

**T.C.  
SAKARYA ÜNİVERSİTESİ  
FEN BİLİMLERİ ENSTİTÜSÜ**

**ELEKTRONİK BİLGİLENDİRME PANOSUNUN  
WEB ÜZERİNDEN KONTROLÜ**

**YÜKSEK LİSANS TEZİ**

**Elektronik Öğretmeni Murat FETTAHOĞLU**

**Enstitü Anabilim Dalı : Elektronik-Bilgisayar Eğitimi**  
**Tez Danışmanı : Yrd. Doç. Dr. Ali Fuat BOZ**

**Haziran 2007**

T.C.  
SAKARYA ÜNİVERSİTESİ  
FEN BİLİMLERİ ENSTİTÜSÜ

**ELEKTRONİK BİLGİLENDİRME PANOSUNUN  
WEB ÜZERİNDEN KONTROLÜ**

**YÜKSEK LİSANS TEZİ**

**Elektronik Öğretmeni Murat FETTAHOĞLU**

**Enstitü Anabilim Dalı : Elektronik ve Bilgisayar Eğitimi**

**Bu tez 14/ 06 /2007 tarihinde aşağıdaki jüri tarafından Oybirliği ile kabul edilmiştir.**

**Yrd. Doç. Dr. Ali Fuat BOZ  
Jüri Başkanı**

**Yrd. Doç. Dr. İlyas ÇANKAYA  
Üye**

**Doç. Dr. Raşit KÖKER  
Üye**

## **TEŐEKKÖR**

Bu tezin hazırlanmasında katkıda bulunan danışmanım sayın Yrd. Doç. Dr. Ali Fuat BOZ 'a (Sakarya Üniversitesi), Sakarya Üniversitesi Teknik Eğitim Fakültesi Dekanı Prof. Dr. Hüseyin EKİZ 'e, elektronik-bilgisayar eğitimi bölümü öğretim üyelerine, Öğr. Gör. Sait Mahmut ÇINAR 'a (Afyon Kocatepe Üniversitesi ), bilgisayar öğretmeni Ali SÖNMEZ' e (Taraklı Lisesi) katkılarından ve yardımlarından dolayı teşekkür ederim.

# İÇİNDEKİLER

TEŞEKKÜR .....	ii
İÇİNDEKİLER .....	iii
SİMGELER VE KISALTMALAR LİSTESİ.....	vi
ŞEKİLLER LİSTESİ .....	vii
TABLolar LİSTESİ .....	viii
ÖZET .....	ix
SUMMARY .....	x

## BÖLÜM 1.

GİRİŞ .....	1
-------------	---

## BÖLÜM 2.

SİSTEM TANIMI .....	3
2.1. PIC Mikrodenetleyiciler .....	4
2.1.1. PIC Mikrodenetleyicinin tercih edilmesinin nedenleri .....	4
2.1.2. Mikrodenetleyiciler hakkında genel bilgiler .....	5
2.1.3. Mikroişlemci ve mikrodenetleyicilerin farkı: .....	6
2.1.4. PIC Programlama .....	7
2.1.4.1. Bilgisayar .....	7
2.1.4.2. PIC compiler programı (pic basic yada pic c).....	7
2.1.4.3. PIC programlayıcı devresi.....	8
2.1.4.4. PIC programlayıcı yazılımı .....	8
2.1.5. PIC donanım özellikleri .....	8
2.1.5.1. PIC çeşitleri .....	8
2.1.5.2. PIC Mikrodenetleyicilerde kullanılan bellek çeşitleri.....	9
2.1.6. PIC Mikrodenetleyicilerinin özellikleri .....	11
2.1.6.1. Güvenirlik .....	11

2.1.6.2. Hız .....	11
2.1.6.3. Komut seti .....	11
2.1.6.4. Statik işlem .....	12
2.1.6.5. Sürme özelliği (sürücü kapasitesi) .....	12
2.1.6.6. Seçenekler .....	12
2.1.6.7. Çok yönlülük .....	12
2.1.6.8. Güvenlik .....	13
2.1.6.9. Geliştirme .....	13
2.1.6.10. Liste dosyası .....	13
2.1.6.11. Diğer dosyalar .....	13
2.2 PIC 18F452 Mikrodenetleyicisinin özellikleri .....	14
2.2.1. Yüksek performanslı RISC mimarili CPU .....	14
2.2.2. Çevresel özellikler .....	15
2.2.3. Analog özellikler .....	15
2.2.4. Diğer özellikler .....	16
2.3. RS-232/V.24 standardı .....	16
2.4. Uzaktan kontrol .....	17
2.4.1. Kablolu uzaktan kontrol .....	17
2.4.2. Kablosuz uzaktan kontrol .....	18
2.4.2.1. Radyo frekansı (RF) ile uzaktan kontrol .....	18
2.4.2.2. Kızıl ötesi ışınlarıyla uzaktan kontrol .....	18
2.4.2.3. Bluetooth teknolojisiyle uzaktan kontrol .....	18
2.4.2.4. GSM şebekeli uzaktan kontrol .....	19
2.5. Active server page, aktif sunucu sayfaları (ASP) .....	19
2.5.1. Bağlantı (client) ve sunucu (server) taraflı scripting .....	19
2.5.2. ASP 'nin kullanım alanları ve avantajları .....	20
2.5.3. ASP dosyalarının hazırlanması ve yayınlanması .....	22
2.5.4. Web sunum mantığı .....	22
2.6. 74LS164 8bit kaymalı kaydedici entegresi .....	26
2.7. MAX232 RS-232 dönüştürücü entegresi .....	28

BÖLÜM 3.	
SİSTEMİN GERÇEKLENMESİ .....	30
BÖLÜM 4.	
SONUÇLAR .....	42
BÖLÜM 5.	
TARTIŞMA VE ÖNERİLER .....	43
KAYNAKLAR .....	44
EKLER.....	45
ÖZGEÇMİŞ .....	75

## SİMGELER VE KISALTMALAR LİSTESİ

ASP	:Active Server Pages – Aktif Sunucu Sayfaları
CPU	:Central Processing Unit – Merkezi İşlem Birimi
DCE	:Data Circuit Terminating Equipment – Veri Devresi Sonlandırma Cihazı
DLL	:Dynamic Link Libary – Dinamik Bağlantı Kütüphaneleri
DNS	:Domain Name System – Alan Adı Sistemi
DTE	:Data Terminal Equipment – Veri Terminal Cihazı
EIA	:Electronics Industries Association
HTML	:Hypertext Markup Language – Üst Metin İşaretleme Dili
HTTP	:Hypertext Transfer Protocol – Üst Metin Transfer Protokolü
IIS	:Internet Information Server – İnternet Bilgi Sunucusu
IP	:Internet Protocol – İnternet Protokol
ISP	:Internet Service Provider – İnternet Servis Sağlayıcısı
LED	:Light Emitting Diode – Işık Yayan Diyot
MCU	:Microcontroller Unit – Mikrodenetleyici Ünitesi
PIC	:Peripheral Interface Controller – Çevresel Aygıt Denetleyici
RS232	:Seri Haberleşme Protokolü
$\Omega$	:Ohm

## ŞEKİLLER LİSTESİ

Şekil 2. 1. Sistemin genel yapısı .....	4
Şekil 2. 2. PIC 18F452 Mikrodenetleyicisinin bacak bağlantıları. ....	14
Şekil 2. 3. RS-232 konnektör çeşitleri (25 ve 9 uçlu konnektörler).....	16
Şekil 2. 4. İnternet servis sağlayıcısının yüklenmesi.....	23
Şekil 2. 5. İnternet Servis Sağlayıcısı ayarlarının yapılması.....	24
Şekil 2. 6. ASP programının giriş dizini olarak tanıtılması .....	25
Şekil 2. 7. Giriş dizininde gerekli ayarlamaların yapılması.....	25
Şekil 2. 8. 74LS164 Kaymalı kaydedici entegresinin bacak bağlantıları. ....	26
Şekil 2. 9. 74LS164 Kaymalı kaydedici entegresinin iç yapısı.....	27
Şekil 2. 10. a) MAX232 entegresinin bacak bağlantıları b) İç yapısı.....	28
Şekil 2. 11. PC seiporu ile PIC'in MAX232 ile birbirine bağlanması.....	29
Şekil 3. 1. Delphi arayüz programı bağlantı ayar penceresi .....	31
Şekil 3. 2. Delphi arayüz programının ana penceresi.....	31
Şekil 3.3. Delphi arayüz programının akış şeması .....	32
Şekil 3.4. ASP Programının şifre kontrol bölümü .....	33
Şekil 3. 5. ASP Programının genel görünümü .....	34
Şekil 3. 6. ASP diliyle yazılan programın akış şeması. ....	35
Şekil 3.7. Elektronik bilgilendirme panosunda kullanılacak karakterlerin tanımlanması .....	36
Şekil 3.8. Elektronik bilgilendirme panosunda kullanılan satır sürme devresi.....	37
Şekil 3.9. Elektronik bilgilendirme panosunun led sürücü devrelerle oluşturulması	38
Şekil 3. 10. Pic mikrodenetleyicili satır sürücü devre şeması.....	39
Şekil 3.11. Pic mikrodenetleyicili sürücü devrenin baskı devresi (üst görünüm).....	40
Şekil 3.12. Pic mikrodenetleyicili sürücü devrenin baskı devresi (alt görünüm). ....	40
Şekil 3.13. Satır sürme devresi (üst görünüm).....	41
Şekil 3.14. Satır sürme devresi (alt görünüm).....	41



## **TABLolar LİSTESİ**

Tablo 2.1. PIC 18F serisi mikrodenetleyicilerin bellek kapasiteleri.....	15
Tablo 2.2. 9 uçlu konnektör için uç bağlantısı.....	17
Tablo 2.3. 74LS164 8bit kaymalı kaydedici entegresinin fonksiyon tablosu.....	27

## ÖZET

Anahtar Kelimeler: Web tabanlı kontrol, elektronik bilgilendirme panosu, Mikrodenetleyiciler, PIC18F452

İçinde bulunduğumuz bilişim çağında, bilgi teknolojilerinin gösterdiği hızlı gelişme nedeniyle, “uzaktan kontrol” kavramı hayatımızın pek çok alanına hızla girmektedir. Pek çok uzaktan kontrol yönteminden biri de web üzerinden kontroldür. Web üzerinden kontrolde, kontrolü yapılacak sistem internet üzerinden kontrol edilmektedir.

Bu tez çalışmasında, elektronik bilgilendirme panosunun web üzerinden kontrolü gerçekleştirilmiştir. Böylece, mekan bağımlılığı ortadan kaldırılmış ve elektronik bilgilendirme panosuna erişim uzaktan kontrol ile hızlı, ekonomik ve güvenli bir şekilde sağlanmıştır. Panonun bu nedenle; eğitim kurumları, kamu binaları ve hastaneler gibi pek çok yerde kullanım avantajı sunması beklenmektedir.

Tez içerisinde PIC mikrodenetleyiciler, PIC 18F452 serisi mikrodenetleyici, RS-232 haberleşme protokolü, ASP programlama dili, Delphi programlama dili, PIC C programlama dili, uygulamada kullanılan devre elemanları hakkında genel bilgi verildikten sonra uygulamanın çalışması açıklanmıştır. Yazılan programların akış şemaları ve kaynak kodları, kullanılan devrelerin şemaları verilmiştir.

# **WEB BASED CONTROL OF AN ELECTRONIC NOTICE BOARD**

## **SUMMARY**

Key Words: Web based control, electronic notice board, microcontrollers, PIC18F452

In this information era, due to the fact that rapid progress of information Technologies, “remote control” concept contributes our lives rapidly. One of the many remote control techniques is a web based control. In the web based control, the system that will be controlled is controlled via the internet.

In this thesis, controlling an electronic notice board using via web has been realized. Thus, dependence on place is eliminated and it is provided a fast access to electronic notice board, economically and safely. Hence, it is expected that using the board in education establishments, public buildings, hospitals and so on will get advantage.

This thesis covers knowledge about PIC microcontrollers, PIC 18F452 series microcontrollers, RS-232 communication protocol, ASP programming language, Delphi programming language, PIC C programming language and datasheets of components that are used at implementation. Finally, the implementation is presented and flow chart and codes of written programmes and circuit schematics are given.

## **BÖLÜM 1. GİRİŞ**

Günümüzde bilgi teknolojilerinin gösterdiği hızlı gelişme nedeniyle, evlerimizde, işyerlerinde ve fabrikalarda kullandığımız elektronik cihazlar her geçen gün artmaktadır. Dolayısıyla yeni bir anlayışla bu cihazları birbirine ve bir ana kontrol sistemine bağlamak ve bu sistemleri de uzaktan kontrol edebilmek ihtiyacı doğmuştur. Bu sistemlerin hayata geçirilmesiyle fabrika ve benzer otomasyonların küçük bir modelini bilgisayar ortamında oluşturup, bu sistemlerin internet üzerinden kontrol edilebilmesi sağlanmıştır.

Bu tür yeni akıllı sistemler ev ve işyeri sahiplerine artan bir konfor ve kolaylıklar yanında; enerjinin verimli kullanılması, gerekli arıza ve uyarıların anında ve problemler başlamadan alınması; tehlikeli her durumun ve ne yapılması gerektiğinin programlanması, güvenli bir yaşam can ve mal emniyetini sağlar.

[1] 'de verilen "İnternet Tabanlı Ölçme Ve Kontrol" isimli yüksek lisans tezinde bir ortamın sıcaklık bilgisinin internet üzerinden izlenebilmesini ve sisteme internet üzerinden müdahale edilebilmesi işlemi gerçekleştirmiştir. Bu çalışmada amaç, fabrika ortamındaki bir sistemin tümünün veya herhangi bir parçasının internet üzerinden kontrolünün sağlanmasıdır. Saydam fabrika olarak isimlendirilen bu sistem bir fabrikayı yönetmede mekan bağımlılığını ortadan kaldırmaktadır. Saydam fabrika, Ethernet / IP üzerinden çalışan; otomasyon, üretim ve idari sistemleri arasında iletişim sağlayan, internet tabanlı bir otomasyon sistemi olarak tanımlanır. Bu sistem, yerel ağ veya internet üzerinden üretimin kontrol edilebilmesine olanak tanır. Bu çalışmada saydam fabrika kavramına örnek olarak bir ortamın sıcaklığının ON/ OFF kontrolü sağlanmıştır.

Bu uygulamada, "Web tabanlı elektronik bilgilendirme panosu" gerçekleştirilmiştir. Bunun gerçekleştirilme nedeni olarak, mekan bağımlılığının ortadan kaldırılarak,

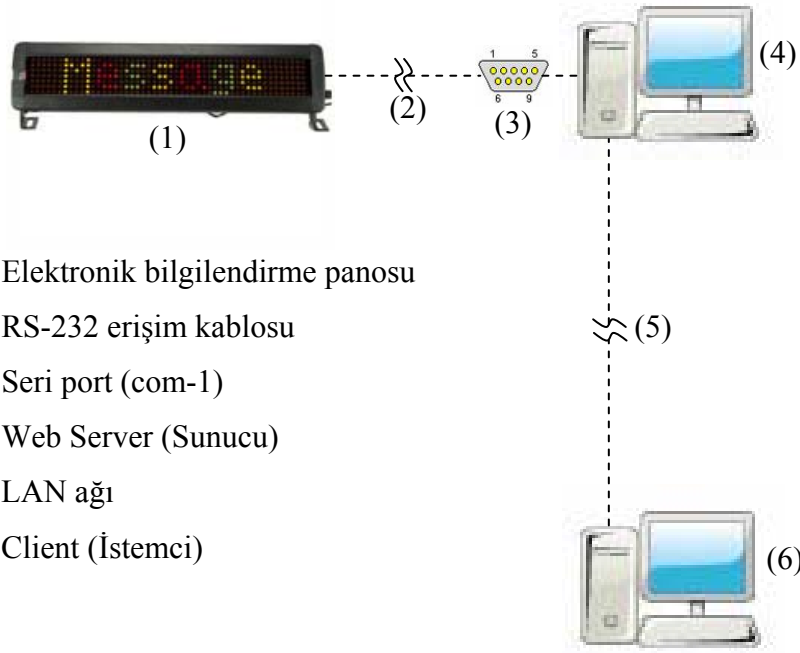
bilgilendirme mesajlarının elektronik bilgilendirme panosuna uzaktan kontrol ile anında, bir çok noktadan ve güvenli bir şekilde iletilmesi sağlanmıştır. Bu pano bilgisayar ağının olduğu; hastanelerde, bekleme salonlarında, eğitim kurumlarında ve benzeri pek çok yerlerde kullanılabilecek şekilde tasarlanmıştır.

Günümüzde; internet tabanlı veri toplama, laboratuvarların internet üzerinden kontrolü, internet kontrollü otomasyon sistemleri gibi internet tabanlı kontrol üzerine çok sayıda çalışma yapılmıştır. Yapmış olduğumuz çalışmada, bir noktada sabit bulunan panoya yetki verilen kişiler tarafından yapmak istedikleri duyuru, haber ve ilan bilgilerini ağ üzerine bağlı herhangi bir bilgisayar yardımıyla, doğrudan, zaman kaybetmeksizin şifreleme mekanizmasının da yardımıyla güvenli bir şekilde göndermeleri, ayrıca panoda bulunan bilgilere de erişebilmelerine olanak sağlanmaktadır.

Bu tez çalışması 5 bölümden meydana gelmektedir. İkinci bölümde; sistem tanımı yapıldıktan sonra öncelikle PIC Mikrodenetleyiciler hakkında genel bilgiler verilerek, elektronik bilgilendirme panosunda kullanılan PIC 18F452 Mikrodenetleyicisinin özellikleri verilmiştir. Ardından uygulamada kullanılan RS 232 haberleşme protokolü hakkında bilgi verilmiştir. Bir sonraki başlıkta ise; Active Server Page, Aktif Sunucu Sayfaları (ASP) ve Delphi programlara dilleri hakkında bilgiler verilmiştir. Bu bölümde son olarak bilgilendirme panosunda kullanılan 74 LS 164 ve MAX 232 entegrelerinin işlevleri açıklanmıştır. Üçüncü bölümde sistemin gerçekleşmesi başlığı altında tüm sistemin birlikte, senkronize çalışması açıklanmıştır. Dördüncü bölümde ise; proje değerlendirilmesine ilişkin bilgiler verilmiştir. Beşinci ve son bölümde çalışmanın eksik yönleri tartışılmış, ve getirilebilecek yeni öneriler üzerinde durulmuştur.

## **BÖLÜM 2. SİSTEM TANIMI**

Gerçekleştirilen bu çalışmada amaç, web üzerinden elektronik bilgilendirme panosunun kontrolünün sağlanmasıdır. Sistemde seri port ile bilgisayarla haberleşebilen ve içerisinde pic mikrodenetleyici kullanılan bir elektronik bilgilendirme panosu yer almaktadır. Panonun seri port ile bağlı bulunduğu Web sunucu bilgisayarda yer alan delphi programlama dili ile yazılmış arayüz yazılımı, belli zaman aralıkları ile (ön tanımlama = 30 saniye) aynı bilgisayarda yer alan veritabanındaki verileri elektronik bilgilendirme panosuna göndermektedir. Veritabanına istemci (client) bilgisayarlardan panonun bağlı bulunduğu web server olarak tanımlı bilgisayarın IP numarası kullanılarak, LAN ağı yada internet üzerinden erişilebilmekte ve veritabanındaki veriler okunup, değiştirilebilmektedir. Güvenlik için veritabanına erişimde şifre ile güvenlik kontrolü sağlanmaktadır. Bu şekilde ağ üzerinden veritabanına erişim sağlanabileceği gibi, ağ bağlantısı kullanılmaksızın aynı bilgisayarda da veritabanına erişilip aynı işlemler yapılabilmektedir. Sistemin genel görünümü Şekil 2.1 'de verilmiştir.



1. Elektronik bilgilendirme panosu
2. RS-232 erişim kablosu
3. Seri port (com-1)
4. Web Server (Sunucu)
5. LAN ağı
6. Client (İstemci)

Şekil 2. 1. Sistemin genel yapısı

## 2.1. PIC Mikrodenetleyiciler

### 2.1.1. PIC Mikrodenetleyicinin tercih edilmesinin nedenleri

Bilgisayar denetimi gerektiren bir uygulamayı geliştirirken seçilecek mikrodenetleyicinin ilk olarak tüm isteklerinizi yerine getirip getirmeyeceğine, daha sonra da maliyetinin düşüklüğüne bakılır. Ayrıca, yapılacak uygulamanın devresini kurmadan önce seçilen mikrodenetleyicinin desteklediği bir yazılım üzerinde simülasyonunu yapıp yapamayacağı da dikkate alınır.

Yukarıda saydığımız özellikleri göz önüne alındığında Microchip 'in ürettiği PIC 'lerin tercih edilme nedenleri şöyle sıralanabilir:

- a) Yazılımın Microchip 'ten veya internetten ücretsiz olarak elde edilebilmesi.
- b) Çok geniş bir kullanıcı kitlesinin bulunması.
- c) PIC 'lerin çok kolaylıkla ve ucuz olarak elde edilebilmesi.
- d) Basit bir devre ile programlanabilmesi

- e) Çok basit reset, clock sinyali ve güç devreleri gerektirmeleri.
- f) Çalışırken, devreden sökülmeden programlanabilmesi.
- g) PC seri portu ile bağlantı için yazılım ve donanım desteği vermesi.

PIC adını İngilizce 'deki Peripheral Interface Controller cümlesindeki kelimelerin baş harflerinden almış olan bir mikrodenetleyicidir. Bu cümleyi Türkçe 'ye, çevresel üniteleri denetleyici arabirim olarak çevrilir. PIC gerçekten de çevresel üniteler adı verilen lamba, motor, röle, ısı ve ışık sensörü gibi I/O elemanların denetimini çok hızlı olarak yapabilecek şekilde dizayn edilmiş bir chiptir. RISC mimarisi adı verilen bir yöntem kullanılarak üretildiklerinden bir PIC 'i programlamak için kullanılacak olan komutlar oldukça basit ve sayı olarak da azdır. 1980 'lerin başından itibaren uygulanan bir tasarım yöntemi olan RISC (Reduced Instruction Set Computer ) mimarisindeki temel düşünce, daha basit ve daha az komut kullanılmasıdır. Örnek olarak PIC16F84 mikrodenetleyicisi toplam 35 komut kullanılarak programlanabilmektedir [2].

### **2.1.2. Mikrodenetleyiciler hakkında genel bilgiler**

Neredeyse her mikroişlemci (CPU) üreticisinin ürettiği birkaç mikrodenetleyicisi bulunmaktadır. Bu denetleyicilerin mimarileri arasında çok küçük farklar olmasına rağmen aşağı yukarı aynı işlemleri yapabilmektedirler. Her firma ürettiği chip 'e bir isim ve özelliklerini birbirinden ayırmak için de parça numarası vermektedir. Örneğin Microchip ürettiklerine PIC adını, parça numarası olarak da 12C508, 16C84, 16F84, 16C711 gibi kodlamalar verir. Intel ise ürettiği mikrodenetleyicilere MCS-51 ailesi adını vermektedir. Genel olarak bu adla anılan mikrodenetleyici ailesine farklı özellikleri bulunan ürünleri birbirinden ayırt etmek için parça numarası olarak da 8031AH, 8051AH, 8052AH, 80C51FA gibi kodlamalar kullanılmaktadır.

Bir uygulamaya başlamadan önce hangi firmanın ürünü kullanılacağına, daha sonrada hangi numaralı denetleyicinin kullanılacağına karar vermek gerekir. Bunun



için mikrodenetleyici gerektiren uygulamada hangi özelliklerin olması gerektiği önceden bilinmesi gereklidir. Aşağıda bu özellikler sıralanmıştır.

- a) Programlanabilir dijital paralel giriş/çıkış.
- b) Programlanabilir analog giriş/çıkış.
- c) Seri giriş/çıkış (senkron, asenkron ve cihaz denetimi gibi).
- d) Motor veya servo kontrol için pals sinyali çıkışı.
- e) Harici giriş vasıtasıyla kesme.
- f) Timer vasıtasıyla kesme.
- g) Harici bellek arabirimi.
- h) Harici bus arabirimi (PCI ISA gibi).
- i) Dahili bellek tipi seçenekleri (ROM, EPROM, PROM, ve EEPROM).
- j) Dahili RAM seçeneği.
- k) Kayan nokta hesaplaması.

Daha da ayrıntıya girecek olursak bu listede sıralanacak özellikler uzayıp gidecektir. Microchip, 8-bit 'lik mikrodenetleyici ve EEPROM üreten bir Amerikan şirkettir. Arizona eyaletinde iki, Tayland ve Tayvan 'da da birer tane olmak üzere toplam dört fabrika ile kendi alanında dünyada söz sahibi olan bir chip üreticisidir [2].

### **2.1.3. Mikroşlemci ve mikrodenetleyicilerin farkı:**

Mikroşlemciler hafıza veya giriş çıkış entegresi gibi çevre elemanlar ile birlikte aritmetik işlemler yapabilen, karar verebilen entegrelerdir. Mikrodenetleyiciler ise mikroşlemcinin yanı sıra kendi program hafızası ve giriş çıkış özelliklerini içinde barındıran entegrelerdir. Bu özellikleri sayesinde yüklü bir programı uygulamaya dönüştürerek bir kontrol sistemi olarak kullanılabilir [2].

#### **2.1.4. PIC Programlama**

Bu bölümde PIC programlamak için nelerin gerektiğini inceleyeceğiz. PIC mikrodeneleyicilerin programlanmasını ve uygulamaların nasıl yapılacağını öğrenmek için, bilinmesi gerekenler aşağıda sıralanmıştır.

- a) Bilgisayar.
- b) PIC compiler programı(pic basic yada pic c).
- c) PIC programlayıcı donanımı.
- d) PIC programlayıcı yazılımı.
- e) PIC mikrodeneleyicisi.

##### **2.1.4.1. Bilgisayar**

Pic basic yada pic c dilinde program yazabilmek için bilgisayar gereklidir. Program yazılıp derlendikten sonra, pic mikrodeneleyiciye gönderilmesi gerekir. Program kodlarının PIC 'e yazdırma işlemi genellikle seri porta bağlanan pic programlayıcı devre aracılığıyla yapılır.

##### **2.1.4.2. PIC compiler programı (pic basic yada pic c)**

PIC Assembly dili kullanılarak program yazılabileceği gibi, yüksek seviyeli diller olan picbasic ve pic c dilleri kullanılarak ta yazılım geliştirilebilir. Pic c ve pic basic compiler programları basic yada c dilinde yazılan programların komutlarını PIC 'in anlayabileceği makine diline çevirerek pic mikrodeneleyiciye yüklenebilecek hale getirir. Yani ayrıca assembler programı kullanmaya gerek kalmamaktadır.

### **2.1.4.3. PIC programlayıcı devresi**

Makine diline çevrilmiş program kodlarını PC 'den alıp PIC microdenetleyicisine yazmak için bir elektronik devreye ihtiyaç vardır. Bu elektronik devre, birçok üretici tarafından piyasada satılmaktadır.

### **2.1.4.4. PIC programlayıcı yazılımı**

Pic c yada pic basic dilinde yazılarak derlenmiş yada assembler parogramı ile makine diline dönüştürülmüş Assembly programı kodlarının PIC 'e yazdırılmasında kullanılan bir programa gereksinim vardır. Programlayıcı yazılımları, PIC 'i programlamak için kullanılan elektronik karta bağımlıdır. Yani her programlayıcı yazılımı ile elinizde bulunan karta kod gönderemeyebilirsiniz. Genellikle programlama kartı üreticileri, ürettikleri karta uygun yazılımı da birlikte sunarlar. Örneğin IC-PROG yazılımı gibi.

### **2.1.5. PIC donanım özellikleri**

#### **2.1.5.1. PIC çeşitleri**

Microchip firması ürettiği mikrodenetleyicileri 4 farklı gruba ayırırlar. Bu gruplara isim verilirken kelime boyu (Word lenght) göz önüne alınmıştır. Bir CPU veya MCU 'nun dahili veri yolu uzunluğuna kelime boyu denir.

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

- a) PIC 16C5XX ailesi 12-bit kelime boyu,
- b) PIC 16CXXX ailesi 14-bit kelime boyu,
- c) PIC 17CXXX ailesi 16-bit kelime boyu,
- d) PIC 12CXXX ailesi 12-bit/14-bit kelime boyuna sahiptir.

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

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

### **2.1.5.2. PIC Mikrodenetleyicilerde kullanılan bellek çeşitleri**

Pic mikrodenetleyicilerin program bellekleri farklı özelliklerde üretilmektedir. Bunlar:

- a) Silinebilir ve programlanabilir bellek (Erasable Programmable Memory-EPROM)
- b) Elektriksel olarak silinebilir ve programlanabilir bellek (Electrically Erasable Programmable Memory -EEPROM). FLASH bellek olarakta adlandırılır.
- c) Sadece okunabilir bellek (Read –Only Memory -ROM)

EPROM bellek hücrelerine elektrik sinyali uygulanarak kayıt yapılır. EPROM üzerindeki enerji kesilse bile bu program bellekte kalır. Ancak silip yeniden başka bir program yazmak için ultraviyole ışını altında bir süre tutmak gerekir (20 dakika

kadar). Bu işlemler EPROM silici denilen özel aygıtlarla yapılır. EPROM bellekli PIC 'ler iki farklı ambalajlı olarak bulunmaktadır:

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

Seramik ambalajlı ve pencereci olan bellek içerisindeki programın silinmemesi için pencere üzerine ışık geçirmeyen bir bant yapıştırılır. Ultraviyole ışığı ile silinmesi istendiğinde bu pencere açılır ve silici aygıt içerisinde belirli bir süre bekletilir. Plastik ambalajlı EPROM 'ların ise programlandıktan sonra silinmesi mümkün değildir ve fiyatı silinebilen tipe göre oldukça ucuzdur. Silinemeyen tipe OTP (One Time Programmable –Bir defa programlanabilir) denilir.

EEPROM belleği bulunan bir PIC içerisine program yazmak için PIC programlayıcı vasıtasıyla elektriksel sinyal gönderilir. EEPROM üzerindeki enerji kesilse bile bu program bellekte kalır. Programı silmek veya farklı yeni bir program yazmak istendiğinde PIC programlayıcıdan elektriksel sinyal gönderilir. Bu tip belleğe sahip olan PIC 'ler genellikle uygulama geliştirme amacıyla kullanılırlar. Microchip bu tip belleği çoğu zaman FLASH bellek olarak da adlandırmaktadır. Fiyatları silinemeyen tiplere göre biraz pahalıdır. Bellek erişim hızları ise EPROM ve ROM 'lara göre daha yavaştır. PIC 16C84 ve PIC16F84 'ler bu tip program belleğine sahiptirler.

ROM program belleğine sahip PIC 'lerin programları fabrikasyon olarak yazılırlar. EPROM ve EEPROM (E<sup>2</sup>PROM olarak ta adlandırılırlar.)eşdeğerlerine nazaran fiyatları oldukça düşüktür. Ancak fiyatının düşüklüğünden dolayı gelen avantaj bazen çok pahalıya da mal olabilir. ROM bellekli PIC programlarının fabrikasyon olarak yazılması nedeniyle PIC 'in elde edilme süresi uzundur. Programda oluşabilecek bir hatanın PIC 'e program yazıldıktan sonra tespit edilmesi, eldeki tüm PIC 'lerin atılmasına neden olabilir. Bu tip PIC 'ler çok miktarda üretilecek bir ürünün maliyetini düşürmek amacıyla seçilir. Program hataları giderilemediği için uygulama geliştirmek için uygun değildir. Microchip, ROM program bellekli PIC 'lere parça numarası verirken "CR" (PIC16CR62, PIC16CR84 gibi) harfleri kullanır.

### **2.1.6. PIC Mikrodenetleyicilerinin özellikleri**

Şimdiye kadar verilen bilgilerin ışığında PIC mikrodenetleyicilerin genel özellikleri şu şekilde sıralanabilir:

#### **2.1.6.1. Güvenirlilik**

PIC komutları bellekte çok az yer kaplarlar. Dolayısıyla bu komutlar 12 veya 14 bitlik bir program bellek sözcüğüne sığarlar. Harvard mimarisi teknolojisi kullanılmayan mikrokontrolörler de yazılım programının veri kısmına atlama yaparak bu verilerin komut gibi çalıştırılmasını sağlamaktadır. Bu da büyük hatalara yol açmaktadır. PIC 'lerde bu durum engellenmiştir.

#### **2.1.6.2. Hız**

PIC oldukça hızlı bir mikrokontrolördür. Her bir komut döngüsü  $1\mu\text{sn}$  'dir. Örneğin 5 milyon komutluk bir programın 20MHz 'lik bir kristalle işletilmesi yalnız 1sn sürer. Bu süre 386SX33 hızının yaklaşık iki katıdır. Ayrıca RISC mimarisi işlemcisi olmasının hıza etkisi oldukça büyüktür.

#### **2.1.6.3. Komut seti**

PIC 'in 16C5X ailesinde bir yazılım yapmak için 33 komuta ihtiyaç duyarken, 16CXX araçları için bu sayı 35 'tir. PIC tarafından kullanılan komutların hepsi yazmaç (register) temellidir. Komutlar 16C5X ailesinde 12 bit, 16CXX ailesindeyse 14 bit uzunluğundadır. PIC 'te CALL, GOTO ve bit test eden BTFSZ ve INCFSSZ gibi komutlar dışında diğer komutlar 1 saykıl içerir. Belirtilen komutlar ise 2 saykıl içerir.

#### **2.1.6.4. Statik işlem**

PIC tamamıyla statik bir işlemcidir. Yani saat durdurulduğunda da tüm yazmaç içeriği korunur. Pratikte bunu tam olarak gerçekleştirebilmek mümkün değildir. PIC mikrodenetleyicisi programı işletilmediği zaman uyuma (sleep) moduna geçirilerek mikrodenetleyicinin çok düşük akım çekmesi sağlanır. PIC uyuma moduna geçirildiğinde, saat durur ve PIC uyuma işleminden önce hangi durumda olduğunu çeşitli bayraklarla ifade eder (elde bayrağı, zero bayrağı...v.b.). PIC uyuma modunda  $1\mu\text{A}$  'den küçük değerlerde akım çeker (Standby akımı).

#### **2.1.6.5. Sürme özelliği (sürücü kapasitesi)**

PIC yüksek çıktı kapasitesine sahiptir. Tek bacadan  $40\text{mA}$  akım çekilebilmekte ve entegre toplam olarak  $150\text{mA}$  akım akıtma kapasitesine sahiptir. Entegrenin  $4\text{MHz}$  osilatör frekansında çektiği akım çalışırken  $2\text{mA}$ , stand-by durumunda ise  $2\mu\text{A}$  kadardır.

#### **2.1.6.6. Seçenekler**

PIC ailesinde her türlü ihtiyaçların karşılanacağı çeşitli hız, sıcaklık, kılıf, I/O hatları, zamanlama (Timer) fonksiyonları, seri iletişim portları, A/D ve bellek kapasite seçenekleri bulunur.

#### **2.1.6.7. Çok yönlülük**

PIC çok yönlü bir mikrodenetleyicidir. Ürünün içinde, yer darlığı durumunda birkaç mantık kapısının yerini değiştirmek için düşük maliyetli bir çözüm bulunur.

### **2.1.6.8. Güvenlik**

PIC endüstride en üstünler arasında yer alan bir kod koruma özelliğine sahiptir. Koruma bitinin programlanmasından itibaren, program belleğinin içeriği, program kodunun yeniden yapılandırılmasına olanak verecek şekilde okunmaz.

### **2.1.6.9. Geliştirme**

PIC program geliştirme amacıyla programlanabilip tekrar silinebilme özelliğine sahiptir (EPROM, EEPROM). Aynı zamanda seri üretim amacıyla bir kere programlanabilir (OTP) özelliğine sahiptir.

### **2.1.6.10. Liste dosyası**

Assembler tarafından oluşturulan ve kaynak dosyadaki tüm komutları heksadesimal sistemdeki değerleri ve tasarımcının yazmış olduğu yorumlarıyla birlikte içeren bir dosyadır. Bir programı bug 'lardan arındırırken araştırılacak en yararlı dosya budur. Çünkü dosyayı izleyerek yazılımlarda neler olup bittiğini anlama şansı kaynak dosyasından daha fazladır. Dosya uzantısı .LST 'dir.

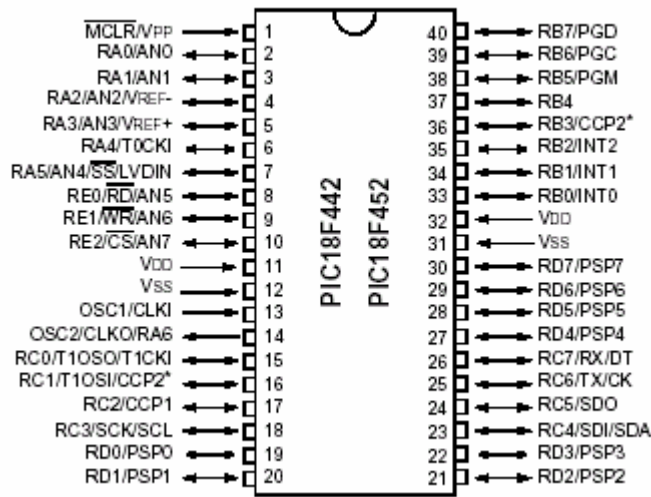
### **2.1.6.11. Diğer dosyalar**

Hata dosyası (Error file: uzantısı .ERR) hataların bir listesini içerir ancak bunların kaynağı hakkında hiçbir bilgi vermez. Uzantısı .COD olan dosyalar emülatör tarafından kullanılırlar.



## 2.2 PIC 18F452 Mikrodenetleyicisinin özellikleri

Elektronik bilgilendirme panosunda yer alan sürücü devrede PIC 18F452 mikrodenetleyici kullanılmıştır. PIC 18F452 mikrodenetleyici, 28/40 pinli, yüksek performanslı, arttırılmış flaş bellekli, 10bit Analog/Dijital çevirici içeren gelişmiş bir modeldir. Şekil 2.2 'de PIC18F452 mikrodenetleyicisinin bacak bağlantıları verilmiştir.



Şekil 2. 2. PIC 18F452 Mikrodenetleyicisinin bacak bağlantıları.

18F452 Mikrodenetleyicisinin teknik özellikleri aşağıda verilmiştir.

### 2.2.1. Yüksek performanslı RISC mimarili CPU

- C derleyicilere uyumlu mimari ve komut seti.
- Komut seti PIC16 ve PIC17 serileri ile uyumludur.
- Doğrudan programlanabilir 32 KByte program belleğine sahiptir.
- Doğrudan adreslenebilir 1.5 KByte data belleğine sahiptir.
- Saniyede 10 milyon işlem yapma kabiliyetine sahiptir.

Tablo 2.1. PIC 18F serisi mikrodenetleyicilerin bellek kapasiteleri.

Mikrodenetleyici	Çip içi Program Hafızası	Çip içi RAM (Byte)	Data EEPROM (Byte)
	FLASH (Byte)		
PIC18F242	16K	768	256
PIC18F252	32K	1536	256
PIC18F442	16K	768	256
PIC18F452	32K	1536	256

### 2.2.2. Çevresel özellikler

- a) Yüksek sink/kaynak akımı (25mA).
- b) 3 adet harici kesme ucu.
- c) Timer 0 Modülü: 8-bit/16-bit zamanlayıcı/sayıcı.
- c) Timer 1 Modülü: 16-bit zamanlayıcı/sayıcı.
- d) Timer 2 Modülü: 8-bit zamanlayıcı/sayıcı.
- e) Timer 3 Modülü: 16-bit zamanlayıcı/sayıcı.
- f) İkincil osilatör clock özelliği.
- g) 2 adet Yakalama/Karşılaştırma/PWM modülü(CCP-Capture/Compare/PWM).
- h) Senkronize seri port modülü(MSSP).
- i) Adreslenebilir USART modülü.
- j) Paralel ikinci port modülü (PSP).

### 2.2.3. Analog özellikler

- a) 10-bit Analog/Dijital dönüştürücü modülü.
- b) Programlanabilir düşük voltaj algılama (PLVD).

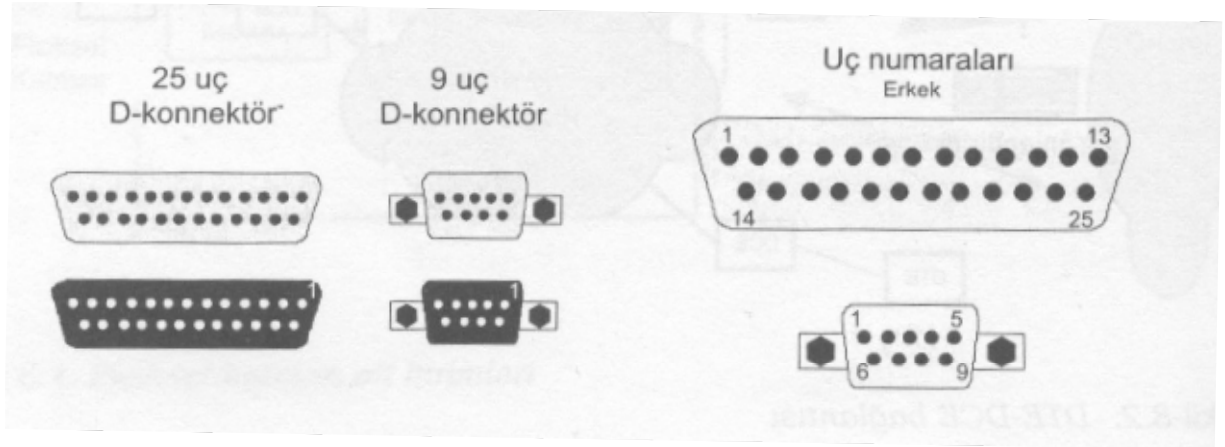
### 2.2.4. Diğer özellikler

- 100000 kez okunup yazılabilen FLASH program belleği.
- 1000000 kez okunup/yazılabilen EEPROM veri belleği.
- 40 yıldan uzun süre silinmeyen FLASH/ Veri EEPROM 'u.
- Watch Dog Timer (WDT), Power-on-Reset (POR), Power-up-Timer (PWRT), ve Oscillator Start-up Timer (OST) özellikleri.
- Uyku modunda az enerji tüketimi.
- Düşük güç tüketimli bellek teknolojisi.
- Tamamen statik dizayn.
- Geniş işlem voltajı aralığı (2.0V ile 5.5V arası).
- Düşük güç tüketimi [7].

### 2.3. RS-232/V.24 standardı

EIA 'nın bilgisayar, terminal ve modem arabağdaşımı için önerdiği bir standarttır. RS-232-C ve RS-232-D gibi çeşitleri vardır.

RS-232 standardında Şekil 2.3 'de görüldüğü gibi 25 veya 9 uçlu D konnektörü kullanılır. RS-232 dengesiz bir arabağdaşım sunar; yani, arabağdaşım devrelerinde tüm işaretler aynı toprak referans alınarak üretilir. Bu özellik, RS-232 arabağdaşımında önemli ölçüde gürültü doğmasına yol açar ve işaretleşme hızı ile işaretleşme uzaklığını sınırlar (20 m için yaklaşık 20 Kbps) [3].



Şekil 2. 3. RS-232 konnektör çeşitleri (25 ve 9 uçlu konnektörler)

Uygulamada genel olarak DTE ve DCE cihazları üzerindeki konnektörler erkek, bu cihazları bağlayan kablo üzerindeki konnektörler ise dişi tipte seçilir.

İşaret düzeyleri aşağıda verilmiştir:

Lojik 1 (mark) ----- -3V ile -15V arası

Lojik 0 (space) ----- +3V ile +15V arası

Buna göre -3V , +3V aralığı geçiş bölgesidir, NRZ-L kodlaması kullanılır.

Tablo 2.2. 9 uçlu konnektör için uç bağlantısı

Uç No:	Uç Adı
1	CD-Carrier Detect
2	RxD- Receive Data
3	TxD- Transmit Data
4	DTR- Data Terminal Ready
5	İşaret Toprağı-Signal Ground
6	DSR- Data Set Ready
7	RTS- Request To Send
8	CTS- Clear To Send
9	Zil işareti – Ring Indicator

## 2.4. Uzaktan kontrol

Herhangi bir makine veya cihazı yanından değil de belirli bir mesafe uzaklıktan çalıştırma, durdurma veya gerekirse programlayabilmek için uygulanmakta olan kontrol sistemidir.

### 2.4.1. Kablolü uzaktan kontrol

Bu kontrol sisteminde her türlü işlem yapılabilir. Ancak kontrol için uzaklık mesafesi kablo mesafesi ile sınırlıdır. Kablo karmaşasında bir başka sakıncasıdır.

Buna ilaveten, diđer uzaktan kontrol sistemlerine gre verilerin (sinyallerin) karışması gibi bir durum söz konusu değildir. Kablolu uzaktan kontrolün direk kablolu uzaktan kontrol ve internet üzerinden kablolu uzaktan kontrol olmak üzere iki çeşidi vardır. Hem kablolu uzaktan kontrol ve hemde mekan sınırlaması ve kablo boyu sınırlaması olmadan yapılabilecek en ideal kontrol sistemi internet üzerinden kablolu uzaktan kontroldür [4]. Bu nedenle tasarımı yapılan bu çalışmanın kontrolü için internet üzerinden kablolu uzaktan kontrol seçilmiştir.

#### **2.4.2. Kablosuz uzaktan kontrol**

##### **2.4.2.1. Radyo frekansı (RF) ile uzaktan kontrol**

Radyo frekansının alıcısı ve vericisi kullanılarak yapılan bir kontrol sistemidir. Kontrol için ilave alt yapıya ihtiyaç duymadan sadece radyo frekansı alıcı ve vericisi ile kontrol imkanı sağlar. Bu kontrol sisteminde frekans alıcı ve vericisinin yayın gücü kontrol mesafesini belirler. Ancak bazı frekansların verilerin (sinyallerin) kaybolmasına veya yanlış gönderilmesine sebep olabilir. Bu kontrol sisteminde programlama yapmak oldukça zordur. Sadece çalıştırmak ve durdurmak için kullanılmak üzere uygun bir kontrol sistemidir.

##### **2.4.2.2. Kızıl ötesi ışınlarıyla uzaktan kontrol**

Radyo frekansına çok benzeyen bir kontrol sistemidir. Ancak bu kontrol sisteminde alıcı ve verici mutlaka birbirlerini görmeleri gerekmektedir.

##### **2.4.2.3. Bluetooth teknolojisiyle uzaktan kontrol**

Bu kontrol sistemi de kızıl ötesi ışınlarıyla uzaktan kontrol sistemleriyle aynı yapıya sahip olup daha uzak mesafeye daha güvenli veriler gönderilebilir.

#### **2.4.2.4. GSM şebekeli uzaktan kontrol**

Bu kontrol sisteminde kontrol edilecek sistemin üzerinde bir GSM kartı bulunan PLC kullanılır. Her hangi bir GSM hattından bu PLC kontrol edilerek sistem kontrol edilmiş olur. GSM şebekesinin veri gönderme özellikleriyle sınırlı bir kontrol sistemidir.

#### **2.5. Active server page, aktif sunucu sayfaları (ASP)**

ASP Active Sever Pages ifadesinin kısaltılmış halidir. Aktif sunucu sayfaları şeklinde Türkçeye çevrilebilir. ASP sunucu taraflı bir scripting dilidir ve web üzerinde kullanılır [5].

##### **2.5.1. Bağlantı (client) ve sunucu (server) taraflı scripting**

Bir scripting dili, web sayfalarında, HTML 'nin yetersiz kaldığı durumlarda kullanılır. Bir anlamda web sayfalarına gömülü programlar yazmamızı sağlar.

Temel olarak iki tür scripting vardır: Client (bağlantı) taraflı ve server (sunucu) taraflı. Client taraflı scriptler, web browser'ı nesne olarak ele alırlar ve alt nesnelere ile birlikte, bir browser'da (web sayfalarında) programlama adına ne gerekiyor ise, onun yapılmasına imkan tanırlar.

Yeni pencereler açılması, mesajların çıkması, web sayfasının renginin değiştirilmesi v.b. Web sayfalarında şekilsel anlamda her şeyi, bağlantı taraflı scriptler yapabilir. Bu scriptler JavaScript, VBScript, Jscript 'tir.

En yaygın kullanılan ve bütün browserlar tarafından işletilebilen scripting dili JavaScript 'tir. Client taraflı scriptler, kullanıcının bilgisayarında çalışırlar. Sunucu, istek geldiğinde, web sayfasını, içindeki scriptler ile birlikte ziyaretçiye gönderir ve görevini tamamlar. Artık ziyaretçi ile hiçbir bağlantısı kalmamıştır. Scriptler ziyaretçinin bilgisayarında işletilir ve sonuçları yine ziyaretçiye gösterilir. Bağlantı taraflı scriptleri, HTML tagları içine gömülmüştür ve bunlar sayfa kodlarına bakılınca gözükürler.

Sunucu taraflı script, farklı olarak sunucu üzerinde çalışır. Özel tag işaretleri arasında yazılan scriptler, sunucu üzerinde işletilir ve sonuçları ziyaretçiye iletilir. Sunucu üzerinde çalışan scriptler, ziyaretçinin bilgisayarına bilgiyi, HTML olarak kodlanmış şekilde gönderir. Ziyaretçi sayfasının kodlarına baktığında, sunucu tarafından gönderilen, işlenmiş verileri HTML olarak görür. Sayfasının kodları içinde, hiçbir şekilde sunucu taraflı bir script ifadesi göremez. Sunucu taraflı scripting, mantık olarak, kullanıcı taraflı scriptin yapamadığı işler için kullanılır. Kullanıcı taraflı script ile yapabileceğimiz hiçbir şeyi, sunucu taraflı script ile yapamamalıyız. Çünkü bu, hem sunucuyu gereksiz yere meşgul eder hem de hızımızı düşürür [5].

ASP sunucu taraflı, en bilinen ve en çok kullanılan scripting dilidir. ASP Microsoft firmasının geliştirdiği bir teknolojidir ve yine Microsoft 'un geliştirdiği ASP destekli sunucular üzerinde çalışır. ASP sayfalarını çalıştırmanın birinci koşulu, ASP 'yi destekleyen bir sunucu ile çalışmaktır.

ASP için WEB ortamının tamamı bir nesnedir. Sunucu, ziyaretçinin bilgisayarını, sunucu tarafında çalışan programlar, iletim ortamı v.b. ASP için birer nesnedir. Bunun anlamı, bu ortam içindeki her şeye müdahale edebileceğimiz ve programlayabileceğimizdir. ASP için sunucu, sunucuda çalıştırılan programlar, ziyaretçiler bir bütündür. Sunucu taraflı bir ortamda çalışması sebebi ile, bağlantı sağlandıktan itibaren karşılıklı etkileşimli (interaktif) bir ortam sağlar. Kullanıcılarını tanıyabilir, gönderdikleri verileri anında değerlendirip cevap verebilir, ziyaretçilerin bilgilerini saklayabilir ve bir dahaki sefer ziyaretçiye iletebilir.

Kullanıcı taraflı scriptler, sayfaya şekilsel anlamda değer katarken, ASP içerik ve karşılıklı etkileşim anlamında sayfalara değer katar.

### **2.5.2. ASP 'nin kullanım alanları ve avantajları**

a) ASP dinamik Web sayfaları oluşturmak için kullanılır. Web sayfalarını canlandırarak bir tekniktir. Web sayfalarına işlem işleme yeteneği sağlar.

b) ASP Web Programcılarında HTML, script dili ve kullanıcıdan bağımsız veritabanı işlemlerini rahatlıkla kullanabilme olanağı sağlar. Bu özgürlük ve platform bağımsızlığı, icra gören tüm işlemlerin sunucu tarafında halledilerek sadece sonuçları üretmesi ile sağlanmaktadır. Sonuç salt HTML kodunda üretildiği için browserlar üzerinde tanımsız komut gibi bir problemle karşılaşmaz.

c) ASP kendi içinde Javascript, VB Script, Jscript, Perl Script gibi geniş bir yelpazedeki ham kodların kullanılmasına olanak tanır. Hatta Windows Scripting host ile uyumlu kalmak şartıyla oluşturulan her yeni script dil, Web server 'a tanırılarak ASP üzerinde kullanılabilir.

d) ASP sunucu tarafında uygulama çalıştırmanıza imkan sağlar. Hazırlanan uygulamaları çalıştıran komutlar sadece sunucu tarafında kalır. İstemci sadece sonuçları görebilir, yani kullanıcılar, istedikleri sayfayı oluşturan komutları görüntüleyemez. Bu yönüyle ASP web programcılarında güvenirlilik ve özgünlük sağlamaktadır.

e) Uygulama ile ilişkilendirilmiş bir database üzerindeki veriler ASP ile istenilen yayınlama stratejisine göre Web sayfalarına aktarılır ve bilginin internet üzerinden güncellenmesi otomatik hale dönüşebilir. Bu özellik ASP 'nin hazırlanan bir formla veri giriş kontrol arayüzü olduğu gibi aynı zamanda hazırlanan Paket programlar ile eş zamanlı çalışacak bir alternatif de sunmaktadır.

f) ASP ile hazırlanan projeler kaynakların yönetimi, kontrolü ve uzaktan sistem yönetimine imkan verir. Şirketler arası bilgi paylaşımı için ideal bir ortam oluşturur. Bugün web programcılarını geliştirdikleri uygulamaları birçok avantajı göz önünde bulundurarak ASP ile tasarlamaktadırlar. İnternet servis sağlayıcılarının çoğu ASP uygulamalarını Host edebilecek şekilde sunucularını düzenlemişlerdir. Birçok yazılım firması ASP 'ye özgün editörler tasarlamaktadır. Geliştirilen paket programların veri giriş arayüzleri alternatifli olarak browser üzerinden de sağlanmaya başlandı. Artık üzerinde aktif bilgi sunumlarının gerçekleştiği, anketlerin çeşitli fonksiyonellikte formların, aritmetik ve lojik hesaplamaların yapıldığı portal sitelerin, üyelik sistemlerinin, forumların, online alışveriş yapılabilen sayfaların sıkça karşımıza çıktığını, firmaların hızla İtranetlerinde yer alan web siteleri üzerinde



firma işleyişini sağlayan uygulamalara yöneldiklerini gözlüyoruz. Elbette ki bu fonksiyonelliğin web sayfalarına yansımada ASP 'nin rolü göz ardı edilemez [1].

### **2.5.3. ASP dosyalarının hazırlanması ve yayınlanması**

ASP ile çalışma yapılırken, bilgisayar üzerinde iki platform kullanılır.

- a) ASP dosyalarının içeriklerinin hazırlanması ve düzenlenmesi için editör
- b) Kodların kontrol edilmesi ve ASP dosyalarının yayınlanması için Web sunucusu

Yukarıda belirtilen ilk işlem için herhangi bir text editörü kullanılabilir. İkinci aşamada ise hazırlanan dosyadaki kodların kontrolü ve yayınlama işlemi yer alır. Windows XP işletim sisteminde yer alan IIS bu işlem için kullanılır.

### **2.5.4. Web sunum mantığı**

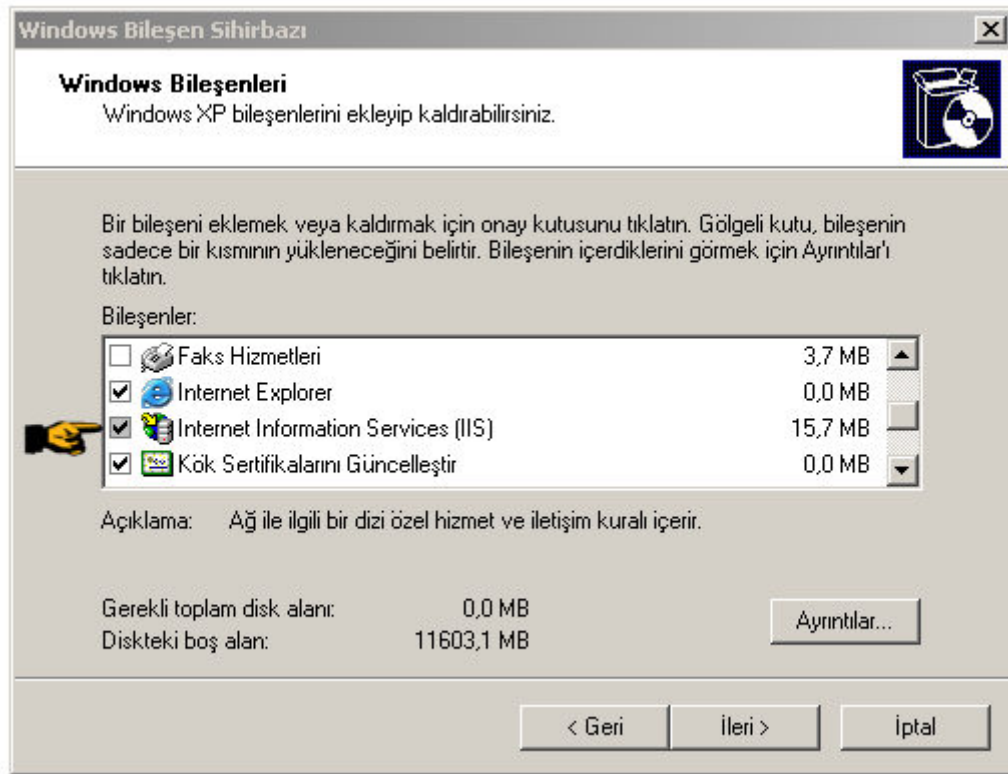
İnternet Information Server, Microsoft firmasının Web sitelerinin yayınlanma ve kontrol edilme konusunda hazırladığı bir uygulamadır. Microsoft XP işletim sisteminin profesyonel sürümünde IIS tanımlıdır.

Web browser işletim sistemi ne olursa olsun Web sitesine müracaat ederken öncelikle Web Server ile bağlantı kurar. Bu bağlantı için kullanılan işletim sistemi veya internet bağlantı şeklimiz fark etmeksizin sunucu ziyaretçinin bilgisayarına HTML kodlarının yer aldığı text dosyayı ve bu dosya içerisinde belirtilmiş nesnelere gönderir. Bu dosyada yer alan HTML komutlarına göre Browser kendisine ulaştırılan text ve çoklu ortam nesnelere görüntüler. Basit anlamda Web iletişimi bu şekilde çalışır.

Bir istemci tarafından bir HTML sayfası çağrıldığında HTML sayfası doğrudan istemciye gönderilir. Tarayıcı tarafından bir ASP sayfası çağrıldığında ise IIS bu isteği ASP motoruna iletir. ASP motoru ASP dosyasını satır satır okur ve içindeki scriptleri çalıştırır. Bu işlemler sonucunda da ASP sayfası düz HTML olarak istemciye gönderilir.

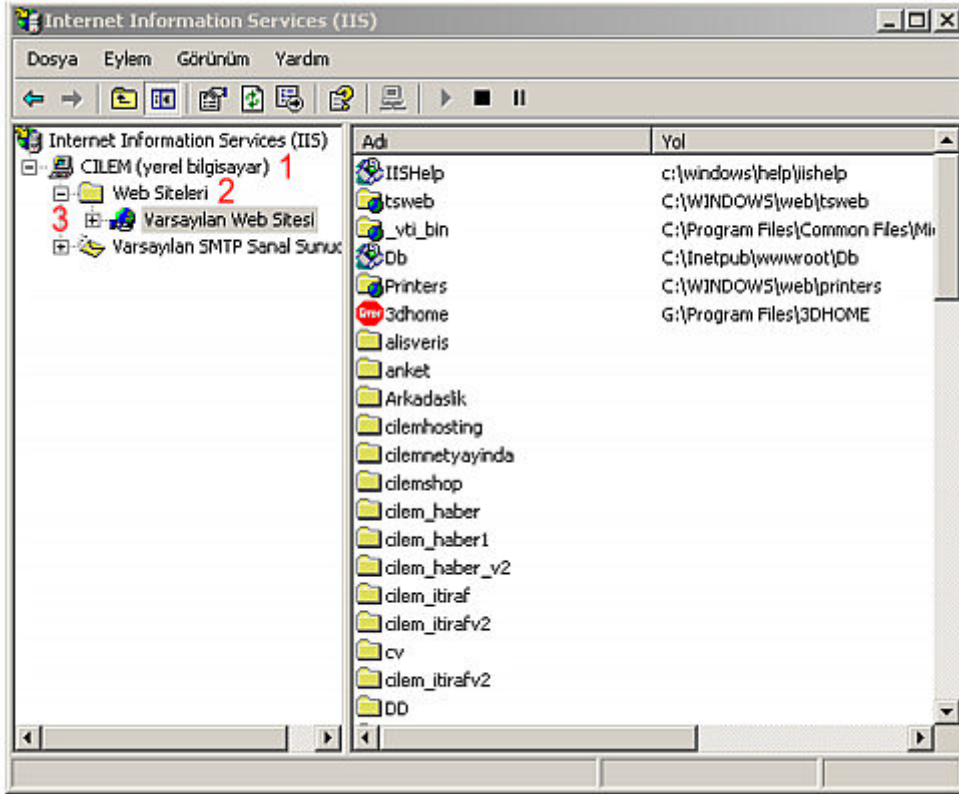
ASP bir sunucu taraflı script olduğu için Web sayfası kullanıcının ekranında görüntülenmeden işlenir, gerekli işlemler yapıldıktan sonra HTML kodları oluşturulur ve kullanıcının bilgisayarına gönderilir. Yani eğer sayfada hepsi varsa önce sunucu taraflı scriptler (ASP, CGI gibi) sonra istemci taraflı scriptler ve en son bunların işlenmesinden oluşan HTML kodları üretildikten sonra yazılan diğer HTML kodları birlikte işletilir [6].

Windows XP Service Pack 2 işletim sistemi yüklü bir bilgisayara İnternet Servis Sağlayıcısı (İnternet Information Server) kurulmak isteniyorsa, öncelikle Windows XP işletim cd'si ile windows bileşenlerini ekle/kaldır (program ekle/kaldır menüsünde yer almaktadır) bölümünden internet servis sağlayıcısı seçilerek yüklenir (Şekil 2.4.).



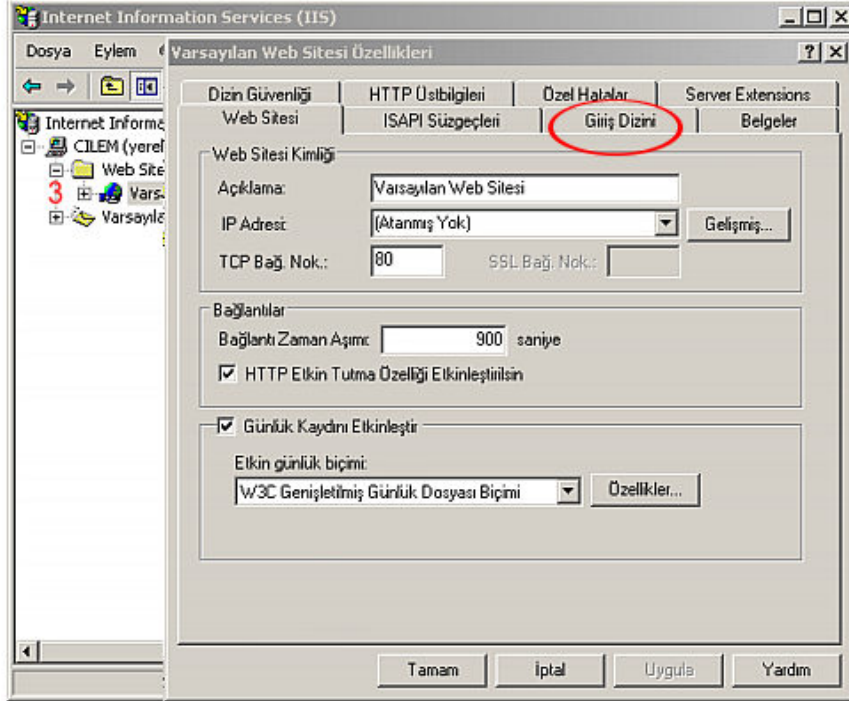
Şekil 2. 4. İnternet sevis sağlayıcısının yüklenmesi

İnternet servis sağlayıcısının yüklenmesiyle c: sürücüsünün içinde inetpub adlı bir klasör otomatik olarak oluşturulur. Tasarımı yapılan asp programı, c:\inetpub\wwwroot\aspdelphi dizinine kopyalanır. Ardından denetim masası\yönetimsel araçlar\internet servis sağlayıcısı bölümü çalıştırılır. Şekil 2.5. 'te verilen ekran karşımıza gelir.

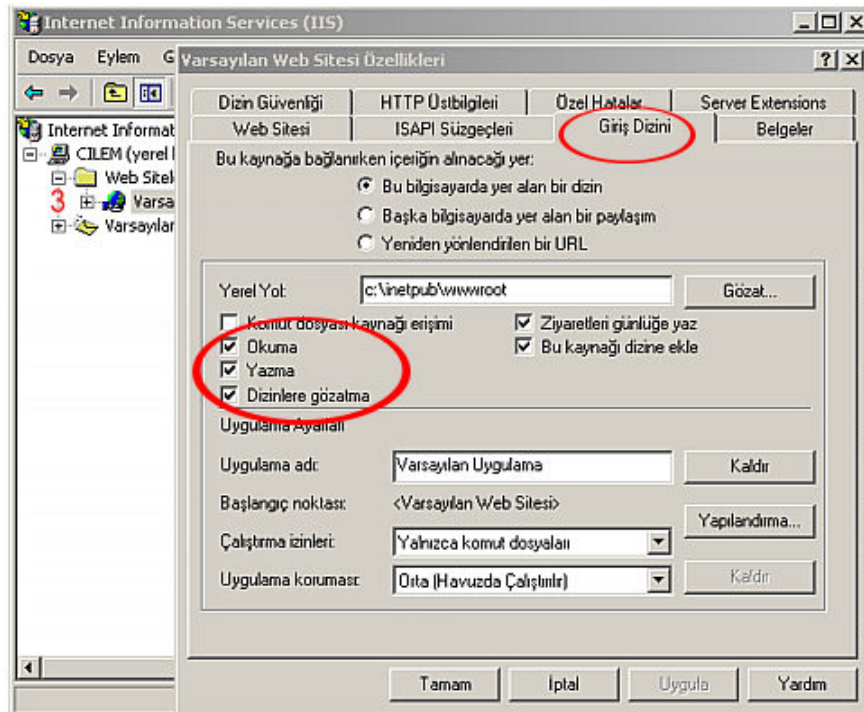


Şekil 2. 5. İnternet Servis Sağlayıcısı ayarlarının yapılması.

Şekilde belirtilen varsayılan web sitesinin özellikleri seçilir ve Şekil 2.6. 'da gösterilen giriş dizini bölümüne girilerek, daha önce aspdelphi dizinine kopyalanmış olan web sitesi giriş dizini olarak tanımlanır. Böylece ağ üzerinde webserver olarak tanımladığımız bilgisayara erişildiğinde (bilgisayar adı veya IP numarası internet Explorer sayfası adres çubuğuna yazılarak erişim sağlanır) tasarımı yapılmış olan asp programı çalıştırılmış olur.



Şekil 2. 6. ASP programının giriş dizini olarak tanıtılması



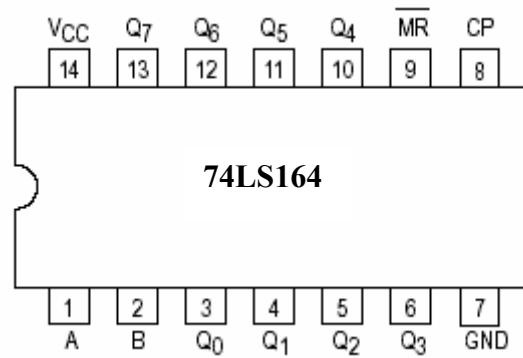
Şekil 2. 7. Giriş dizininde gerekli ayarlamaların yapılması

Son olarak, Şekil 2.7. 'de görüldüğü gibi okuma, yazma ve dizinlere gözetme yetkisi verilmelidir.

Webserver üzerinde ayrıca veritabanı dosyalarının tanıtılması gerekir. Bu işlem, denetim masası\yönetimsel araçlar\Veri kaynakları (ODBC) bölümüne girilerek yapılır. Burada yer alan kullanıcı DSN ve sistemDSN bölümlerine vt isimli veritabanı dosyası eklenir. Bu işlem yapılmaz ise Delphi arayüz programı veritabanına erişim hatası verir ve çalışmaz.

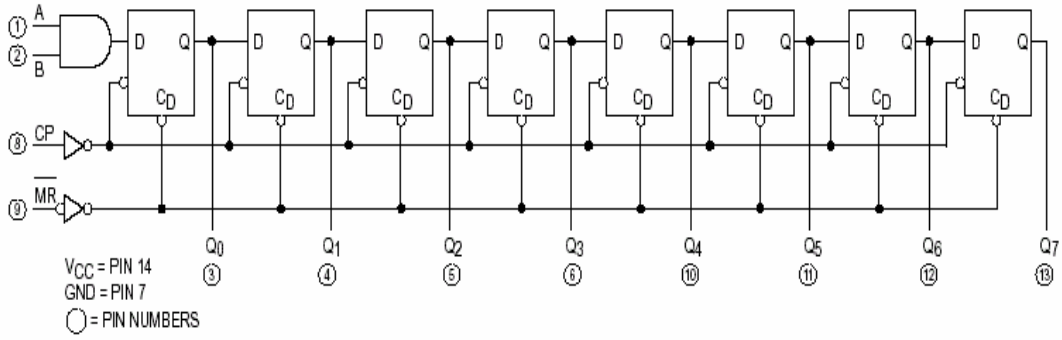
## 2.6. 74LS164 8bit kaymalı kaydedici entegresi

Elektronik bilgilendirme panosunun led sürücü devrelerinde yazının oluşturulmasında, 8 bitlik 74LS164 kaymalı kaydedici entegreleri kullanılmıştır. Bu entegrenin özellikleri verildikten sonra, elektronik bilgilendirme panosunda karakterlerin nasıl oluşturulduğu daha iyi anlaşılacaktır.



Şekil 2. 8. 74LS164 Kaymalı kaydedici entegresinin bacak bağlantıları.

Şekil 2.8 'de 74LS164 kaymalı kaydedici entegresinin bacak bağlantısı verilmiştir. Entegre, 8 bit çıkışa sahiptir. A ve B ile gösterilen 1 ve 2 nolu pinlerden bilgi girişi yapılmaktadır. Bu pinler, entegre içinde ve kapısı ile birleştirilmektedir, biz sadece bu pinlerden 1 tanesini kullanacağız. İstenilirse, pinlerden diğeri giriş kontrol ucu olarak kullanılabilir. 9 nolu pin, reset ucudur. 8 nolu pin ise clock pals ucudur. Entegrenin iç yapısı, Şekil 2.5 'te verilmiştir.



Şekil 2. 9. 74LS164 Kaymalı kaydedici entegresinin iç yapısı.

Entegrenin fonksiyon tablosu Tablo 2.3 'te verilmiştir.

Tablo 2.3. 74LS164 8bit kaymalı kaydedici entegresinin fonksiyon tablosu.

İŞLEM MODU	GİRİŞLER			ÇIKIŞLAR	
	$\overline{MR}$	A	B	$Q_0$	$Q_1-Q_7$
Reset (Clear)	L	X	X	L	L-L
Kayma	H	L	L	L	$Q_0-Q_6$
	H	L	H	L	$Q_0-Q_6$
	H	H	L	L	$Q_0-Q_6$
	H	H	H	H	$Q_0-Q_6$

Açıklama:

L :Düşük Voltaj seviyesi.

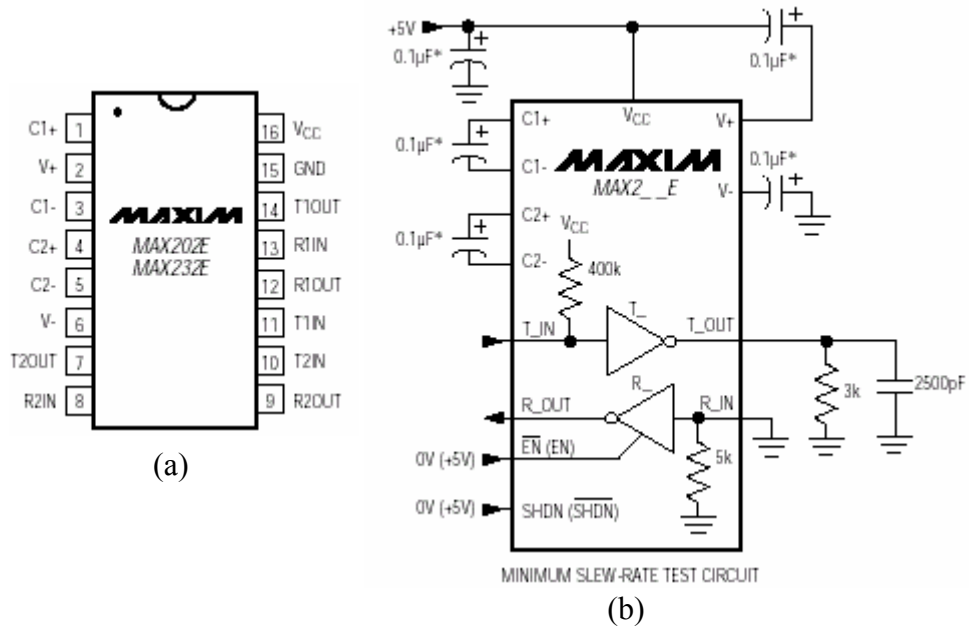
H :Yüksek Voltaj seviyesi.

X :Herhangi bir değer (lojik 0 veya 1).

$Q_n$  Entegrenin n. Kayma çıkışı.

## 2.7. MAX232 RS-232 dönüştürücü entegresi

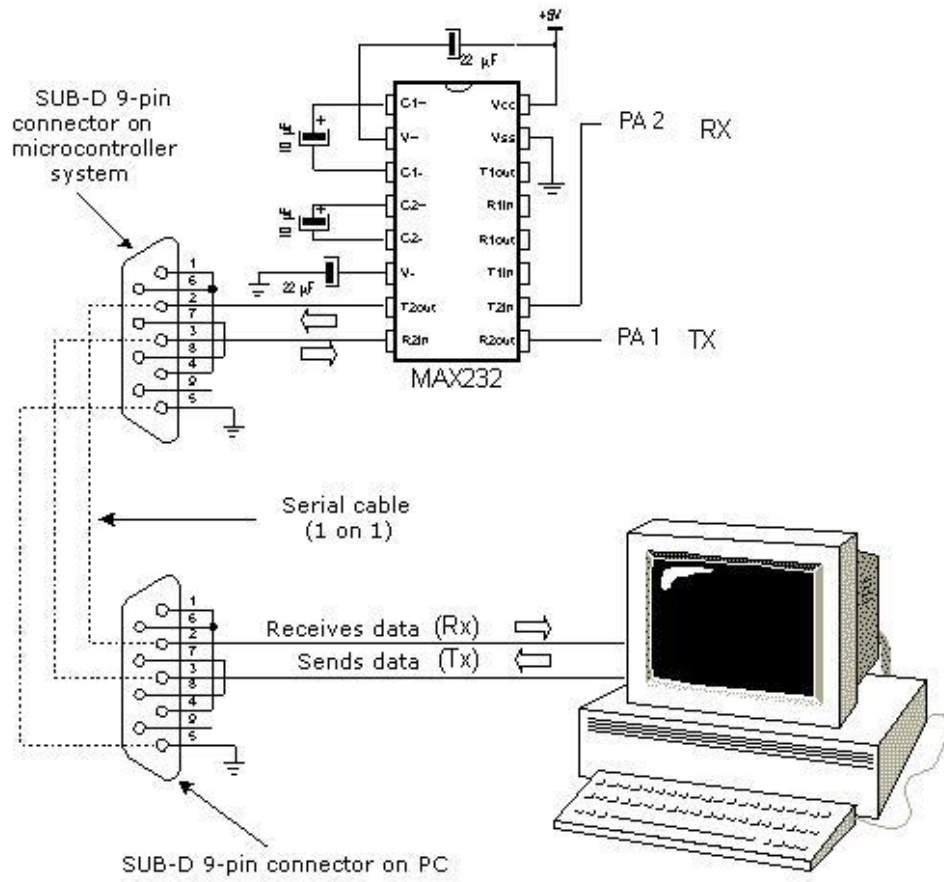
Bilgisayar seri portlarının (comport) gerilim seviyeleri, RS 232 standardından farklıdır. Mikrodenetleyici ile bilgisayar seri port üzerinden RS 232 standardında haberleşirken, gerilim seviyelerindeki bu farklılığı ortadan kaldırmak ve uygunlaştırmak için Maxim firmasının üretmiş olduğu MAX232 entegresi kullanılır. Şekil 2.10 'da MAX232 entegresinin bacak bağlantıları ve iç yapısı görülmektedir [8].



Şekil 2. 10. a) MAX232 entegresinin bacak bağlantıları b) İç yapısı

MAX232 entegresinin içinde tamponlanmış değil kapıları bulunmaktadır.

Şekil 2.11 'de ise PC seiporu ile PIC Mikrodenetleyicinin MAX232 entegresi kullanılarak ile birbirine bağlanması görülmektedir.



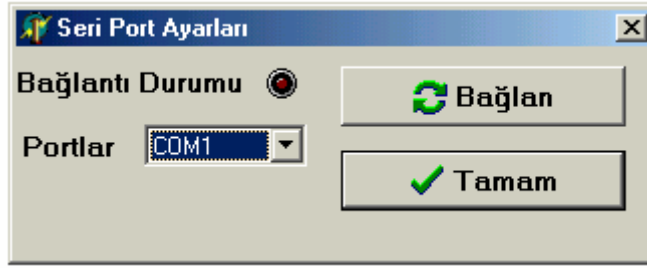
Şekil 2. 11. PC seiporu ile PIC'in MAX232 ile birbirine bağlanması



### **BÖLÜM 3. SİSTEMİN GERÇEKLENMESİ**

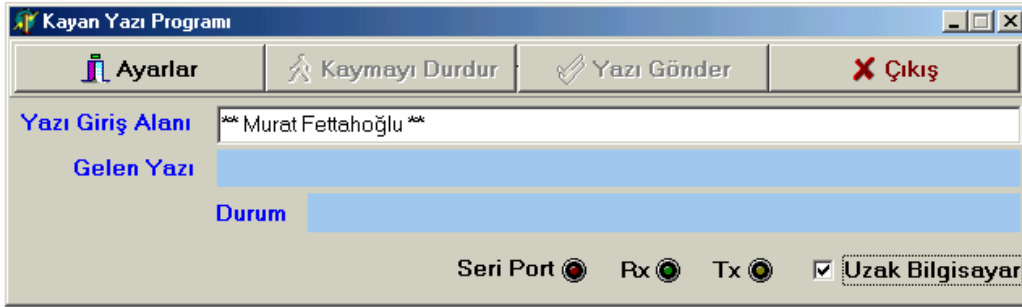
Sistemin çalışabilmesi için ilk olarak, elektronik bilgilendirme panosunun bağlı olduğu bilgisayarın web server olarak tanımlanması gerekmektedir. Web server olarak tanımlama işlemi gerçekleştirildikten sonra ASP kaynak kodu ile yazılmış olan web sayfası arayüzünün ve ilgili veritabanı dosyalarının c:\inetpub\wwwroot\aspdelphi dizini içerisine kopyalanması gerekmektedir.

Delphi programlama diliyle yazılmış olan program ise, bu dizinde yer alan veritabanı dosyasını belirlenen aralıklarla kontrol etmek suretiyle herhangi bir verideğişimi işlemi gerçekleşmişse yeni veriyi RS-232 haberleşme protokolünü kullanarak seri porttan kayan yazının tanımlı protokolüne uygun olarak göndermektedir. Uygulamada veritabanı dosyası çok sık periyotlarda kontrol edilmektedir. Fakat her kontrol etme esnasında veritabanı dosyasında değişim olmaması durumunda elektronik bilgilendirme panosu mevcut durumunu korumaktadır. Delphi dilinde yazılmış arayüz programı webserver olarak tanımlanan ve elektronik bilgilendirme panosunun bağlı bulunduğu bilgisayarın ilk açılışı esnasında otomatik olarak çalıştırılacak şekilde başlangıçta çalıştırılan dosyalar listesine eklenmelidir. Tasarlanmış olan Delphi diliyle yazılmış program ilk çalıştırıldığında, elektronik bilgilendirme panosunun bağlı bulunduğu seri port seçilerek bağlantı için konulan bağlan ikonu ile bağlantı sağlanarak, tamam sekmesi seçilmelidir. Şekil 3.1. 'de bağlantı ayar penceresi görülmektedir.



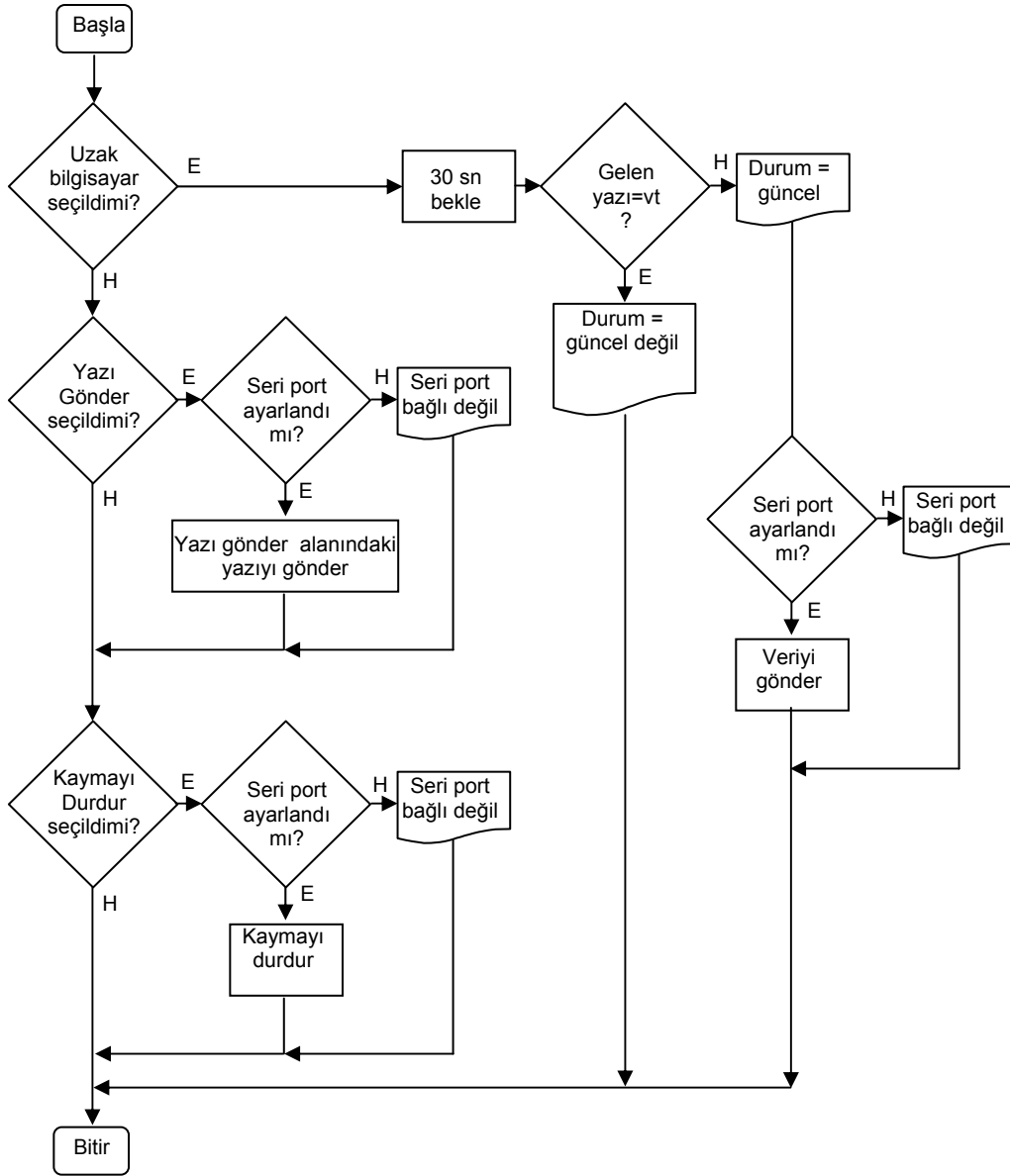
Şekil 3. 1. Delphi arayüz programı bağlantı ayar penceresi

Şekil 3.2. 'de Delphi diliyle yazılmış programa ait arayüz görünmektedir. Kaymayı durdur sekmesi; pano ekranındaki mesajı dondurma işlemini gerçekleştirecek kodları panoda yer alan PIC mikrodenetleyiciye göndererek, yazıyı dondurma işlemini gerçekleştirir. Yazı gönder sekmesi; uzak bilgisayar seçeneği aktif edilmediyse ağ bağlantısı kullanılmaksızın, doğrudan yazı giriş alanı metin kutusuna yazılan metni bilgilendirme panosuna gönderme işlemini gerçekleştirir. Bu işlem veritabanına erişim yapılmaksızın, doğrudan mikrodenetleyici ile haberleşilerek yapılır.



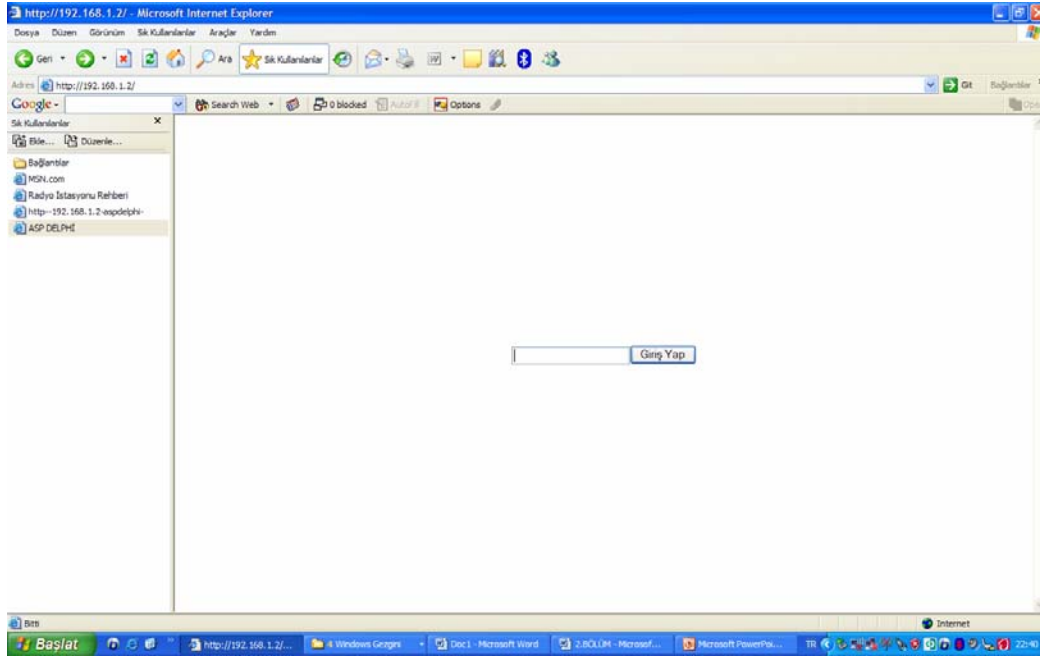
Şekil 3. 2. Delphi arayüz programının ana penceresi

Aşağıdaki şekilde delphi arayüz programının akış diyagramı verilmiştir.



Şekil 3.3. Delphi arayüz programının akış şeması

ASP diliyle yazılmış program ise, web server olarak tanımlanmış, panonun bağlı bulunduğu bilgisayara IP adresi/aspdelphi (Örnek: http://192.168.4.1/aspdelphi) yada İnternet Servis Sağlayıcı ayarlarında giriş dizini olarak aspdelphi dizini tanımlandıysa adres çubuğuna doğrudan webserver bilgisayarın adı yazılarak sunucu bilgisayarla iletişim kurulduğunda çalışır ve sadece yetkili kişilerin girişine olanak sağlamak için ilk olarak şifre doğrulama penceresinin görüntülenmesini sağlar (Şekil 3.4.).

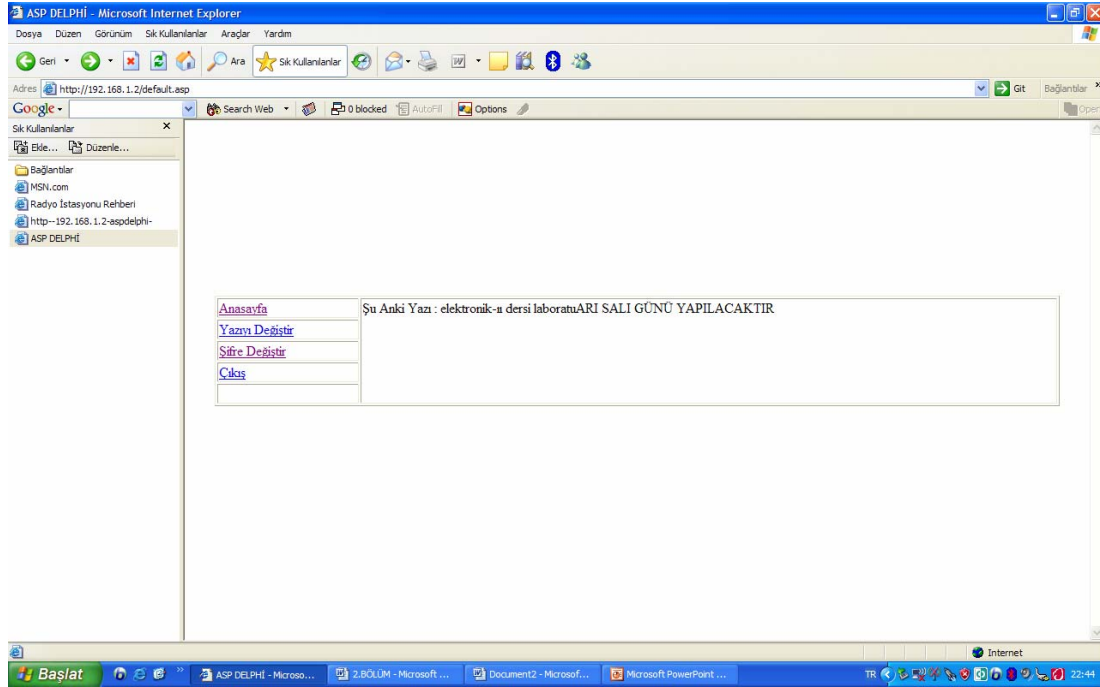


Şekil 3.4. ASP Programının şifre kontrol bölümü

Doğru şifrenin yazılmasıyla, metin gönderme ve okuma penceresi istemci bilgisayarda görüntülenir (Şekil 3.5.).

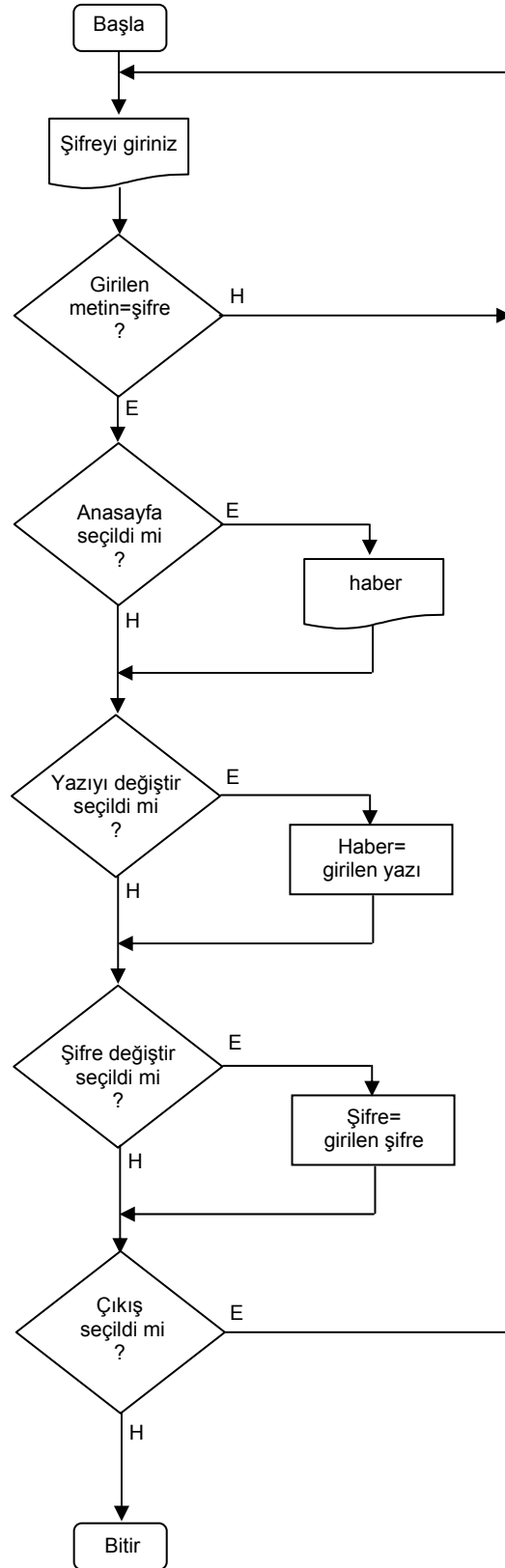
Gelen explorer sayfasında (Şekil 3.5.) görüldüğü gibi; anasayfa, yazıyı değiştir, şifre değiştir ve çıkış bölümleri bulunmaktadır. Anasayfa bölümüne girildiğinde; ekranda veritabanında var olan verinin görüntülenmesi sağlanır. Yazıyı değiştir bölümüne girildiğinde; gelen metin kutusuna yazı yazılır ve kaydet butonuna basıldığı anda yeni veri veri tabanına kaydedilir, bir önceki veri silinir. Şifre değiştir bölümüne girildiğinde; gelen metin kutusuna yeni şifre yazılır ve onaylanırsa şifre değiştirilmiş olur. Çıkış bölümüne girildiğinde ise programdan güvenli bir şekilde çıkmış olur.

Bu şekilde uzak bilgisayardan, kayan yazının bağlı olduğu bilgisayara erişilerek veri okuma , yazma ve şifre değişikliği yapma işlemi gerçekleştirilmiş olur.



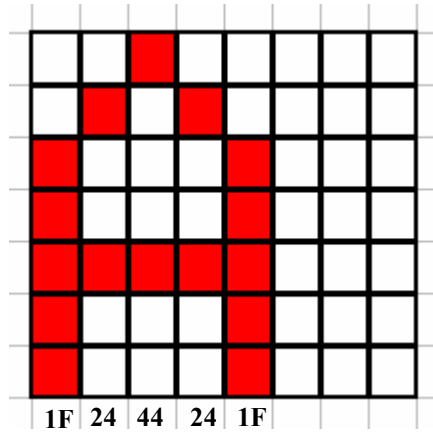
Şekil 3. 5. ASP Programının genel görünümü

Şekil 3.6. 'da ASP diliyle yazılmış olan programın akış şeması verilmiştir. Şemada haber olarak belirtilen değişken, veritabanında haber adıyla kayıtlı bulunan en son veridir. Şifre olarak belirtilen değişken ise veritabanında şifre adıyla kayıtlı bulunan ve içinde en son şifrenin kayıtlı bulunduğu bölümdür.



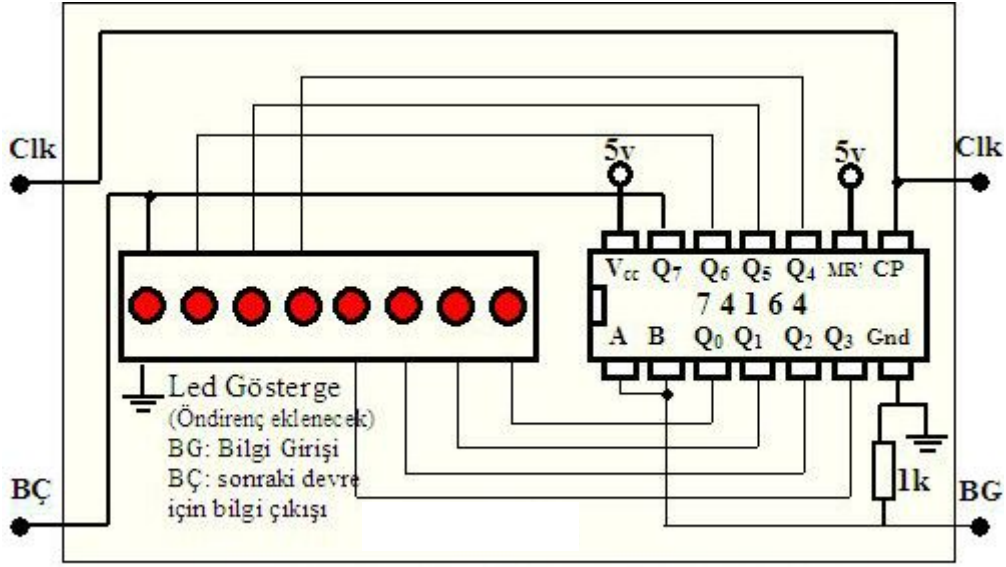
Şekil 3. 6. ASP diliyle yazılan programın akıř řeması.

Elektronik bilgilendirme panosunda karakterlerin tanımlanması işlemi, Şekil 3.7. 'de görüldüğü gibi gerçekleştirilir. Uygulamada kullanılan elektronik bilgilendirme panosu, 7 satırdan oluşmaktadır. Panonun uzunluğu başka bir ifadeyle sütun sayısı istenildiği kadar arttırılabilir. Her bir 8 led diyottan oluşan bir satırlık bölümü oluşturmak için 74ls164 8bit kaymalı kaydedici entegresinden 1 adet kullanılması yeterlidir. Yazılımda kullanılacak karakterler, Şekil 3.7. 'de gösterildiği şekilde teker teker hexadesimal kod olarak tanımlanır. Şekildeki her bir kutucuk bir adet led diyotu ifade eder. Kutucuğun dolu olması led diyotun aktif edileceğini (74ls164 çıkışının lojik "1" olması), boş olması led diyotun sönük durumda olduğunu (lojik "0") ifade eder. 7 satırdan oluşan sütunu mikrodenetleyicinin 8 pinli bir çıkış portunun kontrol edecek olması nedeniyle, her bir sütun iki adet hexadesimal sayı ile ifade edilir. Örneğin şekildeki A karakterini oluşturan ilk sütun 1F hexadesimal sayısı ile tanımlanır. Mikrodenetleyicinin ilgili çıkış portuna 1F hexadesimal sayısı yüklendikten sonra, 74ls164 ile yapılmış sürücü devrelere clock pulsü uygulandığında pano ekranında A karakterinin ilk sütunu görünür. Ardından ikinci sütunu oluşturan karakterler gönderilir, tekrar clock pulsü gönderilir ve bu şekilde istenen karakterler panoda görüntülenir. Panodaki karakterlerin kayma hızı, doğrudan clock pulsünün uygulanma süresine bağlıdır.



Şekil 3.7. Elektronik bilgilendirme panosunda kullanılacak karakterlerin tanımlanması

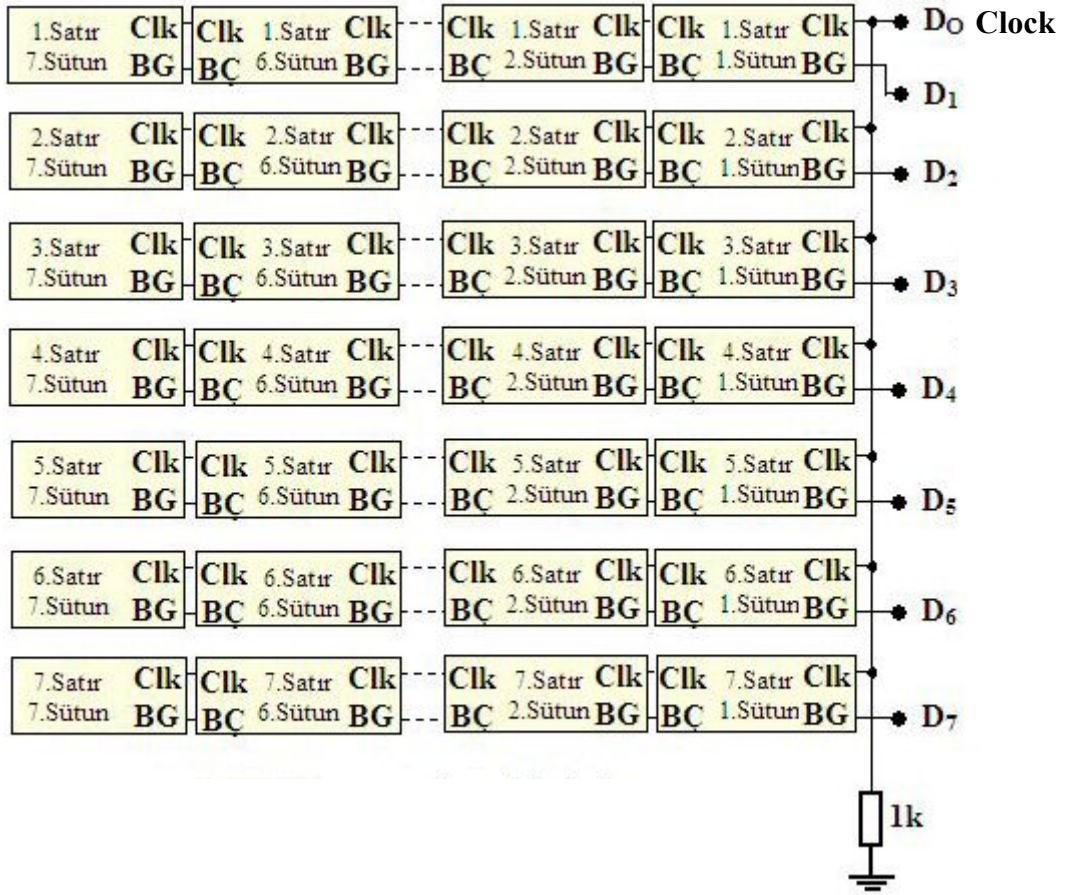
Şekil 3.8. 'de ise elektronik bilgilendirme panosunda kullanılan led sürme devresi verilmiştir. Her bir satırda yer alan 8 led için bu devre kullanılmaktadır.



Şekil 3.8. Elektronik bilgilendirme panosunda kullanılan satır sürme devresi

Şekil 3.8. 'de verilen satır sürme devreleri, Şekil 3.9. 'daki gibi bağlanarak elektronik bilgilendirme panosu oluşturulmuş olur. Panonun uzunluğu için herhangi bir sınırlama yoktur. Sadece devrenin kaynaktan çektiği akım değer sınırlayıcı olabilir. Şekil 3.9. 'da görüldüğü gibi, elektronik bilgilendirme panosunun D<sub>0</sub>, ...D<sub>7</sub> olarak belirtilen satır sürme bilgi girişleri ve clock pulsü girişi olmak üzere 8 adet kontrol girişi mevcuttur.

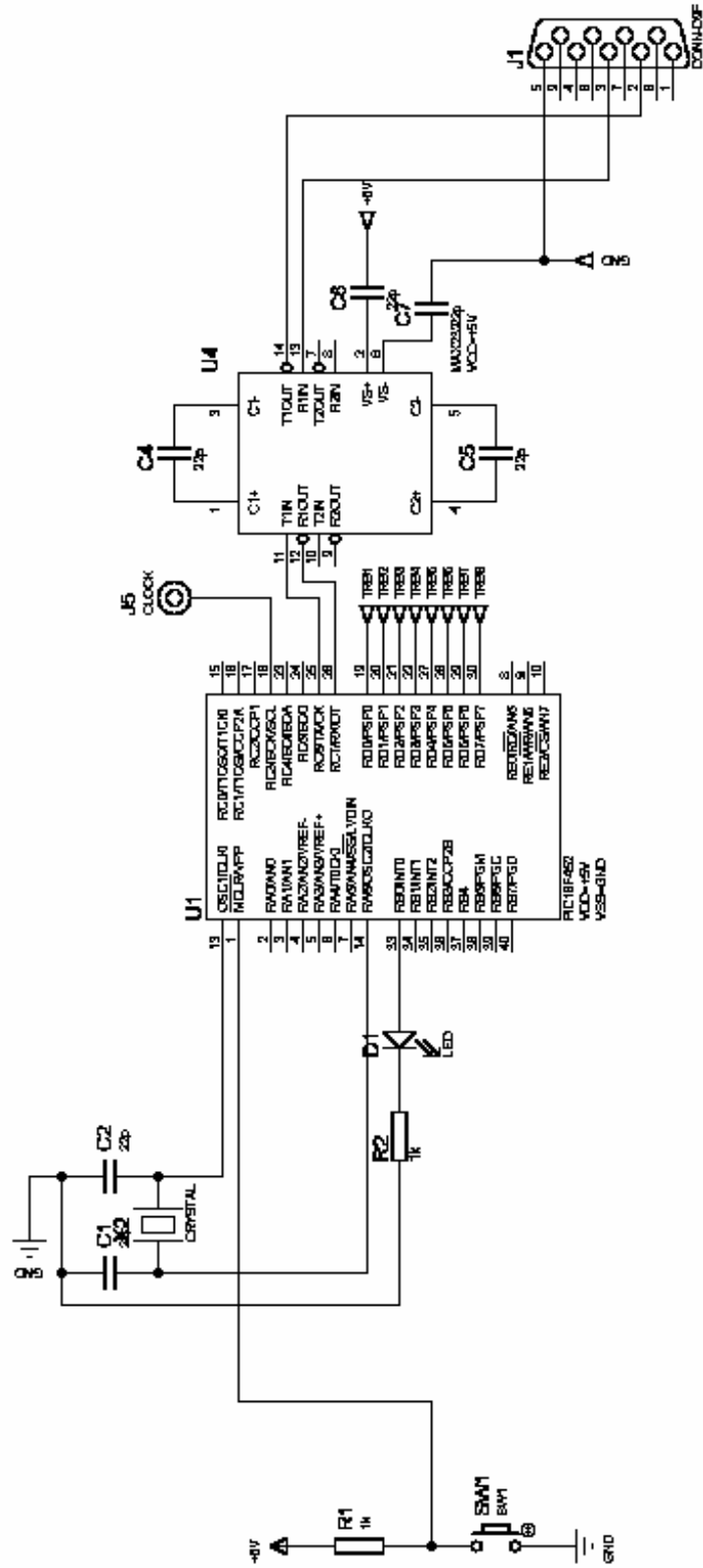




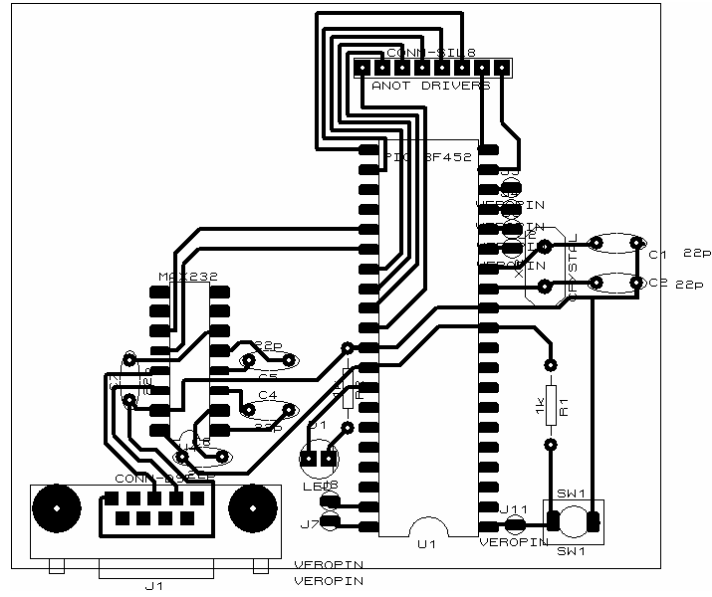
Şekil 3.9. Elektronik bilgilendirme panosunun led sürücü devrelerle oluşturulması

Şekil 3.10. 'da pic mikrodenetleyicili satır sürücü devrenin şeması görülmektedir.

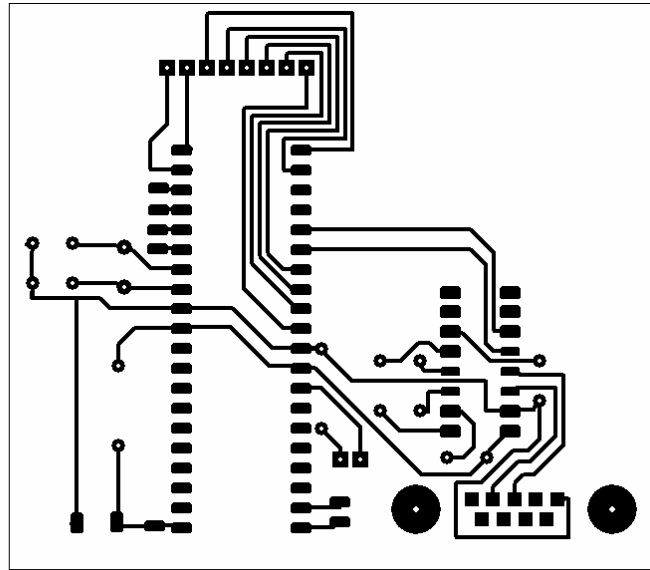
Şekil 3.11. ve Şekil 3.12. 'de ise aynı devrenin baskı devre şemaları verilmiştir.



Şekil 3. 10. Pic mikrodeneetleyicili satır sürücü devre şeması

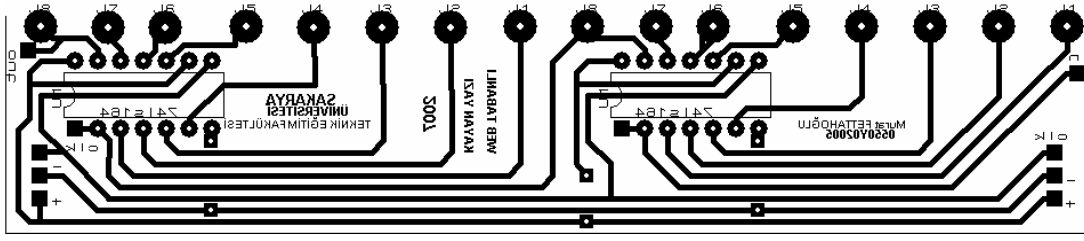


Şekil 3.11. Pic mikrodenetleyicili sürücü devrenin baskı devresi (üst görünüm).

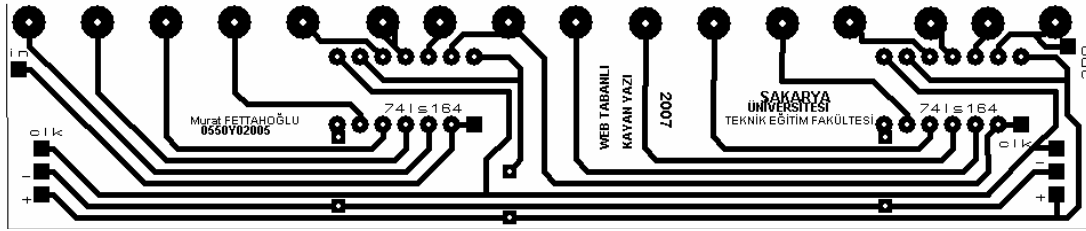


Şekil 3.12. Pic mikrodenetleyicili sürücü devrenin baskı devresi (alt görünüm).

Şekil 3.13. ve Şekil 3.14. 'te satır sürme devresinin baskı devre şemaları verilmiştir.



Şekil 3.13. Satır sürme devresi (üst görünüm).



Şekil 3.14. Satır sürme devresi (alt görünüm).

## **BÖLÜM 4. SONUÇLAR**

Tasarımı yapılan bu çalışmada, web üzerinden elektronik bilgilendirme panosunun kontrolünü sağlanmıştır. Mevcut sistemde RS-232 haberleşme protokolü ile server olarak tanımlı bilgisayarla seri port üzerinden haberleşebilecek şekilde tasarlanmış olan ve içerisinde PIC18F452 mikrodenetleyici kullanılan bir elektronik bilgilendirme panosu yer almaktadır. Web sunucu bilgisayarda çalıştırılan delphi programlama dili ile yazılmış arayüz yazılımı, belli zaman aralıkları ile aynı bilgisayarda yer alan veritabanındaki verileri kontrol ederek panoya göndermektedir. Veritabanına istemci (client) bilgisayarlardan panonun bağlı bulunduğu web server olarak tanımlı bilgisayarın IP numarası kullanılarak ve ilgili dizin seçilerek, LAN ağı yada internet üzerinden erişilebilmekte ve veritabanındaki veriler okunup, değiştirilebilmektedir. Ayrıca sistemde güvenliğin sağlanabilmesi için veritabanına erişimde ilk olarak şifre kontrolü yapılmaktadır. Delphi programında yer alan uzak bilgisayar seçim ikonu kaldırılarak, server üzerinden de doğrudan bilgilendirme panosu kontrol edilebilmektedir.

Uygulamanın kullanıcılara sağladığı en önemli avantaj, panonun bağlı bulunduğu bilgisayara her seferinde veri güncellemek için müdahale etmek yerine ağ üzerinde bağlı herhangi bir bilgisayar vasıtasıyla bu işlemi güvenli bir şekilde ve en kısa sürede sağlamasıdır.

## **BÖLÜM 5. TARTIŞMA VE ÖNERİLER**

Bu uygulamada, iletişim için seri port kullanılmasının nedeni, mikrodenetleyiciler ile kolay iletişim kurulabilmesidir. Bu avantajının yanında, bilgilendirme panosu ile web server olarak tanımlı bilgisayarın seri port üzerinden bağlantı kurması uzun mesafelerde bilginin kaybolmasına veya bozulmasına yol açabilir. Bu sorunun ortadan kaldırılması için mevcut iletişim yerine, güvenli, hızlı ve ucuz iletişim teknolojileri kullanılabilir. Buna örnek olarak (wireless) kablosuz iletişim ağı kullanılabilir. Ayrıca günümüz bilgisayarlarında artık seri port bağlantı noktası yavaş yavaş yerini diğer bağlantı noktalarına bırakmaktadır (USB gibi).

Elektronik bilgilendirme panosunun sürekli bir bilgisayar ile bağlantı halinde olması ihtiyacı bir dezavantaj olarak düşünülebilir. Yeni iletişim teknolojileri kullanılarak bahsedilen bilgisayar kullanılmaksızın doğrudan bilgilendirme panosunda yer alan mikrodenetleyici ile yerel ağ yada internet ortamıyla erişim sağlanabilir.

Uygulama, güncel haber sitelerinin rss formatlı bilgilerini işleyerek, ekranında belirli aralıklarda gösterecek şekilde geliştirilebilir. Ve ayrıca, tarih, saat, sıcaklık, nem gibi değerleri de belirli periyotlarda gösterecek şekilde geliştirilebilir.

Doğrudan webserver olarak tanımlanan bilgisayarın paralel portu kullanılarakta led sürücü devreler kontrol edilebilir. Böylece mikrodenetleyici kullanılmasına gerek kalmayacaktır. Ancak, asp programlama dilinin paralel port erişim fonksiyonları sınırlı kullanım sağlamaktadır ve paralel portta kablo sayısı fazla olduğundan uzak mesafeler için kullanılması maliyeti arttırmaktadır. Ayrıca paralel porta erişimde kullanılacak kablonun mesafesi sınırlıdır.

## KAYNAKLAR

- [1] İSKEFİYELİ, M., “İnternet Tabanlı Ölçme Ve Kontrol”, Yüksek Lisans Tez Çalışması, Sakarya Üniversitesi, 2002.
- [2] ALTINBAŞAK, O., “Mikrodenetleyiciler ve PIC Programlama”, Atlas Yayıncılık, 2003.
- [3] ÇÖLKESEN, R., ÖRENCİK, B., “Bilgisayar Haberleşmesi ve Ağ Teknolojisi”, Papatya Yayıncılık, 1999
- [4] ÇALIŞKAN, A., “Mekatronik Sistemlerde İnternet Tabanlı Kontrol Ve Kartezyen Robot Üzerinde Bir Uygulama”, Yüksek Lisans Tez Çalışması, Sakarya Üniversitesi, 2004
- [5] DEMİRKOL, Z., “ASP İle Web Programcılığı Ve Elektronik Ticaret”, Pusula Yayıncılık, 2000
- [6] ÇUBUKÇU, F., “ASP 3.0 ile Veritabanı Yönetimi”, Alfa Yayınları, 2001.
- [7] [www.microchip.com](http://www.microchip.com)
- [8] [www.maxim.com](http://www.maxim.com)
- [9] FİDAN, G., “İnternet Tabanlı Veri Toplama”, Yüksek Lisans Tez Çalışması, Sakarya Üniversitesi, 2003.
- [10] ÇALIŞKAN, A., “Mekatronik Sistemlerde İnternet Tabanlı Kontrol ve Kartezyen Robot Üzerinde Bir Uygulama”, Yüksek Lisans Tez Çalışması, Sakarya Üniversitesi, 2004.

## **EKLER**

### **EK 1. Delphi programı kaynak kodu**

unit Unit1;

interface

uses

Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls, Forms,  
Dialogs, StdCtrls, Buttons, CPortCtl, CPort, Menus, ExtCtrls, ComCtrls,  
DB, ADODB, Mask, DBCtrls;

type

TForm1 = class(TForm)

  CBUzak: TCheckBox;

  LGelenYazi: TLabel;

  ComLed1: TComLed;

  ComLed2: TComLed;

  Label3: TLabel;

  Label4: TLabel;

  Panel1: TPanel;

  BKayma: TBitBtn;

  BGonder: TBitBtn;

  BCikis: TBitBtn;

  BAyar: TBitBtn;

  SendTimer: TTimer;

  lkayanyazi: TEdit;

  Label2: TLabel;



```
ComLed3: TComLed;
Label5: TLabel;
ADOConnection1: TADOConnection;
ADOTable1: TADOTable;
DataSource1: TDataSource;
DBEGelenYazi: TDBEdit;
Guncel: TTimer;
OtoGonder: TTimer;
LDurum: TLabel;
Label6: TLabel;
Label7: TLabel;

procedure BAYarClick(Sender: TObject);
procedure BKaymaClick(Sender: TObject);
procedure CBUZakClick(Sender: TObject);
procedure FormCreate(Sender: TObject);
procedure BGonderClick(Sender: TObject);
procedure Kay_Dur();
procedure Kay_Basla();
procedure Yazı_Bul();
procedure Yazı_Oku();
//procedure Timer1Timer(Sender: TObject);
procedure BCikisClick(Sender: TObject);
function KodBul(Harf : Char):string;
procedure SendTimerTimer(Sender: TObject);
procedure GuncelTimerTimer(Sender: TObject);
procedure OtoGonderTimerTimer(Sender: TObject);

private

{ Private declarations }
```

```
public
  { Public declarations }
end;

var
  Form1: TForm1;

  KaymaVar      : boolean = true;
  UzakErisim    : boolean = false;
  YazıGonderildi : boolean = false;
  SPGonder      : boolean = false;
  Guncelmi      : boolean = false;
  Gonder        : boolean = false;
  GonderIlk     : boolean = false;
  GonderUzunluk : boolean = false;
  GonderHarf    : boolean = false;
  GonderSon     : boolean = false;
  AraZaman      : boolean = false;

  GonTip       : byte;
  GYUzunlugu   : byte;

  HarfSayaci   : integer;
  AraSayac     : integer;
  DiziUzunlugu : integer;

  DiziUzunluguHex : string;
  Uzunluk         : string;
  GirilenYazi     : string;
  GelenYazi       : string;
  DiziKarakter    : array [1..255] of string;
  Dizi           : array [1..255] of char;
  SeriPortDizisi : array [1..11] of char;
```

```
GelenDizi      : string;
```

```
implementation
```

```
uses unit2;
```

```
{ $R *.dfm }
```

```
//"Ayarlar" butonu alt yordamı
```

```
procedure TForm1.BAyarClick(Sender: TObject);
```

```
begin
```

```
  Form2.show;//Seri port penceresini göster
```

```
end;
```

```
//"Kayma" butonu alt yordamı
```

```
procedure TForm1.BKaymaClick(Sender: TObject);
```

```
begin
```

```
  //Seri port bağlı mı?
```

```
  if Form2.seriport.Connected then
```

```
    begin
```

```
      if KaymaVar then//Kayma var mı?
```

```
        begin
```

```
          Kay_Dur();
```

```
          BGonder.Enabled := true;
```

```
          BKayma.Caption := 'Kaymayı Başlat';
```

```
          KaymaVar := false;
```

```
        end
```

```
      else
```

```
        begin
```

```
          if YazıGonderildi then //Yazı gönderilmiş mi?
```

```
            begin
```

```
              YazıGonderildi := false;
```

```
              BGonder.Enabled := false;
```

```
KaymaVar := true;
BKayma.Caption := 'Kaymayı Durdur';
Kay_Basla();
end
else
  ShowMessage('Yazı Gönderilmemiş!...');
end
end
else
  Showmessage('Seri Port Bağlı Değil!...');
end;

// "Uzak Bilgisayar" seçme kutusu alt yordamı
procedure TForm1.CBUzakClick(Sender: TObject);
begin
  if CBUzak.Checked then // "Uzak Bilgisayar" seçilmiş mi?
  begin
    // Manuel kontrol butonlarını pasif yap
    BKayma.Enabled := false;
    BGonder.Enabled := false;
    // Güncelleme zamanlayıcısını başlat
    Guncel.Enabled := true;
  end
  else
  begin
    // Manuel kontrolleri etkinleştir
    BKayma.Enabled := true;
    BGonder.Enabled := false;
    YazıGonderildi := false;
    KaymaVar := true;
    BKayma.Caption := 'Kaymayı Durdur';
    // Güncelleme zamanlayıcısını durdur
  end
end;
```

```
Guncel.Enabled := false;
end
end;

//Program ilk buradan başlar
procedure TForm1.FormCreate(Sender: TObject);
begin
  if CBUzak.Checked then//Uzak bilgisayar seçilmiş mi?
  begin
    //Manuel kontrol butonlarını pasif yap
    BKayma.Enabled := false;
    BGonder.Enabled := false;
    //Güncelleme zamanlayıcısını başlat
    Guncel.Enabled := true;
  end
  else
  begin
    //Manuel kontrolleri etkinleştir
    BKayma.Enabled := true;
    BGonder.Enabled := false;
    //Güncelleme zamanlayıcısını durdur
    Guncel.Enabled := false;
  end
end;

//"Yazi Gönder" butonu alt yordamı
procedure TForm1.BGonderClick(Sender: TObject);
begin
  //Seri port bağlı mı?
  if Form2.seriport.Connected then
  begin
```

```
if KaymaVar then//Kayma var mı?
begin
  ShowMessage('Önce Kaymayı Durdurun...');
end
else
begin
  BKayma.Enabled := False;
  //Yazi gönderme zamanlayıcısını başlat
  SendTimer.Enabled := true;
  Gonder := true;
  //Yazi gönderme prosedürünü başlat
  GonderIlk := true;
  if YaziGonderildi then//Yazi gönderme tamamlandı mı?
  begin
    //Yazi gönderme zamanlayıcısını durdur
    SendTimer.Enabled := false;
    YaziGonderildi := false;
  end;
end
end
else
  Showmessage('Seri Port Bağlı Değil!...');
end;

//Kayan yazi durdurma alt yordamı
procedure TForm1.Kay_Dur();
begin
  //Seri porta gönderilecek kayma durdurma dizisini oluştur
  SeriPortDizisi[1] := '!';
  GYUzunlugu := 1;
  SPGonder := true;//Kayan yaziya diziyi gönder
end;
```

```

//Kayan yazi başlatma alt yordamı
procedure TForm1.Kay_Basla;
begin
    //Seri porta gönderilecek kayma başlatma dizisini oluştur
    SeriPortDizisi[1] := ':';
    SeriPortDizisi[2] := '0';
    SeriPortDizisi[3] := '1';
    GYUzunlugu := 3; //Kayan yazıya diziyi gönder
    SPGonder := true;
    if CBUzak.Enabled then // "Uzak Bilgisayar" seçilmiş mi?
        OtoGonder.Enabled := false; //Otomatik yazı gönderme zamanlayıcısını durdur
end;

```

```

//Bu fonksiyon kayan yazıya gönderilecek harfe karşılık gelen kodu bulur
function TForm1.KodBul(Harf : Char):string;
begin
    //Harfler
    if (Harf = 'A') or (Harf = 'a') then result := '0000';
    if (Harf = 'B') or (Harf = 'b') then result := '0006';
    if (Harf = 'C') or (Harf = 'c') then result := '000C';
    if (Harf = 'Ç') or (Harf = 'ç') then result := '001E';
    if (Harf = 'D') or (Harf = 'd') then result := '0012';
    if (Harf = 'E') or (Harf = 'e') then result := '0024';
    if (Harf = 'F') or (Harf = 'f') then result := '0060';
    if (Harf = 'G') or (Harf = 'g') then result := '006C';
    if (Harf = 'Ğ') or (Harf = 'ğ') then result := '0072';
    if (Harf = 'H') or (Harf = 'h') then result := '0078';
    if (Harf = 'I') or (Harf = 'ı') then result := '007E';
    if (Harf = 'İ') or (Harf = 'i') then result := '003C';
    if (Harf = 'J') or (Harf = 'j') then result := '0084';
    if (Harf = 'K') or (Harf = 'k') then result := '0030';

```

```
if (Harf = 'L') or (Harf = 'l') then result := '002A';
if (Harf = 'M') or (Harf = 'm') then result := '004E';
if (Harf = 'N') or (Harf = 'n') then result := '0048';
if (Harf = 'O') or (Harf = 'o') then result := '008A';
if (Harf = 'Ö') or (Harf = 'ö') then result := '0090';
if (Harf = 'P') or (Harf = 'p') then result := '0066';
if (Harf = 'Q') or (Harf = 'q') then result := '0196';
if (Harf = 'R') or (Harf = 'r') then result := '0042';
if (Harf = 'S') or (Harf = 's') then result := '005A';
if (Harf = 'Ş') or (Harf = 'ş') then result := '0096';
if (Harf = 'T') or (Harf = 't') then result := '0036';
if (Harf = 'U') or (Harf = 'u') then result := '0054';
if (Harf = 'Ü') or (Harf = 'ü') then result := '009C';
if (Harf = 'V') or (Harf = 'v') then result := '00A2';
if (Harf = 'W') or (Harf = 'w') then result := '019C';
if (Harf = 'X') or (Harf = 'x') then result := '0190';
if (Harf = 'Y') or (Harf = 'y') then result := '00A8';
if (Harf = 'Z') or (Harf = 'z') then result := '00AE';
```

```
//Rakamlar
```

```
if Harf = '1' then result := '00B4';
if Harf = '2' then result := '00BA';
if Harf = '3' then result := '00C0';
if Harf = '4' then result := '00C6';
if Harf = '5' then result := '00CC';
if Harf = '6' then result := '00D2';
if Harf = '7' then result := '00D8';
if Harf = '8' then result := '00DE';
if Harf = '9' then result := '00E4';
if Harf = '0' then result := '00EA';
```

```
//Semboller
```

```
if Harf = ' ' then result := '0018';
```



```
if Harf = '$' then result := '00F0';
if Harf = '!' then result := '0100';
if Harf = '"' then result := '0106';
if Harf = '#' then result := '010C';
if Harf = '%' then result := '0112';
//if Harf = "" then result := '0118';
if Harf = '(' then result := '011E';
if Harf = ')' then result := '0124';
if Harf = '*' then result := '012A';
if Harf = '+' then result := '0130';
if Harf = '-' then result := '0136';
if Harf = ',' then result := '013C';
if Harf = '.' then result := '0142';
if Harf = '/' then result := '0148';
if Harf = ':' then result := '014E';
if Harf = ';' then result := '0154';
if Harf = '<' then result := '015A';
if Harf = '=' then result := '0160';
if Harf = '>' then result := '0166';
if Harf = '?' then result := '016C';
if Harf = '@' then result := '0172';
if Harf = '[' then result := '0178';
if Harf = ']' then result := '017E';
if Harf = '\' then result := '0184';
end;

procedure TForm1.Yazi_Oku();
begin
if CBUzak.Checked then//"Uzak Bilgisayar" seçilmiş mi?
begin
//Veri tabanından alınan yazı uzunluğunu bul
GirilenYazi := GelenYazi;
```

```
    DiziUzunlugu := length(GirilenYazi);
end
else
begin
    //Arayüzden girilen yazı uzunluğunu bul
    GirilenYazi := lkayanyazi.Text;
    DiziUzunlugu := length(GirilenYazi);
end;
end;

//Bu alt yordam yazıdaki harflerin karakterlerinden oluşan
// karakter dizisini üretir
procedure TForm1.Yazi_Bul();
var
    i : byte;
begin
    for i := 1 to DiziUzunlugu do
        begin
            if CBUzak.Checked then/"Uzak Bilgisayar" seçilmiş mi?
                DiziKarakter[i] := Kodbul(GelenYazi[i])
            else
                DiziKarakter[i] := KodBul(GirilenYazi[i])
            end;
        end;
    end;
end;

procedure TForm1.BCikisClick(Sender: TObject);
begin
    Application.Terminate;//Uygulamayı sonlandır
end;
```

```
//Bu zamanlayıcı alt yordamı yazı gönderme sürecini gerçekleştirir
procedure TForm1.SendTimerTimer(Sender: TObject);
var
  y : byte;
  str : string;
begin
  if Gonder then
    begin
      //Yazı gonderme başlangıç kodu oluşturulur
      if GonderIlk then
        begin
          SeriPortDizisi[1] := ':';
          SeriPortDizisi[2] := '0';
          SeriPortDizisi[3] := '0';
          SeriPortDizisi[4] := '0';
          SeriPortDizisi[5] := '9';
          SeriPortDizisi[6] := '0';
          SeriPortDizisi[7] := '0';
          SeriPortDizisi[8] := '0';
          SeriPortDizisi[9] := '1';
          SeriPortDizisi[10] := '3';
          SeriPortDizisi[11] := 'C';
          GYUzunlugu := 11;
          SPGonder := true;//Kayan yazıya diziyi gönder
          GonTip := 1;
        end;
      //Gonderilen yazının uzunluğuna bağlı kod oluşturulur
      if GonderUzunluk then
        begin
          SeriPortDizisi[1] := ':';
          SeriPortDizisi[2] := '0';
          SeriPortDizisi[3] := '2';
          SeriPortDizisi[4] := '0';
```

```

SeriPortDizisi[5] := '1';
Yazi_Oku();
//Dizi uzunluğunun hexadecimal karşılığını hesapla
DiziUzunluguHex := IntToHex(DiziUzunlugu,2);
SeriPortDizisi[6] := DiziUzunluguHex[1];
SeriPortDizisi[7] := DiziUzunluguHex[2];
GYUzunlugu := 7;
SPGonder := true;//Kayan yaziya diziyi gönder
//Yazının ilk harfini göndermek için harf sayacını 1 yap
HarfSayaci := 1;
GonTip := 2
end;
if GonderHarf then
begin
  Yazi_Oku();
  Yazi_Bul();
  //Yazının ilk HarfSayaci'nci harfine ait kodu
  //      str geçici dizisine at
  str := DiziKarakter[HarfSayaci];
  SeriPortDizisi[1] := ':';
  SeriPortDizisi[2] := '0';
  SeriPortDizisi[3] := '2';
  //Seri port dizise 4. hücrenden itibaren harf kodlarını yaz
  for y := 1 to length(str) do
    SeriPortDizisi[y+3] := str[y];
  GYUzunlugu := 7;
  SPGonder := true;//Kayan yaziya diziyi gönder
  //Harflerin tamamı gönderildi mi?
  if HarfSayaci = DiziUzunlugu then
  begin
    HarfSayaci := 0;
    GonTip := 3;
  end;
end;

```

```
end;
//Yazı gönderme sonlandırma kodu oluşturulur
if GonderSon then
begin
    SeriPortDizisi[1] := ':';
    SeriPortDizisi[2] := '0';
    SeriPortDizisi[3] := '0';
    SeriPortDizisi[4] := '0';
    SeriPortDizisi[5] := '8';
    SeriPortDizisi[6] := '0';
    SeriPortDizisi[7] := '1';
    SeriPortDizisi[8] := '0';
    SeriPortDizisi[9] := '9';
    SeriPortDizisi[10] := '0';
    SeriPortDizisi[11] := '0';
    GYUzunlugu := 11;
    SPGonder := true;//Kayan yazıya diziyi gönder
    GonTip := 4;
end;
end;
end;

//Yazı güncelleme zamanlayıcısı
procedure TForm1.GuncelTimerTimer(Sender: TObject);

begin
    ADOTable1.Open;//Veri tabanı dosyasını aç
    //Veri tabanındaki geçerli yazıyı ekrana yaz
    LGelenYazi.Caption := DBEGelenYazi.EditText;
    //Veri tabanındaki yazı güncel mi?
    if not (DBEGelenYazi.EditText = GelenYazi) then
        begin
```

```
AraSayac := 0;
Guncelmi := true;
GelenYazi := DBEGelenYazi.EditText;
DiziUzunlugu := length(GelenYazi);
OtoGonder.Enabled := true;
LDurum.Caption := 'Yazı güncel!...';
end
else
begin
Guncelmi := false;
OtoGonder.Enabled := false;
LDurum.Caption := 'Yazı güncel değil!...';
end;
ADOTable1.Close;//Veri tabanı dosyasını kapat
end;

//Otomatik yazi gönderme zamanlayıcısı
procedure TForm1.OtoGonderTimerTimer(Sender: TObject);
begin
if AraSayac < 1301 then inc(AraSayac);
if Form2.seriport.Connected then
begin
//Kayma durduruluyor
if AraSayac = 10 then//100ms bekle
Kay_Dur();
//Yazi gönderiliyor
if AraSayac = 50 then//500ms bekle
begin
SendTimer.Enabled := true;
Gonder := true;
GonderIlk := true;
LDurum.Caption := 'Kayma durduruldu!...';
```

```
end;
//Kayma başlatılıyor
if AraSayac = 1300 then//13s bekle (255 karakterlik yazi göndermek için)
begin
  if YazıGonderildi then
    begin
      YazıGonderildi := false;
      SendTimer.Enabled := false;
      Gonder := false;
      LDurum.Caption := 'Yazı gönderildi!...!';
    end;
    Kay_Basla();
  end;
end
else
  Showmessage('Seri Port Bağlı Değil!...');
end;
end.
```

```
unit Unit2;
```

```
interface
```

```
uses
```

```
Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls, Forms,
Dialogs, StdCtrls, Buttons, CPortCtl, CPort, ExtCtrls;
```

```
type
```

```
TForm2 = class(TForm)
```

```
  ComLed1: TComLed;
```

```
  bbaglan: TBitBtn;
```

```
  BitBtn2: TBitBtn;
```

```
  Label1: TLabel;
```

```
Label2: TLabel;
seriport: TComPort;
ComDataPacket1: TComDataPacket;
ComComboBox1: TComComboBox;
Timer1: TTimer;
procedure BitBtn2Click(Sender: TObject);
procedure bbaglanClick(Sender: TObject);
procedure FormCreate(Sender: TObject);
procedure ComComboBox1Change(Sender: TObject);
procedure Timer1Timer(Sender: TObject);
procedure ComDataPacket1Packet(Sender: TObject; const Str: String);

private
  { Private declarations }
public
  { Public declarations }
end;

var
  Form2: TForm2;

implementation

  uses unit1;
  {$R *.dfm}

procedure TForm2.BitBtn2Click(Sender: TObject);
begin
  Form2.Close;
end;
```



```
procedure TForm2.bbaglanClick(Sender: TObject);
begin
  if not SeriPort.Connected then
  begin
    BBaglan.Caption := 'Bağlantıyı Kes';
    SeriPort.Open;
    ComDataPacket1.StartString := #1;
    ComDataPacket1.Size := 9;
  end
  else
  begin
    BBaglan.Caption := 'Bağlan';
    SeriPort.Close;
  end
end;

procedure TForm2.FormCreate(Sender: TObject);
begin
  EnumComPorts(ComComboBox1.Items);
end;

procedure TForm2.ComComboBox1Change(Sender: TObject);
begin
  if ComComboBox1.ItemIndex > -1 then
    SeriPort.Port := ComComboBox1.Items[ComComboBox1.ItemIndex];
end;

procedure TForm2.Timer1Timer(Sender: TObject);
begin
  if SeriPort.Connected then
  begin
    if SPGonder then
    begin
      SeriPort.Write(SeriPortDizisi,GYUzunlugu);
      SPGonder := false;
    end
  end
end;
```

```
if GonderHarf then inc(HarfSayaci);
case GonTip of
  1 : begin
    GonderIlk := false;
    GonderUzunluk := true;
  end;
  2 : begin
    GonderUzunluk := false;
    GonderHarf := true;
  end;
  3 : begin
    GonderHarf := false;
    GonderSon := true;
  end;
  4 : begin
    YazıGonderildi := true;
    if not Form1.CBUzak.Checked then Form1.BKayma.Enabled := true;
    GonderSon := false;
    GonTip := 0
  end;
end;
end;
end;
end;
procedure TForm2.ComDataPacket1Packet(Sender: TObject; const Str: String);
var
  sayac : byte;
begin
  for sayac := 1 to length(Str) do
    GelenDizi := GelenDizi + str[sayac];
  end;
end.
end.
```

## EK 2. Asp programı kaynak kodu

```

<html>
<%pass="fettahoglu"%>
<%if session("otu")="Acik" then%>
<%
Set Sur = Server.CreateObject("ADODB.Connection")
Sur.Open "DRIVER={Microsoft Access Driver (*.mdb)}; DBQ=" &
Server.MapPath("vt.mdb")
%>

<head>
<meta http-equiv="Content-Language" content="tr">
<meta http-equiv="Content-Type" content="text/html; charset=windows-1254">
<title>ASP DELPHİ</title>
</head>

<body style="text-align: center">

<p align="center">&nbsp;</p>
<p align="center">&nbsp;</p>
<p align="center">&nbsp;</p>
<p align="center">&nbsp;</p>
<p align="center">&nbsp;</p>
<p align="center">&nbsp;</p>
<table border="1" width="100%">
<%if Request("islem")="kaydet" then%>

    <tr>

        <td>

<p align="center">
<% Set agi82 = Server.CreateObject("ADODB.Recordset")

```

```

sor = "Select * from haber "
agi82.Open sor,Sur,1,3
if request.Form("baslik")="" then
response.Write "Boş Gonderemezsiniz"
else
agi82("baslik")=request.Form("baslik")
agi82("tarih")=Date()
agi82.Update
Response.Write "Yazı Kaydedildi."
goster="hayir"
end if
%>
&nbsp;</td>
</tr>
<%End if%>
<%if not goster="hayir" then%>
<tr>
<td>
<p align="center">Şu Anki Yazı : <% Set agi82 =
Server.CreateObject("ADODB.Recordset")
sor = "Select * from haber order by kimlik desc "
agi82.Open sor,Sur,1,3
%>
<%=agi82("baslik")%>
<%agi82.close%> </td>
</tr>
<form action="default.asp?islem=kaydet" method="post" name="asdfg">
<tr>
<td>
<p align="center"><input type="text" name="baslik"
size="20"><input type="submit" value="KAYDET" name="B1"></td>
</tr>
</form>

```

```

<%End if%>
    <tr>
        <td>&nbsp;</td>
    </tr>
</table>
<p align="center">&nbsp;</p>
<p align="center">&nbsp;</p>
<p align="center">&nbsp;</p>
<p align="center">&nbsp;</p>
</body>

</html>
<p align="center">
<%else%>
&nbsp;</p>
<p align="center">&nbsp;</p>
<p align="center">&nbsp;</p>
<p align="center">&nbsp;</p>
<p align="center">&nbsp;</p>
<p align="center">&nbsp;</p>
<p align="center">&nbsp;</p>
</p>

<form action="default.asp?islem=giris" method="post" name="sdfsdf">
<p align="center">
<input type="password" name="sifre" size="20"><input type="submit" value="Giriş
Yap" name="B2">
</p>
</form>
&nbsp;<p align="center">&nbsp;</p>
<p align="center">&nbsp;</p>
<p align="center">&nbsp;</p>
<p align="center">&nbsp;</p>

```

```
<p align="center">&nbsp;</p>  
<%End if%><%if Request("islem")="giris" then%>  
<%  
if request.Form("sifre")=""&pass&"" then  
session("otu")="Acik"  
response.Redirect "default.asp"  
else  
Response.Write "Şifre Yanlış"  
End if  
%>  
<%End if%>
```

**EK 3. PIC c kaynak kodu**

```

#include "1.h"
//ekranda gösterilen
unsigned long chcnt;
unsigned long mesaj[512];
//seri porttan alınan
unsigned long gchcnt;
unsigned int gost;
//mesaj rsden geldikten sonra eski mesaja devam etmesin göstermeye diye
int1 yenidenbasla;

unsigned int getbyte(void){
char g1,g2;
unsigned int sonuc;
    g1=getch(); printf("%c",g1);
    g2=getch(); printf("%c",g2);

    if(g1<'A') g1=g1-0x30; else g1=g1-'A'+10;
    if(g2<'A') g2=g2-0x30; else g2=g2-'A'+10;
    sonuc=(g1<<4) | (g2& 0x0F);
    return sonuc;
}
#INT_RDA
void getmessage(void){
unsigned int gelen,j;
unsigned long tmp,i;
    if(gost!=23) return;// nokta geldimi?
    gelen=getch(); printf("%c",gelen);
    if(gelen !='.') return;//.

```

```

//MESAJ ALMAYA BAŞLANACAKMI?
//:000900013C
    gelen=getch();    printf("%c",gelen);
    if(gelen !=':') return;//:
    gelen=getbyte(); if(gelen != 0x00) return;//00
    gelen=getbyte(); if(gelen != 0x09) return;//09
    gelen=getbyte(); if(gelen != 0x00) return;//00
    gelen=getbyte(); if(gelen != 0x01) return;//00
    gelen=getbyte(); if(gelen != 0x3C) return;//00
    printf("\n\rok");
//EVET ŞİMDİ MESAJ UZUNLUĞUNU AL
//:020101
    gelen=getch(); printf("%c",gelen);    if(gelen !=':') return;//:
    gelen=getbyte(); if(gelen != 0x02) return;//02
    gelen=getbyte(); if(gelen != 0x01) return;//01
    gchcnt=getbyte();
    printf("\n\rok");
        mesaj[0]=gchcnt;
    write_program_memory(0x2F00,mesaj,2);

//şimdi her bir karakteri alacağız
//:020000
    for(i=0;i<gchcnt;i++){
        gelen=getch(); printf("%c",gelen);    if(gelen !=':') return;//:
        gelen=getbyte(); if(gelen != 0x02) return;//02
        gelen=getbyte();
        j=getbyte();
        printf("\n\rok");
        tmp=gelen;
        tmp=tmp<<8;
        mesaj[i]=tmp +j;
    }

```



```

//şimdi mesaj sonunu al
//:0008010900
    gelen=getch(); printf("%c",gelen);    if(gelen !=':') return;//:
    gelen=getbyte(); if(gelen != 0x00) return;//00
    gelen=getbyte(); if(gelen != 0x08) return;//08
    gelen=getbyte(); if(gelen != 0x01) return;//01
    gelen=getbyte(); if(gelen != 0x09) return;//09
    gelen=getbyte(); if(gelen != 0x00) return;//00
    printf("\n\nrok");
        for(i=0;i<gchcnt;i=i+4)
            write_program_memory(0x3000+(i*2),&mesaj[i],8);
//şimdi kaydır komutunu bekle
//:01
    gelen=getch(); printf("%c",gelen); if(gelen !=':') return;//:
    gelen=getbyte(); if(gelen != 0x01) return;//01
    printf("\n\nrok");
    chcnt=gchcnt;

    for(i=0;i<chcnt;i++)
        printf("%Lx %Lu",i,mesaj[i]);
    yenidenbasla=1;
}

```

```

void senddata(unsigned int a,unsigned int b,unsigned int c,unsigned int d,unsigned
int e){
    output_d(a);output_low(PIN_C3);output_high(PIN_C3);delay_ms(75);
    output_d(b);output_low(PIN_C3);output_high(PIN_C3);delay_ms(75);
    output_d(c);output_low(PIN_C3);output_high(PIN_C3);delay_ms(75);
    output_d(d);output_low(PIN_C3);output_high(PIN_C3);delay_ms(75);
    output_d(e);output_low(PIN_C3);output_high(PIN_C3);delay_ms(75);
    output_d(0x00);output_low(PIN_C3);output_high(PIN_C3);delay_ms(75);
    output_toggle(PIN_B0);

```

```
}

```

```
void harfkaydir(unsigned long a){
    switch(a){
        case 0x0000: senddata(0x7e,0x11,0x11,0x11,0x7e);break;//A
        case 0x0006: senddata(0x7f,0x49,0x49,0x49,0x36);break;//B
        case 0x000C: senddata(0x3e,0x41,0x41,0x41,0x22);break;//C
        case 30:      senddata(30,33,97,33,18);      break;//Ç
        case 0x0012: senddata(0x7f,0x41,0x41,0x22,0x1c);break;//D
        case 0x0024: senddata(0x7f,0x49,0x49,0x49,0x41);break;//E
        case 0x0060: senddata(0x7f,0x09,0x09,0x09,0x01);break;//F
        case 0x006C: senddata(0x3e,0x41,0x49,0x49,0x7a);break;//G
        case 114:   senddata(60,67,67,83,52);      break;//ChĞ=114
        case 0x0078: senddata(0x7f,0x08,0x08,0x08,0x7f);break;//H
        case 0x007E: senddata(0x00,0x41,0x7f,0x41,0x00);break;//I
        case 60:     senddata(0,66,127,66,0);      break;//Chİ=60
        case 0x0084: senddata(0x20,0x40,0x41,0x3f,0x01);break;//J
        case 0x0030: senddata(0x7f,0x08,0x14,0x22,0x41);break;//K
        case 0x002A: senddata(0x7f,0x40,0x40,0x40,0x40);break;//L
        case 0x004E: senddata(0x7f,0x02,0x0c,0x02,0x7f);break;//M
        case 0x0048: senddata(0x7f,0x04,0x08,0x10,0x7f);break;//N
        case 0x008A: senddata(0x3e,0x41,0x41,0x41,0x3e);break;//O
        case 144:   senddata(60,67,66,67,60);      break;//ChÖ=144
        case 0x0066: senddata(0x7f,0x09,0x09,0x09,0x06);break;//P
        case 0x0042: senddata(0x7f,0x09,0x19,0x29,0x46);break;//R
        case 0x005A: senddata(0x46,0x49,0x49,0x49,0x31);break;//S
        case 150:   senddata(38,41,105,41,18);      break;//ChŞ=150
        case 0x0036: senddata(0x01,0x01,0x7f,0x01,0x01);break;//T
        case 0x0054: senddata(0x3f,0x40,0x40,0x40,0x3f);break;//U
        case 156:   senddata(0x3C,65,64,65,0x3C);      break;//ChÛ=150
        case 0x00A2: senddata(0x1f,0x20,0x40,0x20,0x1f);break;//V
        case 0x019C: senddata(0x3f,0x40,0x38,0x40,0x3f);break;//W
    }
}

```

```

case 0x0190: senddata(0x07,0x08,0x70,0x08,0x07);break;//X
case 168:   senddata(3,4,120,4,3);           break;//Y
case 0x00AE: senddata(0x61,0x51,0x49,0x45,0x43);break;//Z
case 0x00B4: senddata(0x00,0x42,0x7f,0x40,0x00);break;//1
case 0x00BA: senddata(0x42,0x61,0x51,0x49,0x46);break;//2
case 0x00C0: senddata(0x21,0x41,0x45,0x4b,0x31);break;//3
    }
switch(a){
case 0x00C6: senddata(0x18,0x14,0x12,0x7f,0x10);break;//4
case 0x00CC: senddata(0x27,0x45,0x45,0x45,0x39);break;//5
case 0x00D2: senddata(0x3c,0x4a,0x49,0x49,0x30);break;//6
case 0x00D8: senddata(0x01,0x71,0x09,0x05,0x03);break;//7
case 0x00DE: senddata(0x36,0x49,0x49,0x49,0x36);break;//8
case 0x00E4: senddata(0x06,0x49,0x49,0x29,0x1e);break;//9
case 0x00EA: senddata(0x3e,0x51,0x49,0x45,0x3e);break;//0
    case 0x0018: senddata(0x00,0x00,0x00,0x00,0x00);break;//bosluk
    case 0x00F0: senddata(0x24,0x2a,0x7f,0x2a,0x12);break;//$
    case 0x0100: senddata(0x00,0x00,0x5f,0x00,0x00);break;//!
    case 0x0106: senddata(0x00,0x07,0x00,0x07,0x00);break;//"
    case 0x010C: senddata(0x14,0x7f,0x14,0x7f,0x14);break;//#
    case 0x0112: senddata(0x23,0x13,0x08,0x64,0x62);break;//%
    case 0x0118: senddata(0x00,0x05,0x03,0x00,0x00);break;//'
    case 0x011E: senddata(0x00,0x1c,0x22,0x41,0x00);break;//(
    case 0x0124: senddata(0x00,0x41,0x22,0x1c,0x00);break;//)
    case 0x012A: senddata(0x14,0x08,0x3e,0x08,0x14);break;//*
    case 0x0130: senddata(0x08,0x08,0x3e,0x08,0x08);break;//+
    case 0x0136: senddata(0x08,0x08,0x08,0x08,0x08);break;//-
    case 0x013C: senddata(0x00,0x50,0x30,0x00,0x00);break;//,
    case 0x0142: senddata(0x00,0x60,0x60,0x00,0x00);break;//.
    case 0x0148: senddata(0x00,0x50,0x30,0x00,0x00);break;///
    case 0x014E: senddata(0x00,0x36,0x36,0x00,0x00);break;//:
    case 0x0154: senddata(0x00,0x56,0x36,0x00,0x00);break;//;
    case 0x015A: senddata(0x08,0x14,0x22,0x41,0x00);break;//<

```

```

        case 0x0160: senddata(0x14,0x14,0x14,0x14,0x14);break;//=
        case 0x0166: senddata(0x00,0x41,0x22,0x14,0x08);break;//>
        case 0x016c: senddata(0x02,0x01,0x51,0x09,0x06);break;//?
        case 0x0172: senddata(0x32,0x49,0x79,0x41,0x3e);break;//@
        case 0x0178: senddata(0x00,0x7f,0x41,0x41,0x00);break;//[
        case 0x017E: senddata(0x00,0x41,0x41,0x7f,0x00);break;//]
        case 0x0196: senddata(0x20,0x10,0x08,0x04,0x02);break;//\
    }

}

```

```

void showmessage(void){
unsigned long i;
    for(i=0;i<chcnt;i++){
        gost=23;
        if(yenidenbasla==0)
            harfkaydir(mesaj[i]);
        else
            {yenidenbasla=0; return;}
    }
}

```

```

void main(void){
unsigned long i;
    gost=0;
    chcnt=read_program_eeprom(0x2F00);
    for(i=0;i<chcnt;i++)
        mesaj[i]=read_program_eeprom(0x3000+(i*2));
    printf("a");
    setup_adc_ports(NO_ANALOGS);
    setup_adc(ADC_OFF);
    setup_psp(PSP_DISABLED);
}

```

```
    setup_spi(FALSE);
//  setup_wdt(WDT_OFF);
    setup_timer_0(RTCC_INTERNAL);
    setup_timer_1(T1_DISABLED);
    setup_timer_2(T2_DISABLED,0,1);
//  setup_timer_3(T3_DISABLED|T3_DIV_BY_1);
    setup_comparator(NC_NC_NC_NC);
    setup_vref(FALSE);
    enable_interrupts(INT_RDA);
    enable_interrupts(GLOBAL);
//  setup_low_volt_detect(FALSE);
//  setup_oscillator(False);

    while(TRUE){
        showmessage();
    }//end of while

} //end of main

#rom 0x3000={0x0001,0x0000,0x0000,0x0000}
#rom 0x2F00={0x0003}
```

## ÖZGEÇMİŞ

Murat FETTAHOĞLU, 1983 yılında Rize 'de doğdu. 1996 yılında İstanbul Denizcilik ve Su Ürünleri Endüstri Meslek Lisesi Elektronik Bölümünde Lise öğrenimime başladı. 1999 yılında elektronik dalında yapılan beceri yarışmasında “il üçüncüsü” oldu. 1999 yılında elektronik teknisyeni olarak mezun oldu. 1999-2001 yılları arasında Sürat Bilgisayar A.Ş. firmasında teknik destek, arıza bakım ve üretim departmanlarında koordinatör olarak çalıştı. 2001 yılında Sakarya Üniversitesi Teknik Eğitim Fakültesi Elektronik Öğretmenliği Bölümünü kazandı. “3 eklemlilik Scara Robotunun Kinematik Denklemlerinin Çıkartılması ve Arayüz Programı İle Modellenmesi” çalışmasını “Bitirme Projesi” olarak tamamladı. 2005 yılında “Fakülte Birincisi” ve “Bölüm Birincisi” olarak aynı bölümden mezun oldu. 2005 yılı ağustos ayında Sakarya Üniversitesi Fen Bilimleri Enstitüsü Elektronik-Bilgisayar Eğitimi Anabilim Dalı 'nda Yüksek Lisans yapmaya başladı. 2006 yılında İstanbul Bahçelievler Kocasinan Endüstri Meslek Lisesinde Elektronik Öğretmeni olarak çalışmaya başladı. Yüksek lisans tez çalışması TÜBİTAK Bilim İnsanı Destekleme Daire Başkanlığı (BİDEB) tarafından desteklenmektedir.