

769357

T.C.
FIRAT ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ

**BİLGİSAYAR AĞLARI VE İNTERNET ÜZERİNDEN
BAŞKA BİR BİLGİSAYARIN KONTROLÜ**

Fatih ERTAM

Tez Yöneticisi
Doç. Dr. Hanifi GÜLDEMİR

YÜKSEK LİSANS TEZİ
ELEKTRONİK ve BİLGİSAYAR EĞİTİMİ ANABİLİM DALI

ELAZIĞ, 2005

T.C.
FIRAT ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ

BİLGİSAYAR AĞLARI VE İNTERNET ÜZERİNDEN BAŞKA BİR
BİLGİSAYARIN KONTROLÜ

Fatih ERTAM

YÜKSEKLİSANS TEZİ
ELEKTRONİK ve BİLGİSAYAR EĞİTİMİ
ANABİLİM DALI

Bu tez, 22.09.2005 tarihinde aşağıda belirtilen jüri tarafından oybirliği /oyçokluğu ile başarılı / başarısız olarak değerlendirilmiştir.

Danışman: Doç. Dr. Hanifi GÜLDEMİR



Üye: Yrd. Doç. Dr. Mehmet GEDİKPINAR



Üye: Yrd. Doç. Dr. Hüseyin Altun 

Bu tezin kabulü, Fen Bilimleri Enstitüsü Yönetim Kurulu'nun 21.09.2005 Tarih ve 2005-32/1 Sayılı kararıyla onaylanmıştır.

TEŞEKKÜR

Bu tez çalışması süresince bana yardımlarını esirgemeyen tez danışmanım Sayın Doç.Dr. Hanifi GÜLDEMİR' e (Fırat Üniversitesi Teknik Eğitim Fakültesi, Elektronik Bilgisayar Eğitimi Bölümü) sonsuz teşekkürlerimi sunarım.

Çalışmam süresince; özellikle uygulama devresinin hazırlanması sırasında bana elinden gelen her türlü desteği sağlayan Sayın Arş. Gör. Yavuz EROL' a (Fırat Üniversitesi Mühendislik Fakültesi, Elektrik Elektronik Mühendisliği Bölümü) , Atilla BİNGÖL' e (Fırat Üniversitesi Bilgi İşlem Dairesi Başkanlığı) Fırat Üniversitesi Enformatik Bölümü ve Fırat Üniversitesi Teknik Eğitim Fakültesi Elektronik Bilgisayar Eğitimi Bölümü altında çalışan tüm arkadaşlarıma ve hocalarıma teşekkür ederim.



İÇİNDEKİLER

TEŞEKKÜR	
İÇİNDEKİLER	I
ŞEKİLLER LİSTESİ	III
TABLolar LİSTESİ	IV
KISALTMALAR LİSTESİ	V
ÖZET	VII
ABSTRACT	VIII
1. GİRİŞ	1
2. VERİ İLETİŞİMİ	3
2.1. Veri İletişimine Giriş	3
2.2. Veri İletişiminin Amacı	3
2.3. Veri İletişim Sistemleri	4
3. BİLGİSAYARLAR ARASI İLETİŞİM	6
3.1. Bilgisayarlar Arası İletişime Giriş	6
3.2. OSI Referans Modeli	6
3.2.1. OSI Referans Modeli Katmanları	7
3.2.1.1. Uygulama Katmanı	8
3.2.1.2. Sunuş Katmanı	8
3.2.1.3. Oturum Katmanı	8
3.2.1.4. Ulaşım Katmanı	8
3.2.1.5. Ağ Katmanı	8
3.2.1.6. Veri Bağı Katmanı	8
3.2.1.7. Fiziksel Katman	8
3.3. İnternet ve Protokoller	9
3.3.1. İnternet ve Protokol Kavramları	9
3.3.2. Point to Point Protokol (PPP) – Noktadan Noktaya Protokol	10
3.3.3. TCP-IP Protokolü	11
3.3.3.1. TCP Protokolü	12
3.3.3.2. IP – İnternet Protokolü	13
3.3.4. UDP – Kullanıcı Datagram Protokolü	14
3.3.5. ARP – Adres Çözümleme Protokolü	15
3.3.6. ICMP – İnternet Mesaj Kontrol Protokolü	15
4. İLETİŞİM YÖNTEMLERİ	17
4.1. İletişim Yöntemlerine Giriş	17
4.2. İletişim Hatları	17
4.2.1. Tek Yönlü İletişim	17
4.2.2. Yarım Çift Yönlü İletişim	18
4.2.3. Çift Yönlü İletişim	18
4.3. İletim Yöntemleri	19
4.3.1. Paralel İletim	19
4.3.2. Seri İletim	20
4.3.2.1. Eşzamansız (Asenkron) Seri İletim	21
4.3.2.2. Eşzamanlı (Senkron) Seri İletim	22
4.4. Seri Portlar ve RS-232 Ara yüzü	23
4.4.1. Evrensel Asenkron Alıcı Verici (Universal Asencronous Reciever Transmitter)	25
5. MİKRO DENETLEYİCİLER	26
5.1. Mikro Denetleyicilere Giriş	26
5.2. Mikro İşlemciler	27
5.3. Mikro İşlemcilere Karşı Mikro Denetleyiciler	27
5.4. PIC Mikro Denetleyicileri	27
5.4.1. PIC 16F84A Özellikleri	28

5.4.2. PIC 16F84A Pin Çıkışları ve Mimari Yapısı	29
5.4.3. PIC 16F84A Hafıza Organizasyonu	30
5.4.3.1. Program Hafızası	31
5.4.3.2. Veri Hafızası	31
6. İNTERNET ÜZERİNDEN MİKRO DENETLEYİCİ KONTROLÜ	34
6.1. İnternet Üzerinden Yapılan Kontrol Sisteminin Genel Yapısı	34
6.2. Uygulama Yazılımı	35
6.2.1. ASP İle Hazırlanan Yazılım	36
6.2.1.1. Form İle Bilgi Gönderilen ASP Dosyası	37
6.2.1.2. Formdan Gelen Bilgilerin İşlendiği ASP Dosyası	37
6.2.2. PHP ile Hazırlanan Yazılım	40
6.2.2.1. Form İle Bilgi Gönderilen PHP Dosyası	41
6.2.2.2. Formdan Gelen Bilgilerin İşlendiği PHP Dosyası	41
6.2.3. Yazılımların Test Edilmesi	44
6.3. Uygulama Devresi	45
6.3.1. PIC 16F84A Mikro Denetleyicisi İçin Geliştirilen Yazılım	45
6.3.2. PIC 16F84A Mikro Denetleyicili Seri Port Kontrol Devresi	49
6.3.2.1. PIC 16F84A Mikro Denetleyicili Kontrol Devresinin Çalışma Mantığı	51
7. SONUÇLAR	54
KAYNAKLAR	55
EKLER	57



ŞEKİLLER LİSTESİ

Şekil 2.1 Basit Bir İletişim Sistemi.....	4
Şekil 3.1 Tipik Bir İletişim Şeması.....	6
Şekil 3.2 OSI Başvuru Modeli Katmanları	7
Şekil 3.3 Seri Hat üzerinden PPP bağlantısı	10
Şekil 3.4 TCP/IP Katmanları.....	11
Şekil 4.1 Tek Yönlü İletişim	18
Şekil 4.2 Yarım Çift Yönlü İletişim.....	18
Şekil 4.3 Çift Yönlü İletişim	19
Şekil 4.4 Paralel İletim.....	20
Şekil 4.5 Seri İletim.....	20
Şekil 4.6 Asenkron İletim Modeli.....	21
Şekil 4.7 Asenkron İletim İle Veri Gönderimi.....	21
Şekil 4.8 Senkron İletim İle Veri Gönderimi	22
Şekil 4.9 DB-9 (Erkek) Seri Port.....	23
Şekil 4.10 DB-25 (Erkek) Seri Port.....	24
Şekil 4.11 Seri port Konektörlerinin Dişi Tarafları.....	25
Şekil 5.1 Bir Mikro Denetleyici Sistemin Blok Diyagramı	26
Şekil 5.2 PIC 16F84A Mikro Denetleyicisinin PIN Diyagramı.....	28
Şekil 5.3 PIC 16F84A Mikro Denetleyicisinin Mimari Yapısı.....	30
Şekil 5.4 PIC 16F84A Mikro Denetleyicisinin Program Hafıza Haritası	31
Şekil 5.5 PIC 16F84A Mikro Denetleyicisinin Yazmaç Haritası	32
Şekil 6.1 İnternet Üzerinden Başka Bir Bilgisayarın Güç Sisteminin Kontrolü	34
Şekil 6.2 Kontrol Sistemi Yazılımının Akış Diyagramı.....	36
Şekil 6.3 Form ASP Dosyası.....	37
Şekil 6.4 Form PHP Dosyası.....	41
Şekil 6.5 ASP ve PHP Yazılımı Test Devresi	44
Şekil 6.6 PIC 16F84A Mikro Denetleyicili Seri Port Kontrol Devre Şeması	50
Şekil 6.7 Hedef Bilgisayarın Ana Kartı Üzerinde Bulunan 'Power Button' Pinleri	51
Şekil 6.8 Elektronik Devrenin Bakır Plaka Üzerindeki Görünümü	52
Şekil 6.9 Hazırlanan Uygulamanın Genel Görünümü.....	53

TABLolar LİSTESİ

Tablo 4.1 DB–9 (Erkek) Seri Portunun Pin Çıkışlarının Açıklaması 23

Tablo 4.2 DB–25 (Erkek) Seri Portunun Pin Çıkışlarının Açıklaması 24

Tablo 6.1 ASP Kodlarının Açıklaması..... 39

Tablo 6.2 PHP Kodlarının Açıklaması..... 43

Tablo 6.3 PIC 16F84A Mikro Denetleyicisi İçin Hazırlanan Programın Açıklaması..... 47

Tablo 6.4 PIC 16F84A Mikro Denetleyicili Kontrol Devresi Malzeme Listesi 52



KISALTMALAR LİSTESİ

ACK	: Onaylamak (Acknowledgement)
ARCNET	: Attached Resource Computing Network
ARP	: Adres Çözümleme Protokolü (Address Resolution Protocol)
ARPANET	: Advanced Research Projects Agency Net
ASCII	: American Standard Code for Information Interchange
ASP	: Aktif Sunucu Sayfası (Active Server Page)
ATM	: Asenkron Transfer Modu (Asynchronous Transfer Mode)
CPU	: Merkezi İşlem Birimi (Central Processing Unit)
CSMA/CD	: Taşıyıcı Duyarlı Çoklu Erişim/Çakışma Kontrolü (Carrier Sense Multiple Access/Collision Detection)
DIX	: DEC, Intel ve Xerox kelimelerinin kısaltılmışı
DTE	: Veri Terminal Aygıtı (Data Terminal Equipment)
EBCDIC	: Extended Binary Coded Decimal Interchange Code
FDDI	: Fiber Dağıtımli Data Arabirimi (Fiber Distrubited Data Interface)
FTP	: Dosya Transfer Protokolü (File Transfer Protocol)
HDLC	: Yüksek Seviyeli Veri Link Kontrolü (High-Level Data Link Control)
HTTP	: Hiper Text Transfer Protokolü (Hyper Text Transfer Protocol)
ICMP	: İnternet Kontrol Mesaj Protokolü (Internet Control Message Protocol)
IEEE	: Elektrik ve Elektronik Mühendisleri Enstitüsü (Institute of Electrical and Electronics Engineers)
IEEE 802.3	: IEEE'nin Ethernet mimarisini açıklayan standardı
IP	: İnternet Protokolü (Internet Protocol)
ISO	: Uluslararası Standartlar Organizasyonu (Internetal Organization for Standardization)
LAN	: Yerel Alan Ağı (Local Area Network)
LCP	: Bağlantı Kontrol Protokolü (Link Control Protocol)
MAC	: Ortam Erişim Kontrolü (Media Access Control)
Mbps	: Saniyedeki mega bit miktarı (Mega Bit Per Second)
NCP	: Ağ Kontrol Protokolü (Network Control Protocol)
NSF	: Milli Bilim Kurumu (National Science Foundation)
OSI	: Açık Sistemler Ara bağlaşımı (Open Systems Interconnectivity)
PDN	: Kamusal Veri Ağları (Public Data Network)
PHP	: Kişisel Ev Sayfası (Personal Home Page)
PIC	: Çevresel Arabirim Kontrolcü (Peripheral interface controller)

PPP	: Uçtan uca Bağlantı Protokolü (Point-To-Point Protocol)
RAM	: Rasgele Erişimli Bellek (Random Access Memory)
RFC	: Request For Comment
SDLC	: Eş zamanlı Veri Link Kontrolü (Synchronous Data Link Control)
SMS	: Kısa Mesaj Servisi (Short Message Service)
SMTP	: Temel Posta Transfer Protokolü (Simple Mail Transfer Protocol)
SNMP	: Basit Ağ Yönetim Protokolü (Simple Network Management Protocol)
STP	: Korumalı Dolambaçlı Çift (Shielded Twisted Pair)
TCP	: Aktarım Kontrol Protokolü (Transmission Control Protocol)
TCP/IP	: Transmission Control Protocol/Internet Protocol
UART	: Evrensel Asenkron Alıcı Verici (Universal Asynchronous Receiver Transmitter)
UDP	: Kullanıcı Data gram Protokolü (User Datagram Protocol)
UNI	: Kullanıcı Ağ Ara yüzü (User Network Interface)
UTP	: Korumasız Dolambaçlı Çift (Unshielded Twisted Pair)
WAN	: Geniş Alan Ağı (Wide Area Network)
WAP	: Kablosuz Uygulama Protokolü
X.25	: Uluslararası Dar bantlı Paket Anahtarlama standardı.

ÖZET

BİLGİSAYAR AĞLARI VE İNTERNET ÜZERİNDEN BAŞKA BİR BİLGİSAYARIN KONTROLÜ

Fatih ERTAM

Fırat Üniversitesi
Fen Bilimleri Enstitüsü
Elektronik-Bilgisayar Eğitimi Anabilim Dalı

2005, Sayfa: 55

Bu tez çalışmasında, bilgisayar ağları ve internet üzerinden uzaktaki bir bilgisayarın güç sistemi kontrol edilerek; kapalı durumda bulunan bilgisayarın açılması, açık olan bilgisayarın kapatılması ya da yeniden başlatılması sağlanmıştır. Bu uygulamanın hazırlanması yazılım ve donanım olmak üzere iki aşamada gerçekleştirilmiştir. Yazılım olarak web tabanlı programlama dillerinden ASP ve PHP ile ayrı ayrı iki yazılım geliştirilmiştir. Bu yazılımlardan herhangi birisi kullanılarak uzaktaki sunucu bilgisayarın seri portuna, gerçekleştirilmek istenilen bilgi gönderilmiştir. İkinci aşamada uygulamanın donanım kısmı hazırlanmıştır. Sunucu bilgisayarın seri portuna gelen bilgiyi alıp işleyebilecek mikro denetleyicili bir devre tasarımı gerçekleştirilerek bağlanmıştır. Gelen veri işlendikten sonra bu devreye bağlı hedef bilgisayarın ana kartı üzerindeki güç sistemini harekete geçirecek pinlerine istenilen işlem komutları gönderilmiş ve uygulama başarıyla tamamlanmıştır.

Anahtar Kelimeler: Uzaktan kontrol, mikro denetleyici, veri iletişimi, bilgisayar ağları

ABSTRACT

REMOTE CONTROL OF A COMPUTER VIA INTERNET AND COMPUTER NETWORKS

Fatih ERTAM

Firat University

Graduate School of Natural and Applied Sciences

Department of Electronic and Computer Science

2005, Page: 55

In this thesis, a computer is controlled, turned on, turned off or rebooted, by a remote computer via internet. This is achieved in two stages as software and hardware. Software is developed by using web based programming languages ASP and PHP.

The required command is sent to the serial port of the remote host computer by using one of the developed software.

A Hardware is designed and connected to the serial port of the host computer. This hardware consists of a microcontroller and receives and process the data received. When the data is received and processed one of the commands turn on, turn off or reboot is sent to the pins of the power system on the main board of the target computer.

Keywords: Remote control, microcontroller, data communication, computer networks

1. GİRİŞ

Bilişim sektörünün ve buna bağlı olarak da iletişim sektörünün ne kadar hızlı bir şekilde geliştiği bu sektörün içinde olan, olmayan herkes tarafından kabul edilen bir gerçektir. Bu hızlı gelişmeye paralel olarak insanların istekleri de artmakta ve daha önce sahip olduklarından daha fazlasını istemeye başlamaktadırlar.

İletişim sektöründeki bu hızlı gelişmeye paralel olarak artık mekândan bağımsız olarak kontrol etme isteğimiz engellenemez olmaktadır. İnsanlar evlerinin dışında iken dahi evlerindeki elektronik eşyaları kontrol etmek istemektedirler. Örneğin eve gelmeden evdeki buzdolabının soğutma derecesini artırabilmek veya fırında bulunan yemeği ısıtabilmek için fırını uzaktan çalıştırabilmeyi arzu etmektedirler. Bu şekilde gerçekleştirilen akıllı evler alanında birçok çalışma yapılmaktadır.

Bilişim sektörünün içerisinde ise bilgisayar sistemleri arasında uzaktan kontrol ile birçok işlem gerçekleştirilmeye çalışılmıştır. Bunların içerisinde belki de en önemlisi uzaktan eğitim sistemleri olmuştur. Bu sistemler ile mekândan bağımsız bir şekilde birçok bilgi paylaşımında bulunulabilmektedir. Hastanelerde uzaktan kontrol sistemleri kullanılmaya çalışılmıştır. Bir hastanede bulunan bir hasta, çok uzaklarda bulunan bir doktor tarafından uzaktan kontrollü robot kolları ile ameliyat edilmeye çalışılmıştır. Bu sistemler gün geçtikçe daha da başarılı bir şekilde hazırlanmakta ve daha kontrollü bir şekilde kullanılmaktadır. Robotların uzaktan kontrolü ile yapılmaya çalışılan birçok proje ve sistem gerçekleştirilmektedir[1].

İlgi çekici başka bir olay ise bilgisayar kontrollü makineler olmaktadır [2, 3]. Uzaktan kontrol ile çalıştırılabilen ve kontrol edilebilen makineler insanlar için birçok işlemi kolaylaştırmayı sağlamaktadır. Mekanik sistemlerin internet üzerinden kontrol edilebilmesi üzerine yapılan çalışmalar ile mikro bilgisayarlar kontrol sürecinde yeni bir çağ oluşturmuşlardır. Artık insanlar bürolarına bile gitmeden birçok işlemi evlerinden gerçekleştirebilmektedirler. Son zamanlarda cep telefonu iletişiminin gelişmesiyle insanlar evlerinde bir bilgisayar ağı kurarak WAP ve SMS ile evlerindeki birçok elektronik aygıtı kontrol edebilmişlerdir[4].

Bu gelişmelerin ışığında internet üzerinden başka bir bilgisayarı kontrol edilebilmesi mümkün olabilmektedir. Bu sadece bilgisayar yazılımı kullanılarak gerçekleştirilebilecek bir çalışma olmadığından bu işlemi yapabilmek için kontrol edilecek bilgisayarda bir mikro denetleyici kullanılarak kapalı bir bilgisayarın açılabilmesi mümkün olabilmektedir. Sistemi açık olan bir bilgisayarı kapatmayı başarmak için internet protokollerine uyumlu bir yazılım yazılması yeterli olabilecektir. Tamamen kapalı bir bilgisayar sistemini açabilmek için arada bir

mikro denetleyicili devre kartı hazırlanarak hedef bilgisayarın güç sisteminin kontrol edilebilmesi sağlanmıştır.

Bu tez çalışmasında internet üzerinden – TCP/IP protokol kümesi kullanılarak – başka bir bilgisayarın güç sisteminin kontrol edilebilmesi için öncelikle ASP ve PHP gibi web tabanlı programlama dilleri kullanılarak sunucu bilgisayarın seri portuna veri gönderilmiştir. Hazırlanmış olan yazılım hem ASP dilinde hem de PHP dilinde ayrı ayrı yazılarak test edilmiştir. ASP dosyaları Microsoft Firmasının üretmiş olduğu Windows işletim sistemlerinde bulunan Sunucu yazılımı olan internet Information Services (IIS) üzerinde sağlıklı bir şekilde çalışmaktadırlar. PHP dosyaları ise daha çok Linux işletim sistemi yüklü bilgisayarlar üzerinde çalışan Apache Sunucuları üzerinde sağlıklı bir şekilde çalışmaktadır. Bu sebeple işletim sistemine bağımlı kalınmaması için her iki web tabanlı script dili kullanılarak yazılımlar hazırlanmıştır.

Uygun yazılımlar hazırlandıktan sonra tez çalışmasının ikinci aşamasına geçilmiş ve sunucu bilgisayarın seri portuna bağlanarak gelen verileri işleyecek olan mikro denetleyicili devre tasarımının oluşturulmasına geçilmiştir.

Hazırlanması düşünülen devre kartı içindeki en önemli yapı Microchip Firmasının geliştirmiş olduğu PIC 16F84A Mikro denetleyicisidir. Bu mikro denetleyicisi için PIC C derleyicisi kullanılarak gelecek olan veriyi nasıl işleyeceğine dair uygun bir yazılım hazırlanmıştır. Hazırlanmış olan yazılım derlendikten sonra oluşan dosya Seri porttan beslenen bir PIC programlayıcı kartı kullanılarak IC-PROG yazılımı ile PIC 16F84A mikro denetleyicisine yüklenmiştir.

PIC 16F84A Mikro denetleyicisi yüklendikten sonra devre board üzerinde test edilmiş ve sağlıklı bir şekilde çalıştığına kanaat getirildikten sonra devre uygun büyüklükteki bir bakır plaka üzerine basılmıştır.

Kısaca bu tez çalışmasında uzakta bulunan önem derecesi yüksek hedef bilgisayarın güç sisteminin kontrol edilmesi tasarlanmıştır. Önem derecesi yüksek hedef bilgisayarda sistemin kilitlenmesi, enerjinin kesilerek sistemin kapanması gibi herhangi bir problem meydana gelmesi mümkündür. Bu şekilde bir problem ile karşılaşılması durumunda hazırlanmış olan devre üzerinde bulunan klemens uçları ile hedef bilgisayarın ana kartı üzerinde bulunan güç sistemini tetikleyecek 'power button' pinleri arasında bulunan kablolar yardımıyla devreden gelen sinyal hedef bilgisayara iletilmektedir. Böylece hedef bilgisayar sisteminin; kapalı durumda ise açılması açık durumda ise kapatılması ya da yeniden başlatılması işlemi gerçekleştirilmektedir.

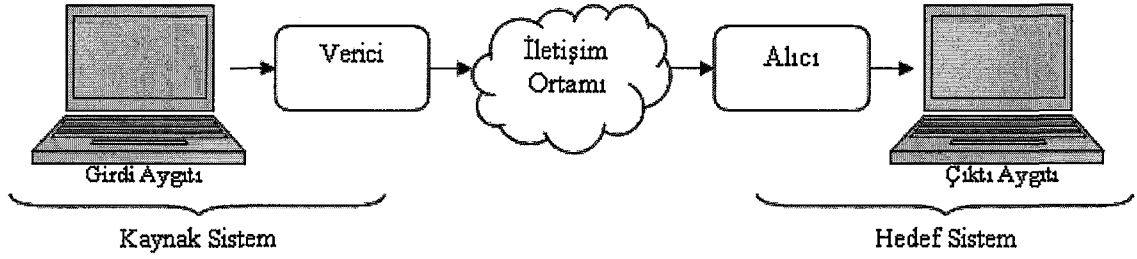
2. VERİ İLETİŞİMİ

2.1. Veri İletişimine Giriş

1980 sonrasında bilişim teknolojisinde çok hızlı gelişmeler yaşanmıştır. Buna bağlı olarak veri iletişimi daha da hızlanmış ve donanım ürünlerinin piyasadaki sayısının artması sebebiyle teknoloji ucuzlamaya başlamıştır. Teknolojinin ucuzlaması yaşantımıza daha fazla girmesine sebep olmaktadır. Kişisel bilgisayarların (PC) sayısında büyük bir artış gözlenmiş; işyerlerinde verimliliği artırabilmek için küçük ağlar (LAN) tercih edilmeye başlanmıştır. Kişisel bilgisayarların bu şekilde hızla artması, insanların gereksinimlerini de artırmıştır; telefon hatları ve modemler vasıtasıyla dosyalar bilgisayarlar arasında gidip gelmeye başlamış, kurumlar arasında iletişimi hızlandırmak için elektronik mektup kullanımı çok hızlı bir şekilde benimsenmiş ve artmıştır. Aynı şirket içerisinde çalışanlar ortak dosyalar kullanmaya başlamışlardır. Bankalarda bilgisayarlar arasında para aktarımı yapılabilmiş, uzaktan rezervasyon işlemleri veya alış veriş işlemleri yapılabilmıştır. Günlük hayatımızdaki birçok işlem bilgisayarlar arasında haberleşme ve veri iletişiminin gelişmesiyle oldukça olağan bir durum olmaya başlamıştır.

2.2. Veri İletişiminin Amacı

Veri iletişiminin temel amacı bilgi alışverişini sağlamaktır ve en basit biçimiyle; "verilerin bir kaynaktan başka bir kaynağa hatasız olarak aktarılması süreci" olarak tanımlanabilmektedir [5]. Veri bilgisayarlar arasında gidip gelebileceği gibi, verinin gönderildiği sistem bir bilgisayar da olmayabilir. Örneğin bu bir mikro denetleyici sistem olabilmektedir. İletişime geçirilebilecek her şey veri olarak kabul edilebilmektedir. Veri her zaman bir metin olmayabilir, resim, ses, video görüntüsü de veri olarak kabul edilebilmektedir. Bilgi ise insan için anlam ifade eden veriler olarak tanımlanabilir. Veri iletişimde önemli olan konu verilerin bozulmadan istenilen hedefe ulaştırılmasını sağlayabilmektedir. Veri iletişimini sağlamak için üç temel unsur gerekir bunlar: Veriyi aktarmak için bir "verici", veriyi almak için bir "alıcı" ve veriyi alıcı ile verici arasında iletmek için kullanılan bir "iletişim ortamı" olarak verilebilir (Şekil 2.1) [5].



Şekil 2.1 Basit Bir İletişim Sistemi

Bilgisayar ortamı dijital bir ortam olduğu için veri iletişimi sayısal kodlama aracılığıyla yapılmaktadır. Aktarılan veri (metin ya da görüntü) '0' ve '1' biçiminde sayısal olarak kodlanarak aktarılmaktadır. Böylece, bilgisayar terminolojisinde "veri iletişimi" sayısal olarak kodlanmış bir bilginin bilgisayarlar arasındaki değiş tokuşu biçiminde açıklanabilmektedir.

Verilerin bilgisayar ağları üzerinden aktarılabilmesi için bir dizi işlem görmesi ve denetimlerden geçmesi gerekmektedir. Yerine getirilmesi gereken bu işlemler farklı düzeylerde gerçekleşmekte ve oldukça karmaşık uygulamalar gerektirebilmektedir. Bu işlemlerin ve denetimlerin bütünü "veri iletişim sistemi" ni oluşturmaktadır.

2.3. Veri İletişim Sistemleri

Aygıtlar arasında bir iletişim gerçekleşebilmesi için bu aygıtlarının birbirlerini anlıyor olmaları gerekmektedir. Metin olan verinin iletimi için standart bir ikili kodlama tekniği kullanılmaktadır. Sayısal iletişim ikili tabanda kodlanmış bilgi veya verinin sistemler arasında aktarılmasını sağlamaktadır.

Hiçbir aygıt direkt veri iletişiminde bulunamaz, çünkü veri iletişimine geçebilmesi için veri iletişimini sağlayan sistemle arasındaki bağlantı için bir arabirim gerekmektedir. Veri kodlama işlemi daha sonra gelmektedir. Verinin kodlanması demek girdi verisi hangi ortamda iletilecekse o ortamda ilerleyebileceği sayısal işaretlere dönüştürülmesi süreci olarak kabul edilmektedir. Bu işaretler genellikle elektrik işaretleridir ama ışık da iletişim biçimi olarak kullanılabilir. Kodlama veri aktarımı sırasında kullanılmak istenen yöntemle göre değişebilmektedir.

Verilerin iki aygıt arasında aktarımı için sadece kodlama yapmak yeterli değildir. Alıcı birim ile verici birimin birbirlerini anlayabileceği bir ortam oluşturulmalıdır. Bu ortam iletişim teknikleri ve eş zamanlama ile sağlanmaktadır. İletişim teknikleri ile iki aygıt arasında ortaya çıkabilecek hataları ortadan kaldırmak veya en aza indirmek hedeflenmektedir.

Gönderilmiş verinin alıcı tarafından tanınıp alınabilmesi için eş zamanlama tekniklerinin kullanılması gerekmektedir. Eş zamanlama verici ve alıcı arasında kullanılacak

işaretler, veri formatları ve temel aktarım birimlerinin değerlendirme yöntemleri üzerinde anlaşmaları ile sağlanmaktadır.

İletişim ortamında yapılan aktarımlar her zaman sorunsuz bir şekilde gerçekleşmeyebilir. Bazen iletim sırasında hatalar meydana gelebilmektedir. Bu hatalar veya bozulmalar mümkün olduğunca denetim altında tutulmaya çalışılmalıdır. Veri iletişimde hata olabilme ihtimali her zaman göz önünde tutulmalıdır ve hataları düzeltebilmek için çeşitli teknikler kullanılmalıdır.

Alıcı aygıt ile verici aygıt arasındaki iletişimde birbirlerini ne kadar anladıkları kadar birbirlerini anlama hızları da önemlidir yani alıcı aygıtın veriyi alma hızı verici aygıtın veriyi gönderme hızından çok daha düşük olursa gönderilen verilerin sadece bir kısmı alınmış olacak alınamayan veriler ise bozulacak veya kaybolacaktır. Bu durumlarda da çeşitli kurtarma teknikleri kullanılabilmektedir.

Bilgisayar ağlarında verinin kaynaktan hedefe doğru iletilebilmesi için adresleme ve yönlendirme yapılması gerekmektedir. Verinin hangi aygıtı gönderilebileceğini belirtmek için bir adres kullanılması gerekmektedir, bu adres veri ile birlikte gönderilmeli ve verinin ağ üzerinde bu adrese ulaşabileceği en kısa yola yönlendirilmelidir.

Bir sistemde birden fazla aygıt bulunabilir, bulunan tüm aygıtların eşzamanlı olarak kullanılabilmesi gerekmektedir. Bunun için çoklama teknikleri kullanılmaktadır. Mevcut kapasite bu şekilde birden fazla aygıtı dağıtılabilmektedir.

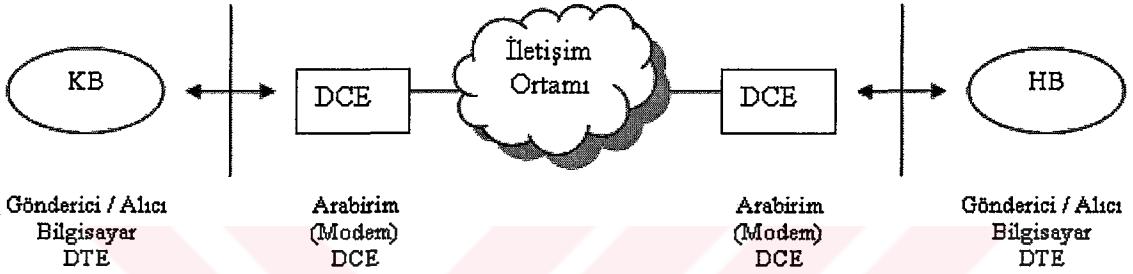
Bir bilgisayar ağında ihtiyaç duyulan temel gereksinim uygun bir veri iletişim sistemi oluşturmaktır.

3. BİLGİSAYARLAR ARASI İLETİŞİM

3.1. Bilgisayarlar Arası İletişime Giriş

Bilgisayarlar arası iletişim için 3 temel öge gereklidir (Şekil 3.1) [5].

- Gönderici (ev sahibi/kaynak bilgisayar) ve alıcı bilgisayar (hedef bilgisayar), DTE (Data Terminal Equipment - Veri Terminal Aygıtı) olarak adlandırılmaktadır.
- Arabirimler (örneğin, modem aygıtları DCE (Data Circuit Terminating Equipment - Veri Devresi Sonlama Aygıtı) olarak adlandırılmaktadır.)
- Veri iletişim ortamı (kablo veya uydu bağlantılarıdır).



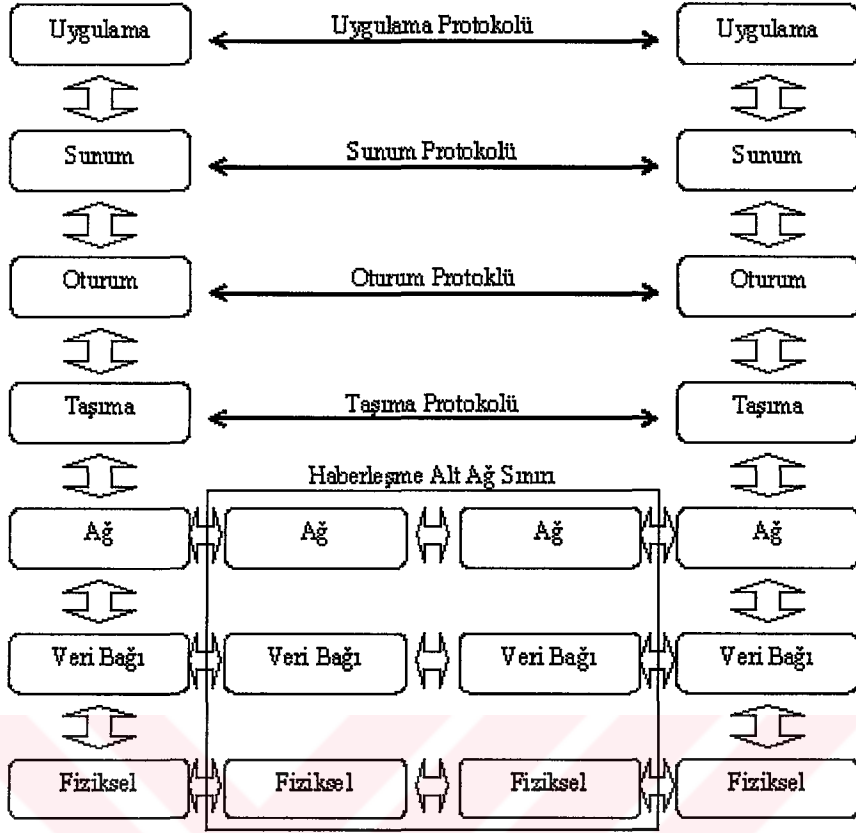
Şekil 3.1 Tipik Bir İletişim Şeması

3.2. OSI Referans Modeli

OSI başvuru modeli ISO tarafından tanımlanmış ve ağ uygulamasında kullanılan örnek bir modeldir denilebilir. Her ne kadar pratikte birebir uygulanmasa da, diğer tüm mimariler OSI başvuru modeli temel alınarak açıklanmaktadır. Ayrıca anahtar (switch), hub, yönlendirici (router), geçit yolu (gateway) gibi ağ cihazlarının fonksiyonu OSI başvuru modeline dayanılarak açıklanmaktadır.

OSI (Open System Interconnection, Açık Sistemler Ara bağlaşımı) başvuru modeli, ağdaki bir uç sistemin veya ara cihazın ağ üzerinden iletişim yapabilmesi için sahip olması gereken işlevleri tanımlamaktadır. Model, Şekil 3.2' de gösterildiği gibi 7 katmana ayrılmıştır. Her katmanın girişi, çıkışı ve görevi belirli olan modüler yapıya sahip olduğu kabul edilebilir.

Bir uç sistemde, örneğin internet'e bağlı bir bilgisayar sisteminde, OSI' nin tanımladığı tüm katmanlara ait işlevlere gereksinim duyulurken, ağ cihazları genel olarak ilk birkaç katmana ait işlevlere sahiptir. Örneğin bir yönlendirici (router) ilk üç, bir anahtar cihazı ilk iki katmana ait fonksiyonlara sahiptir.



Şekil 3.2 OSI Başvuru Modeli Katmanları

3.2.1. OSI Referans Modeli Katmanları

OSI referans modeli 7 katmandan oluşmaktadır. Katman kavramının oluşturulması ve sayısında temel alınan ilkeler şu maddelerde belirtilebilir [6]:

- Yeni bir katman ancak farklı seviyede bir soyutlama gerekiyorsa oluşturulur.
- Her bir katman iyi tanımlanmış bir fonksiyonu yerine getirmelidir.
- Her katmanın yerine getireceği fonksiyon, uluslararası standart protokollerini tanımlamaya yönelik olarak seçilmelidir.
- Arabirimler aracılığıyla bir katmandan diğerine gönderilen bilgiler en aza indirgenmelidir.
- Katman sayısı, aynı katman içerisinde farklı ve çok sayıda fonksiyonu içermeyecek kadar fazla fakat mimariyi çok genişletmeyecek kadar da az sayıda olmalıdır.

OSI referans modelinin kendisi bir ağ mimarisi değildir, çünkü OSI her katmanda kullanılan protokol ve servisleri kesin olarak belirtmemektedir. Sadece her katmanın ne yapması gerektiğini tanımlamaktadır.

3.2.1.1. Uygulama Katmanı

Kullanıcının çalıştırdığı uygulama programları doğrudan bu katmanda tanımlıdır. Dosya aktarımı (FTP), elektronik mektuplaşma (e-mail), ağ yönetimi (SNMP), internet hizmetlerine erişim programlarını örnek göstermek mümkündür.

3.2.1.2. Sunuş Katmanı

Bilginin iletimde kullanılacak biçimde düzenlenmesini sağlamaktadır. Sıkıştırma/açma, şifreleme/şifre çözme, EBCDIC (Extended Binary Coded Decimal Interchange Code) - ASCII (American Standard Code for Information Interchange) dönüşümü ve ters dönüşümü gibi işlevlerin yerine getirilmesini kapsamaktadır.

3.2.1.3. Oturum Katmanı

Uç düğümler arasında gerekli oturumun kurulması, yönetilmesi ve sonlandırılması işlerini kapsamaktadır. İletişimin mantıksal sürekliliğinin sağlanması için, iletişimin kopması durumunda bir eşleme noktasından başlayarak iletimin kaldığı yerden devam etmesini sağlamaktadır.

3.2.1.4. Ulaşım Katmanı

Bilginin son alıcıda her türlü hatadan arındırılmış olarak elde edilebilmesini sağlamaktadır. Ulaşım katmanının oluşturduğu bilgi bloklarına bölüm (segment) denilmektedir.

3.2.1.5. Ağ Katmanı

Veri paketlerinin bir uçtan diğer uca ağdaki çeşitli düğümler (yönlendirici, geçit yolu vs.) üzerinden geçirilip yönlendirilerek alıcısına ulaşmasını sağlayan işlevlere sahip katmandır. Buradaki bilgi bloklarına paket adı verilmektedir. İnternet' in protokol kümesi olan TCP/IP' de IP protokolü bu katmana ait bir protokoldür.

3.2.1.6. Veri Bağı Katmanı

Gönderilecek bilginin hatalara bağışık bir yapıda lojik işaretlere dönüştürülmesi, alıcıda hataların sezilmesi, düzeltilemiyorsa doğrusunun elde edilmesi için göndericinin uyarılması gibi işlevleri yerine getirmeye çalışmaktadır.

3.2.1.7. Fiziksel Katman

Verinin fiziksel olarak hat üzerinden aktarılması için gerekli işlevleri kapsamaktadır. Bu katman için tanımlanan standartlar taşıyıcı işaretin şekli, verici ve alıcı konumundaki uç

noktaların elektriksel ve mekanik özelliklerini belirlemektedir. Kablo, konnektör standartları bu katmanda yapılmaktadır.

3.3. İnternet ve Protokoller

3.3.1. İnternet ve Protokol Kavramları

İnternet'in temeli, 1960'lı yıllarda Amerika Birleşik Devletleri Savunma Bakanlığı'nın, askeri personel ve sivil araştırmacıların savunma ile ilgili projelerde haberleşebilmelerinin sağlanması amacıyla bilgisayar ağlarının kullanılması düşüncesine dayanmaktadır. 1970'li yıllarda ARPANET adlı bir geniş alan ağı ortaya çıkmıştır. ARPANET ile birlikte yerel alan ağlarının geniş alan ağlarına bağlanması kavramı ortaya çıkmış ve birbirine bağlı ağlar arası iletişim (Internetwork) gündeme gelmiştir. Bu ağlar arasındaki uyumsuzluk sorununun giderilmesi amacıyla da TCP/IP (Transmission Control Protocol/ İnternet Protocol) geliştirilmiştir. TCP/IP, farklı topoloji ve protokollere sahip bilgisayar ağlarını birbirlerine bağlamak için kullanılan bir protokoller dizisidir.

İnternet, birbirine bağlanmış çok sayıda bağımsız bilgisayar ağından oluşmaktadır. İnternet' te TCP/IP grubu protokoller kullanılır. Herhangi bir ağ üzerindeki kullanıcı, diğer uçtaki bir başka kullanıcıya bir ileti göndermek istediğinde, bu ileti alıcısına ulaşmadan önce birçok ağ üzerinden geçmektedir. İletiyi gönderen kullanıcının bu geçişlerden haberi olmamaktadır. Kullanıcının bilmesi gereken, yalnızca alıcının internet adresidir. Bu adres, 32 ikil uzunluğunda bir ikil dizgesidir ve 8 ikillik 4 ayrı sayıdan oluşmaktadır (193.255.124.66 gibi). Sekiz ikillik bu gruplar "oktet" adını almaktadırlar. İlerleyen zamanlarda bu adresin ihtiyacı karşılayamayacağı düşünüldüğünden 64 bit olarak kullanılması düşünülmekte ve bununla ilgili çalışmalar yapılmaktadır.

Bilgisayar sistemlerinin birbirleriyle iletişime geçebilmeleri için basitçe birbirlerini tanımaları gerektiğini söylemek mümkündür, işte bu tanıma olayının gerçekleştirilebilmesi için çeşitli protokoller geliştirilmiştir. Bu protokoller hem sistemlerin birbiriyle iletişime geçmesinde büyük rahatlık sağlamış, hem de yeni geliştirilen ağ donanım aygıtlarının bu protokollere uygun halde üretilmeleri ile uyumluluk sağlanmıştır. Örneğin TCP (Transmission Control Protocol) [RFC 793] [7] geliştirilmiş, bundan sonra oluşturulan sistemler bu protokolü kullanmaya başlamışlardır. Sistemler bu protokolü kullandığı için de aralarında gerçekleşen bir uyumluluk oluşmuştur.

3.3.2. Point to Point Protokol (PPP) – Noktadan Noktaya Protokol

Noktadan noktaya protokol bir internet standardı olup IP, IPX gibi çeşitli ağ katmanı protokollerine ait paketleri seri hat üzerinden taşınma işlemini yapmaya çalışmaktadır (Şekil 3.3). 1980 yılından sonra internet protokollerini destekleyen sunucu bilgisayarlarının sayısında bir artış yaşanınca buna bağlı olarak internet kullanımında da büyük bir artış görülmüştür. Bu sunucuların büyük çoğunluğu, Ethernet'in en yaygın olarak kullanıldığı değişik tip LAN' lara bağlı idiler. Kalan sunucuların büyük çoğunluğu ise X.25 tarzı kamusal veri ağları (Public Data Network, PDN) gibi WAN' lar aracılığı ile bağlıydılar. Göreceli olarak bu sunucuların pek azı basit uçtan uca (Point-to-Point) seri bağları kullanmaktaydılar. Hala bu uçtan uca bağlantılar veri haberleşmesi için en eski yöntemler arasında bulunmaktadır ve çoğu sunucu bu tip bağlantıları desteklemektedir. Örneğin, asenkron EIA/TIA-232-C (Daha önce RS-232-C) ara yüzü her sunucuda gerekli bir ara yüz olarak kabul edilmektedir [6].

Bu uçtan uca IP bağlantılarının az sayıda olmasının sebebi standart bir kapsülleme protokolünün olmayışından kaynaklanmaktadır. Point-to-Point Protocol (PPP) bu problemi çözmek için geliştirilmiştir. Uçtan uca bağlantılarda kullanılmak üzere standartlaştırılmış bir IP protokolünün boşluğunu doldurmanın yanında PPP, IP adres ataması ve yönetimi, asenkron (Start/Stop) ve bit eğilimli eşzamanlı kapsülleme, ağ protokol çoğullaması (Diğer protokollere aynı bağlantı üzerinden destek), bağ yapılandırması, bağ kalite testi, hata tespiti, ağ katmanı adresi ve sıkıştırması uzlaşısı gibi diğer yetenekler için seçenekler sunmaktadır. PPP bu konulara, seçime bağlı yapılandırma ve hizmet parametrelerinin uzlaşısı için kapsamlı Bağ Denetim Protokolü (LCP, Link Control Protocol) ve bir Ağ Denetim Protokolleri (NCP, Network Control Protocols) ailesini sağlayarak çözüm getirmektedir. Bugün PPP, IP yanında, Internetwork Packet Exchange (IPX) ve DECnet gibi diğer protokolleri de desteklemektedir.

PPP özellikle düşük hızlı eşzamanlı ve eşzamansız seri hatlarda kullanılmak üzere tasarlanmış bir ikinci katman protokolüdür. Sadece noktadan noktaya bağlantılarda kullanılmaktadır. Doğrulama (Authentication) hizmetlerini sunuyor olması ve seçime bağlı olarak veri sıkıştırma yapabilmesi özellikle çevirmeli bağlantılarda çok fazla kullanılmasına sebep olmaktadır.



Şekil 3.3 Seri Hat üzerinden PPP bağlantısı

3.3.3. TCP-IP Protokolü

TCP/IP'nin kökleri 1960'ların sonunda ve 1970'lerin başında Amerikan Savunma Bakanlığı'na bağlı İleri Araştırma Projeleri Ajansının (Advanced Research Projects Agency, ARPA) yürüttüğü paket anahtarlamalı ağ deneylerine kadar uzanmaktadır. TCP/IP' nin yaratılmasını sağlayan proje, ABD'deki bilgisayarların bir felaket anında da ayakta kalabilmesini, birbirleriyle İletişimlerinin devam edebilmesini amaçlamaktaydı.

TCP/IP'nin tarihi aynı zamanda internet'in tarihidir. İnternet ile TCP/IP ayrılmaz kardeşlerdir. TCP/IP, ,internet'in temeli olarak kabul edilebilir [8].

TCP/IP (Transmission Control Protocol / Internet Protocol) bir protokol kümesidir; marka bağımsız bilgisayar sistemlerinin birbirleriyle karşılıklı çalışabilmesi için en yaygın kullanılan protokol kümesidir. İnternet de TCP/IP protokol kümesi kullanılmaktadır; bu nedenledir ki, TCP/IP kullanımı çok yaygınlaşmıştır.

İnternet bir geniş alan ağıdır; ağların ağı da denilebilmektedir, Bünyesinde binlerce LAN, milyonlarca bilgisayar sistemi vardır ve bu bilgisayarların donanımları, işletim sistemleri birbirinden farklı olabilmektedir, UNIX, Windows, NetWare gibi birçok işletim sistemi bulunmaktadır. Bütün bu sistemler, bu kadar çeşitliliğe rağmen tek bir protokol kümesi TCP/IP aracılığıyla birbiriyle iletişim yapabilmektedir. Teknik açıdan bakıldığında, TCP/IP ve internet hemen hemen aynı anlama gelmeye başlamaktadır.

TCP/IP'nin iletişim için kullandığı birim bilgi miktarına "datagram" adı verilir. İletiler bir uçtan diğerine datagramlar biçiminde iletilmektedir.

TCP/IP, OSI başvuru modeli gibi katmanlardan oluşmuş bir protokoldür. Bununla birlikte TCP/IP protokolü, OSI modelinde bulunan her katmanı içermemektedir, ağ katmanı, taşıma katmanı ve üzerlerindeki protokoller tanımlamaktadır. Örneğin, OSI' nin sunum ve oturum katmanları, TCP/IP protokolünde uygulama katmanı içinde yer almaktadır. Şekil 3.4 ' de, TCP/IP katmanları gösterilmektedir.

Uygulama			FTP			
	SMTP	RLOGIN			TELNET	DOMAIN
Taşıma	TCP				UDP	
	IP			ICMP		
Yönlendirme	IEEE 802.2/LAPB//HDLC					
	Ethernet, X.25, andaçlı halka, çevirmeli ağ vb.					
Fiziksel						

Şekil 3.4 TCP/IP Katmanları

Bu protokoller kümesinde, fiziksel katman ve veri bağlantı katmanı için herhangi bir tanımlama yapılmamıştır. Ağ içinde kullanılan diğer alt katman tanımlamaları geçerlidir. Örneğin bu katmanda IEEE 802 protokolleri, IBM jetonlu halka veya Ethernet protokolü (10BASE-T, FDDI, ATM) kullanılabilir. Bu esneklik, kullanılan iletişim ortamına da yansımaktadır. Böylece herhangi bir fiziksel ortam, kablo kullanılmasına olanak tanınmaktadır.

TCP/IP protokol grubunun veri bağlantı ve fiziksel katmandan bağımsız olması, bu kadar sık kullanılmasının ve yaygınlaşmasının en önemli nedenlerinden biridir. Ayrıca, günümüzde birçok işletim sistemi (örneğin UNIX) de TCP/IP protokol gruplarını hazır olarak içermektedir.

TCP/IP Protokol gruplarını şu şekilde gösterebiliriz;

- Uygulama protokolü (bunlar FTP, SMTP, Telnet gibi uygulama protokolleridir)
- TCP protokolü
- IP protokolü
- Fiziksel katman (ağ arabirim katmanı)

3.3.3.1. TCP Protokolü

TCP/IP protokol kümesine ismini veren TCP, bağlantılı (connection-oriented) ve güvenilir (reliable) bir iletişim sağlamaktadır. Buradaki "bağlantılı" ifadesi şu anlama gelmektedir: Bilgisayarlar iletişime geçmeden önce aralarında bir oturum (session) açmaktadırlar. Oturumun açılması sırasında bilgisayarlar kendi iletişim parametrelerini birbirlerine iletmektedirler. İletişim sırasında bu parametrelere göre hareket edilmektedir. Güvenilir iletişim ise; Bir bilgiyi karşı tarafa gönderdiğimizde karşı tarafın eline ulaşmış olduğundan emin olunması anlamına gelmektedir. Bu da gönderdiğimiz bilginin alındığına ilişkin bir onay mesajının (acknowledge, ACK) bize gelmesi ile mümkün olmaktadır. Eğer belli bir süre içinde bu mesajı alırsak bizden çıkan bilginin karşı tarafın eline geçtiğini kabul etmek mümkündür. Eğer belli bir süre içinde böyle bir mesajı alamazsak gönderdiğimiz bilginin başına bir şey geldiği düşünülebilir. Bu durumda bilgi tekrar gönderilmektedir. TCP iletişimi basit bir iletişim değildir. Bilgi paketini oluşturup üstüne gideceği adresi yazmaktan daha fazlasını yapmamız gerekmektedir. Bu fazladan yaptığımız şeyler veri iletişim performansımızı düşürmektedir ama verinin güvenli bir şekilde hedefine ulaştırılmasını da sağlamaktadır. TCP/IP kümesi içinde kritik İşler yapan protokoller ve servisler verilerini TCP ile iletmektedirler. Örneğin 21 No' lu portu kullanan FTP, 23 No' lu portu kullanan Telnet, 53 No' lu portu kullanan DNS örnek verilebilir.

TCP protokolünün görevi şu şekilde sıralanabilir:

- Alıcı tarafa ne yollandığının kayıtlarını tutmaktır.
- Üst katmandan (uygulama katmanı) gelen bilgiyi parçalara ayırmaktır.
- İletişim ortamında kaybolan veya hatalı yollanan iletileri, bilgileri yeniden göndermektir.
- Alıcısına ayrı sıralar halinde gelebilen parçaların doğru sırada ve hatasız olarak ulaşmasını sağlamaktır.

Bu protokol, kısaca parçaların alıcısına hatasız ve sıralı olarak ulaştırılmasından sorumludur denilebilir. TCP, OSI başvuru modelinde anlatılan taşıma katmanının işlevlerini yerine getirmektedir. Bu katman, uçtan uca bağlantının sağlandığı ilk katmandır. TCP protokolü, iletiyi parçalara ayırdıktan sonra her birinin başına bir başlık eklemektedir. Bu başlık, çeşitli alanlar içermektedir (Şekil 3.5). Bu alanlar, gönderici kapısı (kaynak), alıcı kapısı, sıra numarası, alındı onayı (acknowledgement), veri çoğaltımı, veri için yer ayrılması, pencere, sağlama toplamı (checksum), acil işareti ve bilgi olarak sıralanabilir. Kaynak kapı numarası, paketler gönderilmeden önce TCP tarafından her kullanıcıya verilmiş numaradır. Alıcı kapı numarası ise, iletişim kurulduğu zaman alıcı tarafındaki TCP tarafından verilen numaradır. Bu sayının, gönderici TCP tarafından bilinmesi gerekmektedir. Bu nedenle iki nokta arasında iletişim kurulduğunda, kaynak noktasındaki (gönderen) TCP bu numarayı alıcıdan öğrenir ve parça başlıklarına yazar. Sıra numarası alanı ile her parçaya bir sıra numarası verilir. Bu numara ile karşı taraf, doğru parçayı eksiksiz alıp almadığını anlayabilir. Sağlama toplamı, parça içindeki tüm oktetler toplanarak bulunmaktadır.

Kaynak Kapı Numarası			Alıcı Kapı Numarası					
Sıra Numarası								
Onay (Acknowledgement)								
Veri Offset	Ayrılmış	U R G	A C K	P S H	R S T	S Y N	F I N	Pencere (Window)
Sağlama Toplamı		Acil işareti (Urgent Pointer)						
Bilgi diğer 500 oktet								

Şekil 3.5 TCP Segmenti

3.3.3.2. IP – İnternet Protokolü

IP katman protokolü, datagramların tek tek farklı fiziksel katmanlardan geçerek ve doğru yolu bularak yönlendirilmesinden sorumludur. OSI başvuru modelinin ağ katmanına

karşılık gelmektedir. Bu katman, ağlar arasında veri gönderiminin gerçekleştirilmesini sağlayan protokolleri içermektedir.

IP'nin temel işlevi kaynak uçta TCP' den gelen veriyi kabul etmektir. IP daha sonra bir datagram oluşturarak bunun ağ boyunca yönlendirilmesini sağlamaktadır. Alıcı uca paket vardığında IP veriyi alıcı TCP portuna geçirmektedir. IP ayrıca datagramların hangi durumlarda göz ardı edileceğine dair kuralları tanımlamaktadır.

TCP katmanında parçalara ayrılan bilgi, yönlendirilmek üzere IP katmanına yollanmaktadır. Bu katman parçasının içeriğiyle ilgilenmemektedir, yalnızca bu parçayı gideceği adrese yollamakla yükümlüdür. Bunun için, öncelikle bir yol bulmaya çalışmaktadır. Yolun saptanmasından sonra kendi başlığını TCP'den gelen parçaya ekler. Bu başlığın da eklenmesiyle parça artık datagram olarak adlandırılmaktadır. IP başlığının içinde gönderici (kaynak) ve alıcının ,internet adresi, protokol numarası ve sağlama toplamı bulunmaktadır. Alıcının internet adresi, iletinin gönderilmek istendiği bilgisayarın adresidir. Bu adres sayesinde, aradaki yönlendiriciler ve geçit yolları bu datagramı nereye yönlendireceklerini bilmektedirler. Protokol numarası, bu datagramı TCP'ye vermesi gerektiğini söylemektedir. Yine sağlama toplamı, IP başlık bilgisinin yolda bozulup bozulmadığını veya yanlış adrese gidip gitmediğini denetlemek için kullanılmaktadır.

3.3.4. UDP – Kullanıcı Datagram Protokolü

TCP gibi ulaşım katmanında tanımlı bir protokol olan Kullanıcı Datagram Protokolü (User Datagram Protocol, UDP) Birleşik Devletler Savunma Bakanlığı (Department of Defence, DoD) tarafından tanımlanmış bir aktarım katmanı protokolüdür. Bir uç sisteme Best-Effort Datagram ulaştırma hizmeti sunmaktadır.

UDP, TCP/IP ye göre hem bağlantısız hem de güvensiz bir protokoldür [8]. UDP' ye uygun olarak bir veri iletimi gerçekleştirdiğimizde verimizi hazırlayıp, yollayan kısmına kendi adresimizi, alıcı kısmına hedef bilgisayarı yazılıp yollanmaktadır. Bu verinin yerine sağlıklı bir şekilde gidip gitmediği anlaşılamamaktadır. UDP bağlantı kurmayıp, güvenli bir İletim de sağlamamaktadır ama bu eksiklikleri daha hızlı bir iletişim gerçekleşmesini sağlamaktadır. UDP basit bir protokol olduğu için hızlı iletişim kurmamız gereken yerlerde işimize yaramaktadır.

TCP, bilgiyi parçalara bölen ve bunları alıcı tarafında birleştiren bir protokoldür. Bazı uygulamalarda iletilen bilgiler, tek bir datagram içine sığabilecek büyüklüktedir. Bu tür uygulamalara örnek olarak, adres denetimi verilebilir. Kullanıcılar, internet üzerindeki bir bilgisayara ulaşabilmek için o bilgisayarın adını kullanabilirler. Sistem bağlantı kurmadan önce bu bilgisayar adını internet adresine çevirmek durumundadır. İnternet adreslerinin hangi adlara karşılık geldiği, belli bilgisayarlarda tablolar halinde tutulmaktadır, internet adresi, bu

tablolardan sorgulanarak öğrenilebilir. Böyle bir sorgulama tek bir parçaya sığdırılabileceği için, TCP katmanının kullanılması gereksizdir. Bu gibi durumlarda en sık kullanılan protokol, UDP' dir. UDP, uygulamalara bağlantı kurulmayan bir taşıma katmanı işlevi sunmaktadır. Dolayısıyla TCP protokolündeki çoğu mekanizma UDP' de yoktur (bağlantı kurulumu, alındı bildirimi, tıkanıklık denetimi gibi).

UDP, datagramların belirli sıralara konmasının gerekmediği uygulamalarda kullanılmaktadır. UDP protokolünün kullandığı bir başlık bulunmaktadır. Bu başlık, gönderilecek bilginin başına konmakta ve IP katmanına gönderilmektedir. IP katmanı kendi başlık bilgisini ve protokol numarasını ekleyerek göndermektedir.

UDP' nin TCP'den farkı şudur: Bilgi burada datagramlara bölünmemekte ayrıca yollanan paketlerin kaydı da tutulmamaktadır.

3.3.5. ARP – Adres Çözümleme Protokolü

Ethernet günümüzde en çok kullanılan ağ ara yüzü olarak kabul edilebilmektedir. Ethernet ara yüzleri birbirlerine veri paketi gönderebilmek için kendilerine üretim esnasında verilen MAC (Media Access Control) adreslerini kullanmaktadırlar. Bu adresler 48 bitten oluşmaktadır. TCP/IP protokol kümesinin kullandığı ağlarda IP v4 ile kullanılan adresler 32 bitten oluşmaktadır. Fiziksel katmanda Ethernet ara yüzünün kullanılması durumunda IP adresinden fiziksel adrese dönüşümü gerçekleştirebilecek bir protokole ihtiyaç duyulmaktadır. Bu işlemi yapabilmek için ARP (Address Resolution Protocol) geliştirilmiştir. Adres çözümleme Protokolü (ARP), IP protokolü tarafından, veri bağı protokolü tarafından kullanılan donanım adreslerinin (MAC) IP ağ adreslerine eşlenmesi için kullanılmaktadır. Kısacası ARP; IP adresi bilinen bir bilgisayarın MAC adresini bulmaya yaramaktadır. Bilgisayarlar arası iletişim yalnızca MAC adresleri üzerinden gerçekleştirilebilmektedir.

Adres çözümleme, bir ağ içindeki bilgisayarın adresinin bulunmasını ifade etmektedir. Adresin çözümlenmesi, yerel bilgisayardaki bir kullanıcı işlemi tarafından, diğer bir bilgisayardaki sunucu işlemine adres çözümleme ile ilgili talep gönderen bir protokol kullanımı ile gerçekleştirilmektedir. Diğer bilgisayar tarafından alınan bilgi değerlendirilerek talebi gönderen bilgisayara adresi içeren bir cevap gönderilmekte ve böylelikle çözümleme işlemi gerçekleştirilmektedir.

3.3.6. ICMP – İnternet Mesaj Kontrol Protokolü

ICMP (Internet Control Message Protocol) kontrol yapabilmek amacıyla geliştirilmiş bir protokoldür. Genelde sistemler arasında kontrol mesajları iletilecekse bu IP üzerinden değil ICMP üzerinden yapılmaktadır

Bir kaynaktan bir alıcıya bir mesaj yönlendirilirken birçok sorunla karşılaşılabilir. Gönderen cihazın gönderilen bir data gramla ilgili yaşanan problemlerden haberdar olması, ağda yaşanan problemlere göre doğru yönlendirme kararları vermesi için önemlidir. İnternet Denetim Mesajı Protokolü (ICMP) bu amaçla geliştirilmiştir.

ICMP bir hata raporlama sistemidir. IP'nin gerekli bir bileşenidir ve IP uyarlamasına eklenmelidir. Bu değişik IP sürümleri ve değişik işletim sistemleri arasında tutarlı, anlaşılabilir hata mesajı ve işaretlerin alış verişini sağlamaktadır. ICMP IP katmanının haberleşme sistemidir. ICMP mesajları ağ tarafından diğer IP paketlerinden ayırt edilmez ancak IP katmanı yazılımı tarafından farklı yorumlanırlar. Hemen hemen tüm durumda ICMP tarafından gönderilen hata mesajları asıl datagramı gönderen makineye geri yönlendirilirler. Bu yalnızca kaynak ve alıcı adreslerinin başlığa dâhil edilmesinden kaynaklanmaktadır.



4. İLETİŞİM YÖNTEMLERİ

4.1. İletişim Yöntemlerine Giriş

Veri iletişimi sayısal sistemlerin anlayabileceği şekilde kodlanmış olan verilerin iki sistem arasında iletilmesi olarak açıklanabilir. Veriler aktarılırken aktarılan bu verinin bozulmamasına ve doğru aktarılmasına dikkat etmek gereklidir. Verinin doğru bir şekilde alıcı tarafından alınabilmesi için zamanlama, hata ve akış denetimi bilgilerine ihtiyaç duyulmaktadır. Veri içerisinde bulunan tek bir sayısal değerin dahi bozulması verinin sağlıklı bir şekilde alınmasına engel olmaktadır. Veri iletimi yapılırken alıcı ve verici sistem arasındaki fiziksel yolda iletimin nasıl yapılacağı dikkate alınmalıdır. Bu iki sistem arasındaki mesafe bir bilgisayar sisteminin içerisindeki ana kartta bulunan iki yonga arasında gerçekleşebilecek kadar santimetrelerle ölçülebilecek kısa mesafelerde olabilmekte, farklı ağlarda bulunan bilgisayar sistemleri arasındaki millerce uzunluğunda da olabilmektedir.

Veri iletilirken fiziksel ortamda nasıl iletileceğinin belirlenmesi kadar, verinin; biçimi, zamanlaması, gönderim hızının denetimi de önemlidir. Modern bir iletişim sistemi, bilgiyi göndermeden önce bilginin sıraya koyulmasıyla, işlenmesiyle ve korunmasıyla ilgilenmektedir. Gerçek anlamda gönderme daha fazla işleme ve gürültünün süzülmesiyle gerçekleşmektedir. Son olarak, kod çözme, mesajı koruma ve bilgi algılama basamaklarından oluşan alma işlemi gelmektedir.

4.2. İletişim Hatları

Veri iletişim hatlarında bilgi alış verişinde bulunulurken nasıl bir iletişimde bulunulacağını alıcı ve verici arasındaki gereksinimler belirlemektedir. Bazı durumlarda yalnızca veri almak yeterli olurken bazı durumlarda ise hem veri almak hem de veri göndermek gerekmektedir. Veri iletişimi hatlarını üç gruba ayırmak mümkündür, bunlar;

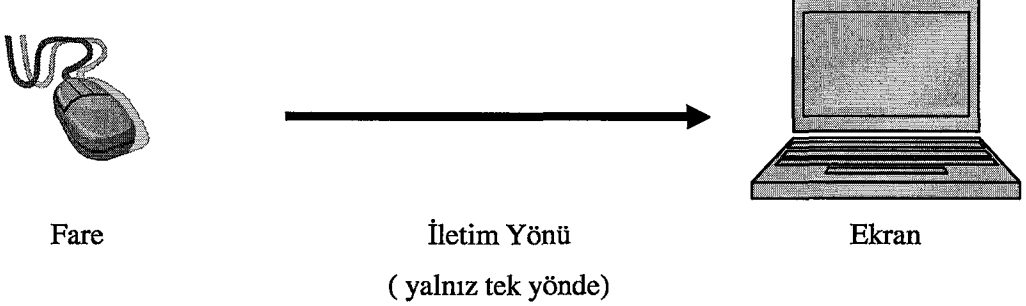
- Tek yönlü iletişim (Simplex)
- Yarım çift yönlü iletişim (Half-duplex)
- Çift yönlü iletişim (Duplex)

Şeklinde gruplanabilmektedir.

4.2.1. Tek Yönlü İletişim

Verinin tek bir yönde iletilmesi yeterli ve gerekli ise bu iletişim hattı seçilmektedir. Tek yönlü iletişim de yalnızca veri gönderme işlemi ya da yalnızca veri alma işlemi gerçekleştirilmektedir. Günlük hayatta izlediğimiz televizyon yayınları tek yönlü iletişime örnek

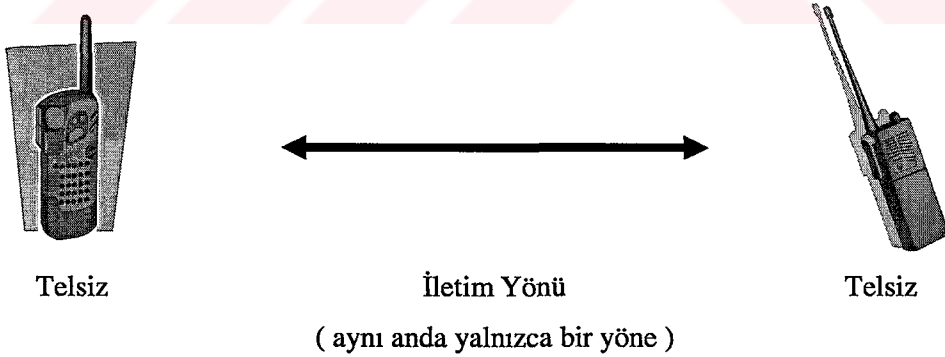
verilebilir. Televizyondan gelen veri alınabilmekte ama televizyon yayınına herhangi bir müdahale hakkımız bulunmamaktadır. Aynı şekilde bilgisayar sistemlerinde giriş birimleri olan klavye, fare gibi sistemler ile bilgisayar sistemine veri yollanırken bilgisayar sisteminden herhangi bir veri klavye ya da fareye gönderilmemektedir (Şekil 4.1).



Şekil 4.1 Tek Yönlü İletişim

4.2.2. Yarım Çift Yönlü İletişim

Yarım çift yönlü iletişimde veriler her iki yönlü de taşınabilir. Mesajların iletimi sırasında aynı iletişim hattı kullanıldığı için verilerin iletişim hattında birbirlerini engellemelerini önlemek için iletişim hattı belli bir sırayla kullanılmaktadır. Alıcı ya da verici tarafta bulunan sistemden hangisi o an iletişim hattını kullanıyorsa sistemin karşı tarafı hattı kullanan tarafın işlemini bitirmesini beklemek zorundadır. Yarım çift yönlü iletişime en güzel örnek telsizler ve faks aygıtları verilebilir. (Şekil 4.2)

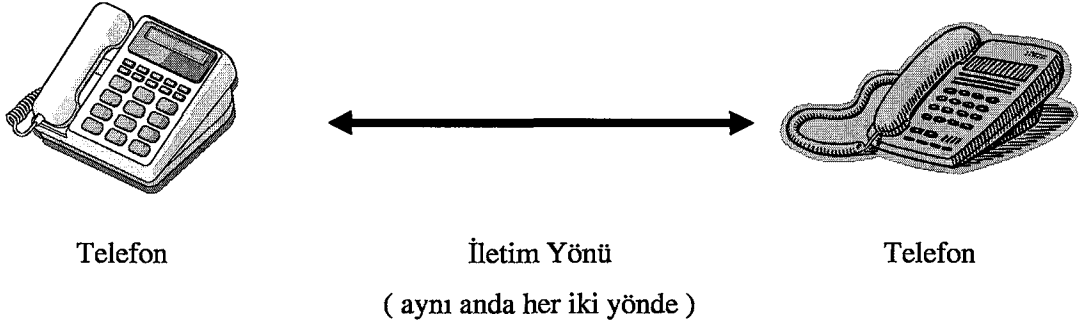


Şekil 4.2 Yarım Çift Yönlü İletişim

4.2.3. Çift Yönlü İletişim

Alıcı ve verici sistemlerden her iki tarafında aynı anda veri alıp gönderebilmesine imkân sağlayan iletişim hattı olarak tanımlamak mümkündür. Günlük hayatta kullandığımız telefonlar bu iletişim sistemi için iyi bir örnektir. Telefon sisteminde aynı anda her iki tarafta

iletiřim kurabilmektedir. Bilgisayar sistemlerinde ise modemler çift yönlü iletiřim kullanarak sistemler arasında iletiřim kurabilmektedirler (řekil 4.3).



řekil 4.3 Çift Yönlü İletişim

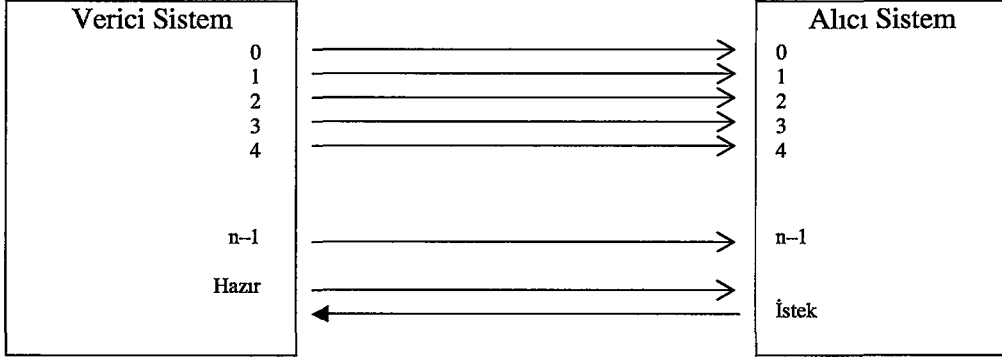
4.3. İletim Yöntemleri

Verileri alıcı ve verici noktaları arasında iletmek için Paralel ve seri iletim şeklinde iki yöntem kullanılmaktadır.

4.3.1. Paralel İletim

Paralel iletim alıcı ve verici sistem arasındaki bilgi alışverişinde veri paketindeki bütün sayısal bilgilerin aynı anda iletilmesi sistemine dayanmaktadır. Alıcı ve verici arasında veri iletilirken problem çıkmaması için vericinin alıcıya yolladığı kodlar ile ilgili bitleri çıkardığını belirtmek için veri hazır (data ready) ve alıcının vericiye veri alabileceğini belirten istek (demand) belirtme yollarına gerek olmaktadır [9].

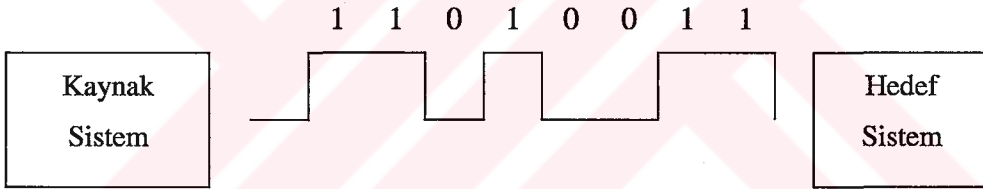
Paralel iletimde bitlerin iletimi, karşı tarafa gönderilecek kaç tane bit varsa o kadar sayıda iletim hattı üzerinden yapılır (řekil 4.4). Bilgisayar sistemlerinin içerisindeki mesafeler kısa olduğu için maliyet çok fazla olmayacağından iletim yöntemi olarak paralel iletim seçilmektedir. Paralel iletimde veriler gönderilecek bit sayısı kadar hat üzerinden iletildiği için hızlı bir iletim gerçekleşmektedir.



Şekil 4.4 Paralel İletim

4.3.2. Seri İletim

Seri iletim, verilerin tek bir hat ya da devre üzerinden aynı anda tek bir bit şeklinde ardı sıra aktarılmasıdır (Şekil 4.5). Bilgisayar ağları üzerinden veri alış verişi seri iletim ile gerçekleştirilmektedir [9]. Seri port bilgisayar veri yolundan aldığı 8,16 veya 32' lik paralel bit paketlerini 8,16 veya 32 lik seri bit akımlarına dönüştürmektedir. Seri haberleşme ismini bu prensipten almaktadır. İletilecek bilginin her biti bir yerden diğerine seri işareti iletmek için yalnızca bir işaret hattı ve toprak olmak üzere iki kabloya ihtiyaç duymaktadır [6].



Şekil 4.5 Seri İletim

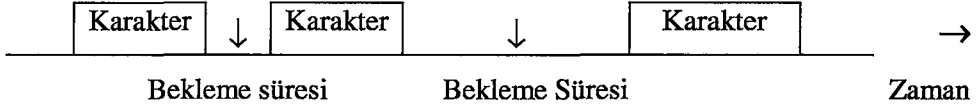
Seri iletim ile paralel iletim arasındaki farklılıkları şu şekilde belirtmek mümkündür;

- Paralel iletimde 8 bitlik veri alıcıya ulaştırılırken toprak ve eşleme de düşünülürse en az 10 hat kullanmak gerekmektedir. Seri iletimde ise iki hat bunun için yeterlidir
- Paralel iletimde veriler her bit için ayrı bir hat üzerinden iletildiğinden seri iletime göre daha masraflıdır.
- Paralel iletimde veriler ayrı hatlardan iletildiğinden seri iletime göre çok daha hızlıdır.
- Paralel iletim daha çok kısa mesafelerde, seri iletim ise daha uzak mesafelerde kullanılır.

Seri iletimi kendi arasında eşzamansız (Asenkron) ve eşzamanlı (Senkron) olmak üzere iki bölüme ayırmak mümkündür.

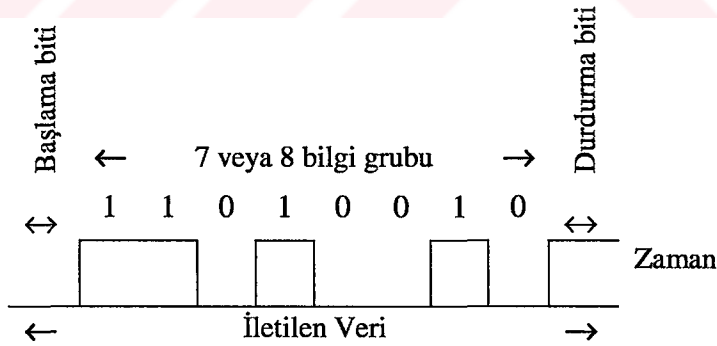
4.3.2.1. Eşzamansız (Asenkron) Seri İletim

Alıcı ve verici sistemin yerel saatlerini ayrı kullanarak çalıştıkları seri iletim şeklidir. Asenkron iletimde; verici sistemden, karakterler birbirlerinden bağımsız olarak iletilmektedir. Alıcı sistem ise aldığı her yeni karakter de yeniden eş zamanlama yapmaktadır (Şekil 4.6). Asenkron iletim eğer veri bir bilgisayardan yazıcıya veri gönderildiği gibi düzensiz aralıklarla gönderiliyorsa tercih edilmektedir. Çünkü bir bilgisayardan yazıcıya ne zaman bilgi gönderileceği veya hangi sıklıkla bilgi gönderileceği tahmin edilememektedir.



Şekil 4.6 Asenkron İletim Modeli

Asenkron veri iletiminde bilgi karakter adı verilen gruplara ayrılarak gönderilmektedir. Her bir grup 7 veya 8 bitten meydana gelmektedir. Gönderilen veri bu bit sayısı ile sınırlı kalmamaktadır. Gönderilecek olan karakterin başına özel bir başla (Start) biti eklenmektedir. Başla biti "0" olarak gönderilmektedir. Karakterin sonunda ise dur (stop) bitinin eklenmesi gerekmektedir. Dur biti "1" olarak gönderilmektedir. Bazı durumlarda hata sezme için kullanılacak bir eşlik (Parity) biti eklenebilmektedir. Eşlik biti "1" veya "0" olarak seçilebilir. Şekil 4.7' de 11010010 verisinin asenkron iletim ile nasıl gönderildiği gösterilmektedir.



Şekil 4.7 Asenkron İletim İle Veri Gönderimi

Gönderilecek olan son grup dur biti değerini aldıktan sonra verici sistemden yeni bir bilgi gönderilene kadar alıcı sistem hattı dur biti ("1") düzeyinde bekletmektedir. Eğer verici sistem tarafından yeni bir karakter grubu yollanacaksa verici sistem başlama biti ("0") göndererek alıcı sistemin yeni veri grubu geleceğini anlamasını sağlamaktadır. Böylece alıcı

sistem verici sistemden gelen verilerin başlangıç ve sonunu birbirinden ayırt ederek eş zamanlama yapmaktadır.

Eşzamansız veri iletiminin avantajları şu şekilde sıralanabilmektedir:

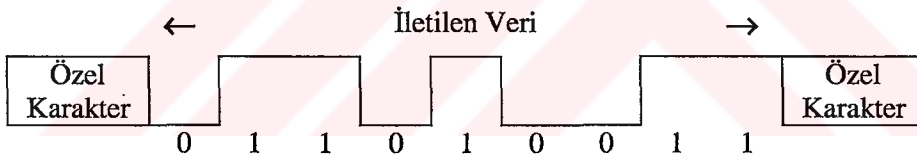
- Düşük donanım bileşenleri ile uygulanabilmektedir.
- ASCII terminaller ve veri girişleri için uygundur.
- Bit hataları hızlı bir şekilde gösterilebilmektedir.
- İletim düşük hızlı olduğu için hata olma olasılığı azalmaktadır.

Eşzamansız veri iletiminin dezavantajları ise şu şekilde sıralanabilmektedir:

- İletim hızı çok yavaş olduğu için verimi fazla değildir.
- Başlama ve Durdurma bitleri de pakete eklendiğinden karakter başına fazladan yük getirmektedir.

4.3.2.2. Eşzamanlı (Senkron) Seri İletim

Eşzamanlı iletim; veriyi içeren tüm paket karakterler arasında herhangi bir gecikme olmadan sürekli bilgi grupları şeklinde gönderilen seri iletim yöntemidir. Eşzamanlı iletimde verilerin başına ve sonuna başlama biti ya da durdurma biti koyulmamaktadır. Eşzamanlı iletimde bilgi çevresinin başına ve sonuna özel karakterler koyularak alıcının bilgiyi doğru alması, gelen verinin nerede başlayıp nerede sonlandığını anlaması sağlanmaktadır (Şekil 4.8).



Şekil 4.8 Senkron İletim İle Veri Gönderimi

Eşzamanlı iletimin bir türevi olarak bir de Isokron iletim bulunmaktadır. Bu iletimde uç sistemlerin birbirleriyle olan haberleşme gereksinimi periyodik olarak karşılanmaktadır; Bu periyotlar ile iletim ihtiyacı olan yol kapasitesi garanti altında tutulmaktadır[9].

Eşzamanlı veri iletiminin avantajları şu şekilde sıralanabilir:

- İletim yüksek hızda gerçekleşmektedir.
- Hata bulma metodu oldukça güvenlidir.

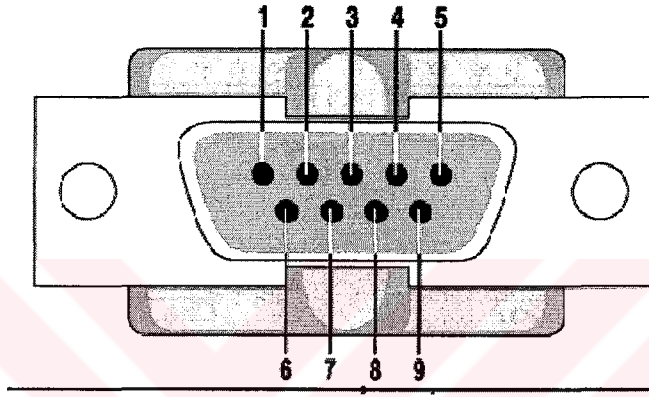
Eşzamanlı veri iletiminin dezavantajları ise şu şekilde sıralanabilir:

- Maliyeti yüksek uygulamalardır.
- İletişim protokolleri her zaman uygun olmalıdır.
- Tek bir bit de bile hata olsa giriş gruplarının yeniden gönderilmesi gerekebilmektedir.

4.4. Seri Portlar ve RS-232 Ara yüzü

COM Portlar olarak da adlandırılabilinecek seri portlar gelen verilerin seri iletim yöntemleriyle her seferinde tek bir bit şeklinde iletildiği yapılardır. Çünkü port her yön için sadece bir tek veri hattına sahip olmaktadır.

RS-232 (Recommended Standart 232) Veri uçbirim elemanı ile veri haberleşme elemanı arasında seri veri alış verişi ile bir arabirim oluşturmak için kullanılmaktadır. RS-232 seri portlarını kullanan aygıtlarda 9 pinli DB-9 (Şekil 4.9) veya 25 pinli DB-25 (Şekil 4.10) konnektörleri kullanılmaktadır. Bu konnektörleri adaptörler yardımıyla biri birlerine dönüştürmek de mümkündür.

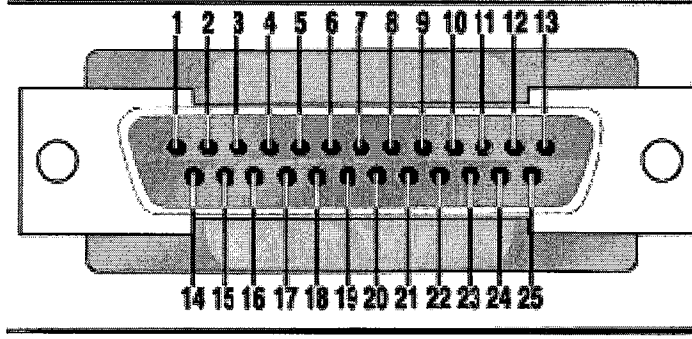


Şekil 4.9 DB-9 (Erkek) Seri Port

Şekil 4.9’ da 9 pinli erkek bir seri port bağlantı bacakları gösterilmiştir. Pinlerin görevleri Tablo 4.1 de belirtilmektedir.

Tablo 4.1 DB-9 (Erkek) Seri Portunun Pin Çıkışlarının Açıklaması

Pin	Açıklama
1	Taşıyıcı Tespiti (Data Carrier Detect)
2	Alınan Veri (Received Data)
3	İletilen Veri (Transmitted Data)
4	Veri Uç birimi Hazır (Data Terminal Ready)
5	İşaret Toprağı (Signal Ground)
6	Veri Kümesi Hazır (Data Set Ready)
7	Gönderme Talebi (Request to send)
8	Gönderme için hazır (Clear to send)
9	Ring Tespitçisi (Ring Detector)



Şekil 4.10 DB–25 (Erkek) Seri Port

Şekil 4.10’ da 25 pinli erkek bir seri port bağlantı bacakları gösterilmiştir. Pinlerin görevleri Tablo 4.2 de belirtilmektedir.

Tablo 4.2 DB–25 (Erkek) Seri Portunun Pin Çıkışlarının Açıklaması

Pin	Açıklama
1	Koruyucu Toprak (Protective Ground)
2	İletilen Veri (Transmitted Data)
3	Alınan Veri (Received Data)
4	Gönderme Talebi (Request to Send)
5	Gönderme İçin Hazır (Clear to Send)
6	Veri Kümesi Hazır (Data Set Ready)
7	İşaret Toprağı (Signal Ground)
8	Taşıyıcı Tespiti (Data Carrier Detect)
9	Pozitif test için kullanılır
10	Negatif test için kullanılır
11	Tanımlı Değil
12	İkincil CD (Secondary Rcvd Line Sig. Detector)
13	İkincil CTS (Secondary Clear to Send)
14	İkincil TX (Secondary Transmitted Data)
15	Gönderici işaret zamanlaması (Transmitter sig. Element Timing)
16	İkincil RTX (Secondary Received Data)
17	Alıcı işaret zamanlaması (Receiver Signal Timing)
18	Tanımlı Değil
19	İkincil RTS (Secondary Request to Send)

20	Veri Uç birimi Hazır (Data Terminal Ready)
21	İşaret Kalitesi Tespiti (Signal Quality Detect)
22	Ring Tespitçisi (Ring Detector)
23	Veri İşaret Oranı Seçicisi (Data signal Rate Selector)
24	Gönderici İşaret Zamanlaması (Transmitter sig. Element Timing)
25	Tanımlı Değil

DB-9 veya DB-25 pinli seri port soketlerinin pinlerinin erkek (Male) olanlarının yerine dişileri (Female) kullanılacak olursa pinlerin görevleri değişmektedir (Şekil 4.11). Özellikle seri portlar üzerinden çalışacak şekilde hazırlanan uygulamalarda veri alıp verirken kullanılan 2. ve 3. pinlerinin yerleri kullanılan konnektörün erkek veya dişi olup olmamasına göre çok önem taşımaktadır.



Şekil 4.11 Seri port Konnektörlerinin Dişi Tarafları

4.4.1. Evrensel Asenkron Alıcı Verici (Universal Asencronous Reciever Transmitter)

Bilgisayar sistemleri baytlar ile çalışmaktadır, fakat seri port bir seferde bir bit iletilmesine izin vermektedir. Tek şeritli bir yoldan sekiz adet arabanın yan yana geçmesinin imkânı olmadığı gibi bir seri porttan da bir baytın geçmesi mümkün olmamaktadır. Bu sebeple Her baytı seri porttan gönderilebilecek şekilde teker teker bitlerine ayırabilecek bir yapıya ihtiyaç duyulmaktadır. Giriş/Çıkış (I/O) kartlarında yerleşik olarak bulunan Evrensel Asenkron Alıcı Verici (Universal Asencronous Reciever Transmitter – UART) bu işlemi gerçekleştirmektedir. UART baytları seri porttan gönderilebilecek seri bitlere dönüştürmektedir. UART ayrıca gelen bitlerin bilgisayar sistemi tarafından işlenebilmesi için bunları baytlara çevirmektedir.

Bilgisayar sistemlerinin UART' ları hem tam çift yönlü hem de yarım çift yönlü iletişim hatlarını desteklemektedir [10].

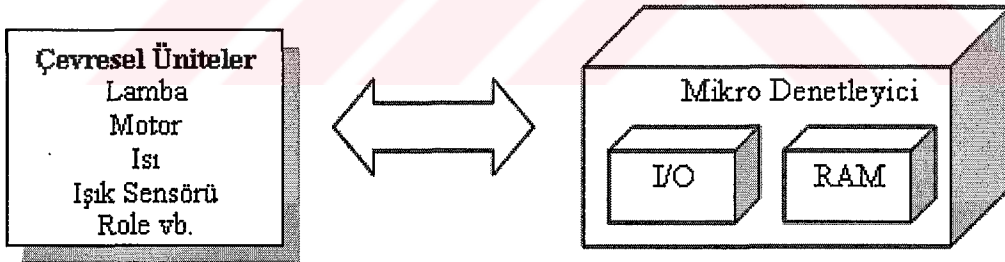
5. MİKRO DENETLEYİCİLER

5.1. Mikro Denetleyicilere Giriş

Mikro denetleyiciler, bugünkü koşullara bütünleşik devre teknolojisini geliştirerek ulaşmışlardır. Bu gelişme, tek bir yonga üzerinde yüzlerce, binlerce transistörün bulunmasını mümkün kılmaktadır. Bu mikro işlemci için bir gereklilik sayılmaktadır ve ilk bilgisayarlar; hafızayı içeren dış devreler, giriş- çıkış birimleri, zamanlayıcılar vb. den yapılmaktaydılar.

Ayrıca küçük denklemlerin seviye içeriği, bütünleşik devrelerin yaratılmasıyla sonuçlanmıştır. Bu bütünleşik devreler, yöntem ve çevrenin her ikisini de içermektedir. Bu, ilk yonganın, nasıl oluyor da bir mikro bilgisayar içerdiğini veya sonrasında mikro denetleyici olarak bilinmesini açıklayabilmektedir [11].

Bir bilgisayar içerisinde bulunması gereken temel bileşenlerden RAM ve I/O ünitesinin tek bir yonga içerisinde üretilmiş biçimine mikro denetleyici (Microcontroller) denilmektedir. Bilgisayar teknolojisi gerektiren uygulamalarda kullanılmak üzere tasarlanmış olan mikro denetleyiciler, mikro işlemcilere göre çok daha basit ve ucuzdurlar. Günümüz mikro denetleyicileri otomobillerde, kameralarda, cep telefonlarında, fax-modem cihazlarında, radyo, televizyon, bazı oyuncaklar gibi sayılamayacak kadar pek çok alanda kullanılmaktadır [12]. Ayrıca uzaktan kontrol sistemlerinde de tercih edilmektedir. Şekil 5.1’ de Bir mikro denetleyici sistemin temel bileşenlerinin bulunduğu blok diyagramı gösterilmektedir.



Şekil 5.1 Bir Mikro Denetleyici Sistemin Blok Diyagramı

Birçok Mikro işlemci üreticisinin ürettiği birkaç mikro denetleyicisi bulunmaktadır. Bu denetleyicilerin mimarileri arasında çok küçük farklar olmasına rağmen hemen hemen aynı işlemleri yapabilmektedirler. Her firma ürettiği yongaya bir isim ve özelliklerini birbirinden ayırmak için de parça numarası vermektedir. Örneğin Microchip Firması ürettiklerine PIC adını, parça numarası olarak da 12C508, 16C84, 16F84A, 16C711 gibi kodlamalar vermektedir. Intel ise ürettiği mikro denetleyicilere MCS-51 ailesi adını vermektedir. Genel olarak bu adla anılan mikro denetleyici ailesinde farklı özellikleri bulunan ürünleri birbirinden ayırt etmek için parça

numarası olarak da 8031 AH, 8051 AH, 8751AHP, 8052AH, 80C51FA gibi kodlamalar kullanılmaktadır.

5.2. Mikro İşlemciler

Mikro işlemci; Ön belleğine yazılan programı işleterek istenilen çıkışlara yönlendiren birimdir. Mikro işlemciler; Merkezi işlemci birimi, Giriş/Çıkış (I/O) birimi ve hafıza olmak üzere üç bölümden ve bunlara ek olarak bazı destek devrelerden oluşmaktadır [13].

5.3. Mikro İşlemcilere Karşı Mikro Denetleyiciler

Mikro işlemci ile kontrol edilecek bir sistemi kurmak için en azından şu üniteler bulunmalıdır; CPU, RAM, I/O ve bu ünitelerin arasındaki veri alış verişini kurmak için veri yolu (data bus) gerekmektedir. Mikro denetleyici ile kontrol edilecek sistemde ise gerekli olan ünitelerin yerine geçecek tek bir yonga (Mikro denetleyici) ve bir de devre kartı kullanmak yetecektir. Tek yonga kullanarak elektronik çözümler üretmenin maliyetinin daha düşük olacağı kesindir [12]. Ayrıca kullanım ve programlama kolaylığı da ikinci bir avantajdır. Bu sebeplerden dolayı bilgisayar kontrollü elektronik sistemlerin tasarımında mikro denetleyicilerden faydalanılmasında büyük artış olmaktadır.

5.4. PIC Mikro Denetleyicileri

Kısaca PIC adı verilen Peripheral Interface Controller (Çevresel Arabirim Kontrolcüsü) bir giriş-çıkış işlemcisidir. 1994 yılında Microchip firması tarafından 16 ve 32 bitlik büyük işlemcilerin, giriş ve çıkışlarındaki yükü azaltmak ve denetlemek amacıyla çok hızlı ve ucuz bir çözüme ihtiyaç duyulduğu için geliştirilmiştir [13].

PIC Mikro denetleyicilerin diğer mikro denetleyicilere göre daha çok tercih edilmelerinin sebepleri ise aşağıdaki gibi sıralanabilir:

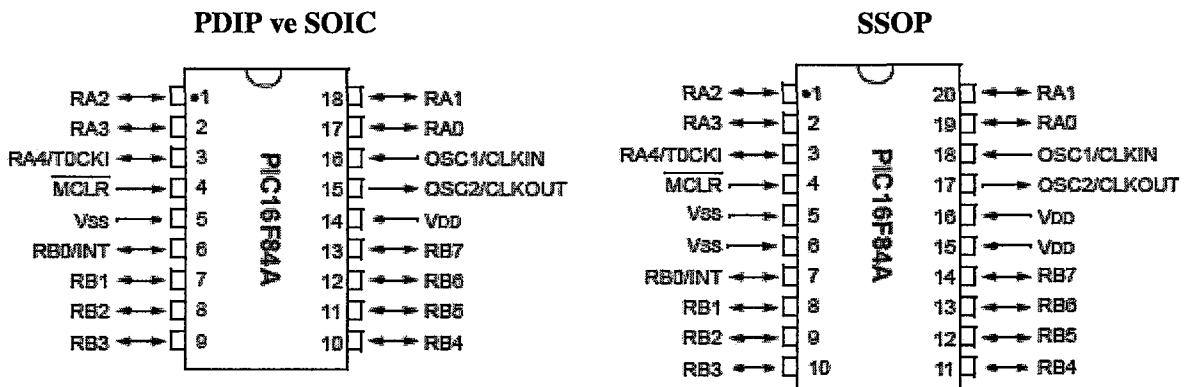
- Fiyatlarının düşük olması.
- Lojik uygulamalarının hızlı olması.
- Veri ve belleğe hızlı bir şekilde ulaşması.
- Herhangi bir ek belleğe ya da I/O elemanına ihtiyaç duymadan çalışabilmesi.
- Bir milyondan daha fazla yazılıp silinebilmeye imkân tanınması.
- Kod sıkıştırma özelliğinin bulunması.
- Birden fazla giriş çıkış portu bulunması.
- Programlanabilmesinin kolay olması.

5.4.1. PIC 16F84A Özellikleri

Microchip firmasının üretmiş olduğu PIC 16F84A modelinin genel özelliklerini şu şekilde belirtmek mümkündür [14]:

- 35 komut kullanılarak programlanabilmesi.
- 68 Bayt Data RAM bulunması.
- 1024 Bayt program hafızasının bulunması.
- 64 Bayt Data EEPROM bulunması.
- Donanım kayıtları için 15 Adet özel fonksiyonunun bulunması.
- Doğrudan, dolaylı ve göreceli adresleme modlarının olması.
- 8 Bit geniş data bayt bulunması.
- 14 Bit geniş yönlendirme Word bulunması.
- 13 Adet Giriş/Çıkış (I/O) Portu bulunması.
- Her bir pin den 25 mA akım verebilmesi.
- 8 Bit programlanabilir Timer bulunması
- 10.000 Defaya kadar silinip yazılabilecek artırılmış FLASH program hafızası bulunması.
- 10.000.000 Defaya kadar silinip yazılabilecek EEPROM Data hafızası bulunması.
- 2 Pin ile seri programlama devresinin gerçekleştirilebilmesi.
- Kod korumasının bulunması.
- Uyku modu ile güç koruması.

PIC 16F84A mikro denetleyicisinin Pin diyagramı Şekil 5.2' de gösterilmektedir. PIC16F84A mikro denetleyicisi PDIP (Plastic Dual In-line Package), SOIC (Small Outline Integrated Circuit) ve SSOP (Shrink Small Outline Package) olmak üzere üç yapıda bulunmaktadır. PDIP ve SOIC aynı pin çıkış görevlerine sahiptir.



Şekil 5.2 PIC 16F84A Mikro Denetleyicisinin PIN Diyagramı

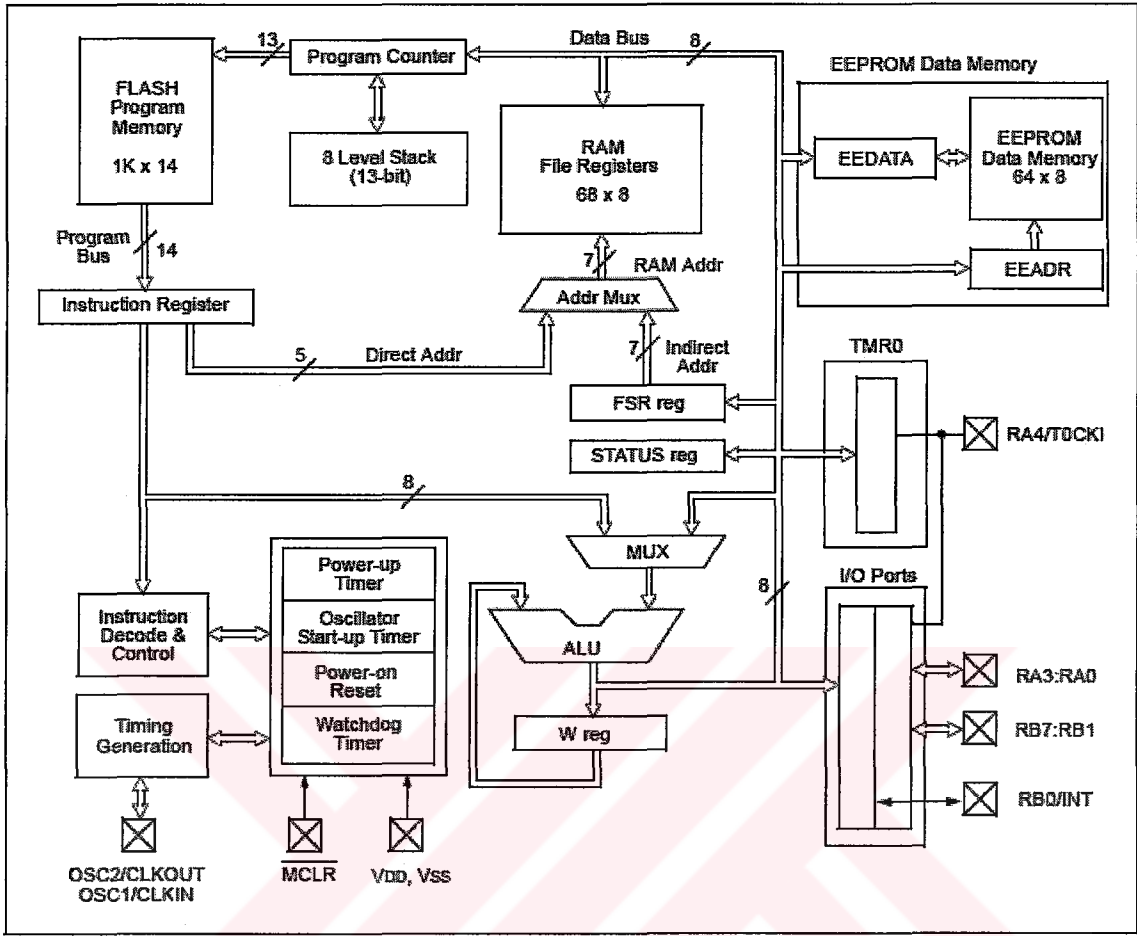
5.4.2. PIC 16F84A Pin Çıkışları ve Mimari Yapısı

PIC 16F84A mikro denetleyicisi 13 adet giriş çıkış portuna sahiptir. Bunlardan 8 bitlik olanına PORTB, 5 bitlik olanına da PORTA denilmektedir. PORTA nın 4 numaralı pini open kolektör olup giriş çıkış yapmak için bir direnç ile +5v a çekmek gerekmektedir. Diğer portlardan hiçbir ek elemana ihtiyaç duymadan giriş/çıkış yapılabilir. PIC 16F84A mikro denetleyicisinin Pin çıkışlarının açıklaması Tablo 6.1 deki gibidir.

Tablo 6.1 PIC 16F84A Pin Çıkışlarının Açıklamaları

Pin Adı	PDIP NO	SOIC NO	SSOP NO	Açıklama
OSC1/CLKIN	16	16	18	Oscillator crystal input/external clock source input. (Osilatör kristal girişi)
OSC2/CLKOUT	15	15	19	Oscillator crystal output. (Osilatör kristal çıkışı)
MCLR	4	4	4	Master Clear (Reset)
RA0	17	17	19	PORTA Giriş/Çıkış Pinleri
RA1	18	18	20	
RA2	1	1	1	
RA3	2	2	2	
RA4/TOCKI	3	3	3	
RB0/NT	6	6	7	PORTB Giriş/Çıkış Pinleri
RB1	7	7	8	
RB2	8	8	9	
RB3	9	9	10	
RB4	10	10	11	
RB5	11	11	12	
RB6	12	12	13	
RB7	13	13	14	
V _{SS}	5	5	5,6	Giriş/Çıkış pinleri için toprak
V _{DD}	14	14	15,16	Giriş/Çıkış pinleri için pozitif güç

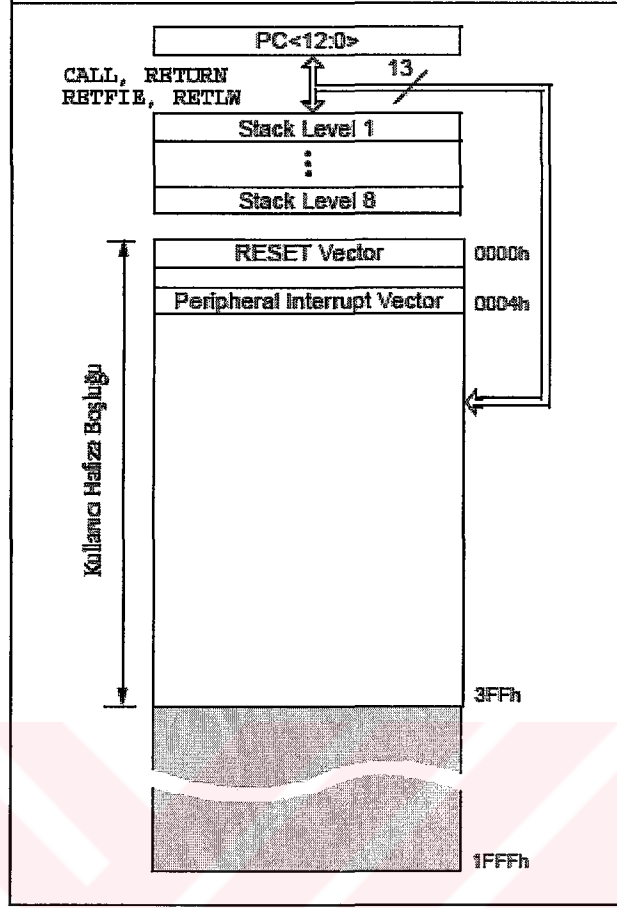
PIC 16F84A Mikro denetleyicisinin iç mimarisi Şekil 5.3 de gösterilmektedir.



Şekil 5.3 PIC 16F84A Mikro Denetleyicisinin Mimari Yapısı

5.4.3. PIC 16F84A Hafıza Organizasyonu

PIC 16F84A mikro denetleyicisi program hafızası (Program Memory) ve veri hafızası (Data Memory) olmak üzere iki adet hafıza bloğuna sahiptir. Her bir bellek kendi taşıyıcısına sahiptir; böylece her bir bloğa erişim aynı osilatör süreci boyunca meydana gelebilmektedir. Bunun ötesinde, veri belleği genel amaçlı RAM ve özel fonksiyon kayıtları (SFRs) olmak üzere ikiye bölünmektedir. SFR'ler her bir bireysel özelleşmiş modülü ele alan bölümde açıklanan özel modülleri kontrol etmek için kullanılmaktadır. Veri belleği EEPROM veri belleğini de içermektedir. Bu bellek, direkt veri belleğine planlanmamış, fakat dolaylı olarak planlanmıştır ve dolaylı adres göstergeleri okumak/yazmak için EEPROM belleğinin adresini belirlemektedir. EEPROM belleği 64 bayt ve 0h-3Fh adres enine sahiptir. PIC 16F84A mikro denetleyicisinin Program hafıza haritası Şekil 5.4' de gösterilmektedir.



Şekil 5.4 PIC 16F84A Mikro Denetleyicisinin Program Hafıza Haritası

5.4.3.1. Program Hafızası

PIC 16F84A mikro denetleyicisi 8K x 14 Program hafıza boşluğunu adresleyebilecek yetenekte 13 bit program sayacına sahiptir. PIC 16F84A mikro denetleyicisi için Şekil 5.4' de gösterildiği gibi ilk 1Kx14 (0000h-03FFh) fiziksel olarak uygulanmaktadır.

5.4.3.2. Veri Hafızası

Veri belleği ikiye ayrılmaktadır. Birincisi özel fonksiyon kayıt alanı (SFR), diğeri ise genel amaçlı kayıt alanıdır. SFR' ler aygıtın işlemini kontrol etmektedirler. Veri belleğinin bölümleri kümelenmiştir. Bu kümeler BANK adını almaktadırlar. Bu hem SFR alanı hem de GPR alanı içinde geçerli olmaktadır. GPR alanı genel amaçlı RAM'in 16 bayt'ından daha fazlasına olanak 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. Veri belleğin tümüne ya direkt her kayıt dosyasının mutlak adreslerini kullanarak ya da dolaylı yoldan dosya seçim kaydı (FSR) üzerinden erişilebilmektedir. Dolaylı adresleme, veri belleğinin kümelenmiş alanına

erişmek için RP1: RPO' nun şimdiki değerlerini kullanmaktadır. Veri belleği genel amaçlı kayıt ve özel fonksiyon kaydını içeren iki kümeye bölünmektedir. RPO bitinin (STATUS <5>) silinmesiyle BANK 0 seçilir. RPO' nun kurulması BANK 1' i seçer. Her bir bank (küme) 7Fh (128 bytes) kadar genişler. Her bir kümenin ilk 12 yerleşimi özel fonksiyon kaydı için rezerve edilmiştir. Kalanı ise statik RAM olarak genel amaçlı kayıt yürütebilmektedir [14]. Şekil 5.5' de Yazmaç (Register) haritası gösterilmektedir.

File Address		File Address	
00h	Dolaylı Adresleme	Dolaylı Adresleme	80h
01h	TMR0	OPTION_REG	81h
02h	PCL	PCL	82h
03h	STATUS	STATUS	83h
04h	FSR	FSR	84h
05h	PORTA	TRISA	85h
06h	PORTB	TRISB	86h
07h	—	—	87h
08h	EEDATA	EECON1	88h
09h	EEADR	EECON2 ⁽¹⁾	89h
0Ah	PCLATH	PCLATH	8Ah
0Bh	INTCON	INTCON	8Bh
0Ch			8Ch
	Genel Amaçlı Kayıtlar (SRAM)	BANK 0 eşleşmeleri	
4Fh			CFh
50h			D0h
7Fh			FFh
Bank 0		Bank 1	

Şekil 5.5 PIC 16F84A Mikro Denetleyicisinin Yazmaç Haritası

Bank 0 da bulunan en önemli yazmaçlar PORTA, PORTB, TMR0 ve INTCON; bank 1 da bulunan en önemli yazmaçlar ise STATUS, TRISA ve TRISB olarak gösterilebilir. Bunlardan PORTA 5 adet; PORTB 8 adet Giriş/Çıkış (I/O) pinine sahiptir.

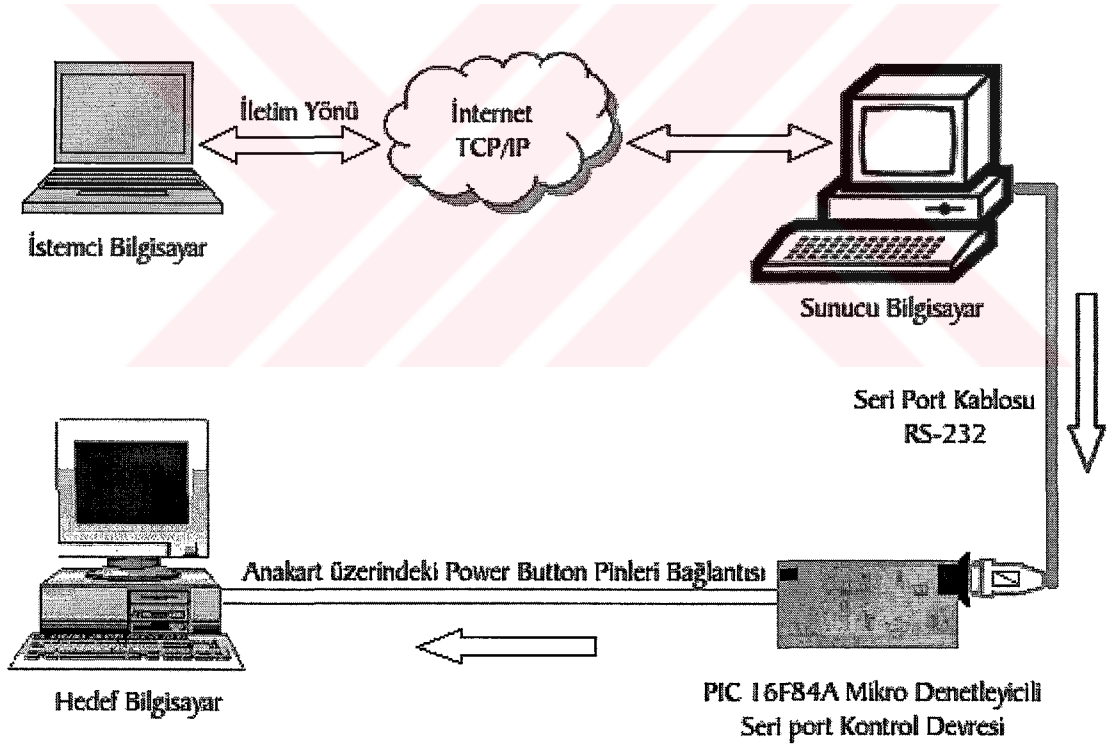
TRISA, PORTA' dan giriş mi yoksa çıkış mı yapılacağını; TRISB ise PORTB' den giriş mi yoksa çıkış mı yapılacağını belirleyen kısımdır. Örnek olarak TRISB=0 (0000 0000) verilirse bu PORTB nin hepsinin çıkış olduğu anlamına gelmektedir. TRISA=4 (0000 0100) verilirse bu PORTA nın 3.Pininin (RA2) çıkış olduğunu göstermektedir.



6. İNTERNET ÜZERİNDEN MİKRO DENETLEYİCİ KONTROLÜ

6.1. İnternet Üzerinden Yapılan Kontrol Sisteminin Genel Yapısı

Günümüzde internet günlük hayatımızın bir parçası olmuştur. Bu kapsamda internet üzerinden bir bilgisayarın güç sistemine ulaşabilmek mümkün olmaktadır. Bunun için öncelikle ASP veya PHP gibi web tabanlı programlama dilleri kullanılarak sunucu bir bilgisayarın seri portuna bilgi gönderilmektedir. Sunucu bilgisayarın seri portuna PIC 16F84A mikro denetleyicili bir devre bağlanarak sunucu bilgisayarın seri portundan gelen bilgi bu devrede işlenmektedir. İşlenilen bilgi iki koblo yardımıyla devrenin klemens uçları tarafından ulaşılacak istenilen hedef bilgisayarın ana kartı üzerindeki power button pinlerine sinyal olarak gönderilmektedir. Hazırlanan devre yardımıyla kapalı durumda bulunan sistem açılabilmiş veya sistem açık ise kapatılabilmiş ya da yeniden başlatılabilmektedir. Yapılan uygulamanın çalışma mantığı Şekil 6.1’ de gösterilmektedir.



Şekil 6.1 İnternet Üzerinden Başka Bir Bilgisayarın Güç Sisteminin Kontrolü

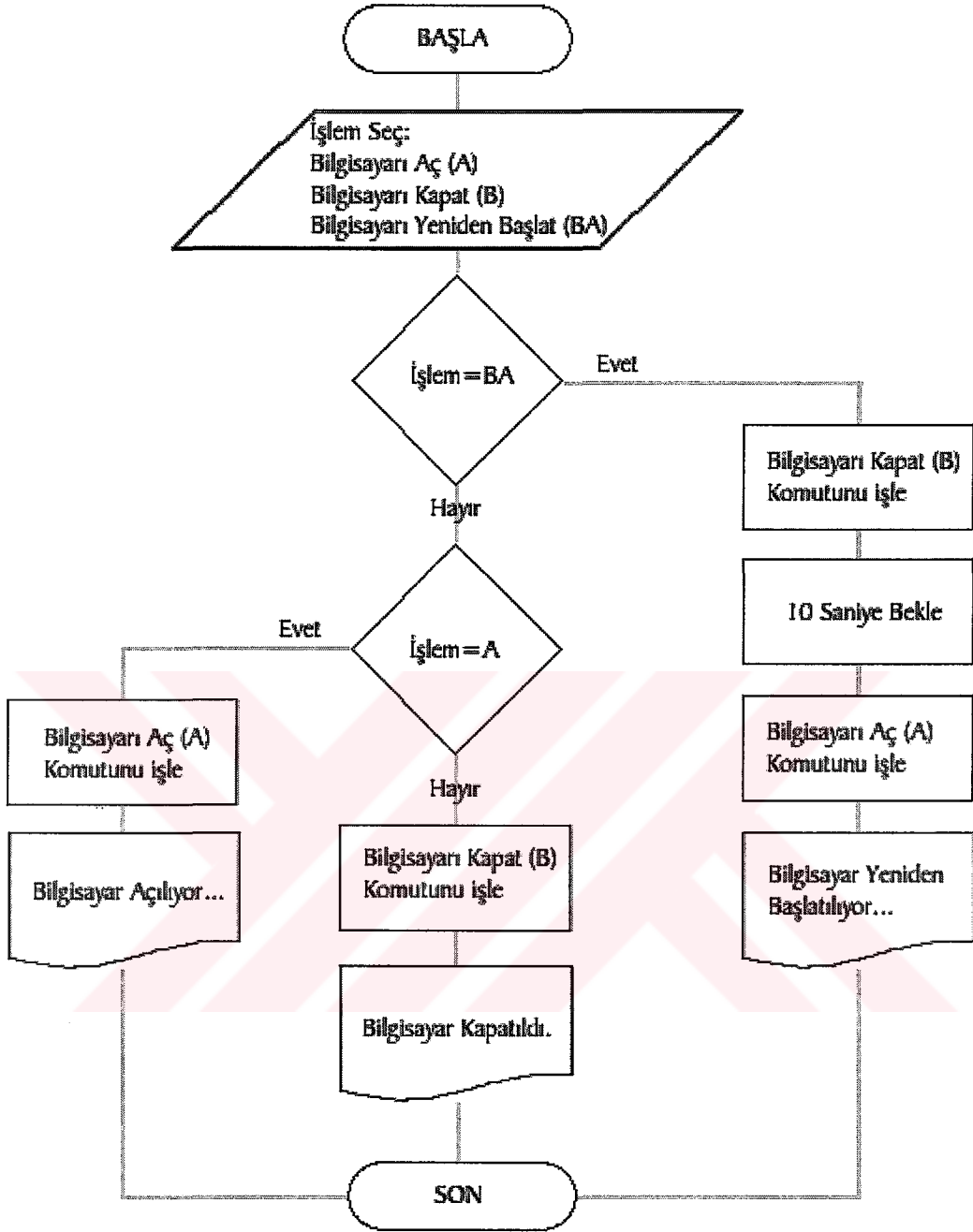
Bu uygulamada sunucu bilgisayarın web sunucusunda çalışan web sayfasına istemci bilgisayar bağlandıktan sonra burada yapmak istediği işlemi – Hedef bilgisayarı açmak, kapatmak veya yeniden başlatmak – seçmektedir.

Bu sistemin oluşturulması iki bölümde gerçekleştirilmiştir. Birinci bölümde web sunucu üzerinde çalışacak web tabanlı yazılım hazırlanmıştır. Bunu gerçekleştirebilmek amacıyla ve her türlü web sunucusunda veya işletim sisteminde çalışabilmesine imkan tanımak için yazılım hem ASP (Active Server Page) script dili hemde PHP (Personal Home Page) script dili kullanılarak iki farklı yöntemle hazırlanmıştır. Bu yazılımlardan istenilen birisi kullanılabilir. Yazılım oluşturulduktan sonra ise uygulamanın donanım kısmının yapımına geçilmiştir. Bunun için PIC 16F84A mikro denetleyicili bir devre tasarlanmıştır.

6.2. Uygulama Yazılımı

Bilgisayar ağları ve internet üzerinden bir web sunucusuna bağlanabilmek için hazırlanması gereken yazılımın öncelikle algoritması ve akış diyagramı oluşturulmuştur. Yazılımın akış diyagramı Şekil 6.2’ de gösterilmektedir

Akış diyagramında gösterildiği gibi öncelikle kullanıcıya bir form dosyası içerisinde gerçekleştirilmek istediği işlemi seçmesi istenmektedir. Kullanıcı ; ‘Bilgisayarı Aç’ , ‘Bilgisayarı Kapat’ ve ‘Bilgisayarı Yeniden Başlat’ seçeneklerinden hangisini seçerse sunucu bilgisayarın seri portuna o işlem ile ilgili olarak bir bilgi gönderilmektedir. Bilgisayarı aç için ‘A’ , Bilgisayarı kapat için ise ‘B’ bilgisi gönderilmektedir. Bilgisayarı yeniden başlatmak için ayrı bir bilgi gönderilmemektedir. Öncelikle sisteme bilgisayarı kapatma sinyali olan ‘B’ gönderilmekte ardından ise belli bir süre beklenerek bilgisayarı açma sinyali olan ‘A’ gönderilmektedir. Belli bir süre beklenmesinin sebebi ise günümüzde kullanılan bilgisayar kasalarında bilgisayarın kapatılabilmesi için açma/kapatma düğmesine belli bir süre boyunca basılması gerekliliğinden kaynaklanmaktadır.



Şekil 6.2 Kontrol Sistemi Yazılımının Akış Diyagramı

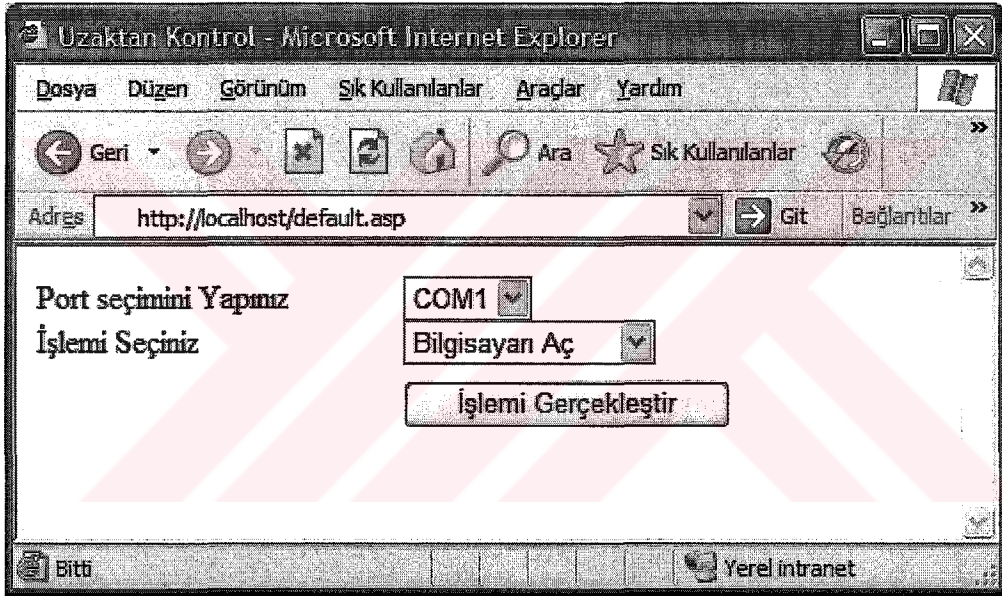
6.2.1. ASP İle Hazırlanan Yazılım

Hazırlanan ASP dosyasında Microsoft firmasının geliştirmiş olduğu MSComm kontrol bileşeni kullanılmıştır. Bu bileşenin kullanılabilmesi için sunucu bilgisayarda lisanslı mscomm32.ocx dosyasının kayıtlı olması gerekmektedir. Hazırlanan ASP yazılımı iki dosyadan oluşmaktadır. Birinci dosyada sunucu bilgisayarda hangi seri portun seçileceğinin ve hangi

işlemin yapılacağıının belirlendiği form dosyası bulunmaktadır. İkinci dosyada ise bu form dosyasından gelen verilerin yorumlanıp işlendiği ve seri porta veri gönderildiği kodlar bulunmaktadır. ASP dosyalarının görüntülenebilmesi için web sunucusunun ASP dosyalarını görüntüleyebilmesi gereklidir. En yaygın kullanılan web sunucu olarak Microsoft Firmasının üretmiş olduğu ve Windows İşletim sistemi üzerinde çalışan Internet Information Server (IIS) gösterilebilmektedir. Hazırlanan yazılım bu web sunucu üzerinde çalıştırılmıştır.

6.2.1.1. Form İle Bilgi Gönderilen ASP Dosyası

Hazırlanmış olan form dosyasının ekran görüntüsü Şekil 6.3’ de gösterilmiştir. Bu form dosyasında port seçimi ve işlem seçildikten sonra seçilmiş olan değerler verilerin işleneceği ASP dosyasına gönderilmektedir. Bu form dosyasının kodları EK-1’ de verilmiştir.



Şekil 6.3 Form ASP Dosyası

6.2.1.2. Formdan Gelen Bilgilerin İşlendiği ASP Dosyası

Form dosyasından gelen verilerin işlendiği dosyadır. Form dosyasında ‘İşlemi Gerçekleştir’ düğmesine basıldığında form etiketinin içerisindeki ‘action’ komutunda yazılan dosyaya formda bulunan ‘port’ ve ‘islem’ değişkenlerinin değerleri gönderilmektedir.

Formdaki kodlar incelendiğinde ‘port’ değişkeni ‘1’ ya da ‘0’ değerini ‘islem’ değişkeninin ise ‘A’, ‘B’ veya ‘BA’ değerlerini alabildikleri görülmektedir. Bu değerlerin işlendiği ASP dosyasının kodları satır numaraları ile birlikte aşağıda gösterildiği gibidir.

Kod Satırı	Program Kodu
1	<% @LANGUAGE="VBSCRIPT" %>
2	<%OPTION EXPLICIT%>
3	<%
4	Response.Buffer = False
5	DIM port,veri,MSComm1,c,basla
6	port=Request.Form("port")
7	veri=Request.Form("islem")
8	SET MSComm1 = Server.CreateObject ("MSCOMMLib.MSComm")
9	MSComm1.CommPort = port
10	MSComm1.Settings = "9600,N,8,1"
11	MSComm1.inputLen = 1
12	MSComm1.PortOpen = True
13	IF veri="BA" THEN CALL yenidenbaslat ELSE CALL ackapa
14	SUB ackapa
15	MScomm1.Output = veri
16	IF veri="A" THEN
17	Response.Write("Sistem Açılıyor")
18	ELSE
19	Response.Write("Sistem Kapatıldı")
20	basla=timer()
21	c=basla+12
22	do until (basla>c)
23	basla=timer()
24	Loop
25	END IF
26	END SUB
27	SUB yenidenbaslat
28	MScomm1.Output = "B"
29	basla=timer()
30	c=basla+12
31	do until (basla>c)
32	basla=timer()
33	Loop

```

34      MSComm1.Output = "A"
35      Response.Write("Sistem Yeniden Başlatılıyor...")
36      END SUB
37      MSComm1.PortOpen = False
38      %>

```

Program kodu satır numaralarına göre Tablo 6.1 de açıklanmıştır

Tablo 6.1 ASP Kodlarının Açıklaması

Kod Satırı	Kodun Açıklaması
1	Kullanılacak script dili olarak vbscript seçilmektedir.
2	Kullanılacak tüm değişkenlere tanımlanma zorunluluğu getirilmektedir.
3	ASP kodlarının başlama etiketi.
4	Hafızada veri tutuluyorsa kullanılmaması sağlanmaktadır.
5	Dosyada kullanılacak değişkenler tanımlanmaktadır.
6	Form dosyasından gelen 'port' değeri 'port' değişkenine aktarılmaktadır.
7	Form dosyasından gelen 'islem' değeri 'veri' değişkenine aktarılmaktadır.
8	MSComm Bileşenine bağlantı oluşturulmaktadır.
9	Porttan gelen değer MSComm bileşeninin CommPort parametresine aktarılıyor. Eğer gelen değer 1 ise COM1; 2 ise COM2 seçilmiş olmaktadır [15].
10	MSComm bileşeninin ayarları yapılmaktadır Baud hızı olarak 9600 seçilmektedir, Parity biti kullanılmamaktadır, data biti 8 seçilmektedir, stop biti ise 1 olarak seçilmektedir.
11	Port için giriş bilgisi uzunluğu 1 seçilmektedir.
12	Seri port açılmaktadır.
13	Veri değişkeninin değeri 'BA' ise form dosyasında sistemin yeniden açılmak istendiği seçilmiştir. Eğer seçim böyle ise 'yenidenbaslat' subrutini çağırılmaktadır. Veri değişkeninin değeri 'BA' dışında bir değer ise – A veya B– 'ackapa' subrutini çağırılmaktadır.
14	'ackapa' subrutini oluşturulmaktadır.
15	MSComm bileşeninin çıkışına veri değişkeninde bulunan değer gönderilmektedir. Böylece sinyal seri porta yollanmış olmaktadır.
16,17	Eğer 'veri' değişkeninin değeri 'A' ise sistemin başlatılmak istendiği seçilmiştir ekrana 'Sistem Açılıyor' yazısı yazdırılmaktadır.
18,19	Eğer veri değişkeninin değeri A değilse – B ise – Ekrana 'Sistem

	Kapatıldı' yazısı yazdırılmaktadır.
20,21	Timer fonksiyonu kullanılarak sistemin o anki süresi alınıyor ve c değişkenine 10 saniyeden büyük – 12 – bir değer aktarılıyor.
22–24	Oluşturulan döngü ile sistemin 12 saniye beklemesi sağlanıyor. Böylece kullanıcı sistemi kapat dedikten hemen sonra mikro denetleyicide sistemi kapat için verilen 10 saniyelik süreyi beklemeden form dosyasına geri dönüp 'Bilgisayarı Aç' komutunu vermesi engellenmiş olmaktadır. Çünkü bu süre beklenmeden 'Bilgisayarı Aç' denilirse sistem bunu yerine getiremeyecektir.
25	IF komutu sonlandırılmaktadır.
26	'ackapa' subrutini kapatılmaktadır.
27	'yenidenbaslat' subrutini oluşturulmaktadır.
28	Sistemin kapatılması için seri port çıkışına 'B' bilgisi gönderilmektedir.
29 – 33	Oluşturulan döngü ile sistemin 12 saniye beklemesi ve bu süre içerisinde hedef bilgisayarın kapatılması sağlanmaktadır.
34	Sistemin yeniden açılması için seri port çıkışına 'A' bilgisi gönderilmektedir.
35	Ekrana 'Sistem Yeniden Başlatılıyor...' yazılmaktadır
36	'yenidenbaslat' subrutini kapatılmaktadır.
37	Seri Port Kapatılmaktadır.
38	ASP kodlarının sonlandığını gösteren etiket.

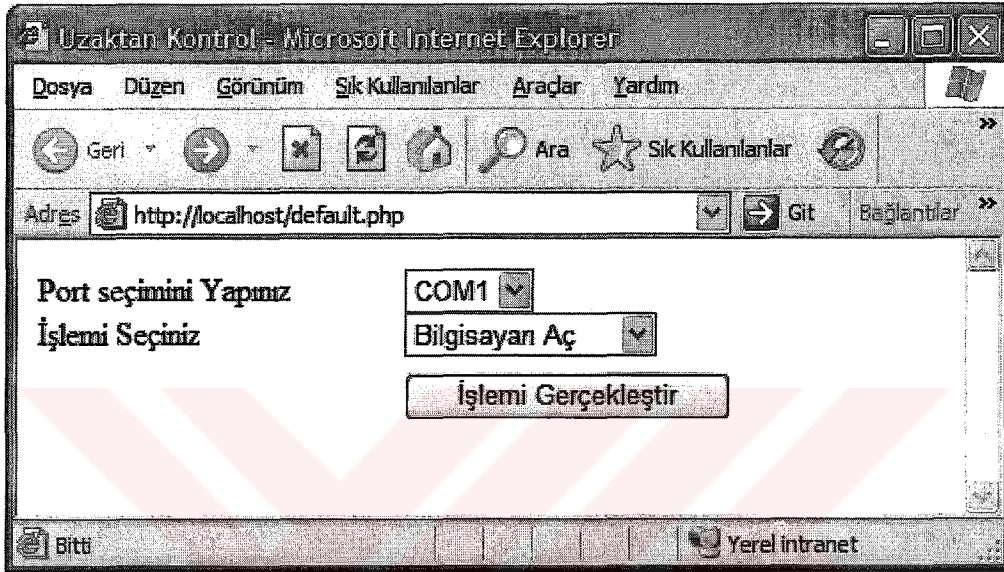
6.2.2. PHP ile Hazırlanan Yazılım

PHP ile hazırlanan yazılım da yine ASP yazılımı ile hazırlanan yazılım gibi iki dosyadan oluşmaktadır. İlk dosya da kullanıcının hangi seri portu seçeceği ve hangi seçimi yapacağı işlemlerini gerçekleştirdiği form dosyası bulunmaktadır. Bu PHP form dosyasının ASP form dosyasından tek farklı yönü seçilen seri portun ASP dosyasında olduğu gibi '1' veya '2' değerlerini işlenecek diğer dosyaya göndermek yerine 'COM1' ve 'COM2' değerlerini göndermesidir. ASP dosyalarını çalıştırmak için kullandığımız Internet Information Services (IIS) ile PHP dosyalarını çalıştıramadığımız için PHP dosyalarını çalıştırabilecek bir sunucuya ihtiyacımız bulunmaktadır. PHP dosyalarını çalıştırabilmek için Apache sunucusu kullanılmıştır. Apache Sunucu Linux işletim sistemi yüklü bilgisayar sistemlerinde kolayca kurulabilmektedir. Microsoft Windows işletim sistemi yüklü bilgisayar sistemlerine yüklenebilmek için ise üçüncü parti programlara ihtiyaç vardır. PHPTriad ya da ApacheTriad gibi yazılımlar ile Microsoft Windows işletim sistemi yüklü bilgisayar sistemlerine Apache

sunucu kurulabilmektedir. Oluşturulan PHP yazılımını denemek için sunucu bilgisayara PHPTriad programı yüklenmiştir.

6.2.2.1. Form İle Bilgi Gönderilen PHP Dosyası

PHP ile Hazırlanmış olan form dosyasının ekran görüntüsü Şekil 6.4' de gösterilmiştir. Bu form dosyasında port seçimi ve işlem seçildikten sonra seçilmiş olan değerler verilerin işleneceği PHP dosyasına gönderilmektedir. Bu form dosyasının kodları EK-2' de verilmiştir.

The image shows a screenshot of a Microsoft Internet Explorer browser window. The title bar reads 'Uzaktan Kontrol - Microsoft Internet Explorer'. The address bar shows 'http://localhost/default.php'. The main content area contains a form with two dropdown menus. The first dropdown is labeled 'Port seçimini Yapınız' and has 'COM1' selected. The second dropdown is labeled 'İşlemi Seçiniz' and has 'Bilgisayarı Aç' selected. Below these dropdowns is a button labeled 'İşlemi Gerçekleştir'. The status bar at the bottom shows 'Yerel intranet'.

Şekil 6.4 Form PHP Dosyası

6.2.2.2. Formdan Gelen Bilgilerin İşlendiği PHP Dosyası

Form dosyasından gelen verilerin işlendiği dosyadır. Form dosyasında 'İşlemi Gerçekleştir' düğmesine basıldığında form etiketinin içerisindeki action komutunda yazılan dosyaya formda bulunan 'port' ve 'işlem' değişkenlerinin değerleri gönderilmektedir. Formdaki kodlar incelendiğinde 'port' değişkeni 'COM1' ya da 'COM2' değerini 'işlem' değişkeni ise 'A', 'B' veya 'BA' değerlerini alabilmektedir. Bu değerlerin işlendiği PHP dosyasının kodları satır numaralarıyla birlikte aşağıdaki gibidir.

Kod Satırı	Program Kodu
1	<?php
2	\$portyaz=\$port;
3	\$islemyaz=\$islem;
4	\$fp = fopen (\$portyaz,"w+");
5	if (!\$fp)
6	{

```

7          $yazi = "port kullanımda veya veri gönderilemiyor";
8          echo $yazi ;
9      }
10     ELSE
11     {
12         if ($islemyaz=="BA")
13         {
14             echo " Sistem yeniden açılıyor..." ;
15             fwrite($fp, B);
16             fflush($fp);
17             sleep(12);
18             fwrite($fp, A);
19         }
20     ELSE
21     {
22         IF ($islemyaz=="A")
23         {
24             sleep(12);
25             echo " Sistem Açılıyor ..." ;
26         }
27     ELSE
28     {
29         echo " Sistem Kapatıldı" ;
30     }
31     fwrite ($fp,$islemyaz);
32     fflush($fp);
33     }
34 }
35 fclose ($fp);
36 ?>

```

Program kodu satır numaralarına göre Tablo 6.2 de açıklanmıştır.

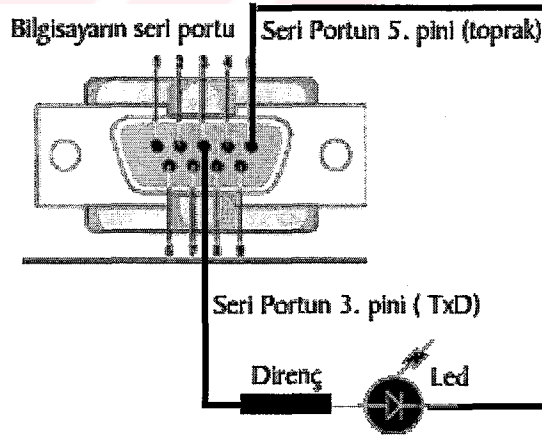
Tablo 6.2 PHP Kodlarının Açıklaması

Kod Satırı	Kodun Açıklaması
1	PHP kodlarının başlama etiketi.
2	Formdan gelen 'port' değişkeni 'portyaz' değişkenine aktarılmaktadır.
3	Formdan gelen 'islem' değişkeni 'islemyaz' değişkenine aktarılmaktadır.
4	'portyaz' değişkeninden gelen port yazmaya açık olarak (w+) açılıyor ve bu değer 'fp' değişkenine aktarılıyor. Burada win32 sistemlerinde w+ yerine wb de kullanılabilinmektedir.
5-9	Eğer port açılmıyorsa ekrana bir hata olduğuna dair 'port kullanımda veya veri gönderilemiyor' yazısı yazdırılmaktadır.
10,11	Eğer bir portu yazmaya açık bir şekilde açmakta bir problem yoksa formdan gelen bilgilerin işleneceği bloğa girilmektedir.
12,13	Eğer 'islemyaz' değişkeninin değeri 'BA' ise form dosyasında hedef bilgisayarın yeniden açılmak istendiği seçeneği seçilmiştir. Yeniden başlatma işleminin gerçekleştirileceği bloğa girilmektedir.
14	Ekrana yeniden 'Sistem yeniden açılıyor...' yazısı yazdırılmaktadır.
15	Açılan porta 'B' sinyali gönderilerek hedef bilgisayarın kapatılması sağlanmaktadır.
16	fflush fonksiyonu ile porta yeniden bilgi gönderilebilinmesi için fp değişkeni ile açılan porta gönderilen veri silinmektedir.
17	Kapatma işleminin gerçekleştirilebilmesi için sistemin 10 saniyeden fazla bir süre -12 saniye- bekletilmesi sağlanıyor. Eğer bekleme yapılmadan aç sinyali - A - gönderilirse mikro denetleyici sistem bu işlemi gerçekleştiremeyecektir. Çünkü mikro denetleyicili devrede bulunan PIC16F84A içerisine yazılan kodda kapatma sinyalini 10 saniye boyunca iletmesi kodlanmaktadır.
18	Porta hedef bilgisayarın açılabilmesi için 'A' sinyali gönderilmektedir.
19	Yeniden açma işleminin gerçekleştirildiği blok kapatılıyor.
20,21	Eğer 'islemyaz' değişkeninin değeri 'BA' değilse - 'A' veya 'B' ise - bu bölümle ilgili işlemlerin gerçekleştirileceği bloğa girilmektedir.
22-23	Eğer 'islemyaz' değişkeninin değeri 'A' ise bu bölümle ilgili işlemlerin yapılacağı bloğa girilmektedir.
24	Blok içerisinde 12 saniye beklenmesi sağlanmaktadır. Bu bekleme ile kullanıcı bilgisayarı kapat komutunu verdikten hemen sonra forma geri dönüp beklenmesi gereken 10 saniyelik süreyi beklemeden tekrar 'Bilgisayarı Aç'

	komutunu verirse bu beklemin sistem tarafından yapılması sağlanmaktadır.
25	Ekrana ‘ Sistem Açılıyor ...’ yazısı yazılmaktadır.
26	Bilgisayarı açma işleminin gerçekleştirildiği blok kapatılmaktadır.
27–30	Eğer ‘islemiyaz’ değişkeninin değeri ‘A’ değilse – ‘B’ – ekrana ‘ Sistem Kapatıldı’ yazısı yazılmaktadır.
31	‘islemiyaz’ değişkeninde bulunan değer porta gönderilmektedir.
32	Porta gönderilen veri silinmektedir.
33	‘islemiyaz’ değişkeninin ‘BA’ olmaması halinde işlenilecek blok kapatılmaktadır.
34	‘fp’ değişkeninin ‘0’ olmaması durumunda – port yazmaya açık olarak açılırken bir hata olmaması durumunda – işletilecek olan blok kapatılıyor.
35	Seri port kapatılmaktadır.
36	PHP kodlarının sonlandığını gösteren etiket.

6.2.3. Yazılımların Test Edilmesi

ASP ve PHP web tabanlı programlama dilleriyle hazırlanmış yazılımların test edilmesi için Şekil 6.5’ de gösterildiği gibi yazılımın çalıştığı bilgisayar sisteminin seri portuna bağlı basit bir devre oluşturulmuştur. Bu devre sadece bir led ve dirençten oluşmaktadır. Bu şekilde asıl mikro denetleyicili kontrol devresi hazırlanmadan oluşturulmuş ASP ve PHP dosyalarının sağlıklı bir şekilde çalışıp çalışmadığı led in yanıp yanmamasına göre test edilmiştir. Direnç konulmasının sebebi ise bilgisayarın seri portuna herhangi bir zarar gelmesini engellemektir.



Şekil 6.5 ASP ve PHP Yazılımı Test Devresi

6.3. Uygulama Devresi

Uygulamanın yazılım aşaması tamamlandıktan sonra donanım aşamasına geçilmiştir. Donanımın en önemli kısmını, sunucu bilgisayarın seri portundan gönderilen verileri değerlendirilen PIC serisi mikro denetleyici oluşturmaktadır. Gerek fiyatının ucuz olması gerekse yeterli donanım özelliklerine sahip olması nedeniyle PIC 16F84A mikro denetleyicisi tercih edilmiştir. PIC 16F84A nın istenen şekilde çalışmasını sağlayan program PIC C derleyicisi kullanılarak yazılmıştır. Üst seviye bir dil olan C dili kullanılarak hem programın daha kısa olması hem de daha anlaşılır olması sağlanmıştır. Derleme işlemi sonrasında oluşan hex uzantılı dosya, IC-PROG yazılımı ile PIC 16F84A mikro denetleyicisine yüklenmiştir. Yükleme için seri porttan beslenen bir PIC programlayıcı kartı kullanılmıştır.

6.3.1. PIC 16F84A Mikro Denetleyicisi İçin Geliştirilen Yazılım

Sunucu bilgisayarın seri portundan gelecek olan bilgiyi alıp işleyecek olan seri port kontrol devresindeki PIC 16F84A mikro denetleyicisi için hazırlanan yazılımın kodları aşağıda gösterilmektedir.

Kod Satırı	Program Kodu
1	#include <pic.h>
2	#include <delay.c>
3	#include <serial.c>
4	#include <stdio.h>
5	void bekle1(void){
6	unsigned char i;
7	for(i=0;i<4;i++){
8	DelayMs(250);
9	}
10	}
11	void bekle10(void){
12	unsigned char i;
13	for(i=0;i<40;i++){
14	DelayMs(250);
15	}
16	}
17	main(void) {
18	unsigned char c,kontrol1,kontrol2;

```

19      TRISB=0;
20      TRISA=4;
21      PORTA=0;
22      PORTB=0;
23      kontrol1=0; kontrol2=0;
24      for(;;){
25          c=getch();
26          if(c=='A'){
27              if(kontrol1==0){
28                  RB1=1;
29                  RB2=0;
30                  RB0=1;
31                  bekle1();
32                  RB0=0;
33                  kontrol1=1;
34                  kontrol2=0;
35              }
36          }
37          if(c=='B'){
38              if(kontrol2==0){
39                  RB1=0;
40                  RB2=1;
41                  RB0=1;
42                  bekle10();
43                  RB0=0;
44                  kontrol1=0;
45                  kontrol2=1;
46              }
47          }
48      }
49  }

```

PIC 16F84A mikro denetleyicisi için hazırlanan programın kodları satır numaralarına göre Tablo 6.3 de açıklanmaktadır.

Tablo 6.3 PIC 16F84A Mikro Denetleyicisi İçin Hazırlanan Programın Açıklaması

Kod Satırı	Kodun Açıklaması
1-4	Program içerisinde kullanılacak olan bazı fonksiyonların tanımlı olduğu kütüphane dosyaları program içerisine dâhil ediliyor.
5-10	Program içerisinde kullanılacak olan 1 saniyelik bekleme için gerekli fonksiyon oluşturuluyor. Fonksiyonda delay.c kütüphanesinde tanımlı DelayMs komutu ile en fazla 255 milisaniyelik gecikme sağlanabildiği için, bir for döngüsü kullanılarak gerekli süre oluşturulmaktadır. For döngüsü içerisinde 4 kez çağrılan DelayMs(250) komutu ile 1 saniyelik bekleme süresi sağlanmaktadır.
11-16	Program içerisinde kullanılacak olan 10 saniyelik bekleme için gerekli fonksiyon oluşturuluyor.
17	Program çalıştırıldığı zaman ilk işlenecek olan ana fonksiyon oluşturuluyor.
18	Programda kullanılacak olan değişkenler tanımlanıyor.
19	PORTB' nin tümünün çıkış olduğunu göstermektedir (0000 0000)
20	PORTA' nın 3. bacağı (RA2) giriş olarak seçilmektedir (0000 0100)
21	PORTA değeri sıfırlanıyor
22	PORTB değeri sıfırlanıyor
23	'kontrol1' ve 'kontrol2' değişkenlerine başlangıçta '0' değeri veriliyor. 'kontrol1' ve 'kontrol2' değişkenlerinin kullanılma sebebi PIC 16F84A mikro denetleyicisinin açma veya kapatma işlemini gerçekleştirdikten sonra tekrar aynı işlemi dikkate almaması içindir. Böylece açma işleminden sonra kapat komutu; kapatma işleminden sonra da açma komutu verilmedikçe program bu komutları işlememektedir. Bu da programın yanlış çalışmasını önlemiş olmaktadır.
24	Sonsuz bir döngü oluşturulmaktadır
25	getch komutu ile seri porttan okunan 8 bitlik veri ('A' veya 'B' karakteri), 'c' değişkenine atanmaktadır.
26	'c' değişkeninin değeri 'A' ise sistemin açılması istenmektedir. Bu işlem ile ilgili blok oluşturulmaktadır.
27	'kontrol1' değişkeninin değerine bakılıyor eğer '0' ise açma işlemi için gerekli kodlar işleniyor. Bu değer 0 değilse aç komutunun tekrar işlenmesi önlenmiş olmaktadır.
28	Yeşil ledin bağlı olduğu RB1 bacağına '1' değeri verilip yeşil led

	yakılarak sisteme aç komutunun gönderilmiş olduğu gösterilmektedir.
29	Kırmızı ledin bağlı olduğu RB2 bacağına '0' değeri verilerek kırmızı led söndürülmektedir.
30	Rölenin bağlı olduğu RB0 bacağına '1' değeri verilerek rölenin enerjilenmesi ve kontağını kapatması sağlanmaktadır.
31	1 saniyelik bekleme süresi için bekle1() fonksiyonu çağırılmaktadır.
32	Rölenin bağlı olduğu RB0 bacağına '0' değeri verilerek rölenin enerjisinin kesilmesi ve kontağının açılması sağlanmaktadır.
33	'kontrol1' değişkenine '1' değeri verilerek bilgisayarı aç işleminin işlendiği belirtilmektedir.
34	'kontrol2' değişkenine '0' değeri verilerek bilgisayarın kapatılmaya hazır olduğu belirtilmektedir.
35	'kontrol1' değişkeni için oluşturulan şart bloğu kapatılmaktadır.
36	'Bilgisayarı Aç' komutunun gönderildiğinin anlaşıldığı şart bloğu kapatılmaktadır.
37	'c' değişkeninin değeri 'B' ise bilgisayarı kapat komutunun gönderilmiş olduğu anlaşılmakta ve bu işlem ile ilgili bloğa girilmektedir.
38	'kontrol2' değişkeninin değeri kontrol edilmektedir. Eğer 'kontrol2' değişkenini değeri '0' ise şu anda bilgisayarın kapalı durumda olmadığı anlaşılmaktadır.
39	Yeşil ledin bağlı olduğu RB1 bacağına '0' değeri gönderilerek yeşil led söndürülmektedir.
40	Kırmızı ledin bağlı olduğu RB2 bacağına '1' değeri gönderilerek kırmızı led yakılmaktadır. Böylece devreye 'Bilgisayarı Kapat' işleminin gönderildiği anlaşılmaktadır.
41	Rölenin bağlı olduğu RB0 bacağına '1' değeri gönderilmektedir.
42	Sistemin kapatılabilmesi için 10 saniye boyunca bekleyecek bekle10() fonksiyonu çağırılmaktadır.
43	Rölenin bağlı olduğu RB0 bacağına '0' değeri gönderilerek sinyal kesilmektedir
44	'kontrol1' değişkenine '0' değeri atanarak sisteme bilgisayarı aç komutunun verilebileceği belirtilmektedir.
45	'kontrol2' değişkenine '1' değeri aktarılarak sisteme bilgisayar kapalı iken yeniden bilgisayarı kapat komutunun verilebilmesi engellenmektedir.
46	'kontrol2' değişkeni için oluşturulan şart bloğu kapatılmaktadır.

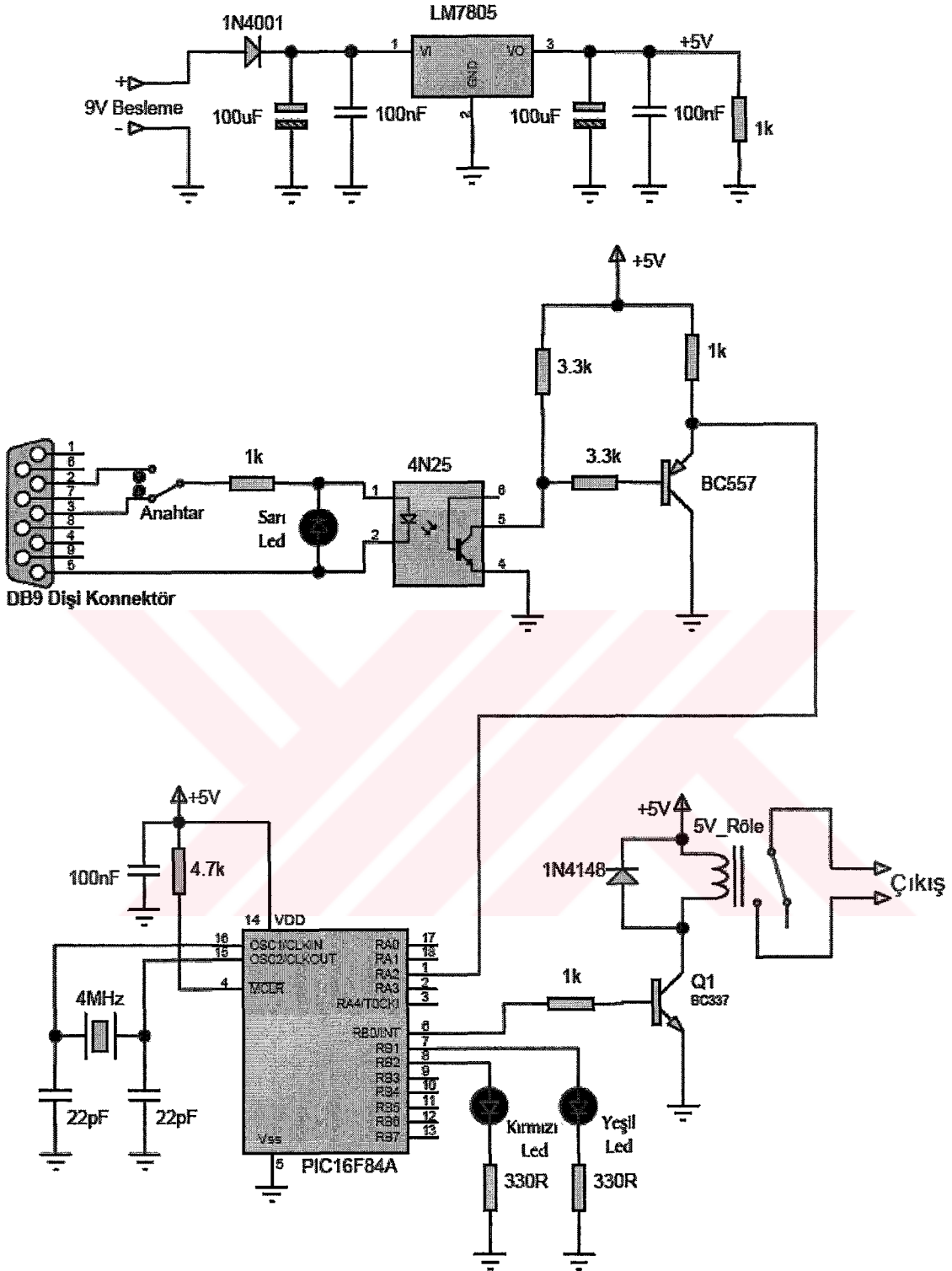
47	'Bilgisayarı Aç' komutunun gönderildiğinin anlaşıldığı şart bloğu kapatılmaktadır.
48	Oluşturulan for döngüsünün bloğu kapatılmaktadır.
49	main fonksiyonu için oluşturulan blok kapatılmaktadır.

Yazılan C programı derlendikten sonra oluşan hex uzantılı dosya EK-3' de verilmiştir. Bu dosyanın PIC 16F84A'ya yüklenmesi ile mikro denetleyicinin istenen şekilde çalışması sağlanmış olacaktır.

6.3.2. PIC 16F84A Mikro Denetleyicili Seri Port Kontrol Devresi

PIC 16F84A mikro denetleyicisinin programlanma işlemi tamamlandıktan sonra seri port kontrol devresi hazırlanmıştır. Tasarlanan elektronik devre Şekil 6.6' da gösterilmektedir. Devrenin doğru çalıştığı öncelikle Proteus programında benzetim yapılarak görülmüş, ardından devre bord üzerine kurularak gerekli testler yapılmıştır.





Şekil 6.6 PIC 16F84A Mikro Denetleyicili Seri Port Kontrol Devre Şeması

6.3.2.1. PIC 16F84A Mikro Denetleyicili Kontrol Devresinin Çalışma Mantığı

DB-9 dişi konnektörü üzerinden gelen sinyal 3. ve 5. pinleri üzerinden devreye gönderilmektedir. Kullanılacak olan seri port kablosunun Şekil 4.11' de gösterildiği gibi dişi veya erkek olma durumuna göre sinyalin geldiği 2. ve 3. pinlerin yerleri değişebileceği için buraya bir anahtar konularak sisteme istenirse 2. pin den gelen veri istenirse 3. pinden gelen veri aktarılmaktadır.

Verinin geldiğinin anlaşılabilmesi için buraya sarı bir led konulmuştur. Sarı ledin görevi devreye sinyalin gönderildiğinin anlaşılabilmesidir.

Seri porttan gönderilen sinyal 4N25 Optocoupler entegresine uygulanmıştır. Böylece seri port ile devrenin diğer kısımları fiziksel olarak birbirinden yalıtılmıştır. Bu da seri portun herhangi bir şekilde zarar görmesini önlemektedir. Port gerilim seviyesi negatif iken optocoupler içerisinde bulunan led ışık yaymayacağı için foto transistör kesimde olacak ve PIC'in RA2 bacağında lojik 1 seviyesi görülecektir. Port gerilimi seviyesi pozitif olduğunda ise optocoupler içerisindeki led ışık yayacağından foto transistör iletme geçecek ve PIC'in RA2 bacağında lojik 0 seviyesi görülecektir. Böylece seri porttan gönderilen sinyal, 0 ve 5V'dan oluşan lojik sinyale dönüştürüldükten sonra PIC'e uygulanmaktadır.

PIC'in kontrol ettiği 5V'luk rölenin normalde açık kontağı bir ikili klemense bağlıdır. Yeterli uzunlukta bir kablo ile bu klemens uçları hedef bilgisayarın ana kartı üzerinde bulunan 'power button' pinleri (Şekil 6.7) arasına bağlanmaktadır. Böylece, röleye enerji geldiği anda normalde açık kontak kapanacak ve hedef bilgisayarın açılması veya kapanması sağlanacaktır.

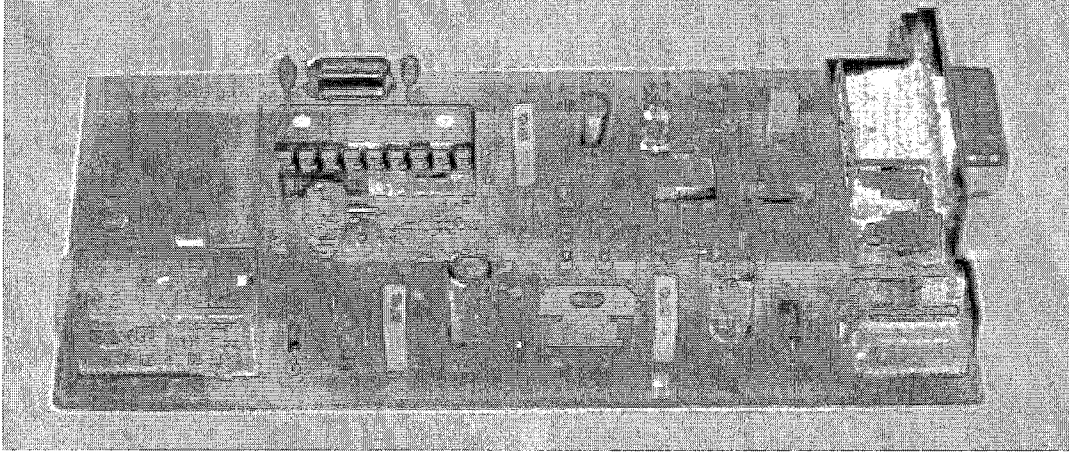


Şekil 6.7 Hedef Bilgisayarın Ana Kartı Üzerinde Bulunan 'Power Button' Pinleri

Devrenin beslemesi LM7805 gerilim regülatörü ile sağlanmıştır. Bu regülatör 9V'luk adaptörden gelen gerilimi 5V'a düşürerek PIC'in ve rölenin çalışmasını sağlamaktadır. Adaptör

soketinin ters bağlanması durumunda devrenin zarar görmemesi için gerilim regülâtörünün girişine 1N4001 diyotu bağlanmıştır.

Tasarlanan elektronik devreye ait baskı devre şeması uygun büyüklükteki bir bakır plakaya basılmıştır. Devrenin bakır plaka üzerindeki görünümü Şekil 6.8’de gösterilmektedir.



Şekil 6.8 Elektronik Devrenin Bakır Plaka Üzerindeki Görünümü

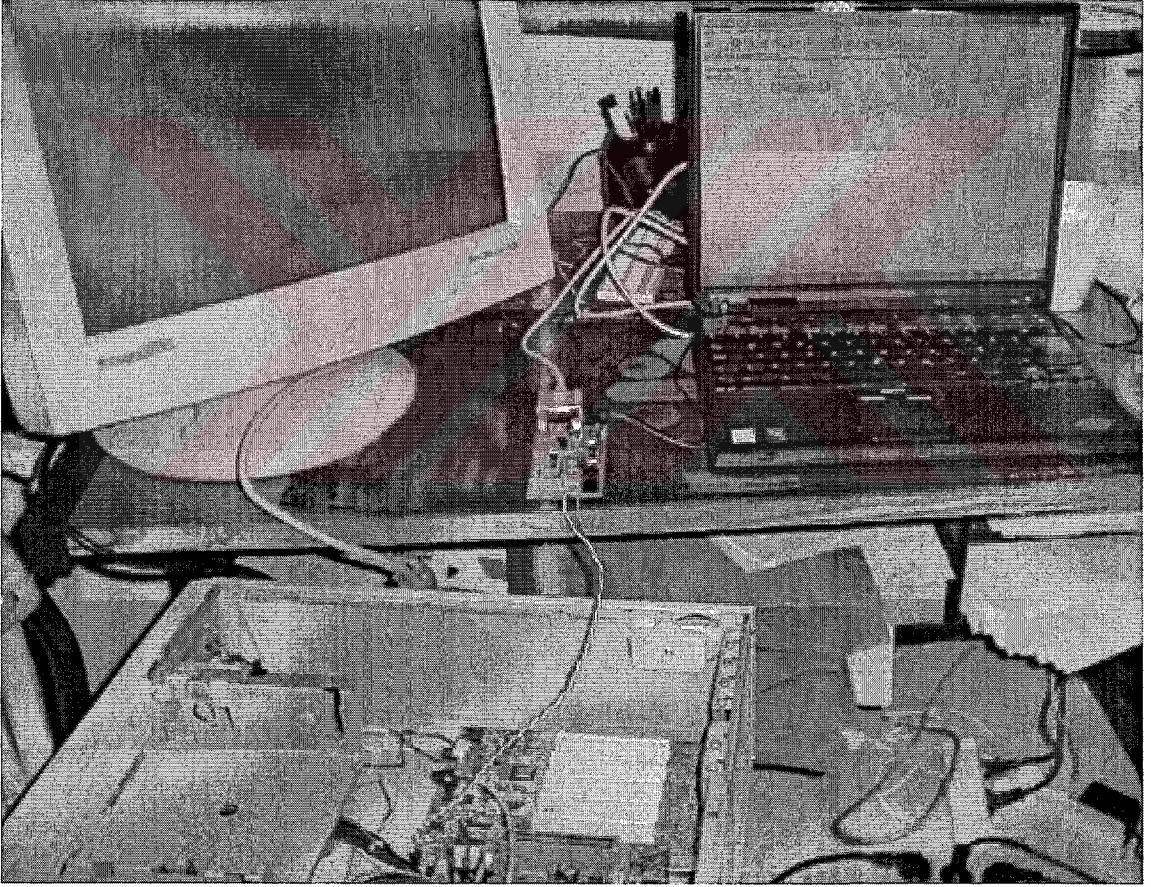
Devrenin yapımı için kullanılan malzemelerin listesi ise Tablo 6.4’ de gösterilmektedir.

Tablo 6.4 PIC 16F84A Mikro Denetleyicili Kontrol Devresi Malzeme Listesi

Malzeme Adı	Özelliği	Adet
Adaptör	9V	1
Seri Port Kablosu	Dişi-Erkek	1
Mikro Denetleyici	PIC16F84A	1
Soket	18 lik	1
Regülâtör	7805	1
Optocoupler	4N25	1
Anahtar	Basmalı	1
Direnç	1K	4
Direnç	4.7 K	1
Direnç	3.3K	2
Direnç	330R	2
Transistör	BC557	1
Transistör	BC337	1
Adaptör Jack		1
Klemens	Vidalı 2:1	1

Kristal	4 MHz	1
Kondansatör	100 uf	2
Kondansatör	100 nf	3
Kondansatör	22 pf	2
Diyot	1N4001	1
Diyot	1N4148	1
Led	Sarı, Yeşil, Kırmızı	3
Röle	5V	1
Bakır Plaket	32 cm ²	1

Hazırlanmış olan PIC 16F84A mikro denetleyicili Seri port kontrol devresinin çalışmaya hazır durumdaki görünümü Şekil 6.9’ da gösterilmektedir.



Şekil 6.9 Hazırlanan Uygulamanın Genel Görünümü

7. SONUÇLAR

Hazırlanmış olan tez çalışması yalnızca başka bir bilgisayarın güç sisteminin kontrol edilebilmesini gerçekleştirme ile sınırlı değildir. Bu uygulama ile hazırlanmış olan sistem 5 Volt gerilim verilerek çalıştırılabilinecek tüm sistemleri de kontrol edebilmektedir. Hazırlanmış olan sistem; gerekli gerilim düzenleme işlemleri yapıldıktan sonra herhangi bir elektronik sistemin güç sisteminin de kontrol edilebilmesine imkân sağlamaktadır. Böylece günümüzde yavaş yavaş günlük hayatımıza girmeye başlayan akıllı ev otomasyonları için bir yöntem oluşturulmuştur.

Bu şekilde hazırlanmış olan tez çalışması istenilirse akla gelebilecek birçok sisteme uyumlu hale getirilebilmektedir. Bu amaçla evde kimse yokken evinizdeki çiçekleri sulayabileceğiniz bir sulama kontrol devresini internet bağlantısı olan herhangi bir yerden çalıştırabilecek veya kapatabilecek bir yapıya dönüştürmek mümkün olabilmektedir. Evinizden çıkmadan önce uygun gerilim düzenlemesi yapılmış işyerinizde bulunan klima sistemini çalıştırabilmek ve işyerinize ulaştığınızda çalışmak istediğiniz ısıya hemen kavuşmak mümkün olmaktadır. Bunun gibi uygulamaları çoğaltmak mümkündür. Kullanıcıların hayal gücü ve sistemin iletişim kurabileceği bir web sunucu bilgisayar ile birçok projenin geliştirilebilmesi mümkündür.

KAYNAKLAR

- [1] Yarım, M.A. , Robot Control Over Internet Using Tcp/Ip Protocol, M.Sc Thesis , Graduate School of Natural and Applied Sciences of Dokuz Eylül University.
- [2] Karazeybek, A. , Computer Controlled Machines, M.Sc Thesis , Graduate School of Natural and Applied Sciences of Dokuz Eylül University.
- [3] Coşkun, O. , Güney, K. , Eren, A. , Telefon Hatları Üzerinden Bilgisayar Yardımıyla Elektronik Sistemlerin Kontrol Edilmesi, ELECO'2000: Elektrik-Elektronik-Bilgisayar Sempozyumu, Elektronik Bilgisayar, 2000, 313–317.
- [4] Wu C. , Jan R. , System Integration Of WAP And SMS For Home Network System , Department of computer and Information Science, Nationan Chiao Tung University , Computer Networks 42 (2003) 493-502
- [5] Baykal, N. , 2001, Bilgisayar Ağları, SAS Bilişim Yayınları, Ankara.
- [6] Kaplan, Y. , 2000, Veri Haberleşmesi Temelleri, Papatya Yayınları, İstanbul.
- [7] <http://www.ietf.org>
- [8] Yıldırımoglu, M., 2000, TCP/IP İnternet'in Evrensel Dili , Pusula Yayıncılık.
- [9] Çölkesen, R. , Örencik B. , 1999, Bilgisayar Haberleşmesi ve Ağ Teknolojileri, İstanbul,
- [10] Axelson, J. , Serial Port Complete, 2000, USA,
- [11] <http://www.mikroelektronika.co.yu>
- [12] Altınbaşak, O. , 2000, Mikrodenetleyiciler ve PIC Programlama, Altaş Yayınları, İstanbul,
- [13] <http://www.antrak.org.tr>
- [14] <http://www.microchip.com>
- [15] <http://msdn.microsoft.com>

ÖZGEÇMİŞ

Fatih ERTAM

fatih@ertam.com

Tlf: 424 2370082

Fırat Üniversitesi Enformatik Bölümü

ELAZIĞ

1978 yılında Elazığ da doğdu. İlk, orta ve lise öğrenimini Elazığ'da tamamladıktan sonra 1996 yılında girdiği Fırat Üniversitesi Teknik Eğitim Fakültesi Elektronik-Bilgisayar Eğitimi Bölümü Bilgisayar Öğretmenliği Bölümünden 2000 yılında mezun oldu. Şubat 2003 de Fırat Üniversitesi Fen Bilimleri Enstitüsü Elektronik Bilgisayar Eğitimi Ana Bilim Dalı Bilgisayar Sistemleri Bilim Dalı'nda yüksek lisans eğitimine başladı. Aralık 2000' den beri Fırat Üniversitesi Enformatik Bölümünde Öğretim elemanı görevini sürdürmektedir. Yabancı dili İngilizcedir.

EKLER

EK – 1 ASP İle Hazırlanan Form Dosyasının Kodları

```
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=iso-8859-9">
<META NAME="description" CONTENT="uzaktan kontrol">
<META NAME="author" CONTENT="Fatih Ertam">
<title>Uzaktan Kontrol</title>
</head>
<body>
<form name="form1" method="post" action="portislem.asp">
<table width="450" border="0" cellpadding="0" cellspacing="0">
<tr><td width="183">Port seçimini Yapınız </td>
<td width="267">
<select name="port">
<option value="1" selected>COM1</option>
<option value="2">COM2</option>
</select></td>
</tr><tr>
<td>İşlemi Seçiniz </td><td>
<select name="islem">
<option value="A" >Bilgisayarı Aç</option>
<option value="B" >Bilgisayarı Kapat</option>
<option value="BA">Yeniden Başlat</option>
</select>
</td>
</tr><tr>
<td height="40"></td>
<td><input type="submit" name="Submit" value="İşlemi Gerçekleştir"></td>
</tr>
</table>
</form>
</body>
</html>
```

EK – 2 PHP İle Hazırlanan Form Dosyasının Kodları

```
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=iso-8859-9">
<META NAME="description" CONTENT="uzaktan kontrol ">
<META NAME="author" CONTENT="Fatih Ertam">
<title>Uzaktan Kontrol</title>
</head>
<body>
<form name="form1" method="post" action="portislem.php">
<table width="450" border="0" cellpadding="0" cellspacing="0">
<tr>
<td width="183">Port Seçimini Yapınız </td>
<td width="267">
<select name="port">
<option value="COM1" selected>COM1</option>
<option value="COM2">COM2</option>
</select></td>
</tr><tr>
<td>İşlemi Seçiniz </td>
<td>
<select name="islem">
<option value="A" selected>Bilgisayarı Aç</option>
<option value="B">Bilgisayarı Kapat</option>
<option value="BA">Yeniden Başlat</option>
</select></td>
</tr><tr>
<td height="40">&nbsp;</td>
<td><input type="submit" name="Submit" value="İşlemi Gerçekleştir"></td>
</tr>
</table>
</form>
</body>
</html>
```

EK – 3 PIC C İle Hazırlanan Programın HEX Kodları

:0C000000830100308A0004288301D62B05
:1006F8008312A5238C00BD230C08080083128F01E8
:10070800FA30831298238F0A04300F020318080066
:10071800842B83128F01FA30831298238F0A283092
:100728000F02031808008F2B8312900004309100E9
:1007380053309200920B9E2B910B9C2B900B9A2B73
:10074800080083120519A62B10309100910BAA2BD3
:100758000519A52B08308F0090011E309100910BD0
:10076800B32B031005190314900C8F0BB12B100831
:10077800080083128E000B308D0007308316850029
:10078800831285110C308D001E308C008C0BCA2B07
:100798000E1885150E18D12B851103148E0C8D0B90
:1007A800C82B080083168601043085008312850152
:1007B80086018D018E018312A5238C00413A031D09
:1007C800F02B8D08031DF02B8614061106148223C6
:1007D80006108D018D0A8E010C08423A031DDF2B8D
:1007E8008E08031DDF2B8610061506148D230610B0
:0807F8008D018E018E0ADF2B3A
:00000001FF