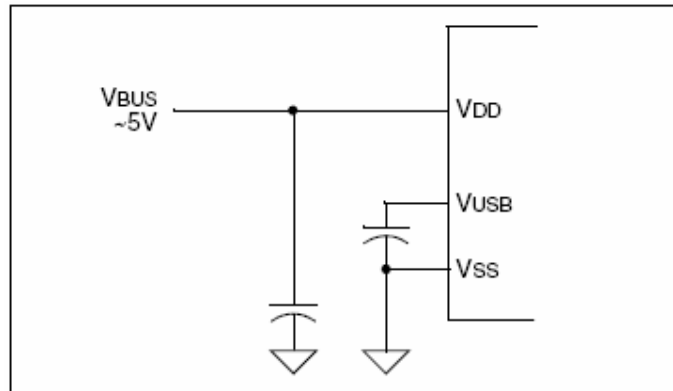
USB bilgisi mikrodenetleyici çekirdeği ve seri USB arayüzü arasında bellek uzayı diye de bilinen USB RAM'ine taşınır. Bu bellek ikili port yapısıyla özel bir bellektir ve normal bellek uzayında 1 Kilobayt için Bank 4'ten Bank 7'ye kadar adreslenir. Şekil 2.17'de de gösterildiği gibi 400h'den 7FFh'ye kadar olan bölüm USB RAM olarak tanımlanır.



Şekil 2.17 : USB RAM.

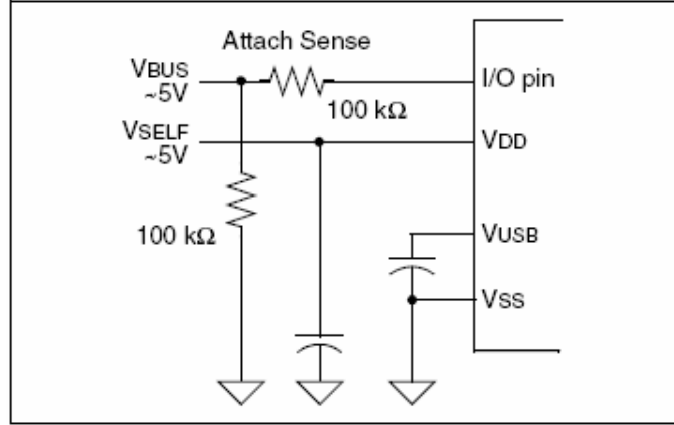
2.2.2.9.2 USB Güç Modları

Birçok USB uygulamaları farklı güç kademelerinde ve farklı ayarlarda yapılmaktadır. En çok kullanılan modları ise yalnızca BUS gücü, yalnızca kendi gücü ve kendi gücünün baskınlığında ikili güç seçenekleridir. Şekil 2.18’de görülen devrede tüm güç USB’den sağlanmaktadır ve bu etkin ve basit bir yöntemdir.



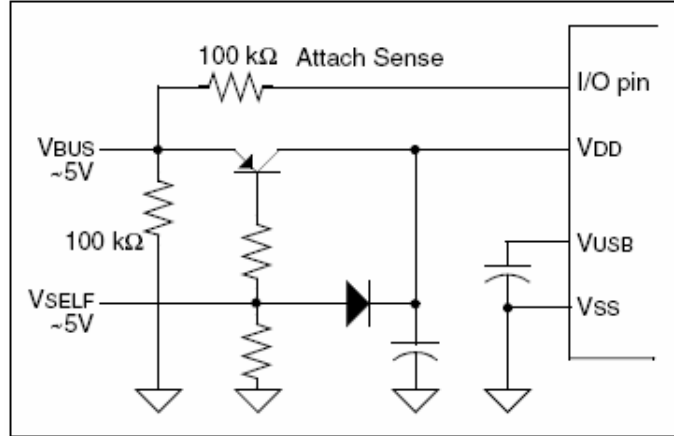
Şekil 2.18 : Yalnızca BUS gücü modu.

Yalnızca kendi gücü modunda Şekil 2.19’da görüleceği üzere USB’den çok küçük bir güç çekilir.



Şekil 2.19 : Yalnızca kendi gücü modu.

Bazı uygulamalarda ikili güç gerekebilir. İkili güç modunda öncelikle bir dahili güç kaynağı kullanılır. Ancak dahili bir güç kaynağı yoksa bu durumda gerekli güç USB’den sağlanır.



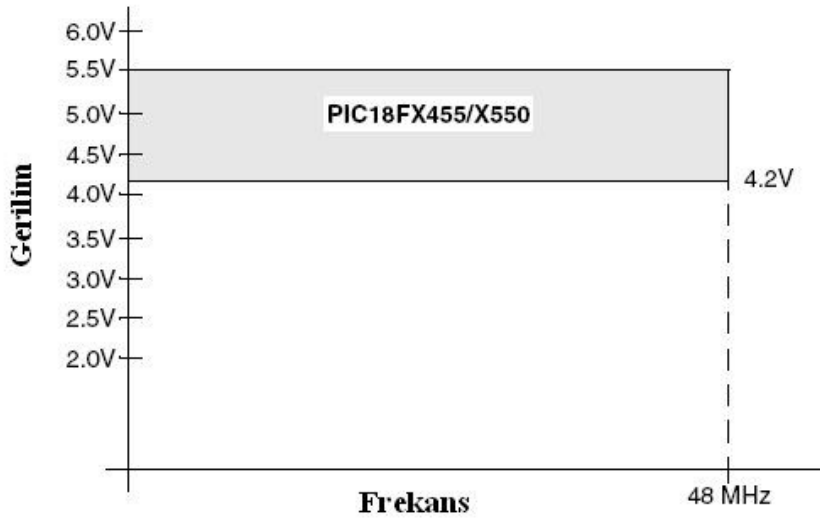
Şekil 2.20 : İkili güç modu.

2.2.2.10 Elektriksel Özellikleri

Sabit gerilim altında çevre sıcaklığı	: 40°C ile +85°C arasında
Depolama durumundaki sıcaklığı	: -65°C ile +150°C arasında
Herhangi bir pinin referans gerilme göre gerilim farkı (VDD, MCLR ve RA4 pinleri dışındaki)	: -0.3V ile (VDD + 0.3V) arasında
VDD ile VSS arasındaki gerilim farkı	: -0.3V ile +7.5V arasında
MCLR ile VSS arasındaki gerilim farkı	: 0V ile +13.25V arasında
Toplam güç kaybı	: 1.0 W

VSS pininin maksimum akımı	: 300 mA
VDD pininin maksimum akımı	: 250 mA
Giriş akım kısılacı, ($V_I < 0$ veya $V_I > V_{DD}$)	: ± 20 mA
Çıkış akım kısılacı, IOK ($V_O < 0$ or $V_O > V_{DD}$)	: ± 20 mA
Herhangi bir I/O pinindeki maksimum akım kaybı	: 25 mA
Tüm portlardaki maksimum akım kaybı	: 200 mA
Tüm portlardaki maksimum akım kazancı	: 200 mA

PIC18F2550'nin çalışma aralığı ise Şekil 2.21'de gösterilmektedir.



Şekil 2.21 : PIC18F2550'nin çalışma aralığı.

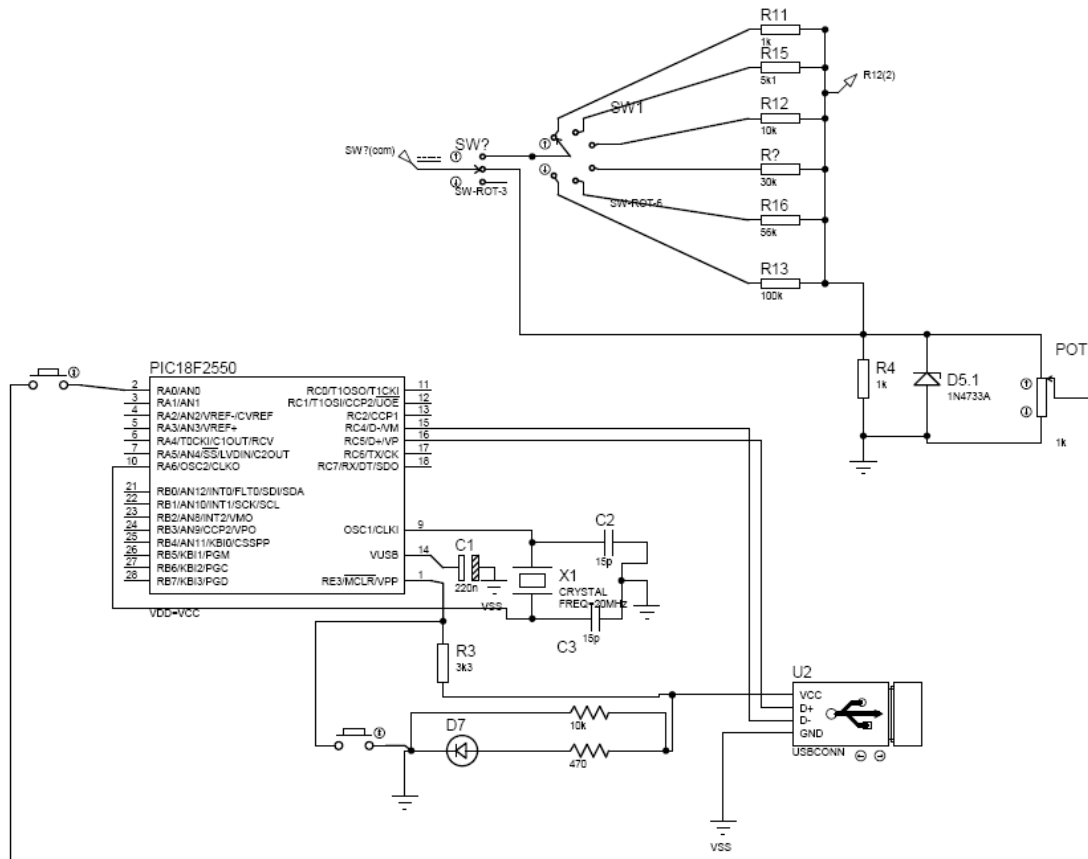
2.3 Bilgisayar Yazılımı

Elektronik alanında C dili vazgeçilmez bir yer edinmiştir. C dili hem insan diline olan yakınlığı hem de makine diline daha az kayıpla çevrilebilme özelliği sayesinde elektronik ürünler ve mikrodenetleyici programlama da oldukça yaygın olarak kullanılmaktadır. Diğer elektronik ürünlerde olduğu gibi Microchip firmasının PIC mikrodenetleyici ürünlerinde de C dili ile programlama yapılabilmektedir. PIC ürünleri için mevcut olan C derleyicileri içinde, PIC ürünlerinin neredeyse tümünü destekleyen, büyük bir oranla ANSI C uyumlu, esnek ve çok kolay bir şekilde mikrodenetleyici programlanmasına izin veren, birçok iletişim protokolü ve çevresel ürünler için hazır kütüphane dosyaları (kontrol fonksiyonları) içeren CCS (Custom Computer Services Inc.) firması ürünü CCS C PIC C Compiler programı oldukça kullanışlıdır. Bu program kullanılarak PIC 18F2550 mikrodenetleyicisinin osiloskop işlevi görmesini sağlayan kodlar oluşturulmuştur.

Arayüz programının hazırlanmasında ise Visual Basic kullanılmıştır. Visual Basic, nesneye dayalı ve olay kontrollü bir programlama dilidir. Ekranda görünen her şey birer nesnedir. Visual Basic tamamen Nesne + Özellik = Değer mantığıyla çalışır. Bu değer nasıl belirtileceği konusunda ise değişkenler yardımcı olurlar. Visual Basic, Microsoft tarafından Basic programlama dili üzerinden geliştirilmiş, olay yönlendirmeli, üst seviye, nesne yönelimli ve görsel bir programlama dilidir.

3. DEVRENİN OLUŞTURULMASI

Öncelikle oluşturulacak devre Proteus programında ISIS bölümü altında oluşturularak benzetimi (simülasyonu) yapılmıştır. Yapılan benzetimde oluşturulan komut dosyası da eklendiğinden eksiksiz bir benzetim olmuştur. Şekil 3.1’de bu çalışma kapsamında tasarlanıp gerçekleştirilen ölçüm cihazının temel devre yapısı gösterilmiştir.



Şekil 3.1 : Ölçüm cihazının temel devre yapısı.

3.1 Gerilim Bölücü ve Aşırı Gerilim Koruması

Mikrodenetleyiciler genellikle 5,5V ve daha alt değerlerdeki gerilimlerde çalışabildiklerinden bu gerilim değerlerinden daha yüksek gerilimleri okuyabilmek için gerilim bölücü devrelere ve mikrodenetleyicinin aşırı gerilime maruz kalmasını önleyecek devrelere ihtiyaç vardır. Aşırı gerilim koruması ve gerilim bölücü işlevini yapan devre sistemi Şekil 3.1’de görülmektedir.

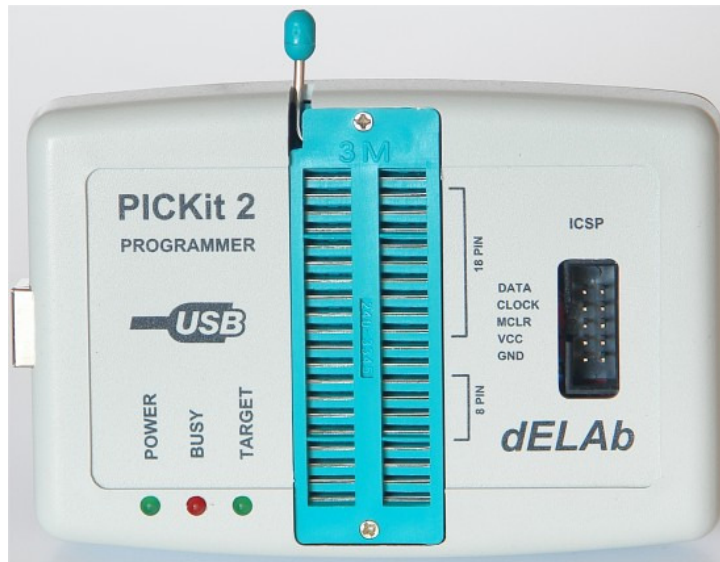
3.2 PIC Mikrokontrolör ve Program Yükleme

PIC18F2550 entegresi Şekil 3.2’de görülmektedir (Microchip, 2009)



Şekil 3.2 : PIC18F2550.

Öncelikle PIC18F2550’ye herhangi bir PIC programlayıcı kullanılarak yazılımın (kaynak kodunun) yüklenmesi gerekmektedir. Şekil 3.3’te görülen PICKit2 adlı PIC programlayıcısı ile programlanmak üzere bilgisayara bağlanır.



Şekil 3.3 : PICKit2 PIC Programlayıcı.

PICKit2 adlı PIC programlayıcısı kullanılarak kaynak dosyası PIC 18F2550 mikrodeneleyicisine gömülür.

3.3 PIC18F2550 İçin Yazılan Kaynak Dosyası

C dilinde yazılan kaynak dosyasına ait komutlar EK.A2’ de verilmiştir. Bu kaynak dosyası içerisinde geçen içerik dosyaları CCS programı kütüphanesinde mevcuttur. Kodlar CCS ile derlenmiş olup hex uzantılı dosya da program tarafından otomatik olarak oluşturulmuştur. Programda baud 115200 ve PLL5 aktif hale getirilerek 20 MHz kristal ile 48 MHz çalışma frekansı elde edilmiştir.

Çizelge 3.1 : Örnekleme frekansı ve hata oranları.

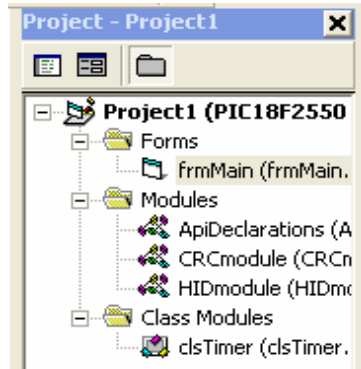
BAUD RATE (K)	SYNC = 0, BRGH = 1, BRG16 = 1 or SYNC = 1, BRG16 = 1											
	Fosc = 40.000 MHz			Fosc = 20.000 MHz			Fosc = 10.000 MHz			Fosc = 8.000 MHz		
	Actual Rate (K)	% Error	SPBRG value (decimal)	Actual Rate (K)	% Error	SPBRG value (decimal)	Actual Rate (K)	% Error	SPBRG value (decimal)	Actual Rate (K)	% Error	SPBRG value (decimal)
0.3	0.300	0.00	33332	0.300	0.00	16665	0.300	0.00	8332	300	-0.01	6665
1.2	1.200	0.00	8332	1.200	0.02	4165	1.200	0.02	2082	1200	-0.04	1665
2.4	2.400	0.02	4165	2.400	0.02	2082	2.402	0.06	1040	2400	-0.04	832
9.6	9.606	0.06	1040	9.596	-0.03	520	9.615	0.16	259	9615	-0.16	207
19.2	19.193	-0.03	520	19.231	0.16	259	19.231	0.16	129	19230	-0.16	103
57.6	57.803	0.35	172	57.471	-0.22	86	58.140	0.94	42	57142	0.79	34
115.2	114.943	-0.22	86	116.279	0.94	42	113.636	-1.36	21	117647	-2.12	16

Çizelge 3.2 : 20 MHz osilatör frekansı ve PLL÷5 ile CPU frekansı.

Input Oscillator Frequency	PLL Division (PLLDIV2:PLLDIV0)	Clock Mode (FOSC3:FOSC0)	MCU Clock Division (CPUDIV1:CPUDIV0)	Microcontroller Clock Frequency
20 MHz	÷5 (100)	HS, EC, ECIO	None (00)	20 MHz
			+2 (01)	10 MHz
			+3 (10)	6.67 MHz
			+4 (11)	5 MHz
		HSPLL, ECPLL, ECPIO	+2 (00)	48 MHz
			+3 (01)	32 MHz
			+4 (10)	24 MHz
			+6 (11)	16 MHz

3.4 PIC-PC İletişim Arayüzü Yazılımı

Visual Basic programında nesnesel olarak yapılan programın genel çalışma şeması aşağıda görülmektedir (Şekil 3.6)



Şekil 3.4 : Geliştirilen programın Visual Basic proje penceresi.

3.5 USB Bağlantısı

Devrede PIC18F2550 USB bağlantısı ile bilgisayara bağlanmakta ve aynı zamanda kaynak gerilimini de bu bağlantı yolu ile sağlamaktadır.

4. ÜRETİM MALİYETİ

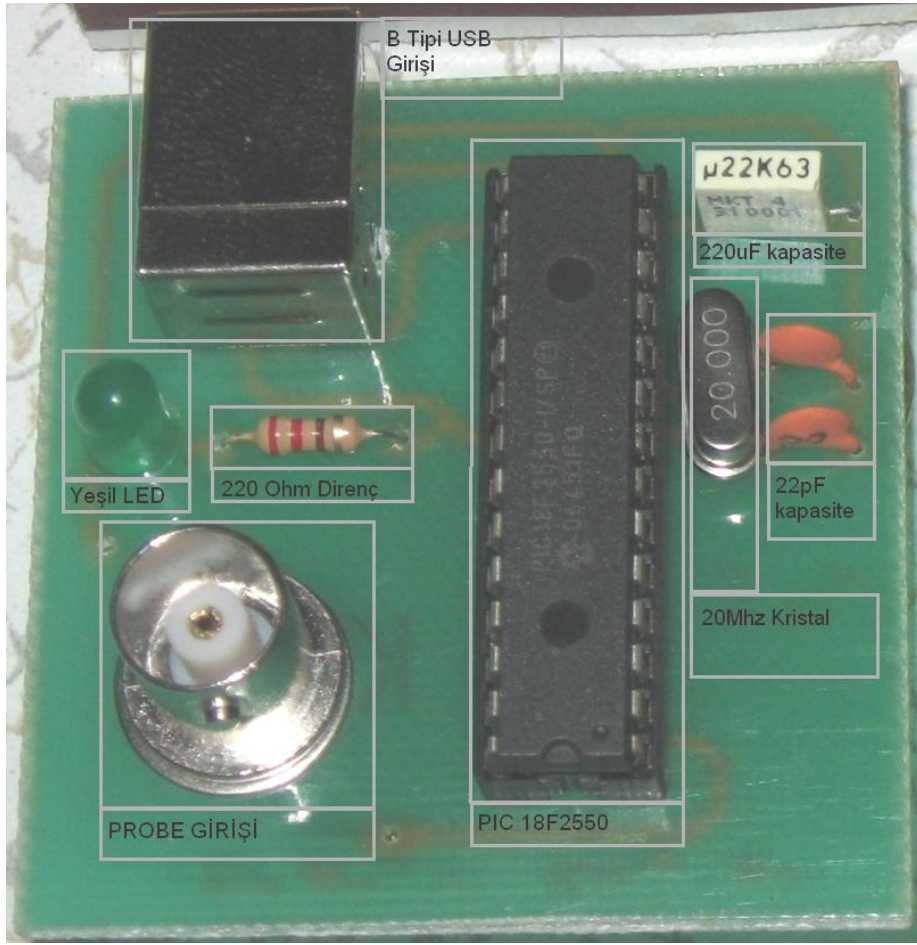
Devrenin üretilmesi için yapılan harcamalar USD (ABD) para birimi cinsinden aşağıda verilmiştir.

Çizelge 4.1 : Yapılan harcamalar.

Malzemenin İsmi	Fiyatı (USD, \$)
PIC18F2550	11
Devre oluşturma yüzeyi (Board)	2
Osiloskop Probu ve Girişi	13
USB Ara kablosu (1 metre)	2
Led	0.25
20 MHz Osilatör	0.75
Dirençler	1
Kapasiteler	1
Toplam	31

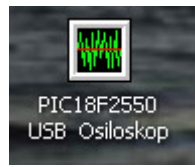
Burada 1 USD (ABD) = 1,60 TL alınarak ve %18 KDV oranı hesaba katılarak yapılan harcamaların toplam tutarı Türk Lirası cinsinden yaklaşık 60 TL'dir.

Devre üzerindeki elemanlar ise şekil 5.3' te görülmektedir.



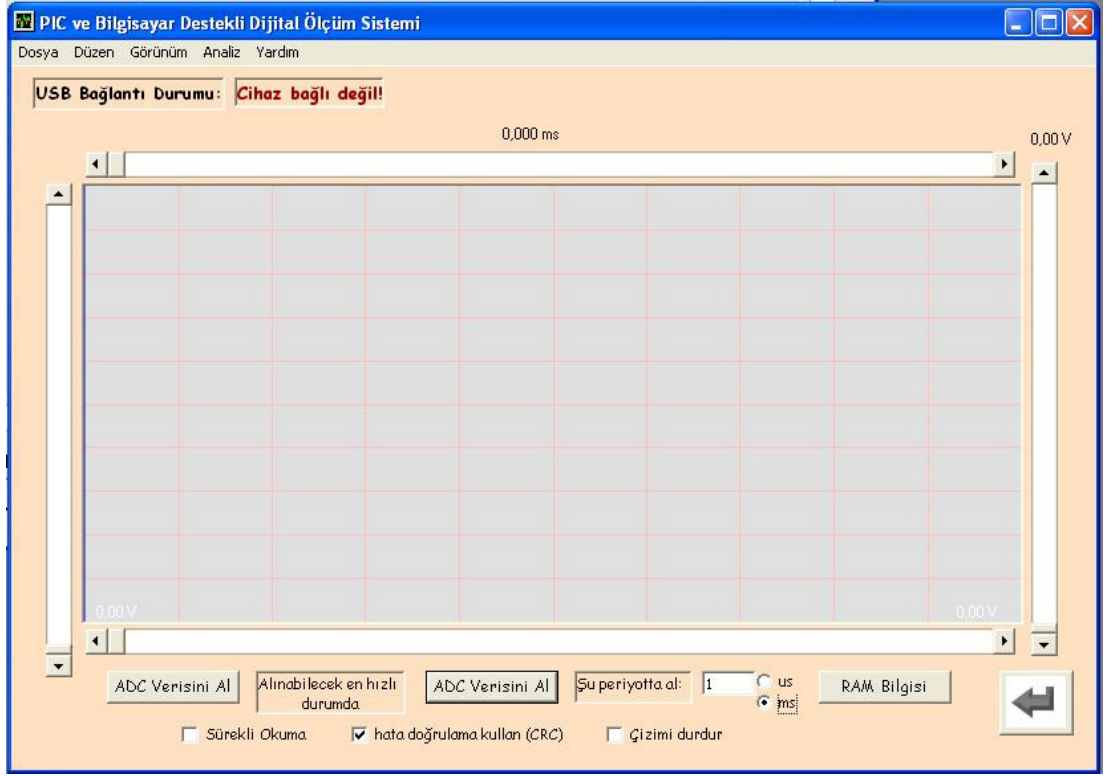
Şekil 5.3 : Baskı devre ve elemanları

Yazılan yazılımın (programın) masaüstündeki görünümü Şekil 5.4 'teki gibidir.



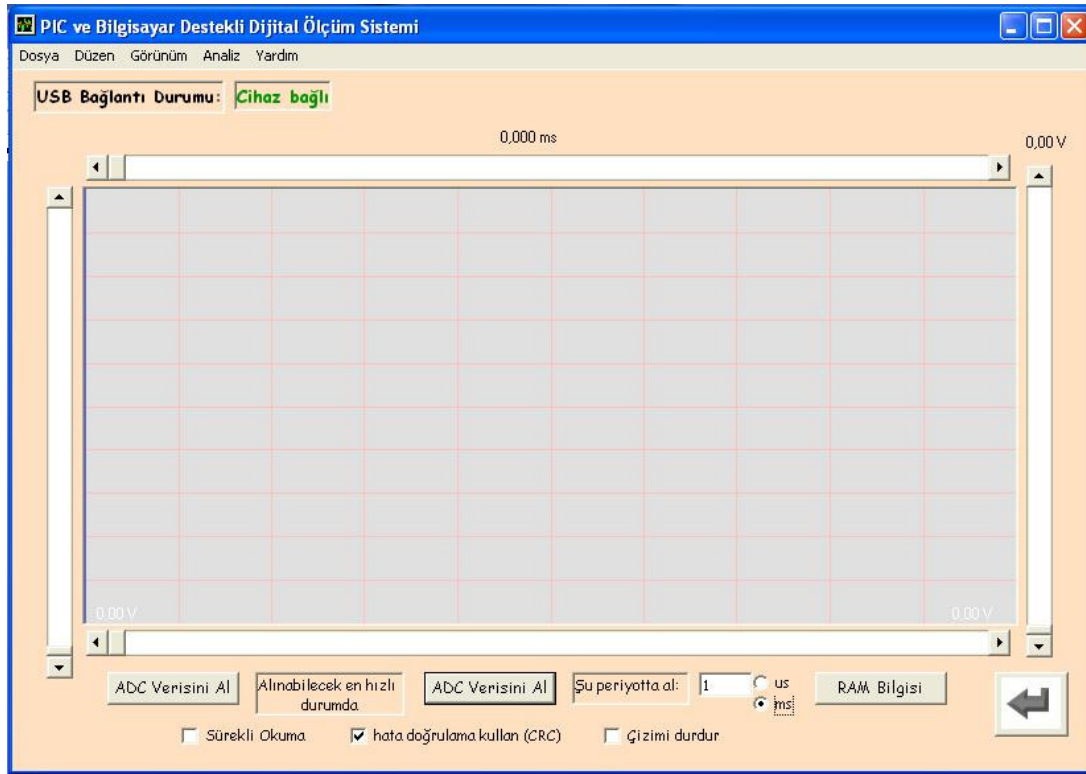
Şekil 5.4 : Programın masaüstü kısayol ikonu.

Programın ikonuna çift tıklayarak program çalıştırılır ve ardından açılan pencere Şekil 5.5'teki penceredir.

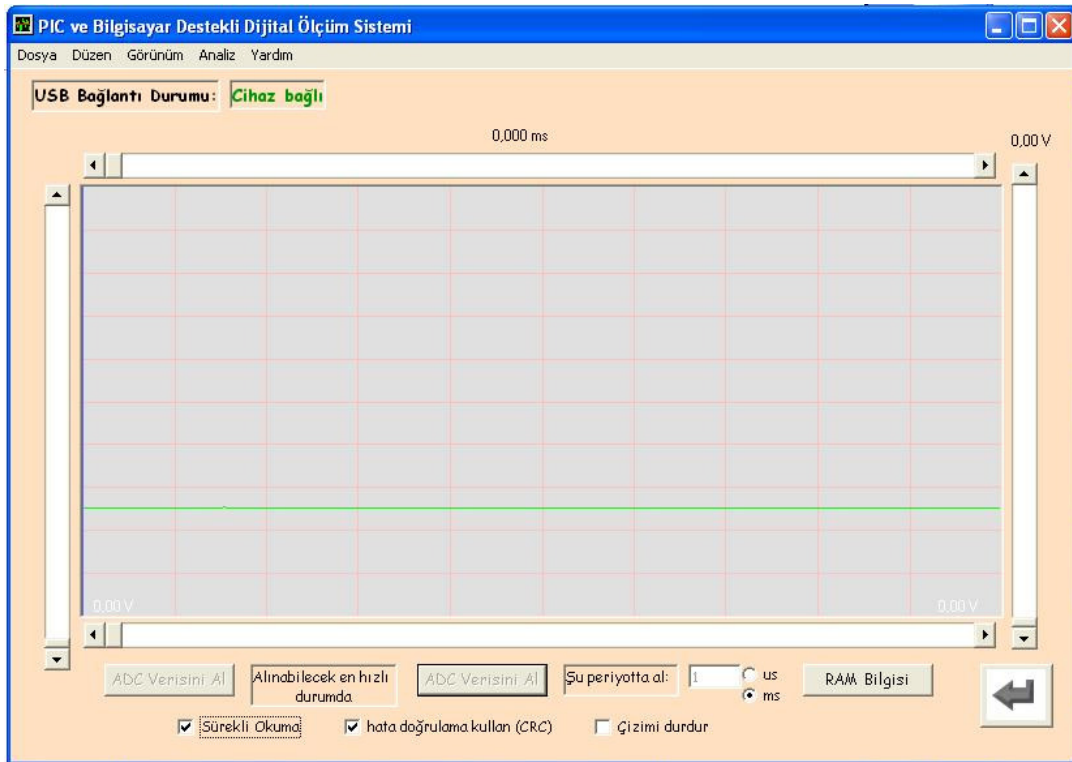


Şekil 5.5 : Cihazın bağlı olmadığı konumda yazılım görünümü.

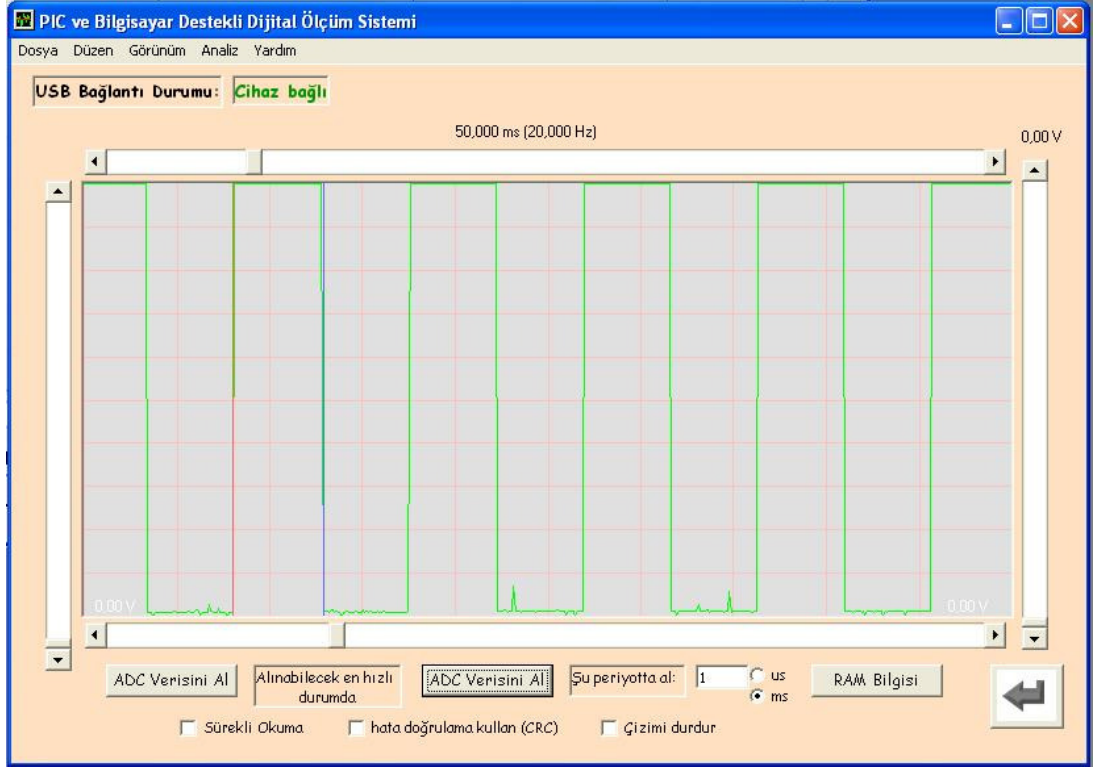
USB-Osiloskop bilgisayara bağlanıp tanındıktan sonra bağlantı durumu sekmesinin karşısındaki “cihaz bağlı değil” ibaresi “cihaz bağlı” ibaresine dönüşür (Şekil 5.6). Artık USB-Osiloskop kullanıma hazırdır. Alınan dalga şekilleri kaydedilebileceği gibi kaydedilen veriler de daha sonra kaydedilen yerden çağırılıp görüntülenebilmektedir. Şekil 5.7 ve Şekil 5.8’de bu düzenle kaydedilen iki dalga şekli gösterilmektedir.



Şekil 5.6 : Cihaz bağlandığı konumda yazılım görünümü.



Şekil 5.7 : 1,2 V değerinde 1 adet AAA pil ile deney.



Şekil 5.8 : 20 kHz 5 V kare dalga.

Diğer örnek ölçümler EK A1’de gösterilmektedir.

5.1 Gerçekleştirilen USB Osiloskobun Özellikleri

- DC 100 kHz,1 Kanallı
- Örnekleme Hızı:1MS/s
- Dikey ölçülebilirlik: 5V
- Yatay tarama:1µs
- Otomatik trigger
- Otomatik tanıma (plug and play)
- Kayıt/Tekrar görüntüleme

Sonuç olarak bilgisayara USB bağlantısı ile bağlanabilen, yapılan deneysel ölçümlerle de en çok 5 V genlikli ve 100 kHz frekanslı işaretlerin ölçümünü yapabilen dijital osiloskop yapısı gerçekleştirilmiştir. Bu devrenin işareti alan kısmından önce bir max114 analog dijital çeviricisi konarak, devre 1 MHz ölçüm yapabilir hale getirilebilmektedir. Ancak PIC mikrodenetleyicisinin RAM’e kaydetme hızı 1024 kps olduğundan maksimum 1 MHz frekanslı işaretleri ölçebilen bir osiloskop yapılabilir.

KAYNAKLAR

- [1] **Anand, M. M. V.**, 2002. *PC-based oscilloscope (Electronics for you)*, SANI THEO Press.
- [2] **Axelson, J.**, “*USB Code Firmware and Related Applications*”
<http://www.lvr.com/usbcodes.htm>, alındığı tarih 03.11.2008.
- [3] **Bredendiek, J.**, “*USB-Interface-Beispiel*”,
<http://www.sprut.de/electronic/pic/8bit/18f/programm/usb2550/usb2550.htm>,
alındığı tarih 12.12.2008.
- [4] **Cholewiak, S.**, “*PIC18F2550 USB HID Oscilloscope*”,
<http://www.semifluid.com/?p=24>, alındığı tarih 05.01.2009.
- [5] **Çiçek, S.**, 2008. *CCS C İle PIC Programlama* İstanbul: Altaş Yayıncılık.
- [6] **Dhananjay, V. G.**, 2000. *Programming the Parallel Port: Interfacing the PC for Data Acquisition & Process Control*, PAP Press.
- [7] **Engdahl, T.**, 2005. Parallel port interfacing made easy: Simple circuits and programs to show how to use PC parallel port output capabilities.
- [8] **Gallagher, R. S.**, 1994. *Computer Visualization: Graphics Techniques for Engineering and Scientific Analysis*, CRC Press.
- [9] **Gerçek, C.**, 2002. *Her Yönüyle USB*. İstanbul: Bileşim Yayıncılık.
- [10] **Göksu, S.**, 2002. *Adım Adım Microsoft Visual Basic 6.0*, Arkadaş Yayınevi, Ankara.
- [11] **Grocutt, T.**, “*Digital Storage Oscilloscope*”,
http://www.chocbar.demon.co.uk/download/docs/dso_doc.pdf, alındığı tarih 21.03.2008.
- [12] **Gropp, W., Lusk, E., Skjellum, A.**, 1999. *Using MPI: Portable Parallel Programming with the Message Passing Interface (Scientific and Engineering Computation Series)*, MIT Press, Paperback, 2nd edition, 258-371.
- [13] **Insam, E.**, 2002. “A Simple USB Oscilloscope For The PC”, Published In Electronics World: February/March 2002
- [14] **McCord, C. K.**, “*Low cost PC-based quad channel real-time / storage oscilloscope*”, http://www.cmccord.co.uk/FYP/Download/Final_Report.pdf,
alındığı tarih 10.03.2009.
- [15] **Microchip**, PIC18F2550 Datasheet
<http://ww1.microchip.com/downloads/en/DeviceDoc/39632D.pdf>, alındığı tarih 07.03.2008.
- [16] **Microsoft Official Curriculum**, 2002. *Mastering Visual Basic 6 Beginner*, official complete course notes.

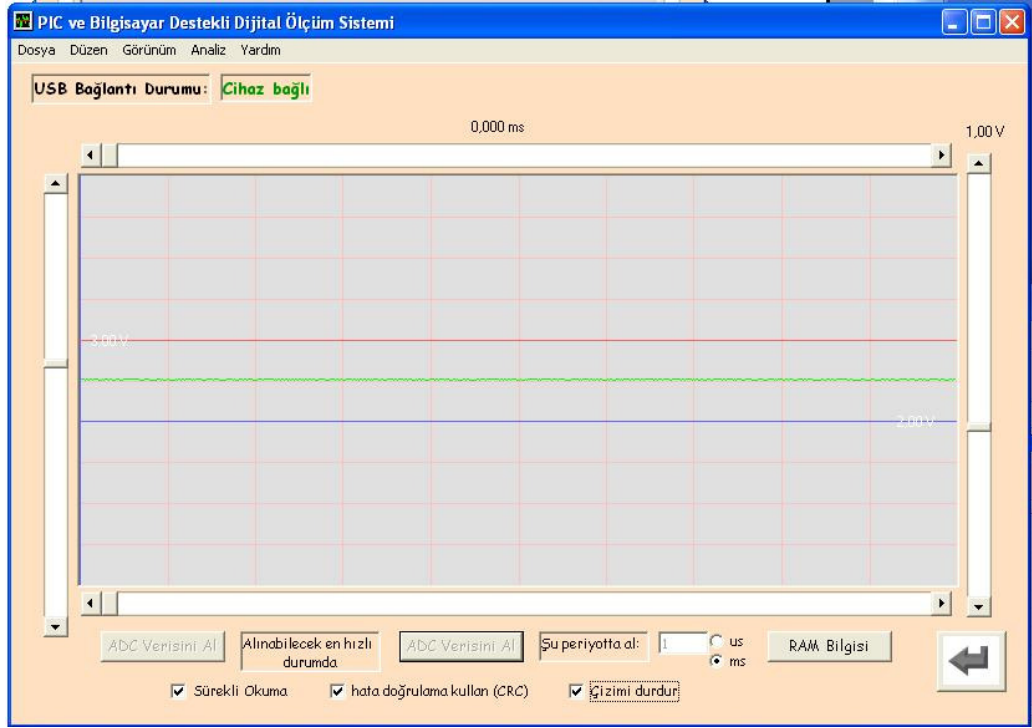
- [17] **Microsoft Official Curriculum**, 2002. *Mastering Visual Basic 6 Development*, official complete course notes.
- [18] **Microsoft Visual Basic**, www.bilgisayarogren.com/basic1.doc alındığı tarih 03.04.2009.
- [19] **Plassche, R.**, 1994. *Integrated Analog-to-Digital and Digital-to-Analog Converters*, Kluwer Academic Publishers.
- [20] **Proakis, J.**, Manolakis D., 1992. *Digital Signal Processing: Principles, Algorithms, and Applications*, New York, Macmillan Publishing Company.
- [21] **Razavi, B.**, 1995. *Principles of Data Conversion System Design*, IEEE Press, Inc., New York.
- [22] **Rozental, A., Lempart, L., Ferrara, P.**, “*PC-Based Digital Oscilloscope*“
http://courses.ece.uiuc.edu/ece445/projects/spring2006/project25_final_paper.doc, alındığı tarih 05.01.2009.
- [23] **Wieser, W.**, “*USB Live Oscilloscope*”,
<http://www.cip.physik.uni-muenchen.de/~wwieser/elec/analysis/USB-LiveOsci/>, alındığı tarih 21.03.2008.

EKLER

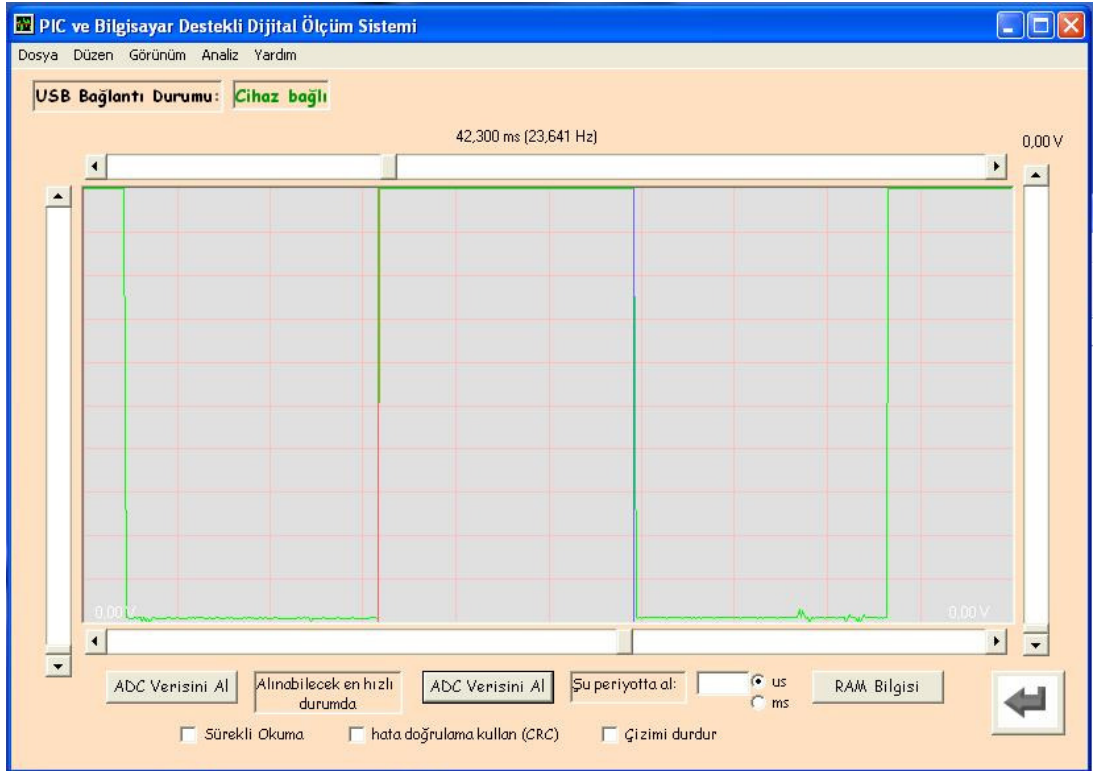
EK A.1 : Örnek Ölçüm Sonuçları

EK A.2 : PIC18F2550 Kaynak Kodları

EK A.1 : Örnek Ölçüm Sonuçları

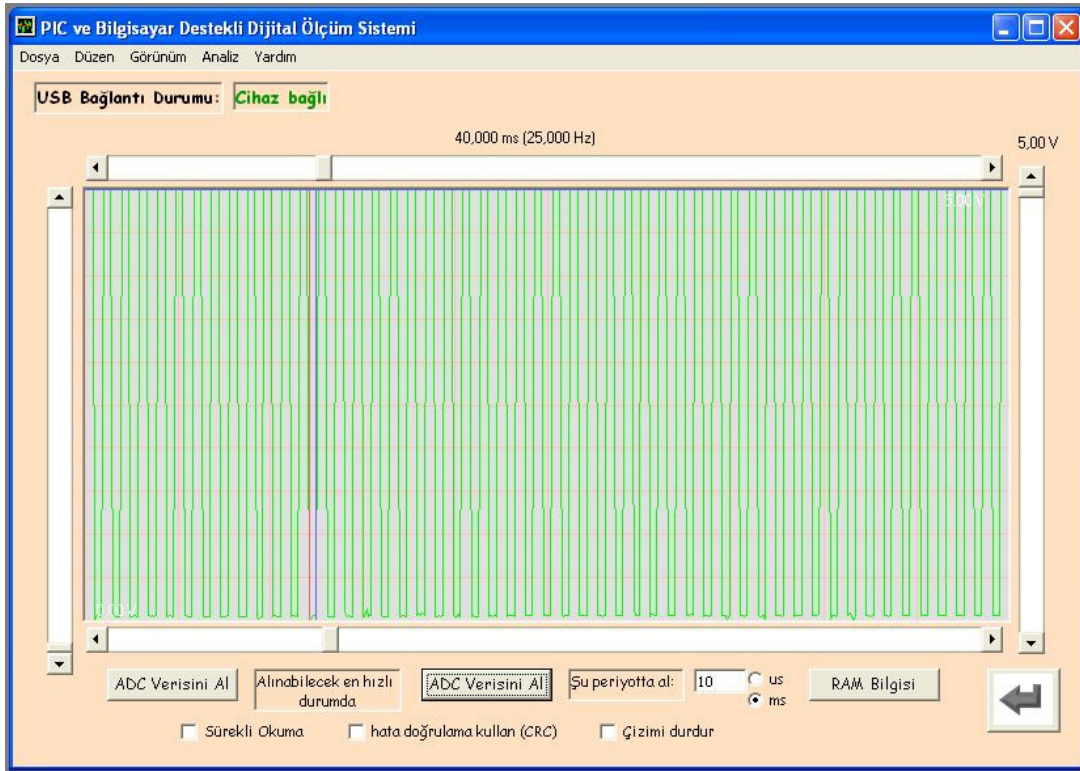


Şekil A.1.1 : 1.2V değerinde 2 adet AAA pil ile deney.

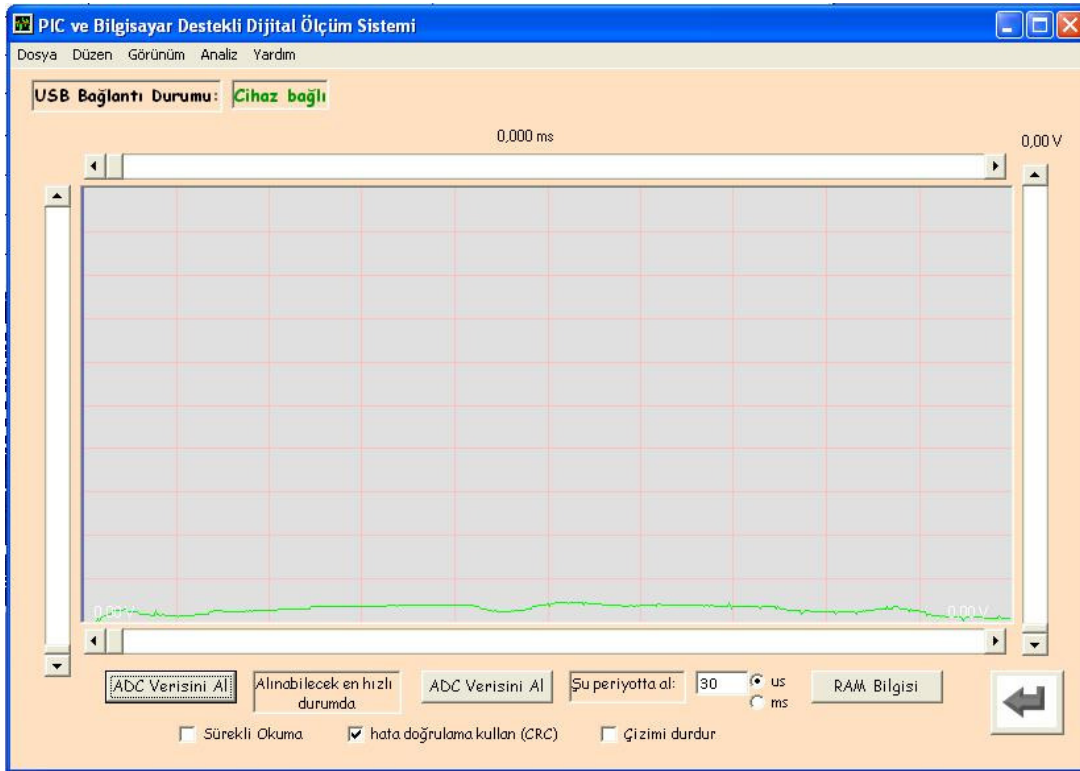


Şekil A.1.2 : 23 kHz 5 V kare dalga.

Farklı frekanslardaki kare dalga ölçüm sonuçları ise aşağıdaki gibidir.

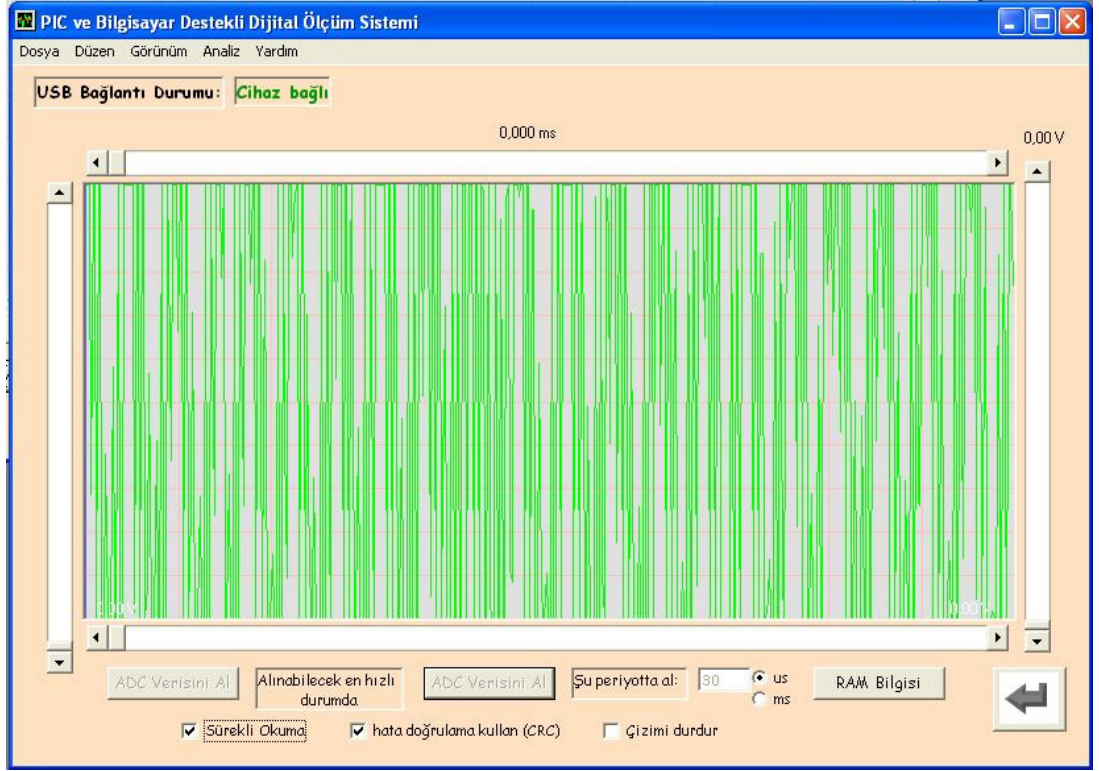


Şekil A.1.3 : 25 kHz 5 V kare dalga.



Şekil A.1.4 : ADC verisinin alınabilecek en hızlı durumda alınması.

Sürekli okuma durumunda cihaz herhangi bir elektriksel büyüklüğe bağlı değilse ortamdaki mevcut gürültüyü okur ve gürültüyü görmemize olanak sağlar.



Şekil A.1.5 : Sürekli okuma durumu.

EK A.2 : PIC18F2550 Kaynak Kodları

```
////////////////////////////////////////
//          PIC18F2550 USB Osiloskop
// Filename   : 18F2550 USB Osiloskop.c
// Programmer: İbrahim SÖYLER 504061032
// Version    : Version 1.0
////////////////////////////////////////

#define __USB_PIC_PERIF__ 1
#include <18F2550.h>
#define ADC=10
#define fuses HSPLL, NOWDT, NOPROTECT, NOLVP, NODEBUG, USBDIV, PLL5,
CPUDIV1, VREGEN
#define delay(clock=48000000)
#define use rs232(stream=PC, baud=115200, xmit=PIN_C6, rcv=PIN_C7, ERRORS)

// CCS kütüphanesi tanımlamaları
#define USB_HID_DEVICE TRUE
#define USB_EP1_TX_ENABLE USB_ENABLE_INTERRUPT
#define USB_EP1_TX_SIZE 64
#define USB_EP1_RX_ENABLE USB_ENABLE_INTERRUPT
#define USB_EP1_RX_SIZE
// CCS USB kütüphanesi tanımlamaları
#include <pic18_usb.h>
#include <usb_desc_hid 8-byte.h>
#include <usb.c>

void usb_debug_task(void) {
    static int8 last_connected;
    static int8 last_enumerated;
    int8 new_connected;
    int8 new_enumerated;

    new_connected=usb_attached();
    new_enumerated=usb_enumerated();
}
```

```

if (new_connected && !last_connected) {
    printf("\r\n\nUSB connected, waiting for enumeration...");}
if (!new_connected && last_connected) {
    printf("\r\n\nUSB disconnected, waiting for connection...");}
if (new_enumerated && !last_enumerated) {
    printf("\r\n\nUSB enumerated by PC/HOST");}
if (!new_enumerated && last_enumerated) {
    printf("\r\n\nUSB unenumerated by PC/HOST, waiting for enumeration...");}

last_connected=new_connected;
last_enumerated=new_enumerated;
}
#INT_RDA
void serial_isr()
{
    int8 uReceive;
    disable_interrupts(GLOBAL);
    uReceive = fgetc(PC);
    switch (uReceive) {
        case 0x12: {
            if (fgetc(PC) == 0x34 & fgetc(PC) == 0x56 & fgetc(PC) == 0x78 &
fgetc(PC) == 0x90) #asm reset #endasm
                }
            break;
        }
    }
    enable_interrupts(GLOBAL);
}

int calc_crc(int oldcrc, int newbyte) {
    int shift_reg, data_bit, sr_lsb, fb_bit, j;
    shift_reg=oldcrc;
    for(j=0; j<8; j++) { // for each bit
        data_bit = (newbyte >> j) & 0x01;
        sr_lsb = shift_reg & 0x01;
        fb_bit = (data_bit ^ sr_lsb) & 0x01;

```

```

    shift_reg = shift_reg >> 1;
    if (fb_bit)
        shift_reg = shift_reg ^ 0x8c;
    }
    return(shift_reg);
}

#define theSampleSize          512
#define usbConfirmAction       0
#define usbCheckStatus         1
#define usbReadRam              2
#define usbWriteRam             3
#define usbReadADC              4
#define usbReadADCnTimes        5
#define usbReadADCPeriod        6
#define usbReadADCnTimesMS      7
#define usbClearRam             8
#define usbSetRamByte           9
#define usbSetUseCRC            10
#define usbClearUseCRC          11
#define usbReadADCnTimesUS      12
#define usbError                66

void main() {
    int1 useCRC;
    int8 in_data[8];
    int8 out_data[8];
    int8 adcData[theSampleSize];
    int8 theCRC, tempADC;
    int16 n, approxUS, approxMS, period;

    SETUP_ADC_PORTS(AN0);
    SETUP_ADC(ADC_CLOCK_DIV_64);
    SETUP_TIMER_0(RTCC_INTERNAL|RTCC_DIV_1);
    SETUP_TIMER_1(T1_DISABLED);

```

```

SETUP_TIMER_2(T2_DISABLED, 127, 1);
SETUP_TIMER_3(T3_INTERNAL | T3_DIV_BY_8);
SETUP_CCP1(CCP_OFF);
SETUP_CCP2(CCP_OFF);
enable_interrupts(INT_RDA);
enable_interrupts(GLOBAL);
usb_init();
useCRC = true;
set_adc_channel(0);
delay_ms(1);
while (TRUE) {
    usb_task();
    usb_debug_task();
    if (usb_enumerated()) {
        if (usb_kbhit(1)) {
            usb_get_packet(1, in_data, 8);
            if (useCRC) {
                theCRC = 0;
                theCRC = calc_crc(theCRC,in_data[0]);
                theCRC = calc_crc(theCRC,in_data[1]);
                theCRC = calc_crc(theCRC,in_data[2]);
                theCRC = calc_crc(theCRC,in_data[3]);
                theCRC = calc_crc(theCRC,in_data[4]);
                theCRC = calc_crc(theCRC,in_data[5]);
                theCRC = calc_crc(theCRC,in_data[6]);
            }
            else {
                theCRC = in_data[7];
            }
            if (theCRC == in_data[7]) {
                out_data[0] = 255;
                out_data[1] = 255;
                out_data[2] = 255;
                out_data[3] = 255;
            }
        }
    }
}

```

```

out_data[4] = 255;
out_data[5] = 255;
out_data[6] = 255;
switch (in_data[0]) {
    case usbReadRam: {
        if (make16(in_data[1],in_data[2]) <= theSampleSize) {
            out_data[0] = usbConfirmAction;
            out_data[1] = usbReadRam;
            out_data[2] = in_data[1];
            out_data[3] = in_data[2];
            out_data[4] = adcData[make16(in_data[1],in_data[2])];
        }
        else {
            out_data[0] = usbError;
            out_data[1] = usbReadRam;
            out_data[2] = in_data[1];
            out_data[3] = in_data[2];
        }
    }
    break;
    case usbWriteRam: {
        if (make16(in_data[1],in_data[2]) <= theSampleSize) {
            adcData[make16(in_data[1],in_data[2])] = in_data[3];
            out_data[0] = usbConfirmAction;
            out_data[1] = usbWriteRam;
            out_data[2] = in_data[1];
            out_data[3] = in_data[2];
            out_data[4] = in_data[3];
        }
        else {
            out_data[0] = usbError;
            out_data[1] = usbWriteRam;
            out_data[2] = in_data[1];
            out_data[3] = in_data[2];
        }
    }
}

```

```

        out_data[4] = in_data[3];
    }
}
break;
case usbReadADC: {
    tempADC = READ_ADC();
    out_data[0] = usbConfirmAction;
    out_data[1] = usbReadADC;
    out_data[2] = tempADC;
}
break;
case usbReadADCnTimes: {
    if (make16(in_data[1],in_data[2]) <= theSampleSize) {
        set_timer3(0);
        for (n=0;n<make16(in_data[1],in_data[2]);n++)
        {
            adcData[n] = READ_ADC();
        }
        period = get_timer3(); // 1000/((clock/4)/8) for ms
        out_data[0] = usbConfirmAction;
        out_data[1] = usbReadADCnTimes;
        out_data[2] = in_data[1];
        out_data[3] = in_data[2];
    }
    else {
        out_data[0] = usbError;
        out_data[1] = usbReadADCnTimes;
        out_data[2] = in_data[1];
        out_data[3] = in_data[2];
    }
}
break;
case usbReadADCPeriod: {
    out_data[0] = usbConfirmAction;

```

```

    out_data[1] = usbReadADCPeriod;
    out_data[2] = make8(period,1);
    out_data[3] = make8(period,0);
}
break;
case usbReadADCnTimesUS: {
    if (make16(in_data[1],in_data[2]) <= theSampleSize) {
        approxUS = make16(in_data[3],in_data[4]);
        for (n=0;n<make16(in_data[1],in_data[2]);n++)
        {
            set_timer3(0);
            adcData[n] = READ_ADC();
            while (get_timer3() * 2/3 < approxUS);
        }
        out_data[0] = usbConfirmAction;
        out_data[1] = usbReadADCnTimesUS;
        out_data[2] = in_data[1];
        out_data[3] = in_data[2];
        out_data[4] = in_data[3];
        out_data[5] = in_data[4];
    }
    else {
        out_data[0] = usbError;
        out_data[1] = usbReadADCnTimesUS;
        out_data[2] = in_data[1];
        out_data[3] = in_data[2];
        out_data[4] = in_data[3];
        out_data[5] = in_data[4];
    }
}
break;
case usbReadADCnTimesMS: {
    if (make16(in_data[1],in_data[2]) <= theSampleSize) {
        approxMS = make16(in_data[3],in_data[4]);

```

```

    for (n=0;n<make16(in_data[1],in_data[2]);n++)
    {
        set_timer3(0);
        adcData[n] = READ_ADC();
        while (get_timer3() * 1/1500 < approxMS);
    }
    out_data[0] = usbConfirmAction;
    out_data[1] = usbReadADCnTimesMS;
    out_data[2] = in_data[1];
    out_data[3] = in_data[2];
    out_data[4] = in_data[3];
    out_data[5] = in_data[4];
}
else {
    out_data[0] = usbError;
    out_data[1] = usbReadADCnTimesMS;
    out_data[2] = in_data[1];
    out_data[3] = in_data[2];
    out_data[4] = in_data[3];
    out_data[5] = in_data[4];
}
}
break;
case usbClearRam: {
    for (n=0;n<512;n++)
    {
        adcData[n] = 0;
    }
    out_data[0] = usbConfirmAction;
    out_data[1] = usbClearRam;
}
break;
case usbSetRamByte: {
    for (n=0;n<512;n++)

```

```

        {
            adcData[n] = in_data[1];
        }
        out_data[0] = usbConfirmAction;
        out_data[1] = usbSetRamByte;
        out_data[2] = in_data[1];
    }
    break;
case usbSetUseCRC: {
    useCRC = true;
    out_data[0] = usbConfirmAction;
    out_data[1] = usbSetUseCRC;
}
break;
case usbClearUseCRC: {
    useCRC = false;
    out_data[0] = usbConfirmAction;
    out_data[1] = usbClearUseCRC;
}
break;
}
if (useCRC) {
    theCRC = 0;
    theCRC = calc_crc(theCRC,out_data[0]);
    theCRC = calc_crc(theCRC,out_data[1]);
    theCRC = calc_crc(theCRC,out_data[2]);
    theCRC = calc_crc(theCRC,out_data[3]);
    theCRC = calc_crc(theCRC,out_data[4]);
    theCRC = calc_crc(theCRC,out_data[5]);
    theCRC = calc_crc(theCRC,out_data[6]);
    out_data[7] = theCRC;
}
usb_put_packet(1, out_data, 8, USB_DTS_TOGGLE);
}

```

```
        delay_ms(1);  
    }  
}  
}  
}
```

ÖZGEÇMİŞ

İbrahim Söyler, 1984 yılında Biga’da doğdu. İlköğrenimine Balıkesir’in Gömeç ilçesinde başladı ve Burhaniye ilçesinde tamamladı. Ortaöğrenim ve lise öğrenimini ise 2002 yılında Burhaniye Anadolu Lisesi’nde okul birinciliği ile tamamladı. Aynı yıl İstanbul Teknik Üniversitesi Elektrik Mühendisliği programına puan sıralamasında birinci sırada girmiş ve 3,5 yılda tamamlayarak bir dönem erken bitirmiştir. Halen Fen Bilimleri Enstitüsü Elektrik Mühendisliği Bölümü son sınıf öğrencisidir ve 2008 yılından bu yana İstanbul İl Sağlık Müdürlüğünde Elektrik Mühendisi olarak görev yapmaktadır.