

**T.C.**  
**NİĞDE ÜNİVERSİTESİ**  
**MÜHENDİSLİK MİMARLIK FAKÜLTESİ**  
**ELEKTRİK ELEKTRONİK MÜHENDİSLİĞİ BÖLÜMÜ**

**BITİRME ÖDEVİ**

**İNTERNET ÜZERİNDEN EV OTOMASYONUNUN**  
**GERÇEKLEŞTİRİLMESİ**

**DANIŞMAN**  
**Yrd. Doç. Dr. Murat UZAM**

**HAZIRLAYAN**  
**İrfan AKTAN**

**NİĞDE 2003**

**TÜRKİYE CUMHURİYETİ**  
**NİĞDE ÜNİVERSİTESİ**  
**MÜHENDİSLİK MİMARLIK FAKÜLTESİ**  
**ELEKTRİK ELEKTRONİK MÜHENDİSLİĞİ BÖLÜMÜ**

**BİTİRME ÖDEVİ DERSİ SINAV TUTANAĞI**

**ÖĞRENCİNİN :**

**Adı ve Soyadı** :.....  
**Sınıf ve Numarası** :.....  
**Bitirme Ödevinin Adı** :.....  
:.....

**Bitirme Ödevini Yöneten**

**Öğretim Üyesinin Adı ve Soyadı** :.....

**BİTİRME ÖDEVİ SINAV JÜRİSİ**

**Başkan** :.....  
**Üye** :.....  
**Üye** :.....  
**Üye** :.....

**Bu çalışma ...../...../..... tarihinde yapılan sınav sonucunda jürimiz tarafından oy birliği/çokluğu ile başarılı/başarısız bulunmuştur.**

**UYGUNDUR**

...../...../2002

**Doç. Dr. Saadetdin HERDEM**

**Bölüm Başkanı**

## ÖZET

Bu çalışma çok yaygın olarak kullanılan internetin, bir elektronik ortamın (bir devre, ev yada fabrika) kontrolünde nasıl kullanılabileceği ve bunun getireceği avantajları incelemeyi amaçlamıştır. Basitce, internete bağlı herhangi bir bilgisayarla, kurduğumuz siteye bırakılan kontrol bilgisinin kaydedilmesi ve kontrol(röle) devresinin bağlı bulunduğu bilgisayarca bu bilginin alınıp, devreye aktarılması şeklinde çalışan sistemimizde kontrol devresinin bağlı bulunduğu bilgisayar da doğal olarak internete bağlı olmalıdır.

Bu işlem temelde paralel port, ActiveServerPage dosyaları, internet ve ilerleyen bölümlerde açık devre şemasını vereceğimiz kontrol devresiyle gerçekleştirilmektedir.

İnternet kullanıcısı tarafından (belli bir şifre girerek) verilen emirler, ASP dosyalar vasıtasıyla internet üzerindeki bir hafıza bölgesine(bize bu bölgeyi sağlayabilen birçok internet sayfası var) kaydedilmektedir. Kaydedilen bu veriler, kontrol edilecek devrenin bağlı bulunduğu bilgisayar tarafından Visual Basic programı yardımıyla yazdığımız bir programca belli süreler zarfında kontrol edilip işlenmektedir. Ve verilen emirler yine Visual Basic ile yazdığımız bir program vasıtasıyla paralel porta aktarılmaktadır. Geriye sadece bilgisayarın paralel portundan aldığımız veriyi, kullanabileceğimiz bir seviyeye yükselterek istediğimiz uygulamada kullanmak kalmaktadır. Bunu da az önce bahsettiğimiz devre gerçekleştirmektedir.

## TEŞEKKÜR

Öncelikle bu çalışma boyunca bana maddi ve manevi destek sağlayan aileme, saygıdeğer danışman hocam Yrd. Doç. Dr. Murat UZAM'a, sağladıkları destekle bu günlere gelmemi sağlayan tüm bölüm hocalarıma, derin programcılık bilgisiyle her ihtiyacım olduğunda yanımda bulunan Cafer Gazi KOÇ'a, bana destek olan tüm dostlarıma ve verdikleri vergilerle bizlerin üniversite okumamıza katkı sağlayan ülkemizin insanlarına teşekkür ederim.

## İÇİNDEKİLER

|  |      |
|--|------|
| ÖZET.....  | iii  |
| TEŞEKKÜR.....  | v    |
| İÇİNDEKİLER.....   | vi   |
| ŞEKİLLER DİZİNİ.....   | vii  |
| TABLolar DİZİNİ.....   | viii |
| BÖLÜM 1 GİRİŞ.....   | 1    |
| BÖLÜM 2 PARALEL PORT.....  | 3    |
| 2.1. Paralel Veri İletimi Nedir?.....  | 5    |
| 2.2 Paralel Veri İletimi İle Seri Veri İletiminin Karşılaştırılması.....       | 6    |
| 2.2.1. Paralel veri iletiminin avantajları.....                                | 6    |
| 2.2.2. Paralel veri iletiminin dezavantajları.....                             | 6    |
| 2.3. Portlara Erişim.....  | 8    |
| 2.3.1. Sinyaller.....  | 9    |
| 2.4. Paralel Arabirim Ünitelerinde Ortaya Çıkabilecek Problemlerin Çözümü..... | 12   |
| 2.4.1. Paralel arabirimlerle çalışma.....                                      | 12   |
| 2.4.2. Crosstalk Problemi.....   | 13   |
| BÖLÜM 3 İNTERNET.....  | 15   |
| 3.1. İnternette Güvenlik.....  | 15   |
| 3.2. Web Sayfası Tasarımı Ve Teknikleri.....                                   | 16   |
| 3.3. HTML.....   | 18   |
| 3.4. ASP.....  | 21   |
| BÖLÜM 4 PROGRAMLAMA.....   | 24   |
| BÖLÜM 5 HARDWARE (DONANIM).....  | 29   |
| 5.1. Devrenin Açık Şeması.....   | 29   |
| 5.2. Devrenin Çalışması.....   | 29   |
| BÖLÜM 6 SONUÇ.....   | 31   |
| EKLER.....   | 32   |
| KAYNAKLAR.....   | 40   |

## ŞEKİLLER DİZİNİ

|   |    |
|---|----|
| Şekil 1.1. Sistemin çalışma diyagramı.....        | 8  |
| Şekil 2.1. Paralel port.....                      | 10 |
| Şekil 2.2. Bacak Tanımları.....                   | 11 |
| Şekil 2.3. Port tanımları.....                    | 11 |
| Şekil 2.4. Paralel veri iletimi.....              | 12 |
| Şekil 2.5. Paralel port bağlantısı IBM.....       | 14 |
| Şekil 2.6. HandShaking blok diyagram.....         | 15 |
| Şekil 3.1. Şifre sorma sayfası.....               | 24 |
| Şekil 3.2.Emirlerin gönderildiği web sayfası..... | 25 |
| Şekil 3.3. ASP'nin çalışması.....                 | 29 |
| Şekil 5.1.Devrenin açık şeması.....               | 36 |

## TABLolar DİZİNİ

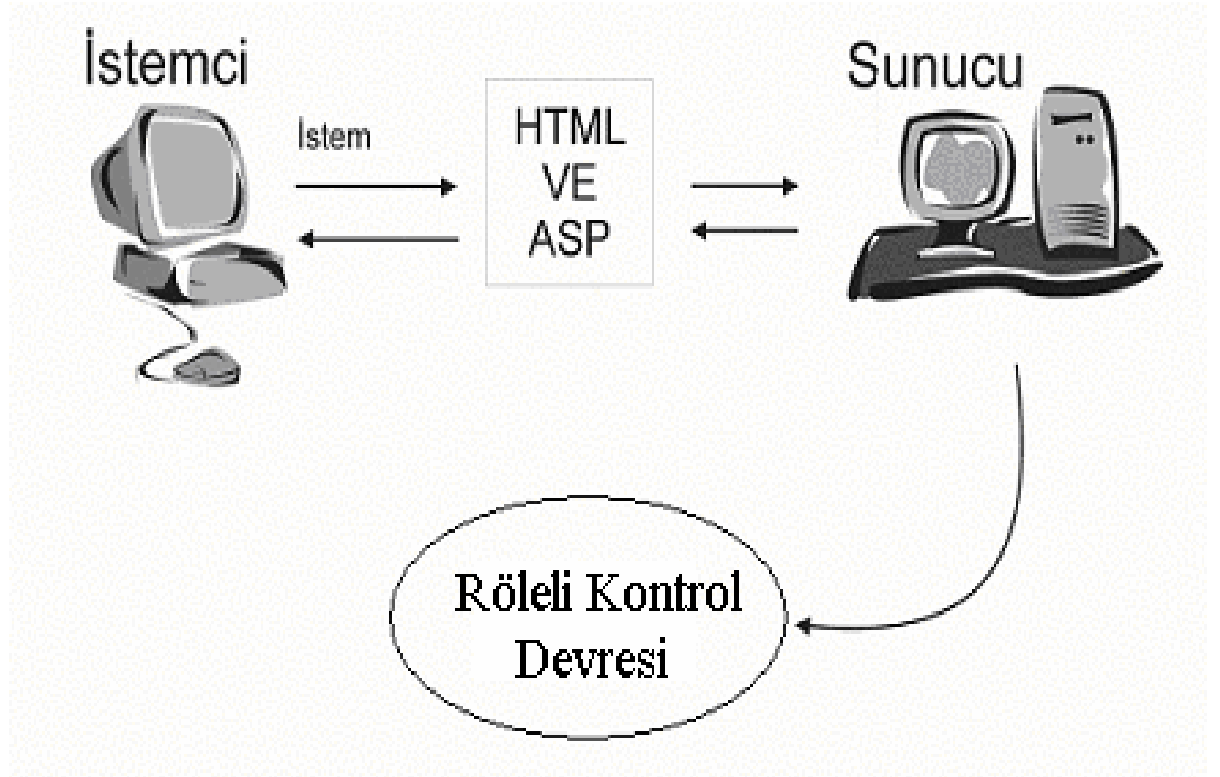
|   |    |
|---|----|
| Tablo 2.1. Port adresleri .....   | 11 |
| Tablo 2.2. Bacak numarasına göre düzenlenmiş paralel port sinyalleri..... | 17 |
| Tablo 2.3. Yazmaca göre düzenlenmiş paralel port bitleri.....             | 19 |
| Tablo 2.4. Centronics paralel port.....                                   | 21 |
| Tablo 4.1. Dosyaların isimleri ve işlevleri.....                          | 36 |

## BÖLÜM1

### GİRİŞ

Sistem temel olarak 8 adet rölenin, internet üzerinden kontrol edilmesini sağlamaktadır. Bu rölelere istenen aygıtlar bağlanarak, istenen donanım rahatlıkla kontrol edilebilir. Sistem paralel port üzerinden çalışmaktadır. İnternet üzerinden Visual Basic'te yazılan program sayesinde alınan veri paralel porta gönderilir ve buradan da devremiz tarafından alınarak röleler kontrol edilir. Ayrıca internet üzerinde yayımlanacak sayfalarda bulunmaktadır. Bu da dünyanın herhangi bir yerinden, şifreyi bilen bir kullanıcının sistem veritabanına emrini bırakabilmesini sağlar. Ayrıca bu veritabanındaki emirleri gösteren bir sayfada mevcuttur. İşte bu sayfadan gelen veri programımız tarafından işlenerek, paralel porta gönderilen veridir.

Tasarlanan sistem temel olarak Şekil 1.1' de görülen yapıya sahiptir.



### Şekil 1.1. Sistemin çalışma diyagramı

Şekilden de görüleceği gibi alınan emirler html ve asp yardımı ile oluşturulan sayfaya verilmekte, buraya kaydedilmekte, ve belli aralıklarla değişimi izleyen sunucu bilgisayar tarafından alınmakta ve işlenmektedir. Devrenin bağlı olduğu bilgisayara sunucu bilgisayar dememizin nedeni aldığı emirleri devreye sunmasıdır.

Ayrıca sistemin kurulabilmesi için paralel port çıkışına ve internet bağlantısına sahip bir bilgisayar, internet bölgesinde yüklü olan ASP ve HTML dosyalar, emirlerin gönderilebileceği bir web sayfası, ve Bölüm 5' te açık şeması verilen devreyi oluşturabilmek için gerekli elemanlara ihtiyaç vardır.

Bölüm 2'de paralel port'un yapısı, çalışması, özellikleri, kullanım alanları, paralel portların yapısı, çalışma prensibi, dış görünüşü, bilgisayara bağlantı biçimleri, paralel veri iletişiminin bize sağlandığı katkılar, seri veri iletişiminden üstünlükleri,

Bölüm 3'de internet ile ilgili genel bilgi, internette güvenlik, web sayfası tasarımı,

Bölüm 4'de internet üzerinden veri alışverişi için gerekenler,

Bölüm 5'de çalışmanın programlama bölümü ile ilgili bilgi,

Bölüm 6'de devrenin şeması ve açıklaması,

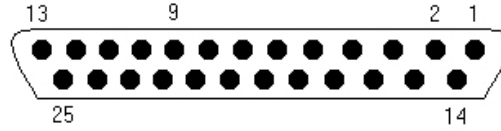
Bölüm 7'de sonuç, bu uygulamanın bize sağladığı faydalar ve kullanım sahası

Hakkında bilgi verilmiştir.

## BÖLÜM 2

### PARALEL PORT

Çevre birimlerinin kontrolü ile ilgili gerçekleştirilecek projeler için, PC'nin paralel portu (printer portu) hala çok ucuz ve güçlü bir platform özelliğindedir. Printer portu bize sekiz TTL çıkış, beş giriş ve dört tane iki yönlü uç sağlar.



Şekil 2.1. Paralel port

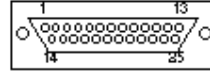
Her bir printer portu üç adresten oluşur: veri, durum ve kontrol portu. Bu adresler sıralı şekildedir. Yani, eğer veri portu adresi 0x0378'de ise ilgili durum portu 0x0379'da ve ilgili kontrol portu da 0x037A'dadır.

Tablo 2.1. Port adresleri

| Printer | Veri Portu | Durum Portu | Kontrol Portu |
|---------|------------|-------------|---------------|
| LPT1    | 0x03BC     | 0x03BD      | 0x03BE        |
| LPT2    | 0x0378     | 0x0379      | 0x037A        |
| LPT3    | 0x0278     | 0x0279      | 0x027A        |

Lütfen Şekil 2.1. Bacak tanımları ve Şekil 2.2. Port tanımları'nı inceleyiniz. Bu iki şekil 25 bacaklı konektörün bacak tanımları ile üç portun bit tanımlarını gösterir.

Şekil 2.2'de paralel portun bacak tanımları verilmiştir. 18-23 arası ground(toprak)'tır.



View is looking at  
Connector side of  
DB-25 Male Connector.

**Pin Description**

| Pin | Description                            |           |
|-----|--|-----------|
| 1   | $\overline{\text{Strobe}}$             | PC Output |
| 2   | Data 0                                 | PC Output |
| 3   | Data 1                                 | PC Output |
| 4   | Data 2                                 | PC Output |
| 5   | Data 3                                 | PC Output |
| 6   | Data 4                                 | PC Output |
| 7   | Data 5                                 | PC Output |
| 8   | Data 6                                 | PC Output |
| 9   | Data 7                                 | PC Output |
| 10  | $\overline{\text{ACK}}$                | PC Input  |
| 11  | Busy                                   | PC Input  |
| 12  | Paper Empty                            | PC Input  |
| 13  | Select                                 | PC Input  |
| 14  | $\overline{\text{Auto Feed}}$          | PC Output |
| 15  | $\overline{\text{Error}}$              | PC Input  |
| 16  | $\overline{\text{Initialize Printer}}$ | PC Output |
| 17  | $\overline{\text{Select Input}}$       | PC Output |

Pin Assignments

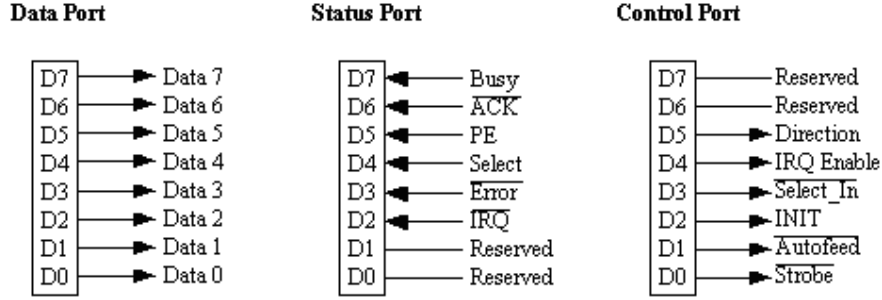
Note: 8 Data Outputs  
4 Misc Other Outputs

5 Data Inputs

Note: Pins 18-25 are  
Ground

Şekil 2.2. Bacak Tanımları

Şekil 2.3'de paralel portun giriş ve çıkış sinyalleri verilmiştir. Görüldüğü gibi 8 bit datalar için kullanılmıştır. Ayrıca 6 bit giriş sinyali, 6 bit de çıkış sinyali olarak göze çarpmaktadır.



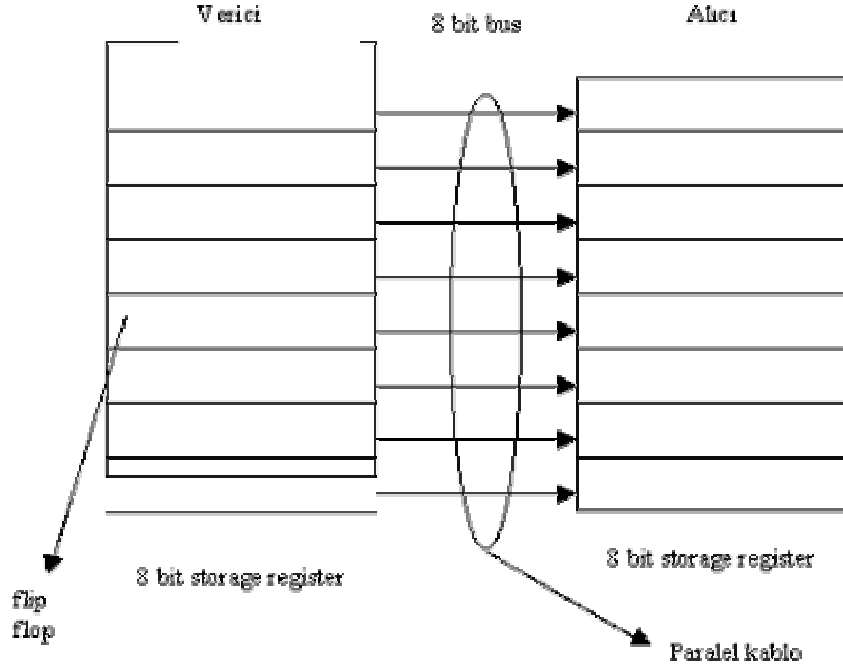
Şekil 2.3. Port tanımları

Veri portunda sekiz çıkış (Data 7(msb) – Data 0) ve Kontrol portunda dört ek çıkış (/SELECT\_IN, INIT, /AUTO FEED ve /STROBE) bulunduğuna dikkat ediniz.

Veri portundaki tüm çıkışlar “true” lojik esasına göre çalışır. Yani, bir bite lojik bir yazmak ilgili çıkışın “high” olmasına neden olur. Fakat Kontrol portundaki /SELECT\_IN, /AUTOFEED ve /STROBE çıkışları ters lojik esasına göre çalışır. Yani, bir bite lojik bir göndermek ilgili çıkışın lojik sıfır olmasına neden olur. Bu, printer portunun kullanımına biraz karmaşıklık getirir. Bununla beraber, çıkıştan önce basitçe bu bitlerin XOR işlemiyle tersini almak bizi çözüme götürür.

## 2.1. Paralel veri iletimi nedir?

Seri veri iletimi yöntemi gibi paralel veri iletimi de veri iletiminde kullanılan temel yöntemlerden birisidir. Digital olarak kodlanmış bilginin tüm bitleri aynı anda transfer ediliyorsa buna “paralel veri iletimi” denir. Bu olay Şekil 2.4’ten daha net olarak anlaşılabilir.



Şekil 2.4. Paralel veri iletimi

Transfer edilecek binary bilgi, içerisinde herbir bit için bir flip flop bulunan bir yazmaç'a yüklenir. Herbir verici flip-flop çıkışı uygun kablo bağlantıları ile alıcı devrenin flip-flop'una bağlanır. Alıcı devresi de genellikle bir yazmaç'dır. Görüldüğü gibi paralel veri iletiminde, iletilecek bilginin her biti için bir kablo bağlantısı vardır. Bu ise, çok kablolu (multi-wire) bir hattın kullanımını gerektirir. Binary bilgi taşıyan çoklu paralel hatlara "data bus" denir.

## 2.2. Paralel Veri İletimi İle Seri Veri İletiminin Karşılaştırılması

### 2.2.1. Paralel veri iletiminin avantajları

Seri veri iletiminde, bir kerede bir karakterin sadece bir biti iletilir. Alıcı makine doğru haberleşme için karakter uzunluğunu, start-stop bitlerini ve iletim hızını bilmek zorundadır. Paralel veri iletiminde, bir karakterin tüm bitleri aynı anda iletildiği için start-stop bitlerine ihtiyaç yoktur. Dolayısıyla doğruluğu daha yüksektir.

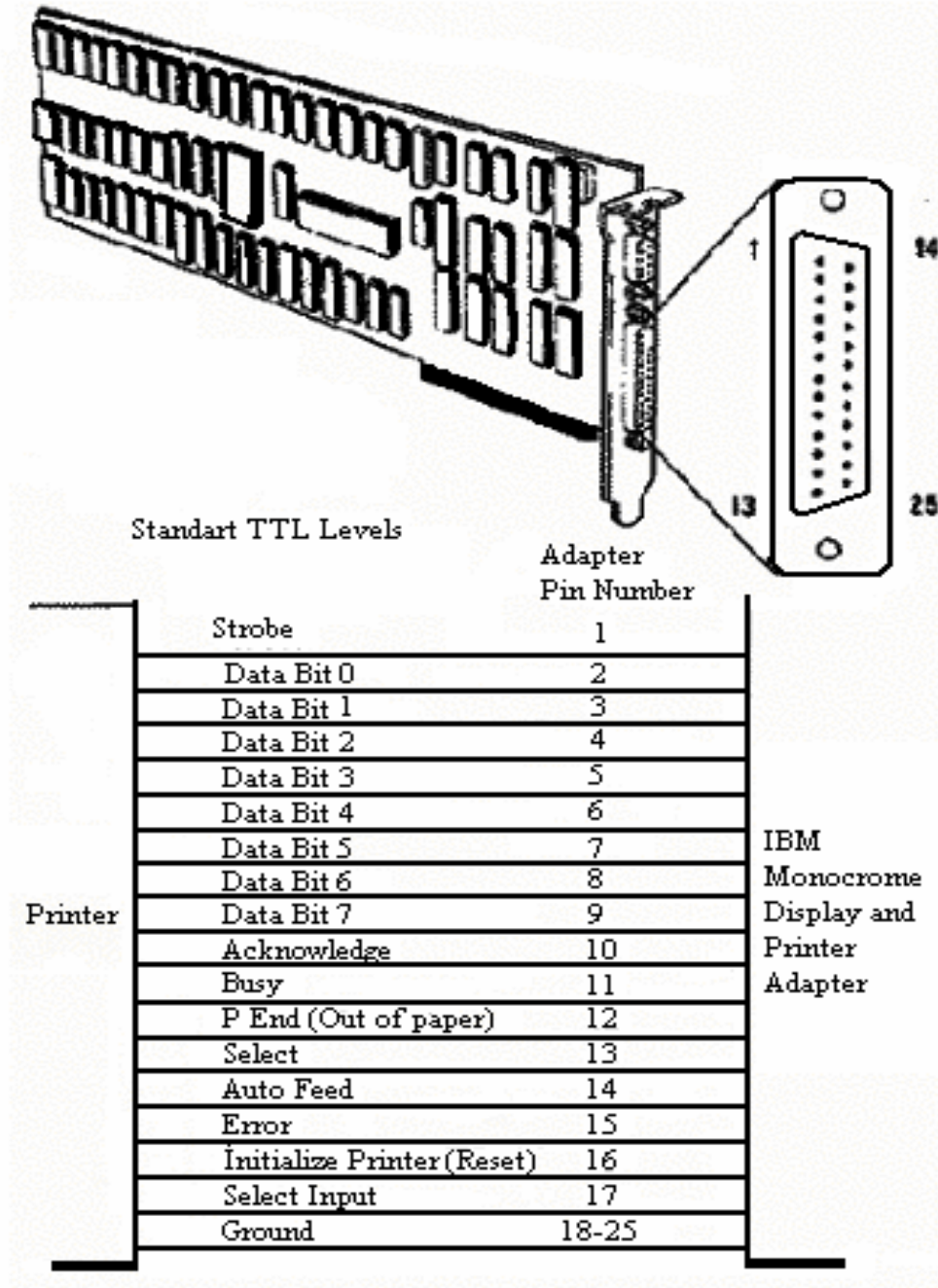
Paralel veri iletimi, bilginin tüm bitlerinin aynı anda iletimi sebebiyle çok hızlıdır. Paralel veri iletiminde, bir kerede bir karakter iletildiği için bilgi iletim hızı "cps" (Character Per Second) olarak bilinir. Sıradan bir paralel portun veri iletim hızı yaklaşık 100.000 cps'dir. Benzer şekilde, sıradan bir seri portun veri iletim hızı yaklaşık 9600 bps'dir. Pek çok paralel port uygulamasında bir karakter yaklaşık olarak birkaç mikro saniyede (10 mikrosaniye) alıcı devreye iletilebilir.

Örnek olarak, 8 bitlik bir ASCII kodunun (1 karakter) paralel iletimini seri iletim ile karşılaştıralım. ASCII kodundaki herbir bitin transferi için 10 mikrosaniye geçtiği düşünülürse, seri haberleşmede 8 bitlik ASCII kodu için 80 mikrosaniye gerekir. Paralel haberleşmede ise bu iletim sadece 10 mikrosaniye'de yapılabilir.

### **2.2.2 Paralel veri iletiminin dezavantajları**

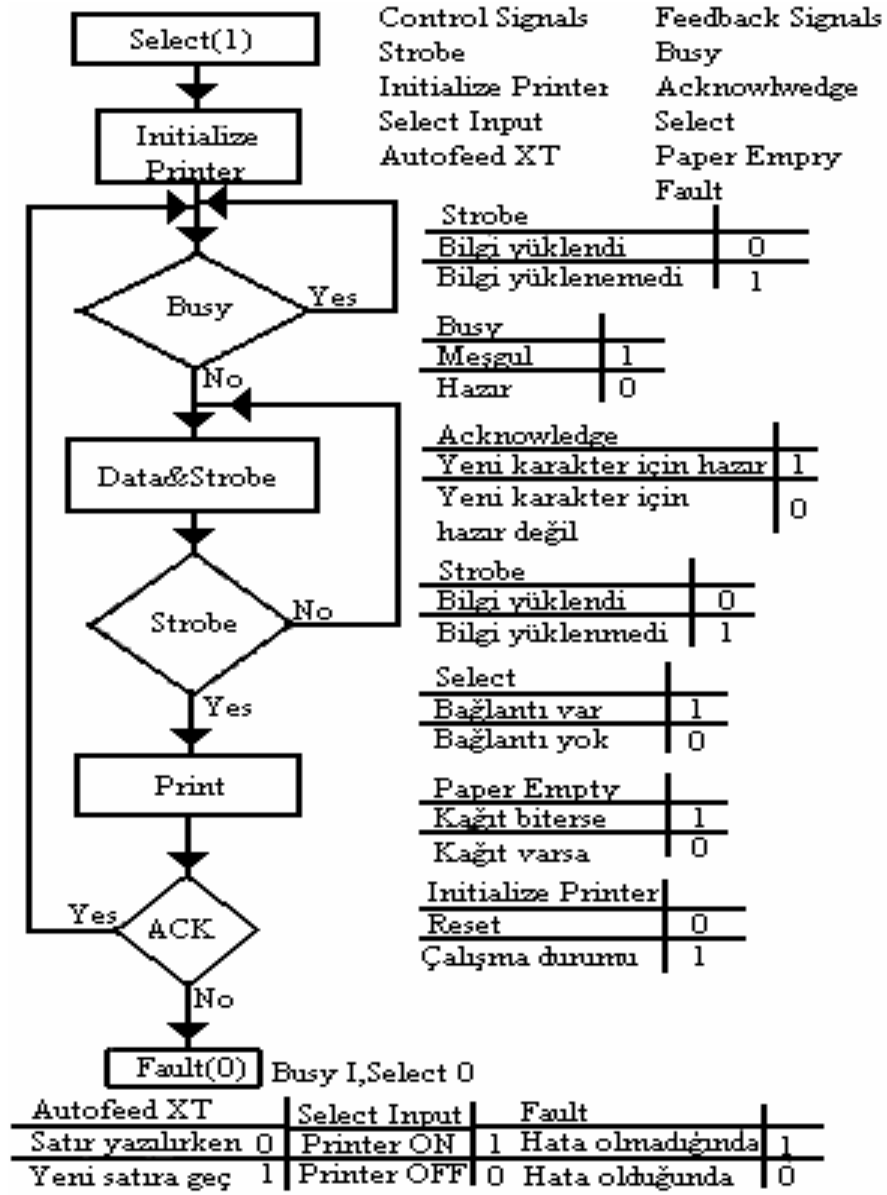
Veri iletiminde kullanılan portlardaki kablolar pahalıdır. Telefon ve seri iletişim bağlantılarında genellikle ikili hatlar kullanılır (örneğin mouse). Tipik paralel port bağlantıları için genellikle 25'li hatlar kullanılır. Ancak bunların sadece 8 tanesi data hattıdır. Geriye kalan diğer hatlardan kontrol sinyalleri iletilir. Çok kablolu hatları pahalı olması, paralel iletişiminin kısa mesafelerde tercih edilmesinin sebeplerinden biridir. Bilgisayarlarda mikroişlemci ile harddisk, printer, scanner vb. elemanlar arasındaki kısa mesafelerde paralel iletişim kullanılır. Örnek olarak Centronix firmasının ürettiği 36 pinli paralel porttan yapılan bir printer bağlantısında, paralel kablunun uzunluğunun 9 feet'den fazla olması önerilmez.

Şekil 2.5'de IBM firmasının ürettiği paralel portun görünüşü ve bacaklara göre kontrol ve data sinyalleri verilmiştir.



Şekil 2.5. Paralel port bağlantısı IBM

Şekil 2.6'da kontrol sinyallerine göre akış diyagramının printerlar için özel bir şekli olan Handshaking blok diyagramı verilmiştir.



Şekil 2.6. HandShaking blok diyagram

### 2.3. Portlara Erişim

Windows, DOS ve Visual Basic (VB), paralel portlara yazma ve okumada çeşitli yollar sunarlar. Burada en dolaysız yol port yazmaçlarına yazmak ya da okumaktır. Hemen hemen tüm programlama dillerinde bu yetenek bulunur en azından programcının

eklemesine imkan verirler. VB (Visual Basic)'in, Printer nesnesi, PrintForm metodu ve Open LPTx gibi, başka seçenekleri de vardır. Windows, LPT portlarına erişmek için API çağrılarını kullanır. 16-bit programlar, LPT erişiminde BIOS ve DOS yazılım kesmelerini (interrupts) kullanabilirler. Bu bölümde paralel port sinyalleri ve hazırlanan programlarla bunlara erişim yolları anlatılacaktır.

### 2.3.1. Sinyaller

Tablo 2-2 paralel port konnektöründe bulunan 25 yuvadan her birinin işlevlerini göstermektedir. Tablo 2.2 , aynı zamanda, sinyaller hakkındaki ilave bilgileri ve karşılık gelen yazmaç bitlerini de içermektedir. Tablo 2.3, yazmaç tarafından düzenlenen bilgiyi sergilemektedir. Tablo 2.3, bacak

Tablo 2.2 . Bacak numarasına göre düzenlenmiş paralel port sinyalleri

| Pin: D-sub | Signal   | Function                              | Source  | Register |      | Inverted at connector? | Pin: Centronics |
|------------|----------|---------------------------------------|---------|----------|------|------------------------|-----------------|
|            |          |                                       |         | Name     | Bit# |                        |                 |
| 1          | nStrobe  | StrobeDO-D7                           | PC1     | Control  | 0    | Y                      | 1               |
| 2          | DO       | Data Bit 0                            | PC2     | Data     | 0    | N                      | 2               |
| 3          | D1       | Data Bit 1                            | PC2     | Data     | 1    | N                      | 3               |
| 4          | D2       | Data Bit 2                            | PC2     | Data     | 2    | N                      | 4               |
| 5          | D3       | Data Bit 3                            | PC2     | Data     | 3    | N                      | 5               |
| 6          | D4       | Data Bit 4                            | PC2     | Data     | 4    | N                      | 6               |
| 7          | D5       | Data Bit 5                            | PC2     | Data     | 5    | N                      | 7               |
| 8          | D6       | Data Bit 6                            | PC2     | Data     | 6    | N                      | 8               |
| 9          | D7       | Data Bit 7                            | PC2     | Data     | 7    | N                      | 9               |
| 10         | nAck     | Acknowledge(ma<br>ytrigger interrupt) | Printer | Status   | 6    | N                      | 10              |
| 11         | Busy     | Printer busy                          | Printer | Status   | 7    | Y                      | 11              |
| 12         | PaperEnd | Paper end.<br>empty(outof             | Printer | Status   | 5    | N                      | 12              |

|    |                    |   |         |         |   |   |          |
|----|--------------------|---|---------|---------|---|---|----------|
|    |                    | paper)  |         |         |   |   |          |
| 13 | Select             | Printerselected<br>(on line)                            | Printer | Status  | 4 | N | 13       |
| 14 | nAutoLF            | Generateautomati<br>clinefeedsafter<br>carriage returns | PC1     | Control | 1 | Y | 14       |
| 15 | nError<br>(nFault) | Error   | Printer | Status  | 3 | N | 32       |
| 16 | nltl               | Initialize printer<br>(Reset)                           | PC1     | Control | 2 | N | 31       |
| 17 | nSelectIn          | Select printer<br>(Place on line)                       | PC1     | Control | 3 | Y | 36       |
| 18 | Gnd                | Ground return for<br>nStrobe, DO                        |         |         |   |   | 19.20    |
| 19 | Gnd                | Ground return for Di, D2                                |         |         |   |   | 21.22    |
| 20 | Gnd                | Ground return for<br>D3, D4                             |         |         |   |   | 23,24    |
| 21 | Gnd                | Ground return for<br>D5. D6                             |         |         |   |   | 25.26    |
| 22 | Gnd                | Ground return for<br>D7, nAck                           |         |         |   |   | 27.28    |
| 23 | Gnd                | Ground return for<br>nSelectIn                          |         |         |   |   | 33       |
| 24 | Gnd                | Ground return for<br>Busy                               |         |         |   |   | 29       |
| 25 | Gnd                | Ground return for<br>until                              |         |         |   |   | 30       |
|    | Chassis            | Chassis ground  |         |         |   |   | 17       |
|    | NC                 | No connection   |         |         |   |   | 15,18,34 |
|    | NC                 | Signal ground   |         |         |   |   | 16       |
|    | NC                 | +5V   | Printer |         |   |   | 35       |

numarasından ziyade, konnektördeki görünmeyen yazmaç bitlerini içermektedir. Sinyal isimleri ve işlevleri, nokta-vuruşlu yazıcıların ilk imalatçılarından Centronics Data Computer Comporation'ın belirlediği şekildedir. Bu firma artık yazıcı üretmiyor, ancak arabirimleri hala bulunmaktadır. Tablolardaki sinyal isimleri orijinal IBM PC'deki paralel port için kullanılanlardır.

Tablo 2.3. Yazmaca göre düzenlenmiş paralel port bitleri

| Data Register (Base Address)  |               |                    |            |                           |                 |
|---|---------------|--------------------|------------|---------------------------|-----------------|
| Bit   | Pin:<br>D-sub | Signal<br>Name     | Source     | Inverted at<br>connector? | Pin: Centronics |
| 0   | 2             | Data bit 0         | PC         | no                        | 2               |
| 1   | 3             | Data bit 1         | PC         | no                        | 3               |
| 2   | 4             | Data bit 2         | PC         | no                        | 4               |
| 3   | 5             | Data bit 3         | PC         | no                        | 5               |
| 4   | 6             | Data bit 4         | PC         | no                        | 6               |
| 5   | 7             | Data bit 5         | PC         | no                        | 7               |
| 6   | 8             | Data bit 6         | PC         | no                        | 8               |
| 7   | 9             | Data bit 7         | PC         | no                        | 9               |
| Some Data ports are bidirectional. (See Control register, bit 5 below.) |               |                    |            |                           |                 |
| Status Register (Base Address +1)                                       |               |                    |            |                           |                 |
| Bit   | Pin:<br>D-sub | Signal<br>Name     | Source     | Inverted at<br>connector? | Pin: Centronics |
| 3   | 15            | nError<br>(nFault) | Peripheral | no                        | 32              |
| 4   | 13            | Select             | Peripheral | no                        | 13              |
| 5   | 12            | PaperEnd           | Peripheral | no                        | 12              |
| 6   | 10            | nAck               | Peripheral | no                        | 10              |
| 7   | 11            | Busy               | Peripheral | yes                       | 11              |
| Additional bits not available at the connector: 0: may indicate ümeout  |               |                    |            |                           |                 |

| (I=timeout). 1, 2: unused.   |               |                |        |                           |                 |
|--|---------------|----------------|--------|---------------------------|-----------------|
| Control Register (Base Address +2)   |               |                |        |                           |                 |
| Bit  | Pin:<br>D-sub | Signal<br>Name | Source | Inverted at<br>connector? | Pin: Centronics |
| 0  | 1             | nStrobe        | PC1    | yes                       | 1               |
| 1  | 14            | nAutoLF        | PC1    | yes                       | 14              |
| 2  | 16            | nlnit          | PC1    | no                        | 31              |
| 3  | 17            | nSetectIn      | PC1    | yes                       | 36              |
| 'When high, PC can read extemal input (SPP only). Additional bits not available at the connector: 4: Interrupt enable. I=IRQs pass from nAck to system's interrupt controller. 0=IRQs do not pass to interrupt controller. 5: Direction control for bidirectional Data ports. 0=outputs enabled. I=outputs disabled; Data port can read external logic voltages. 6,7: unused |               |                |        |                           |                 |

Tablodaki sinyal isimleri orijinal IBM PC'de bulunan paralel port tarafından kullanılmaktadır. Sinyal fonksiyonlarını tanımlayan isimler PC'den çevre birimlerine doğru yapılan transferlere yöneliktir. Diğer modlarda sinyallerin fonksiyonları ve isimleri değişmektedir.

## 2.4. Paralel Arabirim Ünitelerinde Ortaya Çıkabilecek Problemlerin Çözümü

### 2.4.1. Paralel arabirimlerle çalışma

İlk popüler dot-matrix printer Centronics isimli bir Amerikan şirketi tarafından üretildi. Şirket basit tek yönlü bir paralel haberleşme sistemi geliştirdi ve bu sistemin yapısı ile ilgili tüm bilgileri halka sundu. Bu strateji sayesinde Centronics, Epson firmasının daha iyi bir printer üretmesine kadar popülerliğini korudu. Aslında Epson firması Centronics'in paralel arabirim tekniğini kullanmıştı fakat printer daha güvenilir. Tablo 2.4'te tipik Centronics paralel arabirim bağlantıları gösterilmiştir. Tabloda bilgisayar ve printer pinleri ayrı ayrı gösterilmiştir. Tablodaki rakamlar kabloların bağlı olduğu pin numaralarını göstermektedir. Tablodaki asterisk (\*) işaretleri ile belirlenmiş numaralar; printer bu bağlantılar olmadan da çalışır anlamına gelir.

Pin numaraları delikleri ya da konnektörler de numaralandırılmış kısımları gösterir. Örneğin Pin 1, genellikle konnektörün köşesindedir. Kablolar bu pinlere iki yolla bağlanır. Bunlar; alalım ve geçirme yöntemleridir. Geçirme yöntemi çok yaygın olmasına rağmen az güvenilirdir. Alalım yöntemi ise çok pahalı bir yöntemdir. Satın aldığımız bir kablonun alalımla veya geçirme ile pinlere bağlandığını, kablonun pinlerle bağlantısını sıyırıp ayırmadan anlayamayız. Eğer bağlantı kablosunun dış kabuğunu soyarsanız, kablonun konnektöre nasıl bağlandığına dikkat edin. Alalım bağlantıda herhangi bir pürüz olmazken geçirme bağlantıların sonunda metal fazlalıklarla görülür.

Tablo 2.4. Centronics paralel port

| Computer |            | Printer |                                    |
|----------|------------|---------|------------------------------------|
| in       |            |         | in P                               |
|          | Strobe     | 1       | telle printer to latch on to data  |
|          | Data Bit 0 | 2       | Data                               |
|          | Data Bit 1 | 3       | Data                               |
|          | Data Bit 2 | 4       | Data                               |
|          | Data Bit 3 | 5       | Data                               |
|          | Data Bit 4 | 6       | Data                               |
|          | Data Bit 5 | 7       | Data                               |
|          | Data Bit 6 | 8       | Data                               |
|          | Data Bit 7 | 9       | Data                               |
| 0        | Clock      | 1<br>0  | Printer tells computer received    |
| 1        | Busy       | 1<br>1  | Printer tells computer buffer full |
| 2        | Paper Out  | 12 *    | not necessary                      |
| 3        | Select     | 13 *    | not necessary                      |
| 4        | Auto Feed  | 14 †    | not necessary                      |
| 5        | Error      | 15 ‡    | not necessary                      |
| 6        | Initialize | 16 *†   | not necessary                      |
| 7        | Select     | 17 †    | not necessary                      |
| 8-25     | Ground     | 8-30    | not necessary                      |

Paralel arabirim problemlerinin giderilmesi sırasında iki tane kablomuzun olması gerekir. Bu kablolardan birisi yukarıdaki tüm pinleri içeren kablo, diğeri ise asterikslerle işaretlenmiş pinleri içermeyen küçük kablodur. Bağlantı pin sayısı az olan küçük kablo, büyük kablonun çalışmadığı her yerde çalışır. Bir sonraki bölümde bağlantı sayısı az olan küçük kablodaki gereksiz pinlerin neden sökülebileceği açıklanmıştır.

#### **2.4.2. Crosstalk problemi**

Çok kablolu paralel hatlarda bilgi iletiminde, bazı kablolardaki bilgiler diğerlerini magnetik etki yaratarak kötü yönde etkiler. Bu probleme "crosstalk problemi" denir. Crosstalk problemini daha iyi anlamak için bir örnek verelim. Telefon sistemlerinde uzun mesafeli telefon görüşmelerinde bir kullanıcı başkalarının konuşmalarını az da olsa duyar.

Crosstalk problemini önlemek için iki yöntem kullanılır. Bunlardan birinci yöntem, hatla gönderilen bilgilerin akım ve gerilim seviyeleri çok yüksek tutulmaz. Paralel hat uzunluğu arttıkça crosstalk problemi artar.

Eğer bilgisayarımız ve paralel iletişim yaptığı makine arasındaki bağlantı kablosu önerilen uzunluktan fazla olursa sistem tam güvenilirlikte çalışmaz. Örneğin Centronix kendi printer kabloları için bu mesafeyi 9 feet olarak tavsiye etmiştir.

## BÖLÜM 3

### İNTERNET

İnternet, birçok bilgisayar sisteminin birbirine bağlı olduğu, dünya çapında yaygın olan ve sürekli büyüyen bir iletişim ağıdır.

İnternet, insanların her geçen gün gittikçe artan "üretilen bilgiyi saklama, paylaşma ve ona kolayca ulaşma" istekleri sonrasında ortaya çıkmış bir teknolojidir. Bu teknoloji yardımıyla pek çok alandaki bilgilere insanlar kolay, ucuz, hızlı ve güvenli bir şekilde erişebilmektedir. İnternet' i bu haliyle bir bilgi denizine, ya da büyükçe bir kütüphaneye benzetebiliriz.

İnternet teknolojileri sanayi devriminde olduğundan daha önemli bir gelişmeyi temsil etmekte ve yaşamın tüm boyutlarını çalışma, üretim, ticaret, iş yapma, eğlence, öğrenme, yönetim biçimlerini köklü bir şekilde değiştirmekte. İnternet'in temsil ettiği bu değişim, ekonominin yapısını, ülkenin rekabet gücünü, insan gücü gereksinimini, mesleklerin yapısını değiştirmeye başlamış durumdadır.

#### 3.1. İnternette Güvenlik

İnternetin hızla yaygınlaşması ve gelişmesi evlerde, okullarda, kamu kuruluşlarında, finansal işlemlerde, kısaca hayatın her alanında radikal değişiklikleri de beraberinde getirmiştir. İnternet kullanımının artmasıyla beraber başta finansal kurumlar olmak üzere birçok kuruluş ticari işlemlerini İnternet'e taşımaya başlamıştır. Böylece bilgilerin herkese açık bir ağ üzerinde dolaşmasının yarattığı haklı tedirginlik gündeme gelmiş, bu konu yoğun bir şekilde tartışılmaya başlanmıştır.

İnternet'te güvenlik konusu iki temel başlık altında incelenebilir:

1. Yazılım ve ağ altyapısından dolayı meydana gelebilecek sorunlar. Bu kapsamda İnternet'e bağlandığımız servis sağlayıcıların ve kişisel bilgilerinizi verdiğiniz kurumların bu bilgileri sakladıkları ortamlar ve iç güvenlikleri ile bilgisayarların İnternet'e bağlı oldukları sürede maruz kaldıkları riskler ve önlemler yer alır. Bu kurumlar kendi iç ağlarını İnternet yolu ile gelecek tehlikelere karşı korumak için firewall çözümleri, virüs gateway çözümleri gibi teknolojik önlemler de almaktadırlar. Bunlara ilave olarak güvenilir kurumlar, veri tabanlarında tutulan bilgilerden önemlilerini şifrelenmiş olarak tutmakta, bu bilgileri şifrelenmiş olarak şirket içindeki işlemlerde kullanmaktadır.

2. İnternet üzerinde veri akışı sırasında, bilgilerin çeşitli teknik açıklar değerlendirilerek kötü amaçlı kullanım için ele geçirilme tehlikesi Genel bir İnternet kullanıcısının verebileceği bilgilerin arasında kendi güvenliği ve mahremiyeti açısından sakıncalı olabilecek kredi kartı bilgileri, kullanıcı isimleri şifreler, adres ve telefon numaraları bulunmaktadır. Bu bilgiler genellikle, elektronik ticaret veya finansal işlemler (örneğin bireysel bankacılık) sırasında İnternete açılır, ve ele geçirilme olasılıkları belirir. Elektronik ticaret ve finansal işlemlerin yürütüldüğü siteler İnternet'ten bilgi alışverişini şifreleyerek gerçekleştirdikleri için genel olarak güvenlidirler. Şifrelenmiş bilgi İnternet üzerinde iletilirken ele geçirilse bile, şifrenin kırılması çok büyük bir yatırım ve oldukça uzun bir zaman dilimi gerektirdiğinden güvenli olduğu kabul edilebilir Şifreleme amacıyla yaygın olarak SSL (Secure Socket Layer) güvenlik standardı kullanılmaktadır.

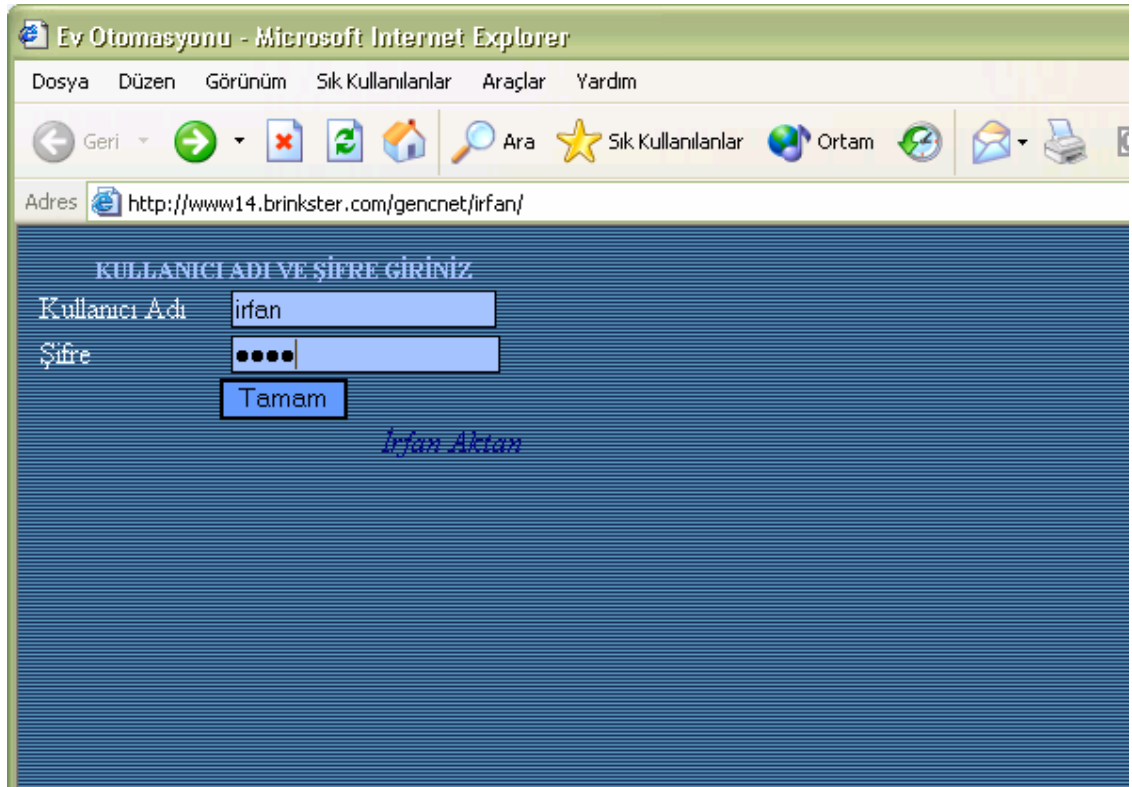
Uygulanacak olan güvenlik politikası ve güvenlik seviyesi ise kurumun ele geçirilebileceğini düşündüğü bilginin değeri ile belirlenir.

### **3.2. Web Sayfası Tasarımı Ve Teknikleri**

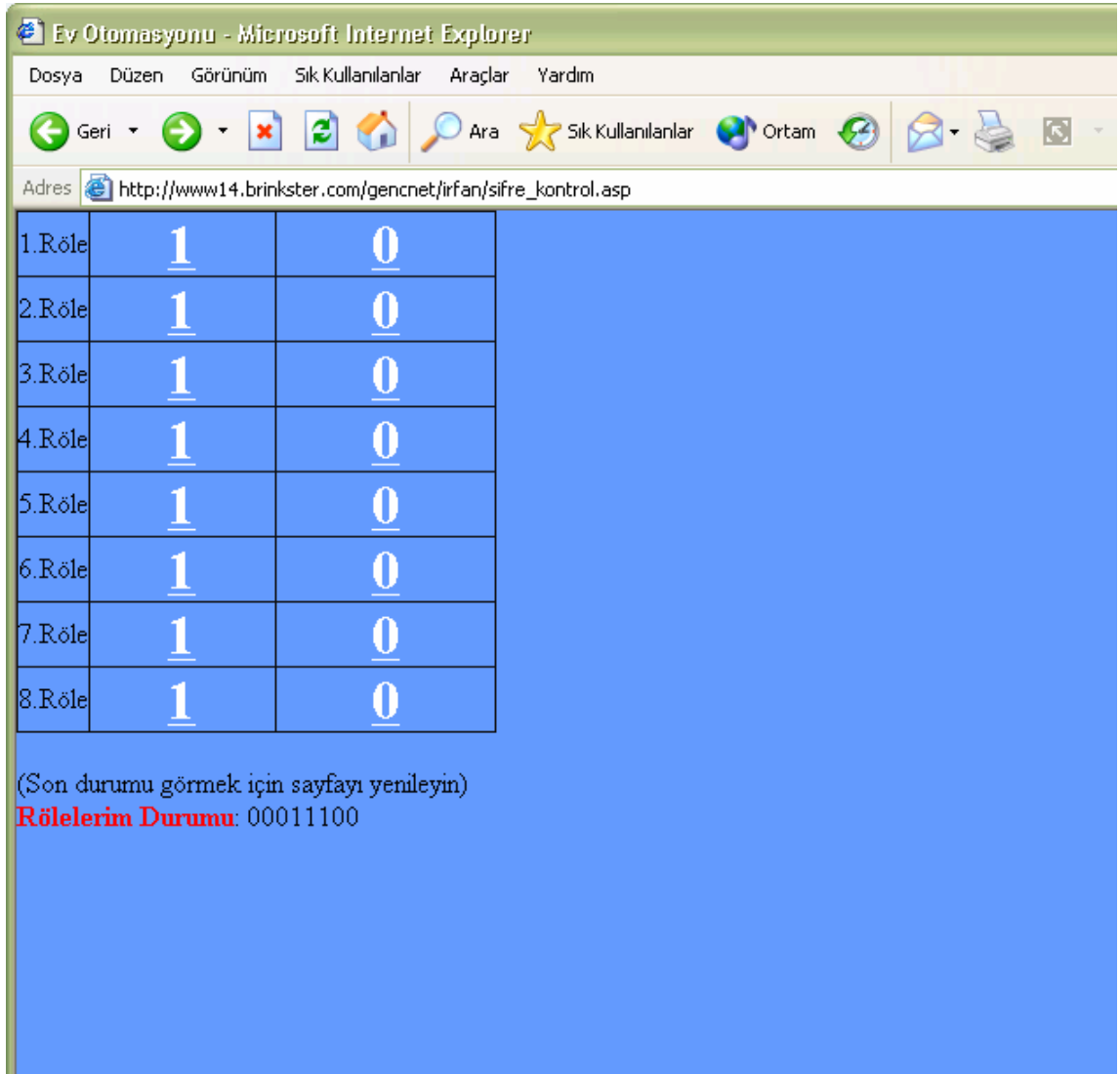
Bir web sayfası dizaynı programlı bir şekilde gereksinimler doğrultusunda elde olan verilerin en iyi şekilde değerlendirilmesinden sonra yapılabilir. Sayfanın işlevsel, abartılmamış bir görselliğe sahip olması gerekir. İlk olarak sayfadan beklenen işlemlerin iyi bir şekilde gerçekleştirilmesi sağlanmalı daha sonra da düzenli bir sunum için görsel öğelere yer verilmelidir.

İşlevsellik konusunda kullanılacak programla sistemlerin ve teknolojinin en uygun biçimde seçilmesi önemli bir noktadır. Bunda da fazla abartıya kaçmadan gereksinimleri en iyi şekilde karşılayacak ve tabiki güvenli bir yol izlenmelidir.

Bu bitirme ödevinde gerçekleştirilen sistemde internet üzerinden uygulama yapılacağı için haliyle HTML kullanılmıştır. Aktif öğelerin programlanmasında da ASP kullanılmıştır. Esas sistem ASP üzerine dayanmaktadır. Ayrıca veritabanlarında veri sorgulaması yapılırken de SQL kullanılmıştır. Bu adı geçen dillere daha ayrıntılı biçimde değinilecektir. Şekil 3.1’de kullanıcının şifre girmesi ve şifre doğru ise emir verebilmesi için tasarlanmış web sayfaları bulunmaktadır. Şekil 3.2.’deki 1’ler ve 0’lar link şeklinde olup bunlardan biri ile emir verildiği zaman bu emir `emirgoster.asp` dosyasında görüntülenmektedir.



Şekil 3.1. Şifre sorma sayfası



Şekil 3.2.Emirlerin gönderildiği web sayfası

### 3.3. HTML

HTML (Hyper Text Markup Language) internet ortamında döküman oluşturmak için geliştirilmiş bir programlama dilidir. HTML dökümanı bir text dosyadır ve tek başına bir yazı olmaktan öteye gidemez. Ancak her hangi bir internet tarayıcısı (internet

explorer, netscape navigator gibi ) yardımıyla çalıştırıldığında içerdiği komutlara göre ve de tarayıcı programın desteklediği özelliklere göre, hazırlanan dökümanlar bir anlam kazanır.

HTML dili programlama mantığından biraz uzak, görsel yönü oldukça önemli olan bir dildir. HTML nin öğrenilmesi, diğer programlama dillerine göre daha kolay olup bazı temel kural ve komutların bilinmesi yeterlidir.

Günümüzde HTML sayfaları hazırlamak için bir çok görsel program vardır (Microsoft FrontPage, HotDog gibi ). Bu tür programlarla HTML sayfaları hazırlamak çok kolay ve hızlı olmakla beraber HTML kodlarını da yine bu programlar üretmektedir.

Hazır programların olması, kodların el değmeden programlar tarafından yazılması daha güzel, daha kolay ve daha hızlıdır. Ancak hazırlanan sayfaların nasıl oluştuğunu, nasıl çalıştığını, arka planda nelerin olduğunu bilmeden, ezberci bir zihniyet kullanmak bir programcı mantığına terstir. Bu tür programları hemen her yerde bulamayabilirsiniz. Dökümanların değiştirilmesi gereken yerlerde eğer hazırladığınız program yoksa ve de HTML dilini biliyorsanız her hangi bir editörden dökümanınıza müdahale edebilirsiniz. HTML bilmenin ikinci bir gerekliliği de ASP kodları HTML kodları arasına gömülü şekilde çalışır. HTML kodlarınının ne işe yaradığını bilemeden ASP kullanılamaz.

HTML dökümanı hazırlamak için her hangi bir text dosya yazabilecek bir editör olması yeterlidir. Bütün yapılması gereken dökümanı HTML kurallarına uygun bir şekilde yazmak ve de dosyayı ".htm" olarak kaydetmektir. Ne derleme ne de ".exe" gibi dosyalara çevirme işlemi yoktur. HTML dosyası hazırdır. Artık tek yapılması gereken hazırlanan bu dosyanın bir internet tarayıcı programı tarafından çağrılmasıdır.

HTML de program yazılırken her satır bir HTML komutu ile başlar ve de komutun geçerliliği biteceği zaman aynı komut isminin başına "/" konularak tekrar yazılır.

<komut>...</komut>

Bir HTML sayfası <Html> komutuyla başlar ve de sayfa sonunda </Html> komutuyla biter. Sayfayla ilgili bilgiler (Başlığı, hangi program kullanılarak yapıldığı...) <Head> komutu yazıldıktan sonra verilir ve de bilgiler bitince </Head> komutuyla kapatılır.

Sayfada gösterilecek bilgiler <Body>...</Body> komutları arasında yazılır.

<Html>

<Head>

<Title>Deneme Sayfası</Title><!--Başlık burada tanımlanır-->

</Head>

<Body><!--Bilgiler burada gösterilir-->

Html öğreniyorum, ilk deneme sayfası

</Body>

</Html>

HTML de açıklama satırı <!-- açıklama --> şeklinde kullanılır. Yazılan açıklama, sadece program satırlarıyla gözüktür. Sayfa çağrıldığında gözükmez. Yazılan program kodlarının ne anlama geldiğinin anlaşılması veya başka biri tarafından bakılan kodların anlaşılabilmesi bakımından bol bol açıklama satırı koyulmalıdır.

<Body> komutu yardımıyla sayfanın arka rengini, arkaya konulacak resmi, bağlantıların (Hyperlinks) rengini, sayfanın yukarı ve sol tarafa göre konumları belirtilebilir. Bir komutun yanına eğer varsa parametreler komut kapatılmadan ">" önce verilir. Komut kapatılıncaya kadar bu parametreler geçerliliğini korur. HTML'de komutların büyük ya da küçük harfle yazılma zorunluluğu yoktur. <br> komutu programda bulunduğu yerden itibaren alt satıra geçişi sağlar. Bu komutun </br> ifadesi yoktur.

HTML dili için görsellik, dolayısıyla da ekran dizaynı çok önemlidir. Dökümanlarınızı en iyi ve en güzel şekilde sunmanız gerekir. Html'de bu dizaynları yapabilmemiz için bazı temel komutlar bulunmaktadır. Yazı veya resimleri kısaca kullanılacak nesnelere ekranın sağına, soluna ve ortasına koyabilmek için <align =...> komutu kullanılır.

<p align = left> </p> Nesnelere ekranın soluna yerleştirir.

<p align = right> </p> Nesnelere ekranın sağına yerleştirir.

<p align = center> </p> Nesnelere ekranın ortasına yerleştirir.

<p> ..</p> komutu arasında kalan tüm nesnelere (yazı, resim, ...) paragraf gibi ayrılır.

Paragrafın başında ve sonunda paragraf olduğunu belirtmek için boşluk bırakılır.

HTML de tüm nesnelere aksi belirtilmedikçe ekranın solundan itibaren gösterilir. Yani soldan başlayarak yazı yazmak için <align = left> komutunu kullanılmasına gerek yoktur.

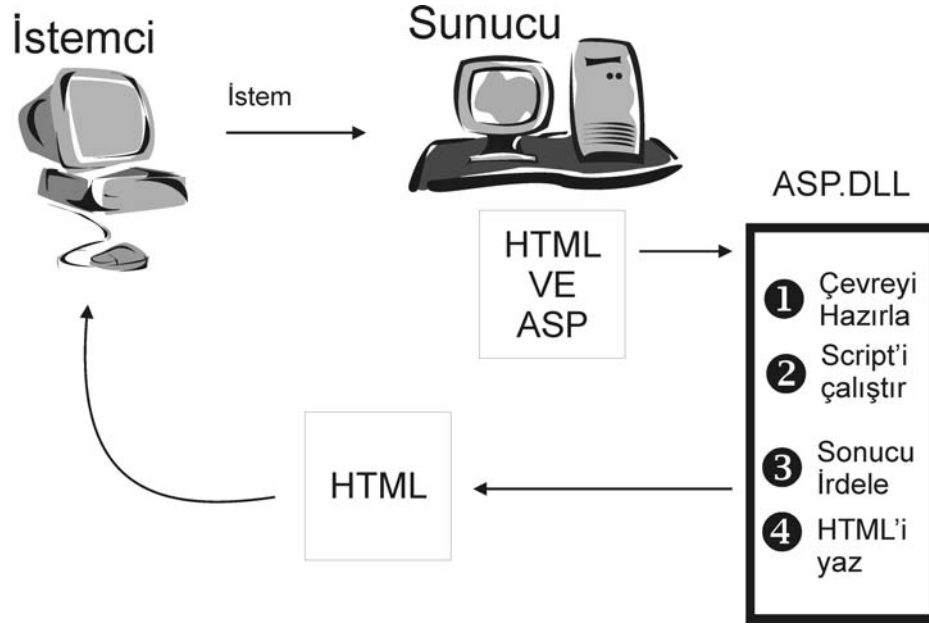
### 3.4. ASP

ASP veya Active Server Pages (Etkin Sunucu Sayfaları) tekniği, web sayfalarını canlandıracak bir tekniktir. Adından anlaşılacağı gibi ASP kodları sunucu kısmında çalışır yani istemciye asp kodları gönderilmez bunun yerine asp kodları çalıştırıldıktan sonra oluşan html dökümanı kullanıcıya iletilecektir. Bu da tasarımların kullanılan browserdan bağımsız olması gibi pozitif bir özellik sağlayacaktır. Diğer diller örneğin Java script kodları sunucu tarafında değil istemci tarafında çalışırlar ve browserın yorumuna göre değişik çıktılar üretebilirler veya hiç çalıştırılmayabilirler. ASP'de ise böyle bir sorun yoktur.

ASP'nin içinde HTML ile kod yan yanadır. ASP yoluyla Web Server'a verilen komutlar, herhangi bir Script dili ile yazılabilir. Bu VBScript olabilir; Javascript veya Jscript olabilir hatta, Microsoft firmasının yeni teknolojisi olan Windows Scripting Host ile uyumlu olmak koşuluyla, kişisel Script dilleri icat edip, bunu Web Server'a öğretecek plug-in (ek program) geliştirilebilir. İlk dönemlerinde ASP kodlarını VBScript ile yazmak adeta şarttı. Bu sistemde de VBscript kullanılmıştır. Bir ASP

sayfası herhangi bir html dökümanı gibi bilgisayarda çalışmayacaktır. Bu kodların çalışması için herhangi bir web server programının kurulu olması gerekir. (Personal Web Server veya Internet Information Service). ASP teknolojisi ile oluşturulan sayfaları, yani içinde ASP teknolojisinin gerektirdiği kodlar bulunan HTML dosyaları .asp uzatmasıyla kaydedilir. Bu dosyalar talep edildiğinde, Web Server programı, eğer ASP teknolojisini tanıyor, talep edilenin ASP sayfası olduğunu dosya adının uzantısından anlar; ve bu sayfayı ziyaretçiye yollamadan önce kendisi işleme tabi tutar. ASP sayfasındaki kodlar büyük bir ihtimalle bir takım dinamik işler yapacaktır, örneğin hesaplamalar, araştırmalar, veri tabanından veri çekme, dosya sisteminden dosya isteme gibi görevlerini yerine getirecek, ve ortaya çıkacak olan HTML dosyasını ziyaretçiye gönderecektir

İstemci Browser, URL hanesine adı yazılan HTML dosyasını yine adresteki sunucu Web Server'dan ister. Web Server da bu dosyayı bulur ve içinde kayıtlı resim ve diğer unsurlarla birlikte istek sahibine gönderir. Fakat kimi zaman Server'a sadece bir dosyayı göndermesini değil, fakat bu dosyanın içinde kayıtlı komutlar varsa onları icra etmesi de bildirilebilir. Bunu yapmanın bir yolu CGI programlarıdır. Bir diğer yolu ise ASP'dir. Web Server, kendisinden bir ASP belgesi talep edildiğinde, durur ve bunu ASP.DLL programına gönderir. Ondan sonra işine devam eder. Şekil 3.3'de ASP uzantılı dosyaların çalışması diyagram halinde verilmiştir.



Şekil 3.3. ASP'nin çalışması

ASP.DLL, kendisine bir .asp dosyasının geldiğini görünce, hemen ASP'lerin kuralları olan global.asp' nin çalışıp çalışmadığına bakar. global.asp, tıpkı diğer ASP dosyaları gibi bir düz yazı dosyasıdır ve ASP programlarının çalışma koşullarını düzenleyen kuralları içerir. Yukarıdaki örnekte görüldüğü gibi ASP dosyası hem HTML kodları içerir, hem de içinde bir Script diliyle yazılmış kodlar vardır. ASP'ye "program" özelliği kazandıran bu Script dili ile yazılmış kodlardır. ASP.DLL, önce gelen .asp dosyasında hangi Script dilinin kullanıldığına bakar ve bunun için gerekli ortamı oluşturur; yani bu Script dilini yorumlayacak programı çalıştırır; bu program Script'i yorumlar ve icra edilecek komutları icra eder; ASP.DLL, icra edilen komutlar, işletim sisteminin yardımını istiyorsa (örneğin bir veritabanından veri çekmek gibi, veya dosya sistemine bir dosya açtırmak, yazdırmak, sildirmek gibi) bu yardımın edinilmesini de sağlar. Bütün bu işlerin sonunda yazılan HTML kodlarına ek yapmak (örneğin bir tablonun içeriğini, çekilen verilerle doldurmak veya dosya sisteminden edinilen bir dosyanın içeriğini sayfaya aktarmak gibi) gerekiyorsa bu ekleri ASP.DLL yapar.

ASP.DLL, kendisine sevk edilen dosyayı satır satır okur ve önce dosyadaki ASP kodlarının gerektirdiği HTML değerlerini bulur; yani önce ASP icra edilir, gereği yerine

getirilir. Sonra HTML bölümleri ile birleştirilip sonuçta sunucuya saf ve temiz bir HTML sayfası gönderilir. Bu sayfanın içinde bir satır bile ASP kodu bulunmaz. Eğer sayfaya ziyaretçinin Browser' ında çalışması amacıyla Javascript veya VBScript kodları koyulmuşsa, elbette bu kodlar HTML' in parçası olarak ziyaretçiye gidecektir. Fakat giden sayfada artık ASP'ye ilişkin hiç bir şey kalmamış olacaktır.

## BÖLÜM 4

### PROGRAMLAMA

Bu bölümde internet üzerinden veriyi alan ve paralel porta aktaran, visual basic ile yazmış olduğumuz programa yer verilmiştir. Programın metni aşağıdaki gibidir. Ayrıca “emirgöster.asp” ve “emirgönder.asp” ve “şifrekontrol.asp” dosyalarının içerikleri ve programın diğer kısmı da EKLER bölümündedir.

```
Begin VB.Form Form1
  BackColor    = &H00FF8080&
  BorderStyle  = 1 'Fixed Single
  Caption      = " Lambalar"
  ClientHeight = 1365
  ClientLeft   = 45
  ClientTop    = 435
  ClientWidth  = 3255
  Icon         = "Form1.frx":0000
  LinkTopic    = "Form1"
  MaxButton    = 0 'False
  MinButton    = 0 'False
  ScaleHeight  = 1365
  ScaleWidth   = 3255
  StartupPosition = 3 'Windows Default
Begin VB.Timer Timer2
  Interval     = 100
  Left         = 2040
  Top          = 240
End
Begin VB.HScrollBar HScroll1
  Height       = 255
  LargeChange  = 3
  Left         = 120
  Max          = 30
  Min          = 1
  TabIndex     = 3
  Top          = 1080
  Value        = 3
  Width        = 1695
End
Begin VB.VScrollBar VScroll1
  Height       = 855
```

```
LargeChange = 20
Left = 2880
Max = 250
Min = 77
SmallChange = 5
TabIndex = 2
Top = 120
Value = 77
Width = 255
End
Begin VB.Timer Timer1
Left = 1080
Top = 240
End
Begin VB.TextBox Text1
Appearance = 0 'Flat
BackColor = &H00FFC0C0&
Height = 285
Left = 120
TabIndex = 0
Text = "http://www14.brinkster.com/genchnet/irfan"
Top = 120
Width = 2655
End
Begin VB.Image Image1
Height = 240
Left = 2880
Picture = "Form1.frx":000C
Stretch = -1 'True
Top = 1080
Width = 240
End
Begin VB.Label Label2
Alignment = 2 'Center
Appearance = 0 'Flat
BackColor = &H80000005&
BackStyle = 0 'Transparent
BorderStyle = 1 'Fixed Single
Caption = "1 Saniye"
ForeColor = &H80000008&
Height = 255
Left = 1920
TabIndex = 4
Top = 1080
Width = 855
End
Begin VB.Label Label1
Alignment = 2 'Center
```

```
Appearance = 0 'Flat
BackColor = &H00FFC0C0&
BorderStyle = 1 'Fixed Single
Caption = "İRFAN"
BeginProperty Font
  Name = "PCLifeLiquidCrystal"
  Size = 18
  Charset = 0
  Weight = 700
  Underline = 0 'False
  Italic = 0 'False
  Strikethrough = 0 'False
EndProperty
ForeColor = &H80000008&
Height = 465
Left = 120
TabIndex = 1
Top = 480
Width = 2655
End
End
Attribute VB_Name = "Form1"
Attribute VB_GlobalNameSpace = False
Attribute VB_Creatable = False
Attribute VB_PredeclaredId = True
Attribute VB_Exposed = False
Dim görünür As Boolean
  Dim TextAddr
  Dim TextValue

Private Sub Form_Load()
  TransparentYap Me.hWnd, 250
  VScroll1.Value = 250
  HScroll1.Value = GetSetting("cgk", "Led", "sure", "5")
  Timer1.Interval = HScroll1.Value * 1000
  Label2.Caption = HScroll1.Value & " Saniye"
  TextAddr = 956
  TextValue = 0
End Sub

Private Sub Form_Unload(Cancel As Integer)

  SaveSetting "cgk", "Led", "sure", HScroll1.Value
End

End Sub

Private Sub HScroll1_Change()
```

```
Label2.Caption = HScroll1.Value & " Saniye"  
Timer1.Interval = HScroll1.Value * 1000
```

```
End Sub
```

```
Private Sub Image1_Click()  
If gorunur = False Then gorunur = True: Form2.Show Else gorunur = False:  
Form2.Hide
```

```
End Sub
```

```
Private Sub Timer1_Timer()  
Dim Value As Long
```

```
sonuc = Mid$(Trim(GetUrlSource(Text1.Text & "/emirgoster.asp")), 1, 8)  
Label1.Caption = sonuc
```

```
If Mid$(sonuc, 1, 1) = "1" Then toplam = toplam + 32  
If Mid$(sonuc, 2, 1) = "1" Then toplam = toplam + 16  
If Mid$(sonuc, 3, 1) = "1" Then toplam = toplam + 128  
If Mid$(sonuc, 4, 1) = "1" Then toplam = toplam + 64  
If Mid$(sonuc, 5, 1) = "1" Then toplam = toplam + 2  
If Mid$(sonuc, 6, 1) = "1" Then toplam = toplam + 1  
If Mid$(sonuc, 7, 1) = "1" Then toplam = toplam + 8  
If Mid$(sonuc, 8, 1) = "1" Then toplam = toplam + 4  
Me.Caption = "(" & sonuc & "/" & toplam & "/" & Hex(toplam) & "H)"  
TextValue = toplam  
Value = Val(TextValue)  
If Value < 0 Then Value = Value + 256  
DllPortWritePortUchar Val(TextAddr), Value  
End Sub
```

```
Private Sub Timer2_Timer()
```

```
If Form1.WindowState = 0 Then  
Form2.Left = Form1.Left + Form1.Width + 100  
Form2.Top = Form1.Top  
End If
```

```
End Sub
```

```
Private Sub VScroll1_Change()  
TransparentYap Me.hWnd, VScroll1.Value
```

```
End Sub
```

“sonuc = Mid\$(Trim(GetUrlSource(Text1.Text & "/emirgoster.asp")), 1, 8)” komutu ile emirgoster.asp dosyasının içeriği belli süreler zarfında alınmakta ve kaydedilmekte. Ve kaydedilen bu son veri paralel porta gönderilmektedir.

Aşağıda internet bölgesie yüklediğimiz ASP dosyalarının görünümü, isimleri ve yerine getirdikleri işlevler tablo halinde verilmiştir.

Tablo 4.1. Dosyaların isimleri ve işlevleri

|                   |   |
|-------------------|---|
| Default.HTM       | Sistem giriş sayfası, site adı yazıldığında ilk bu dosya açılır |
| Emir Göster.ASP   | Visual Basic programı için emirleri görüntüler                  |
| Emir Gönder.ASP   | Veritabanına kullanıcıdan gelen emri yazar                      |
| Şifre Kontrol.ASP | Kullanıcının girdiği şifrenin doğruluğunu kontrol eder.         |

## **BÖLÜM 5**

### **HARDWARE (DONANIM)**

Bu bölümde uygulamanın yapılabilmesi için gereken devrenin şeması ve nasıl çalıştığı anlatılacaktır.

#### **5.1.Devrenin Açık Şeması**

Şekil 4.1’de internet üzerinden donanım kontrolü için gereken devrenin açık şeması ve eleman değerleri verilmiştir



## BÖLÜM 6

### SONUÇ

Sonuç olarak bu çalışma bize, uzaktan kontrolün internetle sağlanabileceğini kanıtlamıştır. Ayrıca internet, giderek gelişen bir sektör olarak karşımıza çıkmaktadır. Bu da bu çalışmanın, ileriye dönük olarak oldukça geniş bir ufuk açtığını göstermektedir. Teknolojik olarak hızla gelişmekte olan ülkemizde dahi, internete ulaşabilen insan sayısının oldukça fazla olması, bunun çok lüks bir çalışma olmadığını, bilakis belki ileride olmazsa olmaz bir sistem olacağını göstermektedir.

Günümüzde bu sistemin henüz evlerimizde kullanılması zor olsa da , fabrika, orta ölçekli işletmeler, sürekli kontrolün gerekli olduğu makine ağırlıklı işletmelerde çok rahatlıkla kullanılabilir.

Ayrıca sistemin maliyeti de oldukça düşük olacaktır. Biz bu basit çalışmada paralel portumuzun 8 bitini kullandık. Eğer büyük bir otomasyon sistemi istenirse, bir bilgisayarla, başka bir deyişle bir paralel portla 256 adet motor, röle, kontaktör vb. çok çeşitli cihaz kontrol edilebilir. Bu irili ufaklı 256 adet elektrikli veya elektronik cihazın kontrolü demektir. Yani bir sunucu bilgisayar ve bu bilgisayarın paralel portu ile, kompleks bir sistemin kontrolü rahatlıkla sağlanabilir. Tabii bilgisayardan aldığımız 5 voltluk sinyali istenen kontrol sinyaline dönüştürecek sistemi kurmak şartı ile.

Sonuç olarak bu çalışmanın bize oldukça büyük bir kolaylık sağlayacağını , uzak mesafeleri internetin hızı sayesinde azaltacağını söylemek mümkündür.

## EKLER:

### Default.htm:

```
<html>

<head>
<meta http-equiv="Content-Language" content="tr">
<meta name="GENERATOR" content="Microsoft FrontPage 5.0">
<meta name="ProgId" content="FrontPage.Editor.Document">
<meta http-equiv="Content-Type" content="text/html; charset=windows-1254">
<title>Ev Otomasyonu</title>
<script LANGUAGE="JavaScript1.2">
function denetle() {
    var num = document.form1.elements.length
    var dogruMu = true
        for (var i=0; i<num; i++) {
            if ((document.form1.elements[i].value == null ||
                document.form1.elements[i].value == "") &&
                (typeof document.form1.elements[i] != 'submit' ||
                typeof document.form1.elements[i] != 'reset'))
            {
                dogruMu = false
                alert("Formdaki " + document.form1.elements[i].name +
                    " alanı doldurulmamış bulunuyor. Lütfen
doldurunuz!")
                break }
            }
        return dogruMu
    }
// -->
</script>
```

```
</head>
```

```
<body background="bg.gif">
```

```
    <form name="form1" method="post" ACTION="sifre_kontrol.asp"
onSubmit="return denetle()">
```

```
    <div align="left">
```

```
        <table border="0" cellspacing="0" style="border-collapse: collapse"
bordercolor="#111111" width="265" id="AutoNumber3">
```

```
        <tr>
```

```
            <td width="261" colspan="2">
```

```
                <p align="center"><b><font color="#A6C2FF" size="2">KULLANICI ADI
```

```
VE
```

```
                ŞİFRE GİRİNİZ</font></b></td>
```

```
            </tr>
```

```
        <tr>
```

```
            <td width="100">
```

```
                <p align="left"><font color="#FFFFFF">Kullanıcı Adı</font></td>
```

```
            <td width="161">
```

```
                <input type="text" name="kullanici" size="20" style="border-style: solid;
border-width: 1; background-color:#A6C2FF"></td>
```

```
            </tr>
```

```
        <tr>
```

```
            <td width="100"><font color="#FFFFFF">Şifre</font></td>
```

```
            <td width="161">
```

```
                <input type="password" name="sifre" size="20" style="border-style: solid;
border-width: 1; background-color:#A6C2FF"></td>
```

```
            </tr>
```

```
        <tr>
```

```
            <td width="261" colspan="2">
```

```
                <p align="center">
```

```
<input type="submit" name="Submit" value="Tamam" style="border-style:
solid; border-width: 1; background-color: #6699FF">
</td>
</tr>
<tr>
<td width="261" colspan="2">
<p align="right"><font color="#000080"><i>İrfan Aktan</i></font></td>
</tr>
</table>
</div>
</form>
```

```
</body>
```

```
</html>
```

### Şifrekontrol

```
<%
```

```
dim veriyolu,veri,gelen,mesaj
```

```
Set veriyolu = Server.CreateObject ("ADODB.Connection")
```

```
veriyolu.Provider = "Microsoft.Jet.OLEDB.4.0"
```

```
veriyolu.Open Server.MapPath("") & "\gencnet\db\ev.mdb"
```

```
%>
```

```
<%
```

```
Dim strkullan, strsifre
```

```
strkullan=Request.form("kullanici")
```

```
strsifre=Request.form("sifre")
```

```
Set veri = veriyolu.Execute("SELECT * FROM ayar WHERE id="" & strkullan & """)
```

```
%>
```

```
<html>
```

```
<head>
<meta http-equiv="Content-Type" content="text/html; charset=windows-1254">
<meta name="GENERATOR" content="Microsoft FrontPage 5.0">

<title>Ev Otomasyonu</title>
</head>
<body bgcolor="#6699FF" topmargin="0" leftmargin="0">
<%if not veri.bof and veri(1)=strsifre then%>
  <table border="1" cellpadding="0" cellspacing="0" style="border-collapse:
collapse" bordercolor="#111111" width="273" id="AutoNumber4" height="153">
    <tr>
      <td width="37" align="center" height="19">1</td>
      <td width="106" align="center" height="19"><a
href="emir_gonder.asp?islem=1">
        <font color="#FFFFFF">Lambayı Yak</font></a></td>
      <td width="126" align="center" height="19"><a
href="emir_gonder.asp?islem=2">
        <font color="#FFFFFF">Lambayı Söndür</font></a></td>
    </tr>
    <tr>
      <td width="37" align="center" height="19">2</td>
      <td width="106" align="center" height="19"><a
href="emir_gonder.asp?islem=3">
        <font color="#FFFFFF">Lambayı Yak</font></a></td>
      <td width="126" align="center" height="19"><a
href="emir_gonder.asp?islem=4">
        <font color="#FFFFFF">Lambayı Söndür</font></a></td>
    </tr>
    <tr>
      <td width="37" align="center" height="13">3</td>
```

```
<td width="106" align="center" height="13"><a
href="emir_gonder.asp?islem=5">
<font color="#FFFFFF">Lambayı Yak</font></a></td>
<td width="126" align="center" height="13"><a
href="emir_gonder.asp?islem=6">
<font color="#FFFFFF">Lambayı Söndür</font></a></td>
</tr>
<tr>
<td width="37" align="center" height="19">4</td>
<td width="106" align="center" height="19"><a
href="emir_gonder.asp?islem=7">
<font color="#FFFFFF">Lambayı Yak</font></a></td>
<td width="126" align="center" height="19"><a
href="emir_gonder.asp?islem=8">
<font color="#FFFFFF">Lambayı Söndür</font></a></td>
</tr>
<tr>
<td width="37" align="center" height="19">5</td>
<td width="106" align="center" height="19"><a
href="emir_gonder.asp?islem=9">
<font color="#FFFFFF">Lambayı Yak</font></a></td>
<td width="126" align="center" height="19"><a
href="emir_gonder.asp?islem=10">
<font color="#FFFFFF">Lambayı Söndür</font></a></td>
</tr>
<tr>
<td width="37" align="center" height="19">6</td>
<td width="106" align="center" height="19"><a
href="emir_gonder.asp?islem=11">
<font color="#FFFFFF">Lambayı Yak</font></a></td>
<td width="126" align="center" height="19"><a
href="emir_gonder.asp?islem=12">
```

```
<font color="#FFFFFF">Lambayı Söndür</font></a></td>
</tr>
<tr>
<td width="37" align="center" height="19">7</td>
<td width="106" align="center" height="19"><a
href="emir_gonder.asp?islem=13">
<font color="#FFFFFF">Lambayı Yak</font></a></td>
<td width="126" align="center" height="19"><a
href="emir_gonder.asp?islem=14">
<font color="#FFFFFF">Lambayı Söndür</font></a></td>
</tr>
<tr>
<td width="37" align="center" height="19">8</td>
<td width="106" align="center" height="19"><a
href="emir_gonder.asp?islem=15">
<font color="#FFFFFF">Lambayı Yak</font></a></td>
<td width="126" align="center" height="19"><a
href="emir_gonder.asp?islem=16">
<font color="#FFFFFF">Lambayı Söndür</font></a></td>
</tr>

</table>

<%else%>
<p align="center">
<b><font color="#FFFFFF">Şifre veya Kullanıcı adı yanlış Girildi <br>
Yeniden Şifre Girebilmek için<br>
Şifre Sorma ekranına</font><font color="#FF0000"> <a
href="default.htm">gitmelisiniz.</a></font></b></p>
<%end if%>
</body>
</html>
```

### Emir göster

```
<%  
dim veriyolu,veri,x  
Set veriyolu = Server.CreateObject ("ADODB.Connection")  
veriyolu.Provider = "Microsoft.Jet.OLEDB.4.0"  
veriyolu.Open Server.MapPath("") & "\gencnet\db\ev.mdb"  
  
Set veri = veriyolu.Execute("SELECT * FROM gelen")  
veri.movefirst  
response.write (veri(0) & veri(1) & veri.fields(2) & veri.fields(3) & veri.fields(4) &  
veri.fields(5) & veri.fields(6) & veri.fields(7))  
  
%>
```

### Emir gonder:

```
<html>  
  
<head>  
<meta name="GENERATOR" content="Microsoft FrontPage 5.0">  
<meta name="ProgId" content="FrontPage.Editor.Document">  
<meta http-equiv="Content-Type" content="text/html; charset=windows-1254">  
<title>New Page 1</title>  
</head>  
<%  
dim veriyolu,veri,islem  
islem = request.querystring("islem")  
Set veriyolu = Server.CreateObject ("ADODB.Connection")  
veriyolu.Provider = "Microsoft.Jet.OLEDB.4.0"
```

```
veriyolu.Open Server.MapPath("") & "\gencnet\db\ev.mdb"
```

```
'set veri=veriyolu.execute("SELECT * FROM gelen")
```

```
'veri.movefirst
```

```
%>
```

```
<html>
```

```
<body onload="javascript:history.back();">
```

```
<%
```

```
if islem=1 then veriyolu.execute("UPDATE gelen SET 1=1")
```

```
if islem=2 then veriyolu.execute("UPDATE gelen SET 1=0")
```

```
if islem=3 then veriyolu.execute("UPDATE gelen SET 2=1")
```

```
if islem=4 then veriyolu.execute("UPDATE gelen SET 2=0")
```

```
if islem=5 then veriyolu.execute("UPDATE gelen SET 3=1")
```

```
if islem=6 then veriyolu.execute("UPDATE gelen SET 3=0")
```

```
if islem=7 then veriyolu.execute("UPDATE gelen SET 4=1")
```

```
if islem=8 then veriyolu.execute("UPDATE gelen SET 4=0")
```

```
if islem=9 then veriyolu.execute("UPDATE gelen SET 5=1")
```

```
if islem=10 then veriyolu.execute("UPDATE gelen SET 5=0")
```

```
if islem=11 then veriyolu.execute("UPDATE gelen SET 6=1")
```

```
if islem=12 then veriyolu.execute("UPDATE gelen SET 6=0")
```

```
if islem=13 then veriyolu.execute("UPDATE gelen SET 7=1")
```

```
if islem=14 then veriyolu.execute("UPDATE gelen SET 7=0")
```

```
if islem=15 then veriyolu.execute("UPDATE gelen SET 8=1")
```

```
if islem=16 then veriyolu.execute("UPDATE gelen SET 8=0")
```

```
%>
```

Emirleriniz Gönderildi. Durum ve Emir Sayfasına Gitmek İçin Browserınızın Geri

Tuşuna Basınız....

```
<br>
```

```
<font color="#FF0000"><b>Lambaların Durumu</b></font>:
```

```
<%
```

```
Set veri = veriyolu.Execute("SELECT * FROM gelen")  
veri.movefirst  
response.write (veri(0) & veri(1) & veri.fields(2) & veri.fields(3) & veri.fields(4) &  
veri.fields(5) & veri.fields(6) & veri.fields(7))  
  
%>  
  
</body>  
  
</html>
```

## **KAYNAKLAR**

- [1] TOKGÖZ, İlker 2002 Bitirme Ödevi, Paralel Portla PC Tabanlı Otomasyon
- [2] KOÇ, Cafer Gazi 2002 Bitirme Ödevi, Web Üzerinden Akademik Kontrol Sistemi
- [3] ÖCAL, Hakkı 2000 ASP Öğreniyorum kitapçığı
- [4] Axelson. J., 1999. Her Yönüyle Paralel Port. ERA Bilgi Sistemleri Yayını, İstanbul.,6-45.