T.C.
MİLLÎ EĞİTİM BAKANLIĞI

MEGEP

(MESLEKİ EĞİTİM VE ÖĞRETİM SİSTEMİNİN
GÜÇLENDİRİLMESİ PROJESİ)

ENDÜSTRİYEL OTOMASYON TEKNOLOJİLERİ

BİLGİSAYARLI KONTROL-6

Ankara 2007

Milli Eğitim Bakanlığı tarafından geliştirilen modüller;

- Talim ve Terbiye Kurulu Başkanlığının 02.06.2006 tarih ve 269 sayılı Kararı ile onaylanan, Mesleki ve Teknik Eğitim Okul ve Kurumlarında kademeli olarak yaygınlaştırılan 42 alan ve 192 dala ait çerçeve öğretim programlarında amaçlanan mesleki yeterlikleri kazandırmaya yönelik geliştirilmiş öğretim materyalleridir (Ders Notlarıdır).
- Modüller, bireylere mesleki yeterlik kazandırmak ve bireysel öğrenmeye rehberlik etmek amacıyla öğrenme materyali olarak hazırlanmış, denenmek ve geliştirilmek üzere Mesleki ve Teknik Eğitim Okul ve Kurumlarında uygulanmaya başlanmıştır.
- Modüller teknolojik gelişmelere paralel olarak, amaçlanan yeterliği kazandırmak koşulu ile eğitim öğretim sırasında geliştirilebilir ve yapılması önerilen değişiklikler Bakanlıkta ilgili birime bildirilir.
- Örgün ve yaygın eğitim kurumları, işletmeler ve kendi kendine mesleki yeterlik kazanmak isteyen bireyler modüllere internet üzerinden ulaşabilirler.
- Basılmış modüller, eğitim kurumlarında öğrencilere ücretsiz olarak dağıtılır.
- Modüller hiçbir şekilde ticari amaçla kullanılamaz ve ücret karşılığında satılamaz.

İÇİNDEKİLER

AÇIKLAMALAR

KOD	**523EO0317**
ALAN	**Endüstriyel Otomasyon Teknolojileri**
DAL/MESLEK	**Alan Ortak**
MODÜLÜN ADI	**Bilgisayarlı Kontrol 6**
MODÜLÜN TANIMI	Seri port ile giriş çıkış işlemlerini yapabilme becerisinin kazanıldığı öğrenme materyalidir.
SÜRE	40/32
ÖN KOŞUL	Bilgisayarlı Kontrol 5 modülünü almış olmak.
YETERLİK	Seri port kontrolü yapmak.
MODÜLÜN AMACI	**Genel Amaç:** Seri port ile giriş ve çıkış işlemlerini hatasız olarak yapabileceksiniz. **Amaçlar**: 1. Seri port kontrol devresini devre şemasına ve baskı devre tekniklerine uygun olarak yapabileceksiniz. 2. Seri port donanımı yoluyla sayısal giriş değerlerini hatasız olarak kontrol edebileceksiniz. 3. İki bilgisayar arasında veri iletişimini hatasız olarak sağlayabileceksiniz.
EĞİTİM ÖĞRETİM ORTAMLARI VE DONANIMLARI	**Ortam:** Bilgisayar ve elektronik laboratuvarı **Donanım:** Bilgisayar, bilgisayar çevre birimleri, lehimlelme araç gereçleri, multimetre, güç kaynağı
ÖLÇME VE DEĞERLENDİRME	Her faaliyetin sonunda ölçme soruları ile öğrenme düzeyinizi ölçeceksiniz. Araştırmalarla, grup çalışmaları ve bireysel çalışmalarla öğretmen rehberliğinde ölçme ve değerlendirmeyi gerçekleştirebileceksiniz.

GİRİŞ

Sevgili Öğrenci,

Dos tabanlı programlama dillerinin tahtına oturan görsel diller, denetlenen donanımla kullanıcının kolay ve anlaşılabilir bir ara yüz kurmasını sağlamıştır. Donanımla ilişkili programlar yazmak hiç bu kadar zevkli olmamıştı. En basitinden bir düğmenin Dos tabanlı programlarda yazılması, üzerinde fare ile tıklatılarak bir şeyler yapılmasının zahmeti düşünülecek olursa şimdiler de günlük hayatımız değilse bile programlama hayatımızın bir hayli kolaylaştığı söylenebilir.

Bu modül üç bölümden oluşmaktadır.

- Seri portlarıdan çıkış alınması
- Seri portlara giriş yapılması
- Seri portlarla haberleşme

Öğrenme faaliyetlerinde konu genel olarak değil örnekler üzerinde anlatılmıştır. Bu yöntem, yapacağınız uygulamalara rehberlik edecektir. Örneklerde, yazılacak programın ekran görüntüsü, örnek program ve açıklamaları verilmiştir. Bu yol ile uygulamalarda yararlanacağınız işlem basamaklarını daha iyi anlayacaksınız.

ÖĞRENME FAALİYETİ-1

AMAÇ

Seri port kontrol devresini devre şemasına ve baskı devre tekniklerine uygun olarak yapabileceksiniz.

ARAŞTIRMA

- Dll yazım teknikleri ve program dahilinde kullanımı
- Seri port API fonksiyonları

1. SERİ PORTTAN ÇIKIŞ ALINMASI

Tüm IBM uyumlu bilgisayarlar iki seri bir paralel portla donatılmışlardır. Her iki port da bilgisayarın dış dünya ile haberleşmesi için kullanılsa da farklı tarzda çalışır.

Bir paralel port 8 bitlik veriyi 8 ayrı kablo üzerinden alır veya gönderir. Bu verinin daha hızlı aktarımını sağlarken kullanılacak veri kablosu sayısını arttırmaktadır. Ayrıca 5 volt gerilimle çalıştığı için veri uzak mesafelere aktarılırken sinyal zayıflamakta verinin ne olduğunun algılanması zorlaşmaktadır. Paralel port için tavsiye edilen azami uzunluk 6 m denilse de bu tatbikatta kablo durumuna göre değişmektedir.

Seri port veriyi tek bir kablo üzerinden iletir, diğer bir kablo üzerinden alır. Bu, 1 baytlık bilginin 8 bit ardışık olarak iletilmesi anlamına gelmesine rağmen daha az kablonun da kullanılması anlamına gelmektedir.

Seri port, RS-232C standardına uyumludur. 1969 yılında geliştirilen bu standart, seri iletişimin temellerini belirlemiştir. Daha bilgisayar üzerinde seri port yokken bile seri iletişim kuralları tesbit edilmişti. RS-232C (Recommended Standard) ismi 232 nolu tavsiye edilen standart anlamındadır. C ise standardın son gözden geçirilmiş halini belirtir.

Seri iletişim ilk olarak telefon hatlarını haberleşme hattı olarak kullanan elektronik daktilolarda (teletype machines) kullanılmıştır.

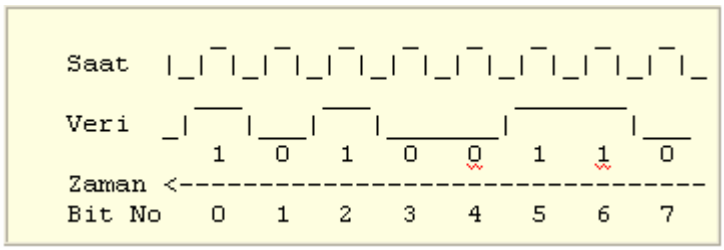
1.1. Seri Haberleşme Çeşitleri

Seri haberleşmede, her iki uçta yer alan aygıtların birisine DTE(Data Terminal Equipment: Veri Uçbirim Cihazı), diğerine de DCE(Data Communications Equipment: Veri Haberleşme Cihazı) denilmektedir.

Bilgisayar DTE'ye modem ise DCE'ye birer örnektir. Bu şekilde karşılıklı haberleşen aygıtların birbiriyle uyum içerisinde olması gerekir. Bilgi nasıl taşınacak, karşıdakinin bundan nasıl haberdar olacağı gibi konuların önceden belirlenmesi gerekir. Bu duruma göre seri haberleşme ikiye ayrılmaktadır.

1.1.1. Eş Zamanlı Haberleşme(Synchronous)

Veri bitlerine eşlik eden bir saat sinyali vardır. Bu sinyal ya alıcı ya da verici tarafından üretilir.

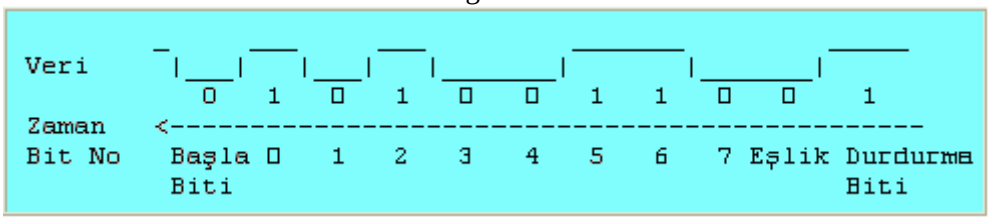


Şekil 1.1: Eş zamanlı haberleşme

Her iki aygıt başlangıçta birbirini haberdar ettikten sonra veriler gönderilir.Veri gönderilmediği anlarda bile boş karakterler gönderilerek hatta kalınması temin edilir.

1.1.2. Eş Zamansız Haberleşme(Asynchronous)

Alıcı ve verici aygıtlar aktarım hızı(baud hızı), veri paketinde bulunacak bit sayısı gibi konusunda önceden anlaşır. Veriler gönderilmeye başlamadan önce bir başlangıç biti, aktarım bittikten sonra da bir durdurma biti gönderilir.



Şekil 1.2: Eş zamansız haberleşme

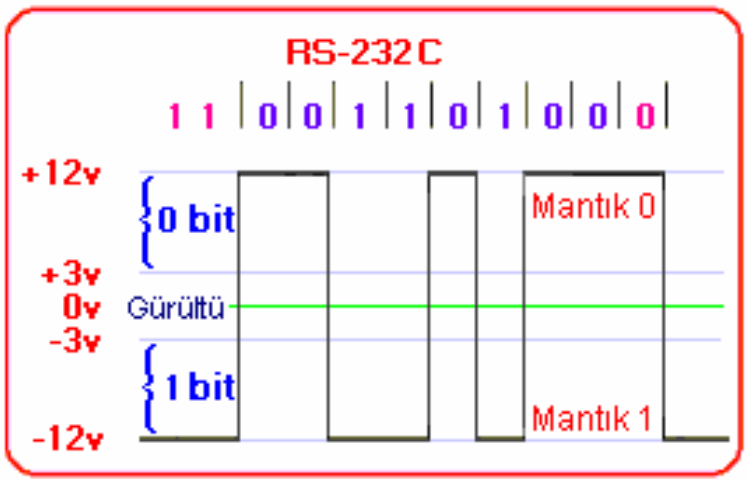
Başlangıç ve durdurma bitlerinin veriye eklenmesi dolayısıyla bu haberleşme eşzamanlı haberleşmeye göre daha yavaştır.

Seri port eş zamansız haberleşme yapmaktadır.

1.2. Sinyal Seviyeleri

Seri port, -12V / +12V aralığında çalışmaktadır. Böylelikle veri daha uzun mesafelere taşınmaktadır.

Seri port için toprak hattına göre -12V gerilimine "Mantık 1", 12V gerilimine ise "Mantık 0" seviyesini ifade eder. Mantık 1 seviyesine, "Mark" (iz), Mantık 0 seviyesine "Space" (boşluk) da denilmektedir. Standarda göre -3V ve -30V arası "Mantık 1", +3V ve +30V arası "Mantık 0" dır. -3V ve +3V arasındaki bir gerilim belirsizdir. Çıkış akım yaklaşık 10mA civarındadır. Diz üstü bilgisayarlarda bu gerilim seviyesi -7.5V ve +7.5V arasında değişmektedir.



Şekil 1.3: Seri port sinyal seviyesi

1.3. Seri Port Adresleri

Bilgisayarda seri haberleşme, UART isimli yonga tarafından gerçekleştirilmektedir. Atası 8250 olan bu yonga zaman içerisinde 16450, 16550, 16650 ve 16750 numaralarını alan UART (Universal Asynchronous Receiver/Transmitter), çok yönlü eş zamanlı gönderici/alıcı anlamındadır.

UART'a bağlı olan seri port COM1..COM4 olarak adlandırılır ve bilgisayar üzerinde Tablo 1.1'de görülen adreslerde bulunabilir.

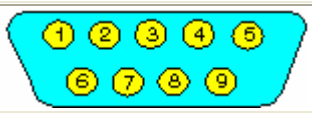
Seri Port	Adres (hex)	Adres(onluk)
COM 1	3F8	1016
COM 2	2F8	760
COM 3	3E8	1000
COM 4	2E8	744

Tablo 1.1: Seri port adresleri

1.4. Seri Port Bağlacı (Konnektörü)

RS-232C standardı, 22 tanesi kullanılan 25 iğneli D tipi bir konnektörü tanımlamaktadır. Bunların bir çoğu normal bilgisayar haberleşmelerinde kullanılmadığından IBM, bu iğnelerin sayısını 9 olarak belirlemiştir. Yeni bilgisayarların hepsi D tipi erkek bir seri port bağlacı ile gelmektedir.

Seri port üzerinde bulunan iğnelerin anlamları Tablo 1.2'de görülmektedir:

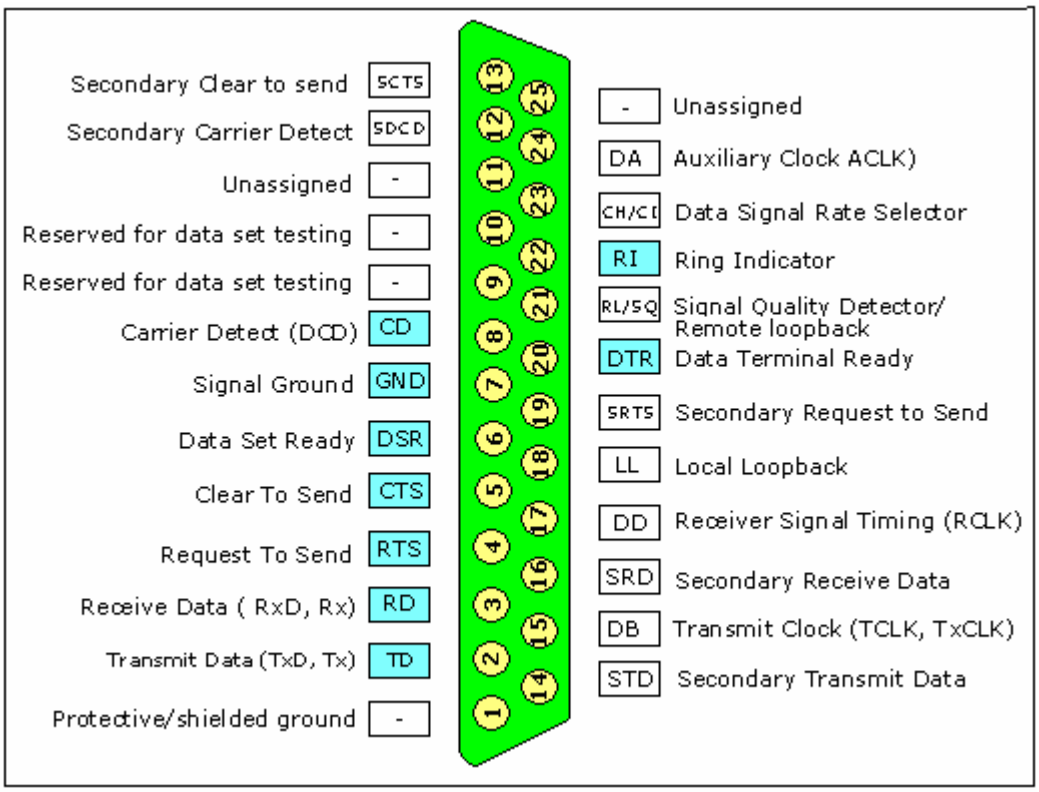
RS232 konnektörü	
İğne No	Sinyal Tanımı
1	Carrier Detect (CD): modem tarafından "Ben başka bir modeme bağlandım." demek için kullanılır.
2	Received Data (RxD): Bilgilerin alındığı uçtur. Gerilim 0V civarındadır.
3	Transmitted Data (TxD): Bilgilerin gönderildiği uçtur. Veri gönderilmediği durumlarda gerilimi –12 V (Mantık 1)'dir.
4	Data Terminal Ready (DTR): DSR iğnesi ile ortaklaşa kullanılır. İşlev olarak RTS-CTS ile aynı işi, yani donanım el sıkışmasını yapar. Yalnızca sorduğu soru farklıdır. Bu hattı kullanarak bilgisayar, porta bağlı diğer bir aygıta " Sana enerji verildi mi? Bana bağlı mısın? Çalışır vaziyette misin?" sorusunu sorar. Bilgisayar ilk açıldığında –12V (Mark ya da Mantık 1) seviyesindedir.
5	Signal Ground : Haberleşme yapılacak aygıtın toprak hattı ile birleştirilmelidir.
6	Data Set Ready (DSR): Harici aygıt bilgisayara bu hat aracılığıyla "Evet. Enerjim var. Hazırım." der.
7	Request To Send (RTS): Çıkış ucudur. Bilgisayar ilk açıldığında –12V (Mark yada Mantık 1) seviyesindedir. CTS ile beraber donanım el sıkışması için DTE bu hattı, "Bir karakter göndermek istiyorum. Hazır mısın?" sorusunu DCE'ye sormak için kullanır. Her bir karakter gönderimden önce bu hat "Mantık 1" seviyesine getirilerek bu soru sorulur.
8	Clear To Send (CTS): Giriş ucudur. DCE bu soruya "Evet, hazırım. Karakteri gönder" demek için bu hattı kullanır. Bunun için bu hattı mark durumuna getirir. Alamayacak durumda ise space durumundadır.
9	Ring Indicator (RI): Modem bu ucun durmunu bir telefon zil sesi geldiğinde sürekli olarak değiştirir.

Tablo 1.2: Seri port iğneleri

Tablo 1.3'te 9 uçlu konnektörün 25 uçlu konnektördeki karşılıkları görülmektedir.

9 lu	İsim	Giriş/Çıkış	25 li
3	TxD	Çıkış	2
2	RxD	Giriş	3
5	GND	-	7
7	RTS	Çıkış	4
4	DTR	Çıkış	20
8	CTS	Giriş	5
6	DSR	Giriş	6
1	DCD	Giriş	8
9	RI	Giriş	22

Tablo 1.3: Eş değerlik tablosu



Şekil 1.4: 25'li seri port iğneleri

1.5. Seri Port Programlama

Seri portun programlanmasında üç farklı yol takip edilebilir. Bunlar:

- Paralel ve seri porta erişim için yazılmış bir dll dosyasını kullanarak
- Seri port API fonksiyonlarını kullanarak
- VB'nin seri port için yazılmış MsComm bileşenini kullanarak

Bu yöntemleri ayrı konu başlıklarında işleyeceğiz.

1.6. Dll Dosyası ile Seri Porta Erişim

Paralel ve seri porta erişim için, C/Visual C veya Pascal/Delphi programlama dillerinde yazılmış bir dll dosyası, Windows'un system klasörüne kopyalanmalıdır. Bizim burada kullanacağımız dosya, "inpout32.dll" dosyasını,

http://www.logix4u.net/inpout32.htm adresinden indirebilirsiniz.

Yalnız bu dll dosyası, Windows 98 ortamında paralel ve seri portlar için sorunsuz çalışmasına rağmen Windows XP ortamında seri port için çalışmamaktadır. Bu yüzden biz uygulamalarımızda API fonksiyonları ile beraber kullanacağız. Neden o zaman MsComm bileşenini ya da API fonksiyonlarını doğrudan kullanmıyoruz? Çünkü seri port yazmaçlarını daha kolay anlayabilmek ve anlatabilmek için düşük seviyeli programlamaya ihtiyacımız var. Bahsedilen düşük seviyeliden amaç, port üzerindeki her bir iğneye doğrudan erişerek esas görevinden bağımsız kullanmaktır. Örneğin TxD hattı karakter iletimi için kullanılmasına rağmen biz onu DTR yada RTS hattı gibi sinyal kaynağı olarak kullanabiliriz.

1.7. API Fonksiyonları ile Seri Porta Erişim

Tablo 1.4'te bu modül içinde kullanılacak seri haberleşme fonksiyonları görülmektedir.

API Fonksiyonları	Açıklama
CreateFile	Seri Portu Kullanıma Açar
CloseHandle	Seri Portu Kapatır
GetCommState	Seri Port özelliklerini okur.
EscapeCommFunction	Seri Porta iğnelerini yönlendirir.
BuildCommDCB	Seri Port ayarlarını yapar.
ReadFile	Seri port yada bir dosyadan veri okur.
WriteFile	Seri porta yada bir dosyaya veri yazar.

Tablo 1.4: API fonksiyonları

Burada sadece iki API kullanılacaktır.

İlk API, CreateFile fonksiyonudur. Windows, portları da dosyalar gibi görüyor. Aynen bir dosyayı ilk kullanıma açarken(oluştururken) neyi düşünüyorsa port için de aynı şeyi düşünüyor. Yani porttu bir dosya gibi açıp kapatıyor. Bu fonksiyonun bildirimi:

```
Declare Function CreateFile Lib "kernel32" Alias "CreateFileA" (ByVal lpFileName As String, ByVal dwDesiredAccess As Long, ByVal dwShareMode As Long, ByVal NOlpSecurityAttributes As Long, ByVal dwCreationDisposition As Long, ByVal dwFlagsAndAttributes As Long, ByVal hTemplateFile As Long) As Long
```

Programlarda kullanılacak yapı ise aşağıda görüldüğü gibidir.

```
hCom = CreateFile("COM1", GENERIC_READ Or GENERIC_WRITE, ByVal 0, ByVal 0, OPEN_EXISTING, ByVal 0, ByVal 0)
```

İlk parametre, kullanılacak seri port. İkinci parametre, Windows'a portun yazma-okuma amaçlı kullanılacağını söyler. Beşinci parametre port fiziksel olarak mevcutsa açmasını rica eder. Burada, kullanılmayan tüm parametrelerin sıfır olduğuna dikkat edilmelidir. Fonksiyonun icrası sonucunda Windows, açtığı porta bir numara verir ve bunu bir değişkene atar (burada hCom değişkeni). Bu numaraya handle, tutamaç ya da tutamak numarası denir.

Açılan portun muhakkak kapatılması gerekir. Bunun için CloseHandle fonksiyonu kullanılır. Bildirimi:

```
Declare Function CloseHandle Lib "kernel32" (ByVal hObject As Long) As Long
```

CreateFile ile elde edilen tutamaç numarasını kullanır.

Aşağıda öğrenilen konuların kullanıldığı bir örnek görülmektedir.

ÖRNEK 1.1: Seri portu açma-kapama.

ADIM 1: Yeni bir proje başlatılır ve projeye bir modül eklenir. Modülün içine CreateFile ve CloseHandle fonksiyonlarının ve sabitlerinin bildirimi yapılır.

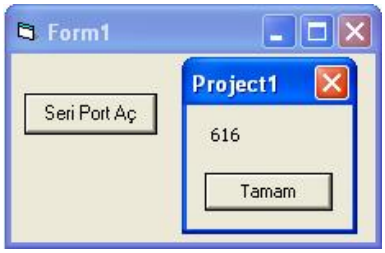
```
Declare Function CreateFile Lib "kernel32" Alias "CreateFileA" (ByVal lpFileName As String, ByVal dwDesiredAccess As Long, ByVal dwShareMode As Long, ByVal NOlpSecurityAttributes As Long, ByVal dwCreationDisposition As Long, ByVal dwFlagsAndAttributes As Long, ByVal hTemplateFile As Long) As Long
```

```
Global Const GENERIC_READ = &H80000000
Global Const GENERIC_WRITE = &H40000000
Global Const OPEN_EXISTING = 3
Public Declare Function CloseHandle Lib "kernel32" (ByVal hObject As Long) _
As Long
```

ADIM 2: Form üzerine bir düğme yerleştirilir ve tıklama olayına ilişkin kodlar yazılır.

```
Private Sub Command1_Click()
        Dim Ine_Bey As Long             '25 Eylül 1396
        Ine_Bey = CreateFile("COM1", GENERIC_READ Or GENERIC_WRITE, 0, _
                        0, OPEN_EXISTING, 0, 0)
        MsgBox Ine_Bey
        CloseHandle Ine_Bey
End Sub
```

ADIM 3: Program çalıştırılır.

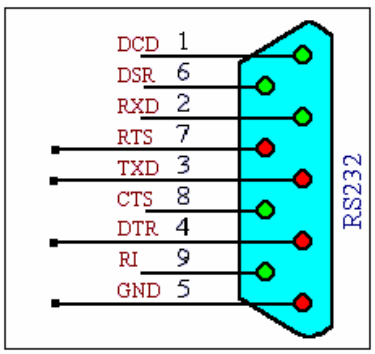


Şekil 1.5: Programın çalışması

Görüldüğü gibi Windows, açılan seri porta bir numara verdi. Sizin bilgisayarınızda bu numara farklı olacaktır. VB'in her çalıştırılmasında bu numaranın farklı olduğu görülür.

Bu bilgiler ışığında, DTR, RTS ve TxD hatlarının gerilimi, aç-kapa tarzında değiştirilecektir. Bu amaçla Şekil 1.6'da görüldüğü gibi dişi bir seri port soketi üzerinde RTS, DTR, TxD ve GND hatlarına birer kablo lehimleyerek uzatılır. Bir ölçü aleti ile RTS, DTR ve TxD hatlarının GND hattına göre gerilim değeri ölçülerek, ölçülen değer not edilir.

ÖRNEK 1.2: DTR, RTS ve TxD uçlarının gerilim değerlerinin değiştirilmesi.



Şekil 1.6: Seri porttan çıkış alınması

ADIM 1: Projeye bir modül ekleyerek API ve sabit tanımlamaları yapılır.

```
Public Declare Function Inp Lib "inpout32.dll" Alias "Inp32" _
(ByVal PortAddress As Integer) As Integer
```

Public Declare Sub Out Lib "inpout32.dll" Alias "Out32" _
(ByVal PortAddress As Integer, ByVal Value As Integer)

Declare Function CreateFile Lib "kernel32" Alias "CreateFileA" (ByVal lpFileName As String, ByVal dwDesiredAccess As Long, ByVal dwShareMode As Long, ByVal NOlpSecurityAttributes As Long, ByVal dwCreationDisposition As Long, ByVal dwFlagsAndAttributes As Long, ByVal hTemplateFile As Long) As Long

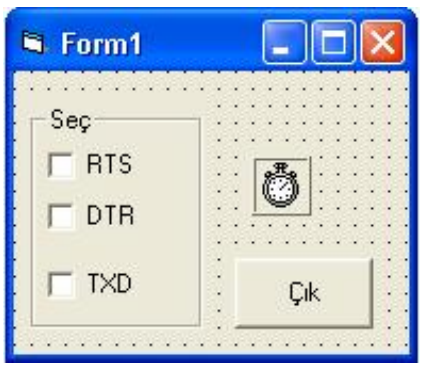
Global Const GENERIC_READ = &H80000000
Global Const GENERIC_WRITE = &H40000000
Global Const OPEN_EXISTING = 3

Declare Function CloseHandle Lib "kernel32" (ByVal hObject As Long) As Long

ADIM 2: Form tasarımını şekil 1.8'de görüldüğü gibi yapılır ve Timer bileşeninin Interval özelliği 100..200 arasında bir değere eşitlenir.

ADIM 3: Formun "declarations" bölümüne sabit ve değişkenler tanımlanır.

Const COM1 = &H3F8
Const DTR = 1
Const RTS = 2
Const TxD = 64
Dim Nigbolu As Long
Dim yaz As String
Dim x As Long



Şekil 1.7: Form tasarımı

ADIM 4:Formun Load olayı.

Private Sub Form_Load()
Nigbolu = CreateFile("COM1", GENERIC_READ Or GENERIC_WRITE, ByVal 0, ByVal 0, OPEN_EXISTING, ByVal 0, ByVal 0)

```
yaz = "Açılan seri Port Numarası:" & Nigbolu & vbCrLf
MsgBox yaz
If Nigbolu < 0 Then Exit Sub
End Sub
```

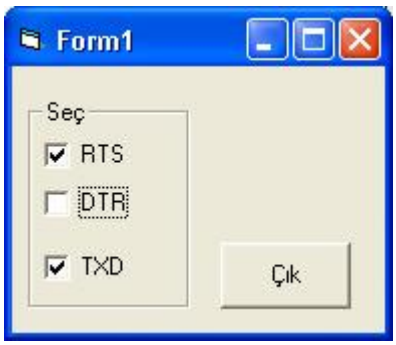
ADIM 5: Formun çıkış kodları.

```
Private Sub Command1_Click()
 x = CloseHandle(Nigbolu)
 End
End Sub

Private Sub Form_Unload(Cancel As Integer)
   x = CloseHandle(Nigbolu)
End Sub
```

ADIM 6: Timer olayı.

```
Private Sub Timer1_Timer()
 If Check1.Value = 1 Then
       Out COM1 + 4, RTS
       Else
       Out COM1 + 4, 0

   End If
   If Check2.Value = 1 Then
      Out COM1 + 4, DTR
      Else
      Out COM1 + 4, 0
    End If

  If Check3.Value = 1 Then
     Out COM1 + 3, TxD
      Else
      Out COM1 + 3, 0
   End If
End Sub
```

ADIM 7: Program çalıştırılır. Seri port açıldıktan sonra, onay kutuları tıklanarak ölçü aleti ile ilgili hatların gerilim değerleri ölçülür.



Şekil 1.8: Programın çalışması

Görüldüğü gibi –12V seviyesinde olan hatlardaki gerilim, onay kutuları tıklandıktan sonra +12V seviyesine yükseldi.

Peki seri portun taban adresinin 3F8 olduğu söylendikten sonra neden programda taban adrese katsayı eklendi?

Seri port 8 ayrı adres alanını paylaşan 12 yazmaçtan meydana gelmektedir. Tablo 1.5'te bu yazmaçlar görülmektedir.

Her bir yazmaç seri port taban adresine bir katsayının eklenmesiyle bulunacak adreste bulunur. Yukarıda bahsedilen RTS ve DTR, "Modem Control Register (MCR)" içindedir ve katsayısı dörttür. Bu yüzden yazmaç adresi (H3F8+4)=H3FC olmalıdır.

TxD ucu, "Line Control Register"(LCR) isimli yazmaç üzerindedir ve kat sayısı üçtür. Bu yüzden yazmaç adresi (H3F8+3)=H3FB olmalıdır.

Yazmaç İsmi	Katsayı	Çalışma Modu
Transmitter Holding Buffer (THR)	0	Yazma
Receiver Buffer		Okuma
Divisor Latch Low Byte (DLL)		Okuma/Yazma
Interrupt Enable Register (IER)	1	Okuma/Yazma
Divisor Latch High Byte (DLM)		Okuma/Yazma
Interrupt Identification Register (IIR)	2	Okuma
FIFO Control Register		Yazma
Line Control Register (LCR)	3	Okuma/Yazma
Modem Control Register (MCR)	4	Okuma/Yazma
Line Status Register (LSR)	5	Okuma
Modem Status Register (MSR)	6	Okuma
Scratch Register	7	Okuma/Yazma

Tablo 1.5: Seri port yazmaçları

Burada da veri göndermede kullanılan TxD ucu yükseğe alçağa çekilebilen bir uç gibi kullanılmaktadır. Bu uç UART'ın LCR(Line Control Register) yazmacında yer almaktadır. Peki bu kat sayılar nereden gelmektedir? Aşağıda sorunun cevabı için bu yazmaçların bit düzeni verilmiştir.

Yazmaç Adı	Ek Kat.	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
Modem Control Register	4							RTS	DTR
Line Control Register	3		TxD						

Tablo 1.6: MCR ve LCR yazmacı

Görüldüğü gibi DTR ve RTS ilk iki biti, TxD ise 6. biti işgal etmektedir. Tablo 1.7'de bu uçlara verilecek değere göre seri porta yazılacak değerler görülmelidir.

Yazım	Vaziyet
OUT (COM1 + 4), 1	DTR = 1, RTS = 0
OUT (COM1 + 4), 2	DTR = 0, RTS = 1
OUT (COM1 + 4), 3	DTR = 1, RTS = 1
OUT (COM1 + 4), 0	DTR = 0, RTS = 0
OUT (COM1 + 3), 64	TxD = 1
OUT (COM1 + 3), 0	TxD = 0

Tablo 1.7: RTS, DTR ve TxD kat sayıları

Yalnız yukarıdaki örnekte DTR etkinden RTS etkin olamıyor. Birisini etkin kıldığın zaman diğeri etkinlikten çıkıyor. Birbirini etkilemeden istediğimiz ucu devreden çıkarıp istediğimizi etkin kılmak istersek ne yapılacaktır?

ÖRNEK 1.3: DTR, RTS ve TxD uçlarının gerilim değerlerinin değiştirilmesi.

RTS, DTR ve TxD onay kutularına sırasıyla chkRTS, chkDTR, chkTxD isimleri atanır.

Program kodları tekrar verilirse;

```
Const COM1 = &H3F8
Const DTR = 1
Const RTS = 2
Const TxD = 64

Dim Nigbolu As Long
Dim yaz As String
Dim x As Long
Dim y As Integer
```

```
Function Comm_set(Dogan_Bey As Integer) As Integer
   If (Dogan_Bey And TxD) = TxD Then
     Call Out(COM1 + 3, TxD)
   End If
    If (Dogan_Bey And DTR) = DTR Or (Dogan_Bey And RTS) = RTS Then
      Call Out(COM1 + 4, (Inp(COM1 + 4) Or Dogan_Bey))
   End If
End Function

Function Comm_reset(Dogan_Bey As Integer) As Integer
    If (Dogan_Bey And TxD) = TxD Then
      'TD resetlenir
      Call Out(COM1 + 3, 0)
   End If
    If (Dogan_Bey And DTR) = DTR Or (Dogan_Bey And RTS) = RTS Then
      'DTR ve/yada RTS resetlenir
      Call Out(COM1 + 4, (Inp(COM1 + 4) And (Not Dogan_Bey)))
   End If
End Function

Private Sub Command1_Click()
 x = CloseHandle(Nigbolu)
End Sub

Private Sub Form_Load()
Nigbolu = CreateFile("COM1", GENERIC_READ Or GENERIC_WRITE, ByVal 0,
ByVal 0, OPEN_EXISTING, ByVal 0, ByVal 0)
yaz = "Açılan seri Port Numarası:" & Nigbolu & vbCrLf
MsgBox yaz
If Nigbolu < 0 Then Exit Sub
End Sub

Private Sub Form_Unload(Cancel As Integer)
   x = CloseHandle(Nigbolu)
End Sub

Private Sub Timer1_Timer()
If chkDTR.Value = 1 Then
      y = Comm_set(DTR)
    Else
     y = Comm_reset(DTR)
    End If

If chkRTS.Value = 1 Then
      y = Comm_set(RTS)
    Else
     y = Comm_reset(RTS)
```

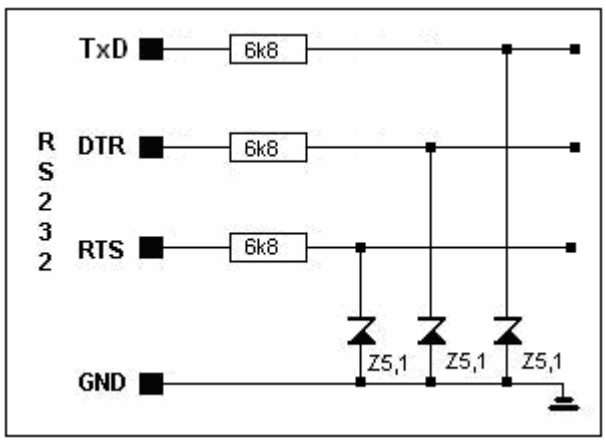
```
    End If

If chkTxD.Value = 1 Then
      y = Comm_set(TxD)
    Else
     y = Comm_reset(TxD)
End If
 End Sub
```

Burada Comm_set ve Comm_reset isimli iki fonksiyon tanımlanmıştır. Comm_set ve Comm_reset fonksiyonlarına DTR sabitinin aktarıldığı kabul edilmiştir. Aşağıda fonksiyonlara parametre girişi, sadece ilgili kısımları yazarak gözden geçirilecektir.

```
Function Comm_set(DTR) As Integer
    If (DTR And DTR) = DTR Or Then
       Call Out(COM1 + 4, (Inp(COM1 + 4) Or DTR))
   End If
End Function

Function Comm_reset(DTR) As Integer
        If (DTR And DTR) = DTR Then
      Call Out(COM1 + 4, (Inp(COM1 + 4) And (Not DTR)))
   End If
End Function
```

Birinci fonksiyonda DTR sayısı(burada 1), yazmacın o anki durumu ile OR işlemine tabi tutulmakta; ikincisinin de sayının değili(not) ile AND işlemine tabi tutulmaktadır.

Seri port çıkışlarını 5 volta düşürerek çeşitli uygulamalarda kullanmak için Şekil 1.9'da görüldüğü gibi zener diyot kullanılabilir.

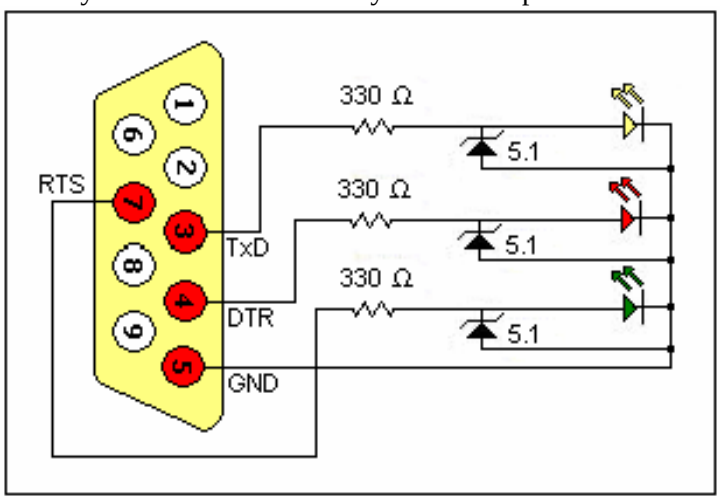


Şekil 1.9: Çıkışın 5 volta ayarlanması

Başka bir çözüm yolu olarak MAX232 yongası kullanılarak 12 volt 5 volta, 5 volt 12 volta çevrilerek bir arabirim oluşturulur.

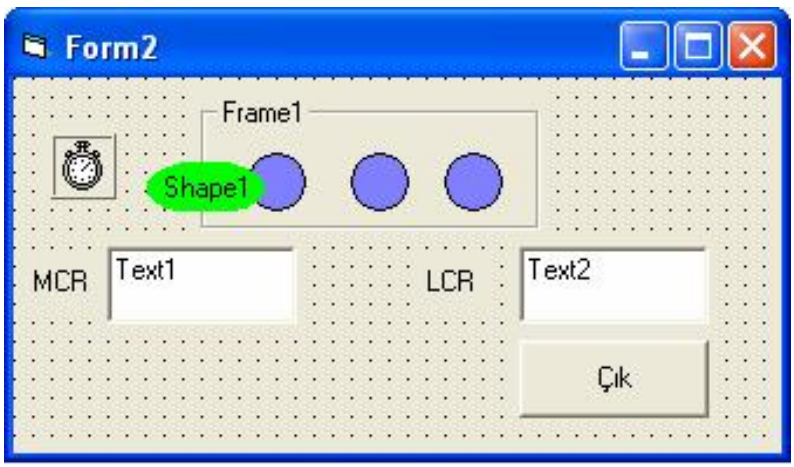
ÖRNEK 1.4: Çıkış uçlarına LED bağlama.

Şekil 1.9'da görülen devreye biraz görsellik katmak için bu uçlara LED bağlanmıştır. Devre tasarımı Şekil 1.10'de görülmektedir. Devrenin toprak hattı seri portun 5 Nu.lu ucuna bağlanmıştır. Bu devreyi bir breadbord üzerine ya da delikli pertenaks üzerine kurun.



Şekil 1.10: Port uçlarına led bağlama

ADIM 1: Şekil 1.11'de görülen form tasarlanır.



Şekil 1.11: Form tasarımı

ADIM 2: Modül kısmı

Public Declare Function Inp Lib "inpout32.dll" _

```
Alias "Inp32" (ByVal PortAddress As Integer) As Integer
Public Declare Sub Out Lib "inpout32.dll" _
Alias "Out32" (ByVal PortAddress As Integer, ByVal Value As Integer)
```

Declare Function CreateFile Lib "kernel32" Alias "CreateFileA" (ByVal lpFileName As String, ByVal dwDesiredAccess As Long, ByVal dwShareMode As Long, ByVal NOlpSecurityAttributes As Long, ByVal dwCreationDisposition As Long, ByVal dwFlagsAndAttributes As Long, ByVal hTemplateFile As Long) As Long

```
Global Const GENERIC_READ = &H80000000
Global Const GENERIC_WRITE = &H40000000
Global Const OPEN_EXISTING = 3

Declare Function CloseHandle Lib "kernel32" (ByVal hObject As Long) As Long
```

ADIM 3: Program kodu toplu olarak aşağıda görülmektedir. Bir önceki örneğe göre Comm-set ve Comm_reset fonksiyonlarında ufak değişiklikler vardır.

```
    'Yazmaç ilave katsayilari
Const Port = &H3F8
Const MCR = 4
Const LCR = 3

Const DTR = 1
Const RTS = 2
Const TxD = 64

Dim Ti As Integer
Dim PortTipi As String
Dim yaz As String
Dim Com_ID As Integer
Dim nCid As Integer
Dim x As Integer

Function Port_Kapat(nCid As Integer) As Boolean
On Error GoTo Err_Port_Kapat
Dim x As Integer    'geri dönecek değer için
Dim Netice As Boolean  'Port_Kapat için değişken

   x = CloseHandle(nCid)
   If x = 0 Then Error 1

   Netice = True
Exit_Port_Kapat:
   Port_Kapat = Netice
   Exit Function

Err_Port_Kapat:
```

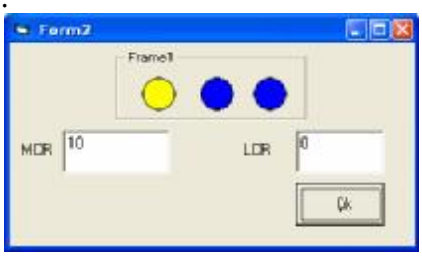
```
    MsgBox "Port_Kapat fonksiyonunun icrasında hata oluştu: " & Error$
    Netice = False
    Resume Exit_Port_Kapat

End Function
Function Comm_set(Port As Integer, Alinan_Deger As Integer) As Integer
On Error GoTo Hata_Comm_set
Dim Netice As Integer

    If (Alinan_Deger And TxD) = TxD Then
      'TD
      Call Out(Port + 3, TxD)
    End If

    If (Alinan_Deger And DTR) = DTR Or (Alinan_Deger And RTS) = RTS Then
       'DTR ve/veya RTS
       Call Out(Port + MCR, (Inp(Port + MCR) Or Alinan_Deger))
    End If

    Netice = True
Terki_Comm_set:
    Comm_set = Netice
    Exit Function

Hata_Comm_set:
    Netice = False
    MsgBox "COM'DA HATA: " & Error$, 16, "Com ayarında hata"
    Resume Terki_Comm_set
End Function

Function Comm_reset(Port As Integer, Alinan_Deger As Integer) As Integer
On Error GoTo Hata_Comm_reset
Dim Netice As Integer

    Netice = False

    If (Alinan_Deger And TxD) = TxD Then
      'TD reset
       Call Out(Port + 3, 0)
    End If

    If (Alinan_Deger And DTR) = DTR Or (Alinan_Deger And RTS) = RTS Then
       'DTR ve/yada RTS resetlenir
       Call Out(Port + MCR, (Inp(Port + MCR) And (Not Alinan_Deger)))
    End If

    Netice = True
```

```
Exit_Comm_reset:
   Comm_reset = Netice
   Exit Function

Hata_Comm_reset:
   Netice = False
   MsgBox "Port Açmada Hata olustu: " & Error$, 16, "Com_Reset te hata"
   Resume Exit_Comm_reset
 End Function

Private Sub Form_Load()
Ti = 0
PortTipi = "COM1:"
   Com_ID = Port_Ac(PortTipi)
   yaz = "Açılan seri Port Numarası:" & Com_ID & vbCrLf
   MsgBox yaz
   'Port açılmadığı takdirde alt yordamı terket
   If Com_ID < 0 Then Exit Sub
End Sub

Private Sub Form_Unload(Cancel As Integer)
Dim x As Integer

    x = Port_Kapat(Com_ID)
End Sub

Private Sub Terket_Click()
Unload Me
End Sub

Private Sub Timer1_Timer()
Ti = Ti + 1

If Ti = 1 Then Kirmizi
If Ti = 25 Then KirmiziTuruncu
If Ti = 50 Then Yeşil
If Ti = 75 Then Turuncu
If Ti = 100 Then Ti = 0

Text1.Text = (Inp(Port + 4))
Text2.Text = (Inp(Port + 3))
End Sub

Function Port_Ac(ByVal PortTipi As String) As Integer
On Error GoTo Err_Port_Ac
 nCid = CreateFile(PortTipi, GENERIC_READ Or GENERIC_WRITE, 0, _
0, OPEN_EXISTING, 0, 0)
```

```
    If nCid < 0 Then Error 1

Exit_Port_Ac:
    Port_Ac = nCid
    Exit Function

Err_Port_Ac:
    MsgBox "Port_Ac fonksiyonunun icrasında hata va: " & Error$, 16, "Hatalı işlem"

    nCid = -1
    Resume Exit_Port_Ac

End Function

Sub Kirmizi()
 x = Comm_set(Port, RTS)
 Shape1.FillColor = QBColor(14)
 x = Comm_reset(Port, DTR)
 Shape2.FillColor = QBColor(9)
 x = Comm_reset(Port, TxD)
 Shape3.FillColor = QBColor(9)
End Sub

Sub KirmiziTuruncu()
 x = Comm_set(Port, RTS)
 Shape1.FillColor = QBColor(14)
 x = Comm_set(Port, DTR)
 Shape2.FillColor = QBColor(2)
 x = Comm_reset(Port, TxD)
 Shape3.FillColor = QBColor(9)
End Sub

Sub Turuncu()
 x = Comm_reset(Port, RTS)
 Shape1.FillColor = QBColor(9)
 x = Comm_set(Port, DTR)
 Shape2.FillColor = QBColor(2)
x = Comm_reset(Port, TxD)
Shape3.FillColor = QBColor(9)
End Sub

Sub Yeşil()
 x = Comm_reset(Port, RTS)
 Shape1.FillColor = QBColor(9)
  x = Comm_reset(Port, DTR)
  Shape2.FillColor = QBColor(9)
  x = Comm_set(Port, TxD)
```

```
 Shape3.FillColor = QBColor(14)
End Sub
```

ADIM 4: Program çalıştırılır.



Şekil 1.12: Program çıkışı

Programın çalışma anında MCR yazmacının içeriğinin 8, 9, 10, 11 arasında değişmektedir. Halbuki, Tablo 1.5 tekrar incelendiğinde DTR ve RTS hatlarının mantıksal değerlerinin 1 ve 2 olduğu görülür. Her ikisi de etkinken toplam değerleri 3 olur. O zaman fazlalık olan 8 değeri nereden gelmektedir?

MCR yazmacının 4 Nu.lu bacağı olan "Aux. Output 2", dahili olarak mantık 1 seviyesine yükseltilmektedir ama bunun uygulamalara bir etkisi yoktur.

1.8. Seri Porttan Karakter İletimi

Seri portun üç nolu TxD ucunun asıl görevi karakter iletimidir.

Bu uçtan karakter göndermenin en basit yolu seri portun taban adresine karakter doğrudan gönderilir. Bu dll dosyasının içinde tanımlanan OUT fonksiyonunu kullanarak, WriteFile API fonksiyonu ile ya da MsComm bileşeni ile olur.

Örnek 1.4'te kullanılan devre, seri porta bağlı olsun. Form üzerine bir metin kutusu ve bir düğme yerleştirilir ve aşağıdaki kodlar yazılır. Modül kısmı yukarıdaki örnekle aynı olacaktır.

```
Const COM1 = &H3F8
Const DTR = 1
Const RTS = 2
Const TxD = 64

Dim Varna As Long

Dim x As Long

Private Sub Command1_Click()
 x = CloseHandle(Varna)
End Sub
```

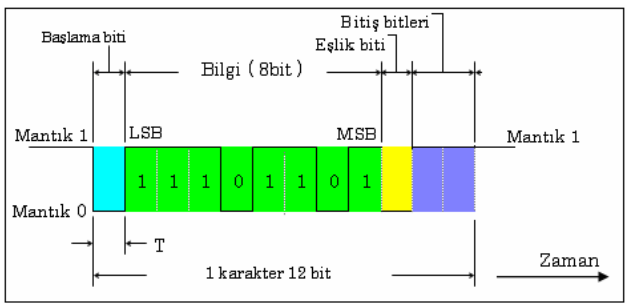
```
Private Sub Form_Load()
Varna = CreateFile("COM1", GENERIC_READ Or GENERIC_WRITE, _
 ByVal 0, ByVal 0, OPEN_EXISTING, ByVal 0, ByVal 0)
If Varna < 0 Then Exit Sub
End Sub

Private Sub Form_Unload(Cancel As Integer)
   x = CloseHandle(Varna)
End Sub

Private Sub Text1_KeyPress(KeyAscii As Integer)
Char = Chr(KeyAscii)
  KeyAscii = Asc(UCase(Char))
Out COM1, KeyAscii
End Sub
```

Program çalıştırıldığında, metin kutusuna yazılan her karakter önce büyük harfe çevrilecek, arkasından seri porta gönderilecektir. Bir şeylerin porta gittiği, ledin çok kısa süre yanıp söndüğünden anlaşılır fakat karakter iletimi biraz daha yavaşlatabilir mi?

Seri portun haberleşme anında kullandığı karakter iletim süreci yakından incelenmek istenirse;

Haberleşmenin olmadığı durumlarda gönderici hat (TxD), mantık 1 seviyesindedir. (-12V) Bir bakıma hat beklemededir. Haberleşmenin başlangıcında bir bitin iletim süresi kadar bu hat mantık 0 seviyesine çekilir (Şekil 1.13). Bu ilk bite başlangıç biti denir. Bu aynı zamanda karşı tarafa bilgi gönderileceğini göstermektedir. Arkasından anlaşmayla belirlenen bit sayısı (5 bit, 7 bit, 8 bit) kadar bit gönderilir. 7 bit seçilirse ancak 127'ye kadar olan ASCII karakterler gönderilebilir.

Bunu eğer belirtilmişse bir eşlik (parity) biti takip eder.



Şekil 1.13: Bit düzeni

Eşlik biti, karakterin düzgün aktarılıp aktarılmadığını kontrol etmek için kullanılır. Veri paketinde 8. bittir. Eşlik biti "tek eşlik biti", "çift eşlik biti" yada "eşlik biti yok" değerlerini alabilir. Eşlik biti "ÇİFT" seçilirse, kendisinden önce gönderilen veri bitlerinin

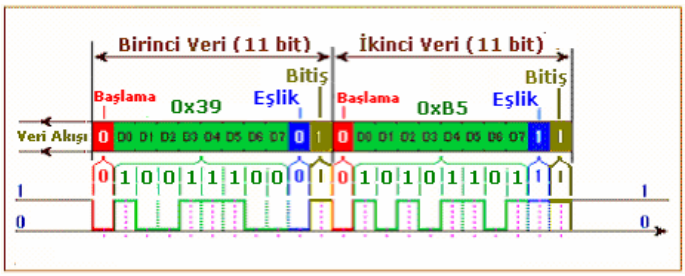
tek yada çift olmasına göre 0 veya 1 değerini alır. Örneğin “0110 0011” düzeninde “1” olan bitlerin sayısı çift sayıdır. Dolayısıyla bu çift sayı düzenini bozmamak için eşlik bitinin değeri “0” olur. Bit düzeni “0100 0011” ise çifte tamamlamak için eşlik bitinin değeri “1” olacaktır.

Gönderilen verilerin ikilik karşılıkları ve eşlik bitleri aşağıdaki Tablo 1.8’de görülmektedir.

Sayı	Veri Biti	Çift Eşlik Biti	Tek Eşlik Biti
&H39	00111001	0	1
&HB5	10110101	1	0
&H1A	00011010	1	0
&H0	00000000	0	1

Tablo 1.8:Eşlik biti

Sonuç olarak gönderici hat mantık 1 seviyesine çekilerek yine anlaşmaya bağlı olarak 1 veya 2 durdurma biti gönderilir. Aktarımı yapılacak diğer karakter için aynı işlem takip edilir. Şekil 1.14’te iki bitin ardışık gönderimi temsil edilmiştir.



Şekil 1.14: Ardışık bitlerin gönderilmesi

Buna göre bir karakterin iletimi için seçime bağlı olarak 10, 11 veya 12 bit gerekiyor. Peki bu bitler hangi hızda iletilecek ve alıcı hangi hızda alacak. Bunun alıcı ve verici tarafından bilinmesi gerekir. Bu hız baud hızı denir.

Baud hızı, ismini, 1890’lı yıllarda Fransız Telgraf İşletmesi’nde çalışmakta olan Jean Maurice Emile Baudot’tan almıştır. Bir hattın saniyedeki durumunu değiştirme oranı olarak tarif edilen baud hızı, günlük hayatta saniyede iletilecek bit sayısı olarak bilinir. Baud hızı 19200 ise saniyede 19200 bit iletilecek yada 19200 defa hat seviye değiştirecek anlamındadır. Baud hızı ile seri haberleşme mesafesi Tablo 1.9’da görülmektedir.

Baud Hızı	Yalıtılmış Kablo [m]	Yalıtılmamış Kablo [m]
110	5000	1000
300	4000	1000
1200	3000	500
2400	2000	500
4800	500	250
9600	250	100

Tablo 1.9: Baud hızına bağlı iletim mesafesi

Hem kullanıcı hem de alıcı baud hızı ile beraber bu veri biti düzeninde de birbiriyle anlaşmış olması gerekir. Aksi takdirde veriler anlamsız ve karmakarışık alınır.

Bilgisayarlarda 1.8432 MHz'lik saat osilatörü bir ön bölücü ile 16'ya bölünerek 115200 Hz elde edilir. Bu değer program yoluyla bölünerek istenilen baud hızı elde edilir. Bölen sayı iki bayt halinde DLM ve DLL yazmaçlarında saklanır. DLM, büyük baytı saklar. Tablo 1.10, bu sayıları göstermektedir.

Baud Hızı	Bölen Sayı	DLM	DLL
110	1047	04H	17H
300	384	01H	80H
600	192	00H	C0H
1200	96	00H	60H
2400	48	00H	30H
3600	32	00H	20H
9600	12	00H	0CH
19200	6	00H	06H
38400	3	00H	03H
57600	2	00H	02H

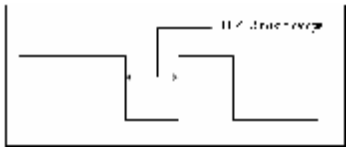
Tablo 1.10: Baud hızı

Baud hızını DLL ve DLM yazmacına yazabilmek için LCR yazmacının son biti, Tablo 1.11'de görüldüğü gibi "1" yapılmalıdır. Bu da OUT (COM1 + 3 , 0x80) komutu ile yapılmaktadır. Daha sonra bu yazmacın alt bitlerine veri biti sayısı, eşlik biti gibi değerler yerleştirilmektedir.

Formun Load olayı aşağıdaki gibi değiştirilerek, program çalıştırılır.

```
Private Sub Form_Load()
Varna = CreateFile("COM1", GENERIC_READ Or GENERIC_WRITE, _
 ByVal 0, ByVal 0, OPEN_EXISTING, ByVal 0, ByVal 0)
If Varna < 0 Then Exit Sub

Out (COM1 + 1), 0
Out (COM1 + 3), &H80
Out (COM1 + 0), &H17
Out (COM1 + 1), &H4     '110 baud hızı
Out (COM1 + 3), &H3     '8 veri biti, eşlik biti yok, 1 stop biti
End Sub
```

Bir örnek vermek gerekirse, DLM yazmacına 00, DLL yazmacına 0C yazıldığında 115200 Hz frekans 12'ye bölünerek 9600 Hz baud hızı elde edilir. Bu da saniyede 9600 ardışık bit iletilebilir anlamına gelmektedir. Bir bitlik süre 1 / 9600 = 104.2 mikro saniyedir. 8 veri biti, 1 başlangıç, 1 sonlandırma bitinden oluşan gönderilecek bir karakter 10 bit uzunluğundadır ve bu bilgi 10 * 104.2 = 1042 mikro saniyede gönderilir. Bu da yaklaşık olarak 1 mili saniye yapar.



Şekil 1.15: Bit süresi

Yukarıdaki programın çalıştırılması sonunda karakterlerin iletimi görülmekedir. Peki UART'ın kabul edeceği en düşük baud hızı nedir? 115200/57600 =2. Buna göre bölen olan 57600 sayısının onaltılık karşılığı E100 sayısını DLM ve DLL yazmaçlarına paylaştırılacaktır.

Aşağıdaki değişiklikler yapılarak, program bir kez daha çalıştırılır.

Out (COM1 + 1), 0
Out (COM1 + 3), &H80
Out (COM1 + 0), &H0
Out (COM1 + 1), &HE1
Out (COM1 + 3), &H3

Artık, basılan tuşun bitleri görülmektedir.

LCR yazmacı, haberleşme için gerekli olan parametreleri ayarlar. Bu yazmaçta verinin bit uzunluğu, eşlik biti sayısı, başlangıç ve bitiş bitleri ayarlanmaktadır. Tablo 1.11, bu bitlerin durumunu göstermektedir.

Bit 7	1	Bölme yazmacı Erişim Biti		
	0	Alıcı ve verici tamponuna, Kesme Yetki Yazmacına Erişim		
Bit 6	Fasıla Yetkisi			
Bit 5 Bit 4 Bit 3	Bit 5	Bit 4	Bit 3	Eşlik Seçimi
	X	X	0	Eşlik Yok
	0	0	1	Tek Eşlik
	0	1	1	Çift Eşlik
	1	0	1	Yüksek Eşlik
	1	1	1	Alçak Eşlik
Bit 2	Durdurma Bitinin Uzunluğu			
	0		Bir adet durdurma biti	
	1		6, 7, 8 bit uzunluğundaki kelime uzunluğu için 2 durdurma biti, 5 bit uzunluğunda kelime için 1.5 durdurma biti	
Bit 1 Bit 0	Bit 1	Bit 0	Kelime uzunluğu	
	0	0	5 bit	
	0	1	6 bit	
	1	0	7 bit	
	1	1	8 bit	

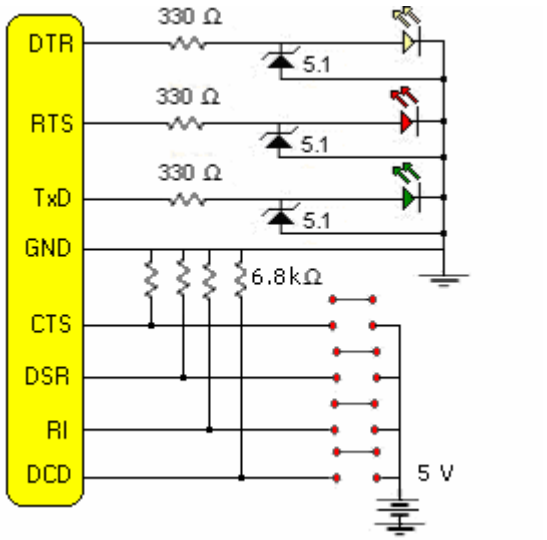
Tablo 1.11: LCR yazmacı

Karakter iletimin de yukarıda yaptığımız işlemi, BuildCommDCB, SetCommState ve GetCommState API fonksiyonları da yapmaktadır.

UYGULAMA FAALİYETİ

Aşağıdaki sorulara ilişkin uygulama faaliyetini yapınız.

- Aşağıda görülen devreyi yaparak genel amaçlı seri port giriş-çıkış devresi yapınız. Burada butonlar yerine sensörler bağlanabilir. Her bir butona basılma sayısını ekranda gösteren programı yazınız.



- DTR, RTS ve TxD hatlarına bağlı ledlerin yarımşar saniye aralıklarla onar kere yanmasını sağlayan programı yazınız.

İşlem Basamakları	Öneriler
Ø Form tasarımını yapınız. Ø Bileşenlere uygun isimler veriniz. Ø Programı yazınız. Ø Yazdığınız programı derleyiniz. Ø Programda hata var ise bunları düzeltiniz. Ø Ekran görüntüsünü kontrol ediniz.	Ø Programda kullanacağınız değişkenlerin tipini belirleyiniz. Ø Değişken isimlendirme kurallarına dikkat ediniz. Ø Program satırlarının düzenli olmasına özen gösteriniz. Ø Karar ifadelerinin belirlenen şartlara uygun olmasına dikkat ediniz.

ÖLÇME VE DEĞERLENDİRME

OBJEKTİF TESTLER (ÖLÇME SORULARI)

1. RS-232 standardı hangi portlar için geliştirilmiştir?

A) Paralel portlar
B) USB portlar
C) Seri portlar
D) EPP portları

2. Aşağıdakilerden hangisi seri port çıkış ucudur?

A) DTR
B) DSR
C) DCD
D) CTS

3. Seri portta Mantık 1'in gerilim değeri kaç volttur?

A) -12
B) 12
C) -3
D) 5

4. COM1'in taban adresi kaçtır?

A) 2F8
B) 3F8
C) 2F9
D) 378

5. DLAB biti hangi yazmaç üzerindedir?

A) MSR
B) MCR
C) LSR
D) LCR

DEĞERLENDİRME

Cevaplarınızı cevap anahtarı ile karşılaştırınız. Doğru cevap sayınızı belirleyerek kendinizi değerlendiriniz. Yanlış cevap verdiğiniz ya da cevap verirken tereddüt yaşadığınız sorularla ilgili konuları faaliyete geri dönerek tekrar inceleyiniz.

ÖĞRENME FAALİYETİ-2

AMAÇ

Seri port donanımı yoluyla sayısal giriş değerlerini hatasız olarak kontrol edebileceksiniz.

ARAŞTIRMA

Bu öğrenme faaliyetinden önce aşağıdaki hazırlıkları yapmalısınız.

- Seri portla cihazların kontrolü

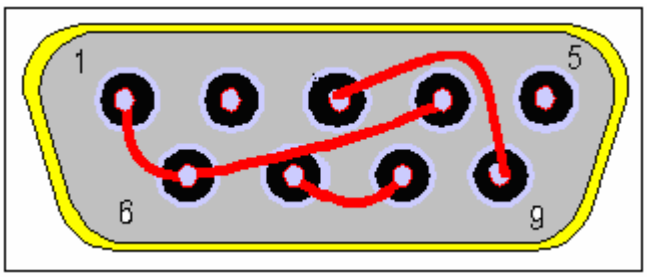
2. SERİ PORTA GİRİŞ YAPILMASI

Bir önceki bölümde seri port uçlarından nasıl çıkış yapıldığını ve nasıl karakter iletimi yapıldığını gördük. Seri port yolundaki yolculuğumuz devam etmektedir.

2.1. Seri Porttan Sinyal Okuma

Seri porta giriş için TTL mantık seviyesi doğru olarak algılandığından giriş gerilimini 12 volta yükseltmeye gerek yoktur. Doğrudan giriş yapılabilir.

Burada seri portun giriş uçlarını kullanmak için seri portun çıkış uçlarından yararlanacağız. Bu bakımdan dişi bir seri port konnektörü üzerinde Şekil 2.1'de görülen bağlantılar yapılarak seri port üzerine takılır.



Şekil 2.1: Giriş bağlantısı

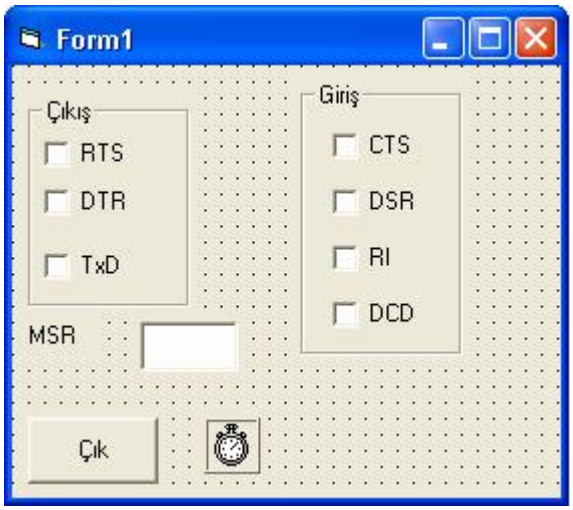
Program yazımına başlamadan önce kısa bir not: Seri portta 2 Nu.lu RxD ucu, genel veri okuma ucudur. Bunun dışındaki 1,6,8 ve 9 Nu.lu uçlar, giriş uçlarıdır. RxD ucundan

farkları, sinyal seviyeleri Mantık 1 ve Mantık 0 arasında değişmesi yani aç-kapa tarzında çalışmasıdır. RxD ucunun çalışması, bir sonraki bölümde görüleceği gibi biraz farklıdır.

ÖRNEK 2.1: Seri port giriş uçlarının kullanılması.

Örnek 1.2'nin üzerine ilaveler yapılacaktır.

ADIM 1: Form tasarımı Şekil 2.1'de görüldüğü gibi yapılır.



Şekil 2.2: Form tasarımı

Bileşenlere, başlıklara uygun olarak chkRTS, chkDTR, chkTxD, chkCTS, chkDSR, chkRI, chkDCD ve txtMSR isimleri atanır. Timer'ın Interval özelliğine de 100 verilir.

ADIM 2: Programın kod bölümü:

```
Const COM1 = &H3F8
Const DTR = 1
Const RTS = 2
Const TxD = 64

Const CTS = 16
Const DSR = 32
Const RI = 64
Const DCD = 128

Dim Nigbolu As Long
Dim yaz As String
Dim x As Long
Dim y As Integer
```

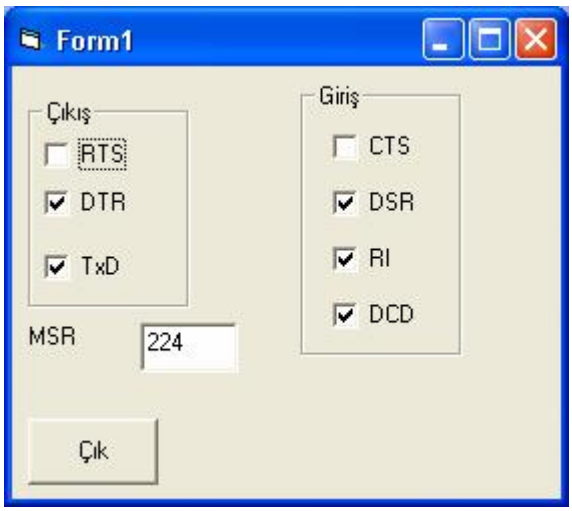
```
Function Comm_set(Dogan_Bey As Integer) As Integer

   If (Dogan_Bey And TxD) = TxD Then
     Call Out(COM1 + 3, TxD)
   End If

   If (Dogan_Bey And DTR) = DTR Or (Dogan_Bey And RTS) = RTS Then
      Call Out(COM1 + 4, (Inp(COM1 + 4) Or Dogan_Bey))
   End If

End Function

Function Comm_reset(Dogan_Bey As Integer) As Integer

   If (Dogan_Bey And TxD) = TxD Then
     Call Out(COM1 + 3, 0)
   End If

   If (Dogan_Bey And DTR) = DTR Or (Dogan_Bey And RTS) = RTS Then
      'DTR ve/veya RTS resetlenir
      Call Out(COM1 + 4, (Inp(COM1 + 4) And (Not Dogan_Bey)))
   End If
End Function

Private Sub Command1_Click()
 x = CloseHandle(Nigbolu)
 End
End Sub

Private Sub Form_Load()
Nigbolu = CreateFile("COM1", GENERIC_READ Or GENERIC_WRITE,
ByVal 0, ByVal 0, OPEN_EXISTING, ByVal 0, ByVal 0)
yaz = "Açılan seri Port Numarası:" & Nigbolu & vbCrLf
MsgBox yaz
If Nigbolu < 0 Then Exit Sub
End Sub

Private Sub Form_Unload(Cancel As Integer)
   x = CloseHandle(Nigbolu)
End Sub

Private Function Comm_Ahval(Minnet_Bey As Integer) As Boolean
On Error GoTo Hatali_Durum
Comm_Ahval = ((Inp(COM1 + 6) And Minnet_Bey) = Minnet_Bey)

Comm_Ahval_Terk:
Exit Function
```

```
Hatali_Durum:
MsgBox "Port durumunu Öğrenirken Hata Oldu. Hata No:" & Error$
Resume Comm_Ahval_Terk
End Function
Private Sub Timer1_Timer()

txtMSR.Text = Inp(COM1 + 6)

If chkDTR.Value = 1 Then
      y = Comm_set(DTR)
    Else
     y = Comm_reset(DTR)
    End If

If chkRTS.Value = 1 Then
      y = Comm_set(RTS)
    Else
     y = Comm_reset(RTS)
    End If

If chkTxD.Value = 1 Then
      y = Comm_set(TxD)
    Else
     y = Comm_reset(TxD)
End If

If Comm_Ahval(CTS) = True Then
  chkCTS.Value = 1
  Else
  chkCTS.Value = 0
  End If

  If Comm_Ahval(DSR) = True Then
  chkDSR.Value = 1
  Else
  chkDSR.Value = 0
  End If

  If Comm_Ahval(RI) = True Then
  chkRI.Value = 1
  Else
  chkRI.Value = 0
  End If

  If Comm_Ahval(DCD) = True Then
  chkDCD.Value = 1
  Else
```

chkDCD.Value = 0
End If

End Sub

ADIM 3: Program çalıştırılır.



Şekil 2.3: Programın çalışması

Burada seri portun çıkış uçları portun giriş uçlarına bağlanarak sinyal okunmuştur. Port giriş uçları Modem Status Register(MSR) üzerindedir. Okunan değer bit düzenine göre "And" işlemine tabi tutularak değerinin bir veya sıfır olduğuna karar verilmektedir.

Yazmaç Adı	İlave Kat.	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
Modem Status Register	6	DCD	RI	DSR	CTS				
Bit Değerleri **à**		128	64	32	16				

Tablo 2.1: MSR yazmacı

Seri portun DTR, RTS ve TxD hatlarını doğrudan süren EscapeCommFunction isimli bir API tanımlanmıştır. Bu seri porta bazı nitelikler yükler. Bildirimi:

```
Declare Function EscapeCommFunction & Lib "kernel32" (ByVal nCid As Long, _
ByVal nFunc As Long)
```

Aldığı parametreler Tablo 2.2'de görülmektedir.

Parametre	Tanım
nCid	Açık olan seri port numarası.
nFunc	İcra edilmek üzere aşağıda niteliği belirtilen fonksiyonu gösteren sabit. • CLRBREAK: ClearCommBreak API fonksiyonu ile aynı işi yapar. TxD hattını açar. • CLRDTR: DTR hattını kapatır. • CLRRTS: RTS hattını kapatır. • SETBREAK: SetCommBreak fonksiyonu aynı işi yapar. TxD hattını kapatır. • SETDTR: DTR ucunu etkin yapar. • SETRTS: RTS ucunu etkin kılar. • SETXON: XOn karakteri alınmış gibi portun davranmasını sağlar. • SETXON: XOff karakteri alınmış gibi portun davranmasını sağlar.

Tablo 2.2: EscapeCommfunction fonksiyonu

Yukarıdaki örnekte MSR yazmacı, adresi doğrudan yazılarak okundu. Bu yazmacı okuyan GetCommModemStatus isimli bir fonksiyon vardır. Bildirimi:

```
Declare Function GetCommModemStatus& Lib "kernel32" _
(ByVal hFile As Long, lpModemStat As Long)
```

Parametreleri tablo 2.3'te görülmektedir.

Parametre	Tanım
hFile	Açık olan seri port numarası.
lpModemStat	Modem durumunu gösteren değişken. Aşağıdaki sabitlerin birleşimi olabilir. • MS_CTS_ON: CTS açık • MS_DSR_ON: DSR açık • MS_RING_ON: RI sinyali hissedildi. • MS_RLSD_ON: DCD sinyali hissedildi.

Tablo 2.3: GetCommModemStatus fonksiyonu

Aşağıda bu fonksiyonlar bir örnekle desteklenmektedir.

ÖRNEK 2.2: Seri porta girişlerini API fonksiyonları ile denetlemek.

Form tasarımı, Örnek 1.1 ile aynı olacak.

ADIM 1: Modül kısmı:

```
Declare Function CreateFile Lib "kernel32" Alias "CreateFileA"_
(ByVal lpFileName As String, ByVal dwDesiredAccess As Long, _
ByVal dwShareMode As Long, ByVal NOlpSecurityAttributes As Long, _
ByVal dwCreationDisposition As Long, ByVal dwFlagsAndAttributes As Long, _
ByVal hTemplateFile As Long) As Long

Global Const GENERIC_READ = &H80000000
Global Const GENERIC_WRITE = &H40000000
Global Const OPEN_EXISTING = 3

Declare Function CloseHandle Lib "kernel32" (ByVal hObject As Long) As Long

Declare Function EscapeCommFunction Lib "kernel32" (ByVal nCid As Long, _
ByVal nFunc As Long) As Long

Declare Function GetCommModemStatus Lib "kernel32" _
(ByVal hFile As Long, lpModemStat As Long) As Long
```

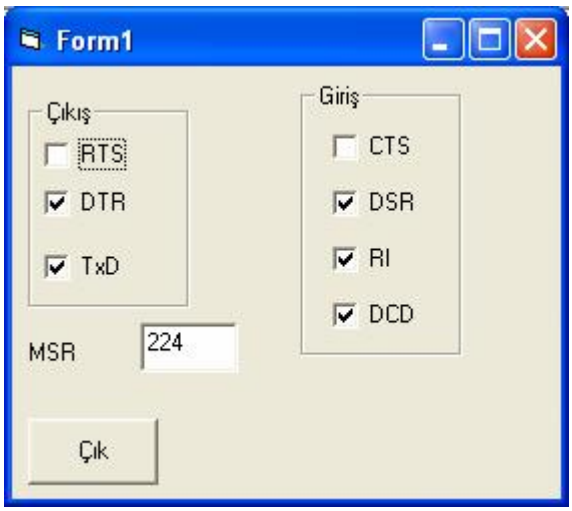
ADIM 2: Aşağıdaki kodlar yazılır.

```
Const SETRTS = 3
Const CLRRTS = 4
Const SETDTR = 5
Const CLRDTR = 6
Const SETBREAK = 8
Const CLRBREAK = 9
Const MS_CTS_ON = &H10
Const MS_DSR_ON = &H20
Const MS_RING_ON = &H40
Const MS_DCD_ON = &H80
Dim Kor_Jan As Long
Dim Minnet_Bey As Integer

Dim Nigbolu As Long
Dim yaz As String
Dim x As Long

Private Sub chkDTR_Click()
```

```
If chkDTR.Value = 1 Then
 Minnet_Bey = EscapeCommFunction(Nigbolu, SETDTR)
    Else
Minnet_Bey = EscapeCommFunction(Nigbolu, CLRDTR)
    End If
End Sub

Private Sub chkRTS_Click()
If chkRTS.Value = 1 Then
   Minnet_Bey = EscapeCommFunction(Nigbolu, SETRTS)
Else
Minnet_Bey = EscapeCommFunction(Nigbolu, CLRRTS)
    End If
End Sub

Private Sub chkTxD_Click()
If chkTxD.Value = 1 Then
Minnet_Bey = EscapeCommFunction(Nigbolu, SETBREAK)
Else
Minnet_Bey = EscapeCommFunction(Nigbolu, CLRBREAK)
End If
End Sub

Private Sub Command1_Click()
 x = CloseHandle(Nigbolu)
 End
End Sub

Private Sub Form_Load()
Nigbolu = CreateFile("COM1", GENERIC_READ Or GENERIC_WRITE, ByVal 0,
ByVal 0, OPEN_EXISTING, ByVal 0, ByVal 0)
yaz = "Açılan seri Port Numarası:" & Nigbolu & vbCrLf
MsgBox yaz
If Nigbolu < 0 Then Exit Sub
End Sub

Private Sub Form_Unload(Cancel As Integer)
   x = CloseHandle(Nigbolu)
End Sub

Private Sub Timer1_Timer()
Minnet_Bey = GetCommModemStatus(Nigbolu, Kor_Jan)
txtMSR.Text = Kor_Jan

If Kor_Jan And MS_CTS_ON Then
chkCTS.Value = 1
Else
```

```
chkCTS.Value = 0
End If

If Kor_Jan And MS_DSR_ON Then
chkDSR.Value = 1
Else
chkDSR.Value = 0
End If

If Kor_Jan And MS_RING_ON Then
chkRI.Value = 1
Else
chkRI.Value = 0
End If

If Kor_Jan And MS_DCD_ON Then
chkDCD.Value = 1
Else
chkDCD.Value = 0
End If
End Sub
```

ADIM 3: Programın çalışma görüntüsü.



Şekil 2.4: Programın çalışması

2.2. MsComm Bileşeni

Visual Basic (VB), diğer dillerden farklı olarak seri port haberleşme için hazır bir bileşen sunar. Microsoft Comm Control olarak adlandırılan bu bileşen seri portla ilgili bir çok ayarı beraberinde getirmektedir.

Comm bileşenini form üzerine getirmek için aşağıdaki yol takip edilir.

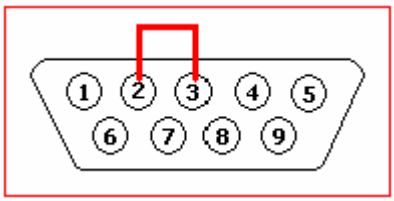
- Projects menüsünden Components alt başlığı seçilir.
- “Controls” sekmesinden “Microsoft Comm Control 5.0 “ yada VB sürümüne bağlı olarak “Microsoft Comm Control 6.0 “ seçilir.

Seçim yapıldıktan sonra alet kutusunun (toolbar) alt tarafında Comm bileşeninin sembolü görünür. Üzerinde telefon resmi olan bileşen seri portla haberleşmede kullanılacak Comm bileşenidir.

Şekil 2.5: MsComm bileşeni

MsComm bileşeni MSCOMM32.OCX dosyasını kullanılır. Bunun \WINDOWS\ SYSTEM dizininde olması gerekiyor. VB, seri haberleşmesi sürücüsü olarak serialui.dll dosyasını, MsComm’un arka planında kullanmaktadır.

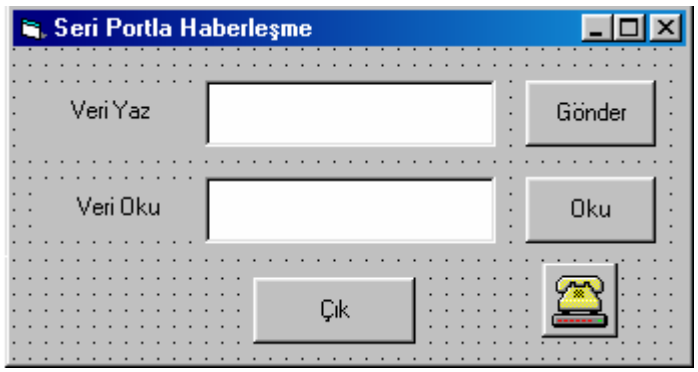
Kullanılacak her bir seri port için form üzerine bir MsComm bileşeni konur. MsComm ile ilgili teorik bilgi bir örnekle anlatılacaktır. Bu amaçla seri port soketinde TxD ve RxD uçlarına Şekil 2.4’te görüldüğü gibi küçük bir kablo lehimlenmelidir.



Şekil 2.6: Basit bağlantı

ÖRNEK 2.3: Gönderilen karakterin geri alınması.

Şekilden de anlaşılacağı gibi çıkış ucu giriş ucuna bağlanmıştır. Bu demektir ki giden her şey geri dönecektir. Bu amaçla Şekil 2.4’te görülen form tasarımını Tablo 2.4’ten yaralanarak yapınız.



Şekil 2.7: Form tasarımı

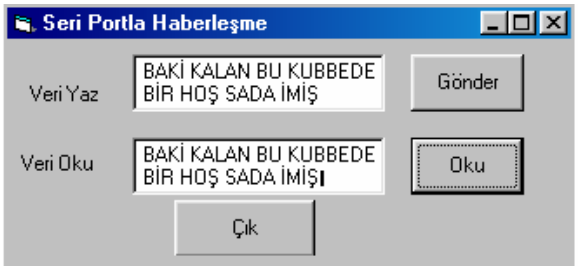
	Name	MultiLine
1. Metin Kutusu	Gonder	True
2. Metin Kutusu	Al	True
Gönder Düğmesi	btnGonder	
Oku Düğmesi	btnAl	
Çık Düğmesi	btnExit	

Tablo 2.4: Form tasarım tablosu

Program kod satırları aşağıdaki gibi düzenlenir.

```
Private Sub btnGonder_Click()
MSComm1.Output = Gonder.Text + Chr$(13)
End Sub

Private Sub btnOku_Click()
Al.Text = MSComm1.Input
End Sub

Private Sub btnExit_Click()
End
End Sub

Private Sub Form_Load()
MSComm1.PortOpen = True
End Sub

Private Sub Form_Unload(Cancel As Integer)
MSComm1.PortOpen = False
End Sub
```

Programı çalıştırma vakti geldi. I. Metin kutusuna bir şeyler yazılarak "Gönder" düğmesine basılır. Akabinde "Oku" düğmesi tıklanarak gönderilen verinin okunması sağlanır.



Şekil 2.8: Programın çalışması

2.2.1. MsComm'un Özellikleri

Program satırlarında MsComm'un üç özelliği kullanıldı. PortOpen, Output ve Input.

PortOpen: "True" değeri verilerek geçerli port kullanıma açılır.

```
MsComm1.PortOpen=True
```

Output: Porta bilgi gönderir. Özellikler penceresinde görünmez.

```
MsComm1.Output="       MAVİ MENEKŞE"
```

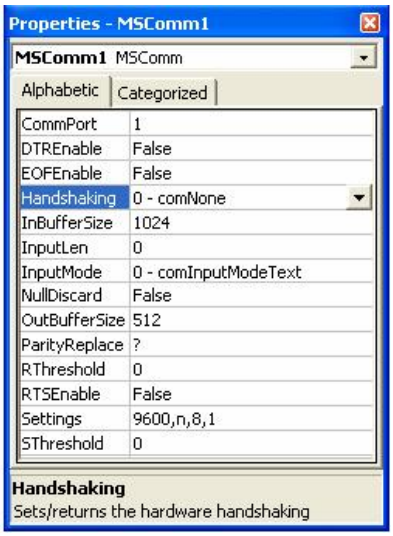
Output özelliği verilerin hem metin hem de ikilik düzende gönderilmesini sağlar. "String" tipinde tanımlanan bir değişkeni metin, bayt dizisi olarak tanımlanan bir değişkeni ikilik düzende gönderir.

```
' İkilik olarak
Dim Talebe() As  Byte
.....
MsComm1.Output=Talebe

'Metin olarak
Dim Talebe() As  String
MsComm1.Output=Talebe

'Modeme bir komut göndermek için
MsComm1.Output="ATDT 123-87654"
```

Input: Belirtilen seri porttan bilgi okur. Okuma işlemi giriş tamponunu da boşaltır.

```
Text1.Text=MsComm1.Input
```

Burada iki özellik kullanılmasına rağmen esasında MsComm'un "özellikler" (properties) kutusunda yazılı olan değerler geçerli kabul edilmişlerdir. İstenirse bu değerler program satırlarında da değiştirilebilir.



Şekil 2.9: MsComm özellikleri

CommPort: Kullanılacak seri port seçilir ve seçilen port numarası okunur.

MsComm1.ComPort=1 'Com1 seçilir

Port numaraları 1-99 arasında olabilir.

Settings: Geçerli port ayarları yapılır ve geçerli ayarlar okunur.

MsComm1.Settings="9600,N,8,1" '9600 baud hızı, Eşlik biti yok, 8 veri biti, 1durdurma biti

MsComm'un kabul ettiği baud hızları: 110, 300, 600, 1200, 2400, 9600, 14400, 19200, 38400, 56000, 128000, 256000. Varsayılan değer 9600 dür.

Geçerli eşlik bitleri: E(Even), M(Mark), N(None), O(Odd), S(Space). Varsayılan olarak None (yok) atanmıştır.

Geçerli bit sayıları: 4, 5, 6, 7, 8. Varsayılan değer 8.

Geçerli durdurma(stop) bitleri: 1, 1.5, 2. Varsayılan değer 1.

InputLen: Seri porta gelen bilgiler tampon belleklerde tutulur. Bu bilgiler okunurken kaç karakterlik bloklar halinde okunacağını belirtir. 0 değeri atanırsa tek seferde bilgiler okunur.

InBufferCount: Kabul edilen her karakter bir alıcı tamponuna konulur ve InBufferCount sayısı bir arttırılır. Tamponda bekleyen karakter sayısı, bu değer okunarak öğrenilir. 0 değeri atanarak tampon boşaltılır.

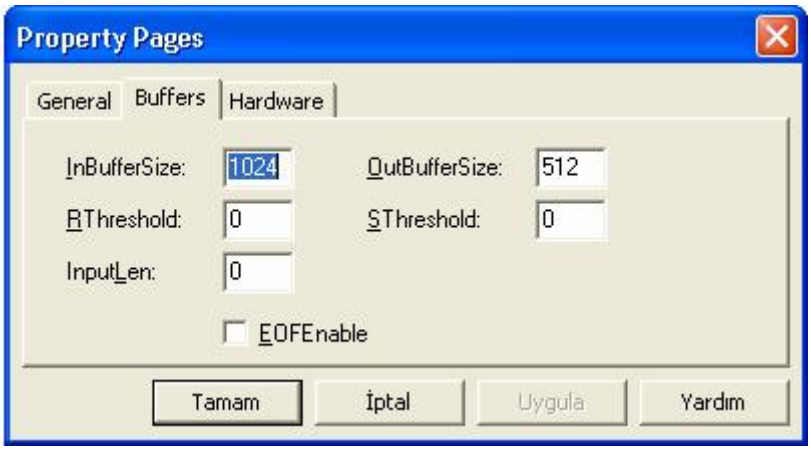
Örnek kod:

```
If MSComm1.InBufferCount Then
   Varna$ = MSComm1.Input
End If
```

OutBufferCount: Gönderici tampon alanında sırasını bekleyen karakter sayısını verir. 0 verilerek gönderici alan silinir.

OutBufferSize: Verici için tampon bellek miktarı. Ne kadar büyük bellek ayrılırsa programın çalışmasına o kadar az yer kalır. Fakat günümüzde bellek kapasitesi yüksek olduğundan programın ihtiyacına göre belirlenir. Varsayılan değer 512'dir. Bu değer azaltılabilir veya arttırılabilir.

Şekil 2.7, form üzerinde bulunan MsComm bileşeni üzerine sağ tıklayarak karşımıza gelen "Property Pages" penceresini göstermektedir. Buffers sekmesine gidilerek burada gerekli ayarlamalar yapılabilir.



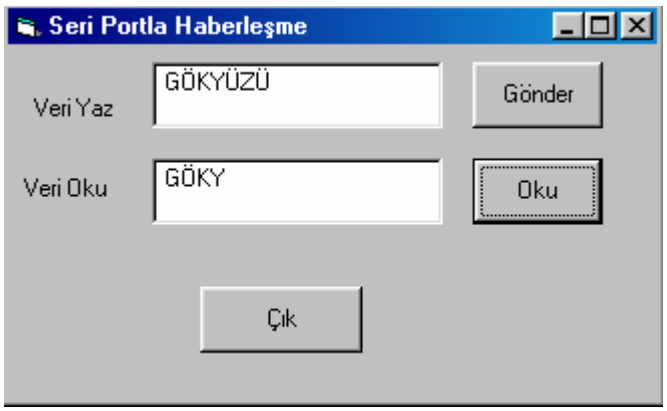
Şekil 2.10: Tampon ayarları

Örnek kod:

```
Private Sub Form_KeyPress(KeyAscii As Integer)
If MSComm1.OutBufferCount < MSComm1.OutBufferSize Then
MSComm1.Output = Chr$(KeyAscii)
End If
End Sub
```

InBufferSize: Tampon uzunluğu 1024 bayttır (1 KB). Bu değer azaltılabilir veya arttırılabilir.

Yukarıdaki örnekte bu özelliğe 5 değeri atanarak yeniden çalıştırılır.



Şekil 2.11: Programın çalışması

Görüldüğü gibi yazılan ilk beş harfi kabul etmektedir.

ÖRNEK 2.4: Gönderilen kelimelerin tersinden geri okunması.

Şekil 2.12: Form tasarımı

Form tasarımını Şekil 2.9' a uygun olarak yaptıktan sonra aşağıdaki kodlar yazılır.

Dim ch As Variant

Private Sub btnExit_Click()
End

```
End Sub

Private Sub btnTemizle_Click()
Text1.Text = ""
Text2.Text = ""
End Sub

Private Sub Form_Load()
MSComm1.CommPort = 1
MsCommSettings = "9600,N,8,1"
MSComm1.PortOpen = True

End Sub

Private Sub Text1_KeyPress(KeyAscii As Integer)
ch = Chr(KeyAscii)
MSComm1.Output = ch
MSComm1.InputLen = 0        'Tampondaki tüm karakterleri al
If MSComm1.InBufferCount > 0 Then ' Tamponda veri bekliyorsa

Text2.Text = MSComm1.Input + Text2.Text
End If
End Sub
```

Program çalıştırılarak, bir kelime yada tümce yazılır.

Şekil 2.13: Programın çalışması

2.2.2. MsComm'un Diğer Özellikleri

Rthreshold: OnComm olayı tetiklenmeden önce alıcı tamponuna konulan karakter sayısını tutar ve ayarlar. Bu özelliği geçerli değer olan 0'da bırakmak, OnComm olayını karakterler alınsa bile devre dışı bırakılmasına sebep olur. 1 yapmak her bir karakterin alıcı tamponuna yerleşmesinden sonra OnComm olayını tetikler. OnComm olayı haberleşme durumlarını gözlemlemek ve olabilecek değişiklere cevap vermek için kullanılır.

Sthreshold: Yine OnComm olayı meydana gelmeden önceki verici tamponuna konulacak asgari karakter sayısını ayarlar. Değeri 1 yapılırsa verici tamponu boşaldığında OnComm olayını tetikler.

EOFEnable: Veri girişi esnasında bir dosya sonu karakteri (EOF-End of Line) işaretinin bulunduğunu gösterir. Değerinin "True" olması veri akışını durdurarak OnComm olayının bu durumu kullanıcıya haber vermek için tetiklenmesine sebep olur.

InputMode: Gelen bilginin "metin " ya da "ikilik" düzende kabul edilmesini bu özellik ayarlar. Veri, katarlar olarak ya da bir bayt dizisinde ikilik veri blokları olarak alınır. İki alt başlığı vardır.

ComInputmodeText:	ANSI karakter setlerinden oluşan veriler için kullanılır.
ComInputModeBinary:	Kontrol karakterlerini de içeren (F1, ALT, Ctrl-Q gibi) diğer veriler için kullanılır.

Break: Çalışma zamanında ayarlanan bir özelliktir. True olması, giden verilerin dondurulması, False olması gönderimin tekrar başlamasını sağlar. Bu daha önceki örneklerde yapıldığı gibi TxD hattını yükseğe yada alçağa çekme işleminde kullanılabilir.

```
Private Sub Form_Load()
  MSComm1.CommPort = 1
  MSComm1.Settings = "9600,N,8,1"
  MSComm1.RThreshold = 1
  MSComm1.PortOpen = True
End Sub

Private Sub Command1_Click()
'TxD hattını aç
MSComm1.Break = True
Bul_Vardal = Timer + 0.5     'Yarım saniye bekle
' Zamanın mürur etmesini bekle
Do Until Timer > Bul_Vardal
  Dummy = DoEvents()
Loop
' TxD hattını kapat
MSComm1.Break = False
End Sub
```

CTSHolding: CTS hattının gözlenip gözlenmeyeceğine karar verir.

True: CTS hattı yüksek
False= CTS Hattı alçak

MsComm1.CTSHolding={True | False }

DSRHolding: DSR hattını yükseğe veya alçağa çeker.

DTREnable: True olursa DTR hattını yetkilendirir.

RTSEnable: True olursa RTS hattını yetkilendirir

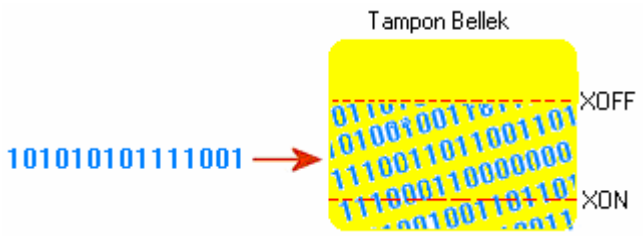
Handshaking:El sıkışma anlaşmasını yürürlüğe koyar ya da kaldırır.

El sıkışma veri gönderici donanım ile alıcı tamponu arasındaki anlaşmadır. Bir veri seri porta geldiğinde haberleşme aygıtı (UART) gelen bu verileri, bir yazılım programının okuyabilmesi için bir tampona yerleştirmek zorundadır. Veriler, UART'ın tampona yerleştirme hızından daha hızlı gelirse birbirinin üzerine yazılır. El sıkışma anlaşması bu verilerin üst üste yazılarak kaybolmamalarını sağlar. VB'de kullanılan el sıkışma anlaşmaları Tablo 2.5'te görülmektedir..

İsim	Değer	Açıklama
comNone	0	El sıkışması yok
comXOnXOff	1	Xon/Xoff el sıkışması
comRTS	2	RTS/CTS (Request To Send/ Clear To Send
ComRTSXOnXOff	3	RTS/CTS ve XonXOff el sıkışmalarının her ikisi birden

Tablo 2.5: El sıkışma türleri

Xon/Xoff, alıcı ve verici arasında bir yazılım anlaşmasıdır. Karakter iletiminin alıcı için çok hızlı olduğu durumlarda, alıcı vericiye ASCII 19 (Xoff) karakterini gönderek veri göndermemesini, kendi tamponunun boşalması durumunda da ASCII 17 (Xon) karakterini göndererek tekrar iletime başlamasını söyler.



Şekil 2.14: Xon/Xoff anlaşması

Xon sinyali gönderilmeden tampon belleğin durumu murakabe edilir.

ComRTS ve ComRTSXOnXOff seçilirse RTSEnabled özelliği True yapılmalıdır. Aksi takdirde bağlantı sağlansa bile veri alınamaz.

2.2.3. OnComm Olayı ve CommEvent Özelliği

Diğer bir aygıtla kurulan bağlantıda meydana gelen olayları veya hataları gözlemlemek için, MsComm bileşeni iki yolla seri haberleşmeyi yürütür.

Olay-Güdümlü: Bir karakterin gelmesi, RTS ya da DCD hattındaki bir değişim bir olayı tetikler. OnComm olayı bu olayları yakalamak için kullanılır.

Gözleme: CommEvents özellikleri bir olayın olup olmadığını test eder durur. Bir döngü içinde, bir karakterin gelip gelmediği gözlenir. Alıcı tamponuna bir karakterin ulaşması, hemen okunmasını tetikler.

Bir haberleşme olayı ya da bir hata olduğunda OnComm olayı tetiklenerek CommEvent özeliği değişir. OnComm'un her meydana gelişte CommEvent'in değeri okunur.

Tablo 2.6, OnComm'u tetikleyen haberleşme olaylarını göstermektedir.

Sabit	Deger	Tanımlama
comEvSend	1	Alıcı tamponunda SThreshold ile tanımlanandan daha az sayıda karakter var
comEvReceive	2	RThreshold sayısı ile tanımlanan sayıda karakter alındı. "Input" özelliği ile alıcı tamponundan karakter okunduğu müddetçe bu olay cereyan eder.
comEvCTS	3	CTS hattında bir değişiklik var
comEvDSR	4	DSR hattında bir değişiklik var. Bu olay sadece DSR, 1'den 0'a değiştiğinde husule gelir.
comEvCD	5	CD hattında bir değişiklik var.
comEvRing	6	Zil çalması algılandı
comEvEOF	7	ASCII 26 sayısı ile tanımlanan bir dosya sonu işareti alındı

Tablo 2.6: OnComm olayı

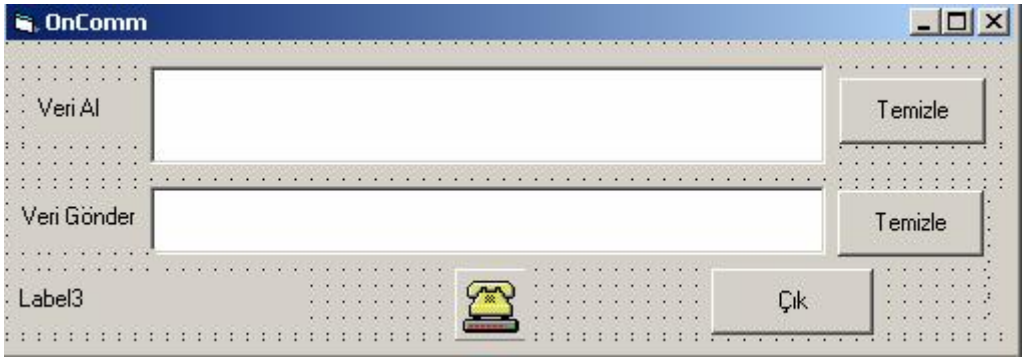
ComEvent'in her hareketinde OnComm olayı tetiklenir. Bu tetiklenmede meydana gelen hatalar sayılarla ifade edilir.

Sabit	Deger	Tanımlama
comEventBreak	1001	Bir ara verme sinyali algılandı
comEventFrame	1004	Çerçeve hatası
comEventOverrun	1006	Üzerine yazma hatası. Bir sonraki karakter geldiğinde bir önceki karakter okunmamışsa önceki kaybolur.
comEventRxOver	1008	Alıcı tamponu taşma hatası. Alıcı tamponunda yeterli yer yok
comEventRxParity	10009	Eşlik hatası
comEventTxFull	1010	Verici tamponu dolu.
comEventDCB	1011	Teşhis edilmeyen hata

Tablo 2.7: CommEvent özellikleri

ÖRNEK: OnComm Olayı.

Şekil 2.12, bu örnek için form tasarımını göstermektedir.



Şekil 2.15: Form tasarımı

Program kodları aşağıdaki gibi yazılır. Şekil 2.3'te görülen soketin hala seri porta takılı olduğundan emin olunmalıdır.

```
Private Sub Form_Load()
  MSComm1.CommPort = 1
  MSComm1.Settings = "9600,N,8,1"
  MSComm1.RThreshold = 1
  MSComm1.PortOpen = True
End Sub

Private Sub MSComm1_OnComm()
        Dim HadiseMsg As String
        Dim Buffer As Variant

        On Error GoTo ErrH

 Select Case MSComm1.CommEvent

      Case comEvReceive
         Buffer = MSComm1.Input
         Text1.Text = Text1.Text & Buffer
         HadiseMsg$ = " Receive "

      Case comEvSend
         HadiseMsg$ = " Send "
   End Select

  If Len(HadiseMsg$) Then
```

```
        Label3.Caption = "Durum : " & HadiseMsg$
      End If
   Exit Sub
ErrH:
   MsgBox Err.Description
End Sub

Private Sub Text2_KeyPress(KeyAscii As Integer)
MSComm1.Output = Chr$(KeyAscii)
End Sub

Private Sub VeriAlTemizle_Click()
Text1.Text = ""
End Sub

Private Sub VeriGonderTemizle_Click()
Text2.Text = ""
End Sub

Private Sub Command1_Click()
Unload Me
End Sub
```

Program çalıştırılır ve bir iki cümle yazılır.

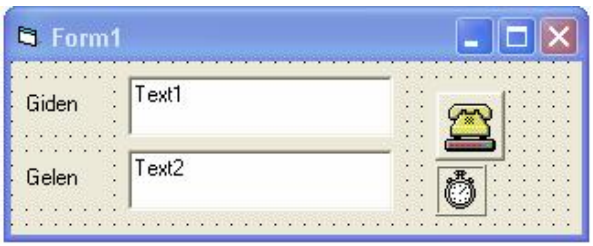
Şekil 2.16: Programın çalışması

OnComm olayındaki Select Case yapısı şu şekilde de tanımlanabilirdi.

```
If (MSComm1.CommEvent = comEvReceive) Then
        Buffer = MSComm1.Input
                Text1.Text = Text1.Text & Buffer
                HadiseMsg$ = " Receive "
End If

If (MSComm1.CommEvent = comEvSend) Then
        HadiseMsg$ = " Send "
End If
```

ÖRNEK 2.6: CommEvent Özelliği

Form tasarımı Şekil 2.14'te görüldüğü gibi yapılır.



Şekil 2.17: Form tasarımı

Timer'ın Interval özelliğini 1 yapın.

Kod listesi:

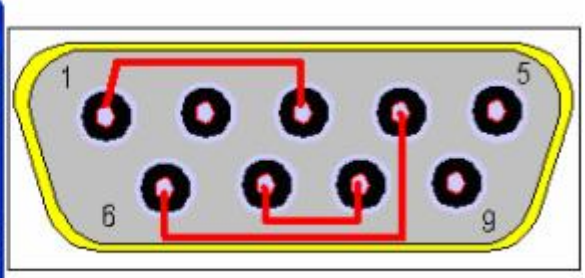
```
Dim Str As String
Private Sub Form_Load()
MSComm1.CommPort = 1
MSComm1.Settings = "9600,N,8,1"
MSComm1.PortOpen = True
MSComm1.InputLen = 0

Text1.Text = "Bulent Vardal"
MSComm1.Output = "Bulent Vardal"
End Sub

Private Sub Form_Unload(Cancel As Integer)
MSComm1.PortOpen = False
End Sub

Private Sub Timer1_Timer()
```

```
    Do  ' Porttan bir haber bekle
        DoEvents
    Loop Until MSComm1.InBufferCount >= 1
    Str = MSComm1.Input
    Text2.Text = Str
End Sub
```

UYGULAMA FAALİYETİ

Aşağıdaki işlem basamaklarına göre uygulama faaliyetini yapınız.

- Sürekli olarak “A” karakterini seri porttan gönderen programı yazınız.
- Sürekli olarak “A-Z” arası karakterleri seri porttan gönderen programı yazınız
- Kullanıcıdan baud hızı, eşlik biti gibi temel RS-232 parametrelerini girmesini isteyen MsComm ile “Endüstriyel Otomasyon” ifadesini gönderen bir program yapınız.



İşlem Basamakları	Öneriler
Ø Form tasarımını yapınız. Ø Bileşenlere uygun isimler veriniz. Ø Programı yazınız. Ø Yazdığınız programı derleyiniz. Ø Programda hata var ise bunları düzeltiniz. Ø Ekran görüntüsünü kontrol ediniz.	Ø Programda kullanacağınız değişkenlerin tipini belirleyiniz. Ø Değişken isimlendirme kurallarına dikkat ediniz. Ø Program satırlarının düzenli olmasına özen gösteriniz. Ø Karar ifadelerinin belirlenen şartlara uygun olmasına dikkat ediniz.

ÖLÇME VE DEĞERLENDİRME

OBJEKTİF TESTLER (ÖLÇME SORULARI)

1. Aşağıdaki uçlardan hangisi Modem Status Register'ın elemanlarından değildir?
   A) CTS
   B) DCD
   C) DSR
   D) DTR

2. Aşağıdakilerden hangisi MsComm'un karakter alım sayısını sınırlar?

   A) InBufferSize
   B) InputLen
   C) RThreshold
   D) InBufferCount

3. Aşağıdakilerden hangisi EscapeCommFunction fonksiyonunu TxD ucu ile ilgili sabitidir?

   A) SetDTR
   B) ClrBreak
   C) ClrDTR
   D) ClrRTS

4. Aşağıdakilerden hangisi MsComm'un haberleşmeyi yönettiği yollardan biridir?

   A) Open
   B) Click
   C) CommEvent
   D) Load

5. Aşağıdakilerden hangisi MsComm'un karakterlerin okunması ile ilgilidir?

   A) SThreshold
   B) InputLen
   C) Custom
   D) Index

DEĞERLENDİRME

Cevaplarınızı cevap anahtarı ile karşılaştırınız. Doğru cevap sayınızı belirleyerek kendinizi değerlendiriniz. Yanlış cevap verdiğiniz ya da cevap verirken tereddüt yaşadığınız sorularla ilgili konuları faaliyete geri dönerek tekrar inceleyiniz

ÖĞRENME FAALİYETİ-3

AMAÇ

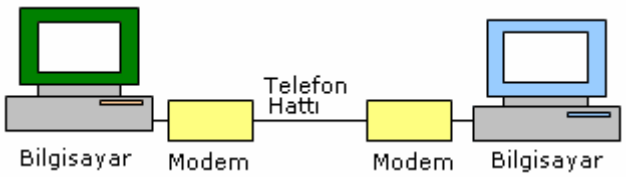
İki bilgisayar arasında veri iletişimini hatasız olarak sağlayabileceksiniz.

ARAŞTIRMA

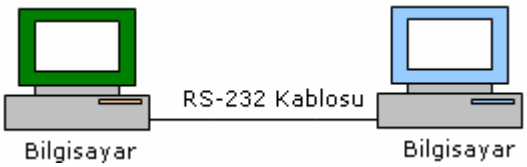
- Bilgisayarlar arası dosya aktarımı
- Z modem protokolü
- CRC hata denetimi

3. İKİ BİLGİSAYARI BAĞLAMAK

Bilgisayarları birbirine bağlayarak veri alışverişi yapmak en heyecan verici uygulamalardan biridir. Mekana ve imkana bağlı olarak bilgisayarlar birbirine çeşitli türlerde bağlanabilmektedir. Şekil 3.1 ve 3.2'te farklı iki tür görülmektedir.



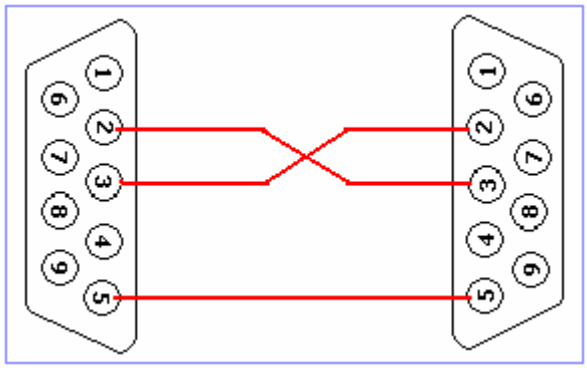
Şekil 3.1: Modem Bağlantısı



Şekil 3.2: Seri kablo bağlantısı

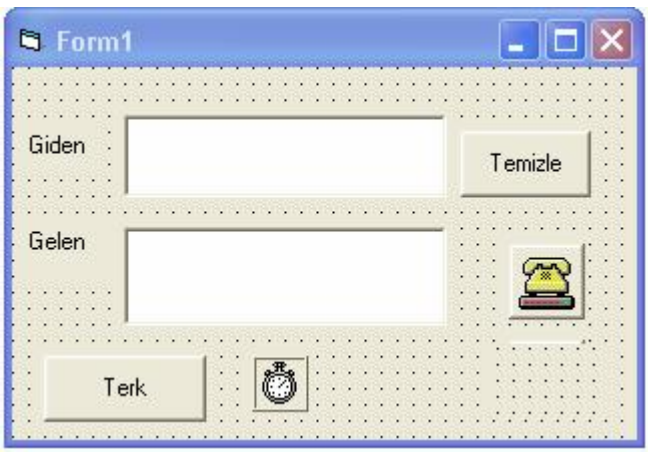
3.1. Bağlantı Şekilleri

Bilgisayarları seri port üzerinden bağlamanın en kısa yolu üç telli bağlamadır. Şekil 3.3'te görüldüğü gibi TxD hattı diğer bilgisayarın RxD hattına, RxD hattı da TxD hattına bağlanır. Toprak hatları birleştirilir.



Şekil 3.3: Üç telli bağlantı

ÖRNEK 3.1: Şekil 3.4'te görülen form tasarımı her iki bilgisayara da yapılır.



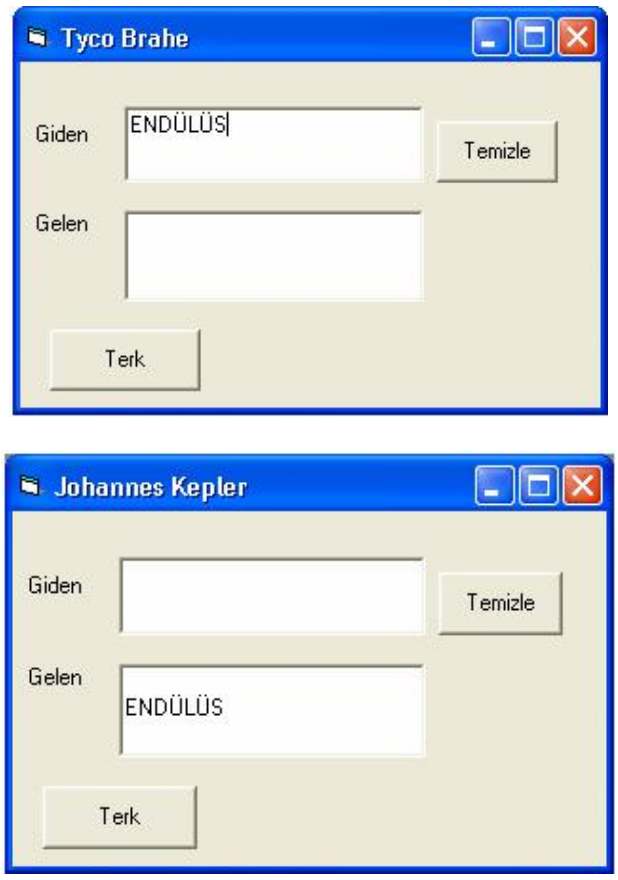
Şekil 3.4: Form tasarımı

Şekil 3.4'te görülen kablo seri portlara takılır.

```
Private Sub Command1_Click()
Unload Me
End Sub

Private Sub Command2_Click()
Text1.Text = ""
MSComm1.Output = "Sil"
End Sub

Private Sub Form_Load()
MSComm1.PortOpen = True
End Sub
```

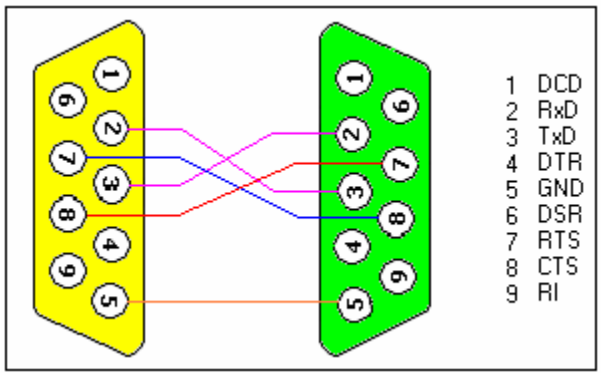
```
Private Sub Form_Unload(Cancel As Integer)
MSComm1.PortOpen = False
End Sub

Private Sub Text1_KeyPress(KeyAscii As Integer)
MSComm1.Output = Chr$(KeyAscii)
End Sub

Private Sub Timer1_Timer()
Selam = MSComm1.Input
If Selam = "Sil" Then
Text2.Text = ""
ElseIf Selam <> "Sil" Then
Text2.Text = Text2.Text + Selam
End If
End Sub
```

Program çalıştırılır.



Şekil 3.5: Programın çalışması

İki bilgisayar arasında basit bir donanım el sıkışmalı bir bağlantı kurulacaksa Şekil 3.6'da görülen kablo bağlantısı kullanılmalıdır.

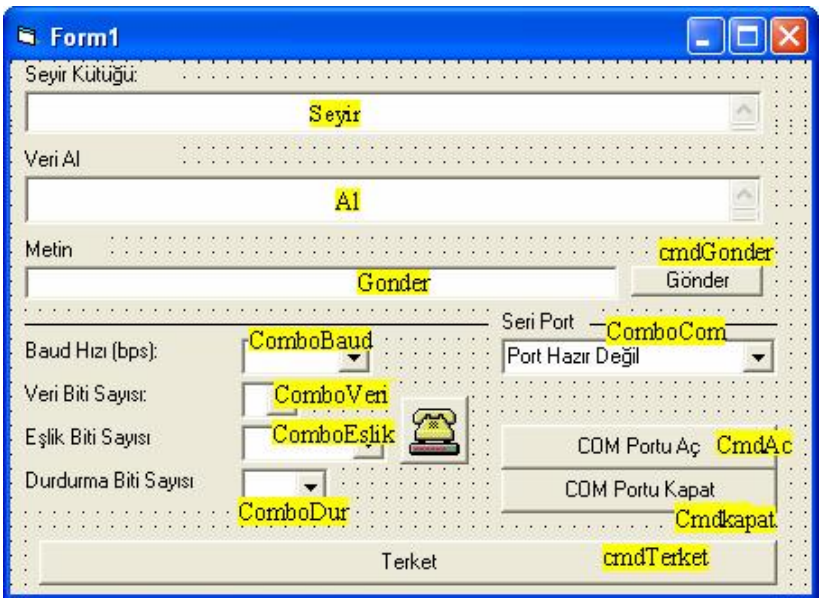


Şekil 3.6: El sıkışmalı bağlantı

ÖRNEK 3.2: Seri portla iki bilgisayarı irtibatlandırma.

Bu amaçla Şekil 3.6'da görülen bağlantıyı uzun kablolarla yaparak bilgisayarların seri portlarına takın.

Form tasarımı Şekil 3.7'de görüldüğü gibi yapılarak gerekli isim atamaları yapılır.



Şekil 3.7: Form Tasarımı

	Name	MultiLine
1. Metin Kutusu	Seyir	True
2. Metin Kutusu	Al	True

Tablo 3.1: Ayar tablosu

Formun kod bölümü:

```
Option Explicit
Dim PortOpen As Boolean

Public Function PortTest(COMPortNummer As Integer) As Boolean
  MSComm1.CommPort = COMPortNummer

  On Error Resume Next

  MSComm1.PortOpen = True
  If Err = 0 Then

    PortTest = True
    MSComm1.PortOpen = False
  Else
    PortTest = False
    MSComm1.PortOpen = False
  End If
End Function

Sub Port_Sına()
Dim Port_Sayisi As Integer

    Seyir.Text = Seyir.Text & Str(Time) & _
            " +++ COM portlar test edildi" & vbCrLf

    ComboCom.Clear

    For Port_Sayisi = 1 To 4
      If PortTest(Port_Sayisi) Then
        ComboCom.AddItem "COM" & Str(Port_Sayisi)
      End If
    Next

    If ComboCom.ListCount = 0 Then
      ComboCom.AddItem "COM Port Hazır Değil"
      Seyir.Text = Seyir.Text & Str(Time) & _
              " +++ COM Port Hazır Değil" & vbCrLf
              Else
```

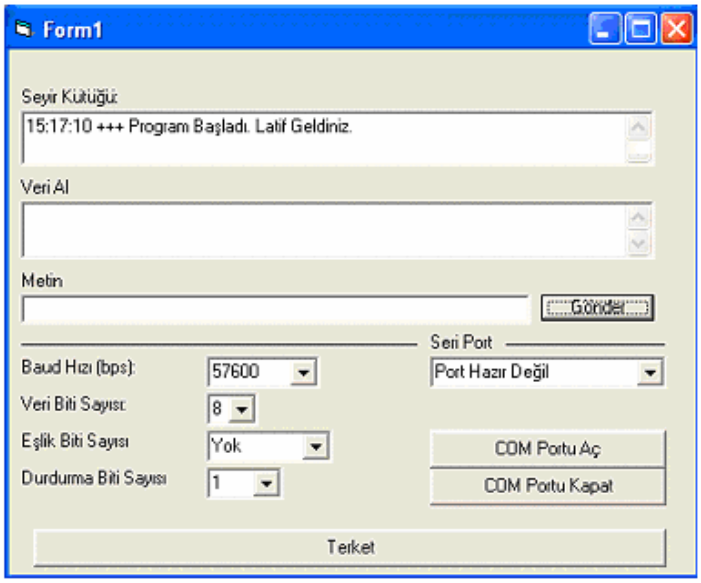
```
    Seyir.Text = Seyir.Text & Str(Time) & _
            " +++" & Str(ComboCom.ListCount) & " COM Port Hazır" & vbCrLf
   End If

   ComboCom.ListIndex = 0
End Sub

Private Sub CmdAc_Click()
  Dim CPort As Byte
  Dim Settings As String

  Port_Sına

  Seyir.Text = Seyir.Text & Str(Time) & _
          " +++ COM Port açıldı" & vbCrLf

   If PortOpen = True Then
    MsgBox "COM Port zaten açık "

    Exit Sub
   End If

   CPort = Val(Mid(ComboCom.List(ComboCom.ListIndex), 4, _
             Len(ComboCom.List(ComboCom.ListIndex))))

   If CPort = 0 Then
    MsgBox "Hata! COM Port mevcut değil"

    Seyir.Text = Seyir.Text & Str(Time) & _
            " +++ COM Port açılamıyor " & vbCrLf
   Else
    MSComm1.CommPort = CPort
    Settings = ComboBaud.List(ComboBaud.ListIndex)
    Select Case ComboEslik.ListIndex
      Case 0: Settings = Settings & ",E"
      Case 1: Settings = Settings & ",M"
      Case 2: Settings = Settings & ",N"
      Case 3: Settings = Settings & ",O"
      Case 4: Settings = Settings & ",S"
    End Select

    Settings = Settings & "," & ComboVeri.List(ComboVeri.ListIndex)

    Settings = Settings & "," & ComboDur.List(ComboDur.ListIndex)

    MSComm1.Settings = Settings
```

```
    On Error Resume Next

    MSComm1.PortOpen = True

    If Err <> 0 Then
      MsgBox "Hata! COM Port mevcut değil"

      Seyir.Text = Seyir.Text & Str(Time) & " +++ COM Port açılamıyor " _
            & vbCrLf
    Else
      MSComm1.RThreshold = 1
      MSComm1.SThreshold = 1
      MSComm1.InputLen = 0

      PortOpen = True
    End If
  End If

End Sub

Private Sub cmdGonder_Click()
  If PortOpen = False Then
    MsgBox "Lütfen önceden portu açınız"
    Seyir.Text = Seyir.Text & Str(Time) & _
          " +++ Gönderim esnasında hata oluştu, Port " & _
          "Port açık değil" & vbCrLf
    Exit Sub
  End If

  If Gonder.Text = "" Then
    MsgBox "Bir metin yazınız"
    Seyir.Text = Seyir.Text & Str(Time) & " +++ Metin Yok" & _
          vbCrLf
    Exit Sub
  End If

  On Error Resume Next

  MSComm1.Output = Gonder.Text + vbCrLf

  If Err <> 0 Then
    MsgBox "Metin gönderilemedi!"
    Seyir.Text = Seyir.Text & Str(Time) & " +++ metin yok " & vbCrLf

  Else
    Seyir.Text = Seyir.Text & Str(Time) & _
          " +++ """ + Gonder.Text & " "" gönderildi" & vbCrLf
```

```
  End If
Gonder.Text = ""
End Sub

Private Sub Cmdkapat_Click()
Seyir.Text = Seyir.Text & Str(Time) & _
          " +++ Port kapatılacak" & vbCrLf

  If PortOpen = False Then
    MsgBox "Açık port yok "


    Seyir.Text = Seyir.Text & Str(Time) & _
            " +++ Port kapatılamıyor " & vbCrLf
            Else
    On Error Resume Next

    MSComm1.PortOpen = False
    If Err <> 0 Then
      MsgBox "Port kapatma esnasında hata oluştu"
      Seyir.Text = Seyir.Text & _
              Str(Time) & " +++ COM Port kapatılamadı " & vbCrLf
    Else
      Seyir.Text = Seyir.Text & Str(Time) & " +++ COM Port kapatıldı " & vbCrLf
      PortOpen = False
    End If
  End If
End Sub
Private Sub cmdTerket_Click()
End
End Sub

Private Sub Form_Load()

  ComboCom.Clear
  ComboCom.AddItem "Port Hazır Değil"
  ComboCom.ListIndex = 0

  ComboBaud.AddItem "4800"
  ComboBaud.AddItem "9600"
  ComboBaud.AddItem "19200"
  ComboBaud.AddItem "38400"
  ComboBaud.AddItem "57600"
  ComboBaud.AddItem "115200"
  ComboBaud.ListIndex = 4

  ComboVeri.AddItem "6"
```

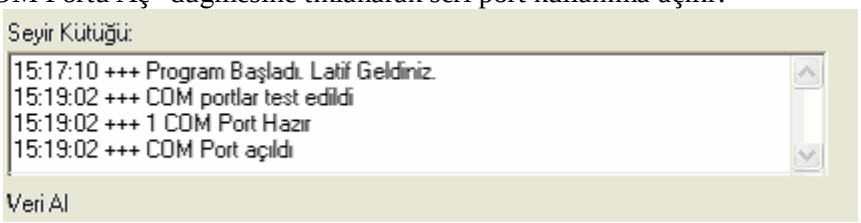
```
  ComboVeri.AddItem "7"
  ComboVeri.AddItem "8"
  ComboVeri.ListIndex = 2

  ComboEslik.AddItem "Tek"
  ComboEslik.AddItem "Çift"
  ComboEslik.AddItem "Yok"
  ComboEslik.AddItem "Mark"
  ComboEslik.AddItem "Space"
  ComboEslik.ListIndex = 2

  ComboDur.AddItem "1"
  ComboDur.AddItem "1.5"
  ComboDur.AddItem "2"
  ComboDur.ListIndex = 0

  PortOpen = False

  Seyir.Text = Str(Time) & _
          " +++ Program Başladı. Latif Geldiniz." & vbCrLf

End Sub

Private Sub MSComm1_OnComm()
Select Case MSComm1.CommEvent
   Case comEventOverrun:   MsgBox "Veri Kayboldu!"
   Case comEvReceive: Al.Text = Al.Text + MSComm1.Input
  End Select

End Sub
```

Program çalıştırıldığındaki ilk ekran görüntüsü:



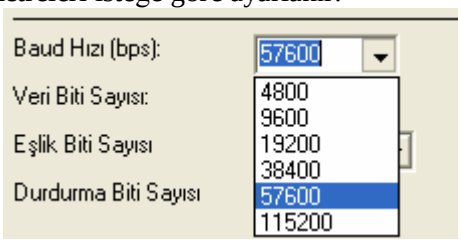
Şekil 3.8: Ekran görüntüsü

"COM Portu Aç" düğmesine tıklanarak seri port kullanıma açılır.



Şekil 3.9: Programın çalışması

Artık seri port parametreleri isteğe göre ayarlanır.

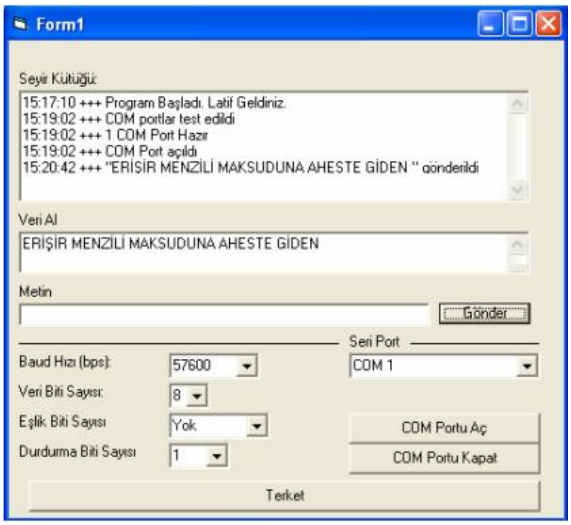


Şekil 3.10: Programın çalışması

İstenen veri hızı yazılır ve düğmeye tıklanır.

Şekil 3.11: Programın çalışması

Görüldüğü gibi yazılan mesaj geri geldi ve seyir günlüğüne kaydedildi.



Şekil 3.12: Programın çalışması

Aynı programı Şekil 2.3'teki soketle deneyiniz.

3.2. Diğer Seri Port Türleri

RS-232 seri haberleşme standardının, EIA (Electronics Industry Association) tarafından 1969 yılında geliştirildiğini daha önce belirtmiştik. Bu standardın ana amacı, farklı firmaların ürünleri arasındaki bağlantı şekillerini basitleştirmekti.

Standart, seri portun elektriksel, mekaniksel ve fonksiyonel niteliklerini tanımlamaktadır. Elektriksel nitelikler, gerilim seviyelerini, kablo empedanslarını tanımlar. Mekaniksel nitelikler, uçlara(pin) yapılacak görev atamalarını tanımlar. Soketin kendisi tanımlanmamıştır. Yani portun 9 ya da 25 bacaklı olacağı kesinleştirilmemiştir. Fonksiyonel nitelikler ise bağlantı şekillerini belirtir.

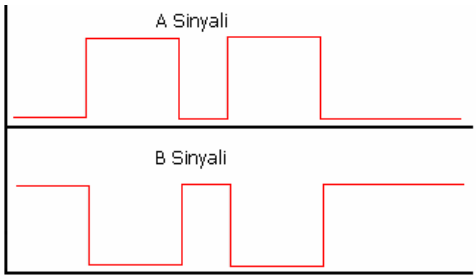
Aynı yıllarda Avrupa'da CCITT (Comité Consultatif International Téléphonique et Télégraphique) heyeti tarafından seri iletişimi tanımlamak için V.24 (fonksiyonel tanım) ve V.28 (elektriksel tanım) standartları geliştirilmiştir. V.90 56.6 kBit'lik veri aktarımını tanımlayan en geniş standarttır. Gerek RS-232 gerekse CCITT birbirine çok benzemesine rağmen RS-232 daha fazla rağbet görmüştür.

RS-232, uzun zaman sanainin ve kişisel kullanıcıların ihtiyaçlarını karşılamıştır. Ama zamanla artan ihtiyaçlar, EIA tarafından yeni standatların geliştirilmesini zorunlu kılmıştır. Bu amaçla RS-422, RS-423, RS-449 ve son olarak da RS-485 geliştirilmiştir. İlerleyen zamanlarda baştaki RS kısaltması EIA olmuş; şimdilerde ise TIA olarak

isimlendirilmektedir. Bu gelişim ile beraber yeni bir sinyal gönderme türü çıkmıştır. Dengeli ya da diferansiyel (fark) sinyal.

RS-232 standardında veri, gerilimi toprak hattına göre değişen bir tek bir tel üzerinde karşı tarafa aktarılır. Aynı zamanda sadece iki cihaz birbirine seri portla bağlanabilir. Bir porta tek bir farenin bağlanabilmesi gibi. Yani bir port için bir aygıt. Buna tek uçlu ya da dengesiz bağlantı denir.

Diferansiyel ki buna dengeli bağlantı denir, iletim iki tel arasındaki gerilim farkına göre veri bilgisini gönderir. Şekil 3.13'te diferansiyel sinyal görülmektedir. A ve B sinyalleri birbirine zıttır. Sinyal seviyesi bir toprak hattına göre değil de topraktan bağımsız iki uç arasındaki gerilime göre tespit edilir. İki uç arasındaki gerilim farkı, iletim hattının uzunluğuna bağlı olarak zayıflasa da hatta kalacağından veri kaybı olmayacaktır. A ve B sinyalinin birbirine göre büyüklüğü bitlerin 1 yada 0 olduğunu belirler.



Şekil 3.13: Diferansiyel sinyal

3.2.1. RS-422 / RS-449

RS-422 dengeli iletimin elektriksel özelliklerini belirler. İğne bağlantılarını ve haberleşme anlaşmasını ise RS-449 standardı belirlemiştir. 12 metrelik bir uzunlukta 10 Mbit'lik bir hıza erişir. İletim uzunluğu 1200 metreye kadar çıkmaktadır. Çok noktalı haberleşmeyi desteklemektedir. Yani bir port birden fazla alıcıyla (azami 10) irtibat halindedir. Yalnız bunlardan sadece birisi verici diğerleri alıcıdır. Alıcılar sadece verici ile haberleşebilir. Kendi arasında ilişki kuramazlar.

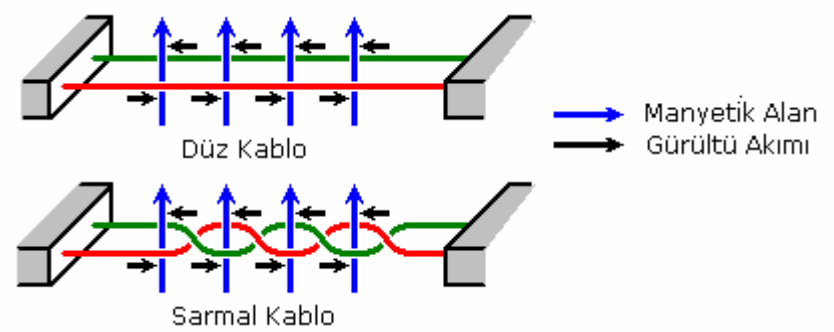
3.2.2. RS-423

RS-423 tek uçlu dengesiz hattır ve RS-232'in gelişmiş biçimidir. Fakat varlığı çok uzun olmamıştır. 12 metredeki hızı 100 kBit/sn'dir.

3.2.3. RS-485

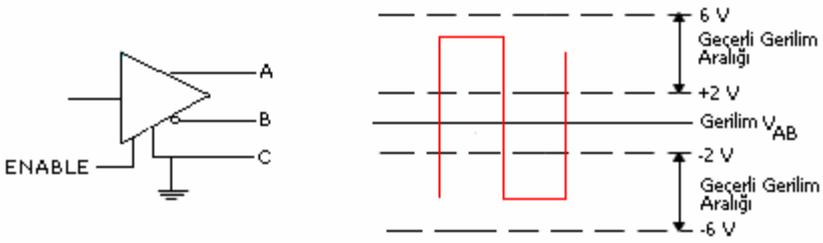
RS-232'nin bir kötü tarafı sinyal hatlarındaki gürültünün mevcudiyetidir. Gürültü, manyetik alan yada başka sebeplerle meydana gelen bozucu etkidir. Azami iletim uzaklığını ve veri iletim hızını sınırlar. Rs-422 / RS-485, iki kablonun birbiri etrafına sarılması ile elde edilen sarmal bir kablo kullanır.

Şekil 3.14'te görüldüğü gibi düz kabloda manyetik alanın meydana getirdiği gürültü akımı aynı yönde akmaktadır. Tıpkı bir transformatördeki döngüsel akım gibi sürekli tekrarlanmaktadır. Sarmal kabloda ise gürültü akımı, sarmalın bir yanıyla diğer tarafı arasında birbirine zıttır. Buna bağlı olarak gürültü akımı düz kablodaki gürültü akımına göre son derece azdır.



Şekil 3.14 : Düz ve sarmal kabloda gürültü

Kablolar arasındaki gerilim farkı veriyi aktarır. Peki kablolar arasındaki gerilim farkı nasıl sağlanıyor? Kablolardan birisi terslenerek. Şekil 3.15'te dengeli bir çıkış sürücüsü görülmektedir. Sürücü, A ve B çıkış uçlarında 2-6 volt arasında bir gerilim üretecektir. Sürücünün toprak hattına bağlanması önemli olmakla beraber RS-485 alıcısı tarafından veri hattının mantık seviyesinin tayininde kullanılmaz. Enable (Yetki) girişi ise girişin çıkışlara aktarılmasına izin verir.

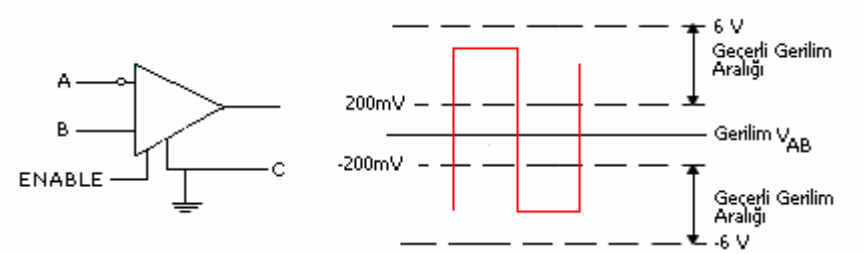


Şekil 3.15 : RS-485 vericisi

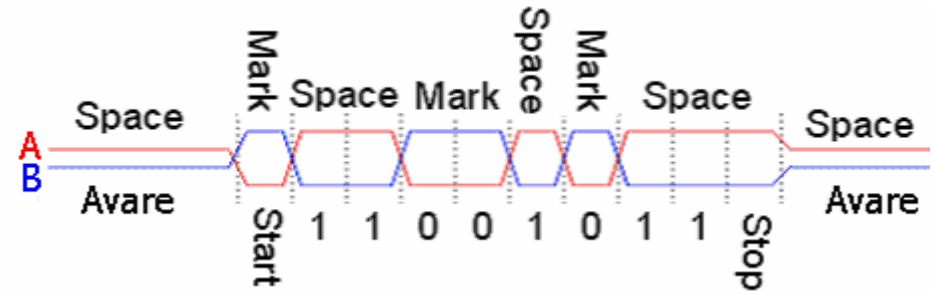
RS-485 alıcısı, A-B giriş telleri üzerindeki gerilim değerlerini algılar. Alıcıda sinyaller, Şekil 3.16'da görüldüğü gibi -6 V / -200mV ve 200mV / 6V aralığında geçerlidir. A ve B sinyalinin durumuna göre op-ampın çıkışı anlam kazanır. Buna göre:

1. Çıkış hattının A ucu, B ucuna göre göre negatifse, alıcıda bu Mantık 1'dir. (Mark ya da OFF durumu)
2. Çıkış hattının A ucu, B ucuna göre göre pozitifse, alıcıda bu Mantık 0'dir. (Space ya da ON durumu)

Şekil 3.17, bu duruma göre bir karakterin bit düzenini göstermektedir.



Şekil 3.16 : RS-485 girişi



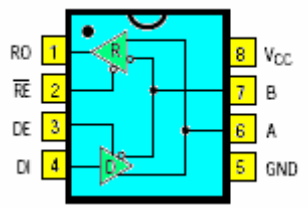
Şekil 3.17 : Giriş mantık seviyeleri

Tablo 3.2, piyasada bulunan RS-485 alıcı-vericilerini ve üreticilerini göstermektedir.

Firma	Ürün
Maxim	MX485
Linear Technology	LCT485
Texas Instruments	SN75176

Tablo 3.2: RS-485 vericileri ve üreticileri

Şekil 3.18, çok kullanılan RS-485 alacı-verici entegresi olan MAX485'in yapısını; Tablo 3.3 bacak yapısını göstermektedir.

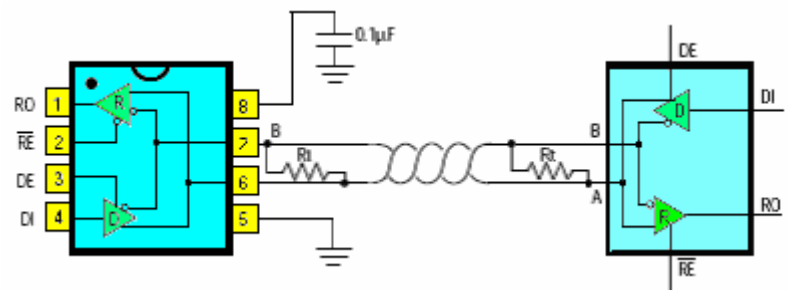


Şekil 3.18 : MAX485 uç yapısı

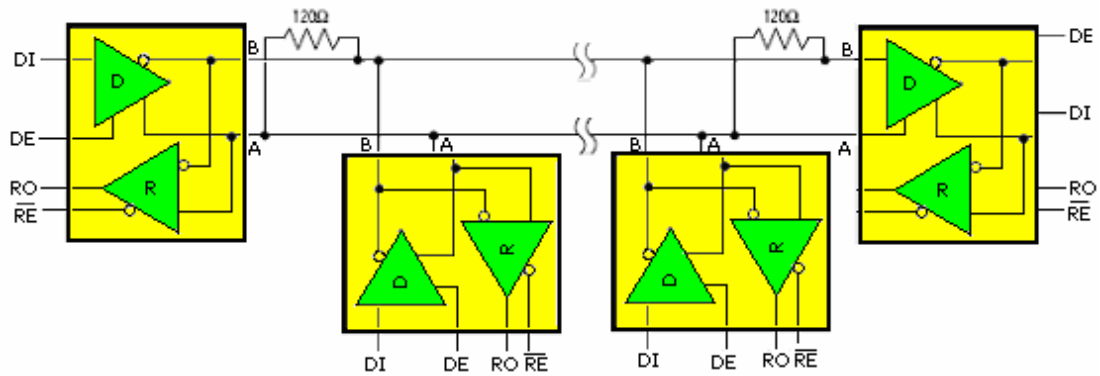
Uç	İsim	İşlev
1	RO	Alıcı Çıkışı. A>B =200 mV olduğunda RO yüksek olacaktır. A<B=200 mV olduğunda RO çıkışı düşük olacaktır.
2	RE	Alıcı çıkışı yetki ucu. RE düşük olduğu zaman RO çıkışı etkin olur.
3	DE	Sürücü çıkış yetki ucu. Yüksek olduğunda B ve A çıkışları etkin olur.
4	DI	Düşük olduğunda A düşük, B yüksek olur.
5	GND	Toprak ucu.
6	A	Terslemeyen sürücü çıkışı ve alıcı girişi
7	B	Tersleyen sürücü çıkışı ve alıcı girişi.
8	Vcc	Besleme. 4.75 < Vcc < 5.25

Tablo 3.3: MAX485 uç açıklamaları

RS-485'in hem cinslerine göre en büyük özelliği 32 alıcının kendi aralarında haberleşmeleridir. Şekil 3.19'da iki sürücünün, Şekil 3.20'de çoklu bağlantı şekilleri görülmektedir.



Şekil 3.19 : İki RS-485 sürücünün bağlantısı

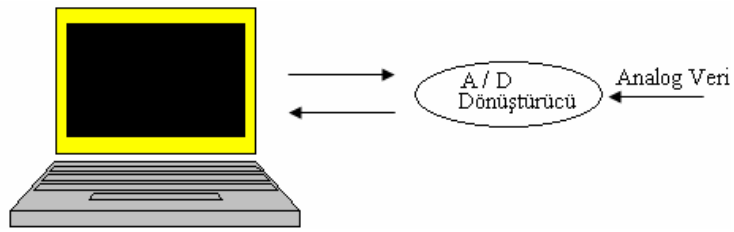


Şekil 3.20 : Çok noktalı bağlantı

RS-485 bağlantısı, PLC'li ve mikroişlemcili otomasyon sistemlerinde çok yaygın olarak kullanılmaktadır. Mikroişlemci ve PLC konularını şu anki sınıfınız itibarıyla görmediğinizden bu modülde RS-485 uygulaması yapılmayacaktır.

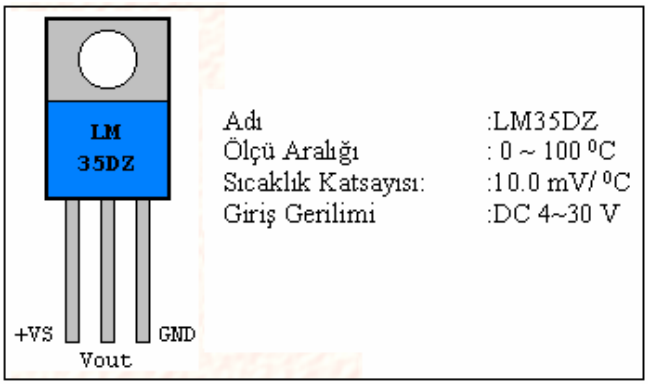
3.3. Seri Port ile Sıcaklık Ölçümü

Seri port ile sıcaklık ölçümünde bir sıcaklık sensörü ve bu sensörden gelen analog bilgiyi seri porta çevirecek bir analog/sayısal dönüştürücü kullanılacaktır. Sensör - A/D ve bilgisayar arasındaki veri alışverişi Şekil 3.21'de görüldüğü gibi olacaktır.



Şekil 3.21: Sensör- bilgisayar ilişkisi

Sıcaklık sensörü olarak LM35DZ kullanılacaktır.



Şekil 3.22: LM35 DZ ve özellikleri

LM35'in çıkış gerilimi ile sıcaklık arasındaki denklem:

Sıcaklık (^{0}C) = Vout * (100 ^{0}C / V)

Vout = 1 V ise Sıcaklık 100 ^{0}C'dir. Çıkış gerilimi sıcaklıkla doğrusal olarak değişir.

Sıcaklık LM 35 ile termistörden daha hassas ölçülür. Termokupl'dan daha yüksek gerilim verdiğinden bir op-amp ile çıkışı yükseltmeye gerek yoktur.

Devrede kullanılacak A/D Çevirici MAX 186'dır.

Özellikleri:

Adı : MAX186(Maxim Integrated Product)

Seri Çıkış :12 Bit
Giriş Gerilimi :5 V
Referans Gerilimi :4.096 V
Kanal Sayısı :8

A/D, 12 bit olduğundan çözünürlüğü:

0000 0000 0000

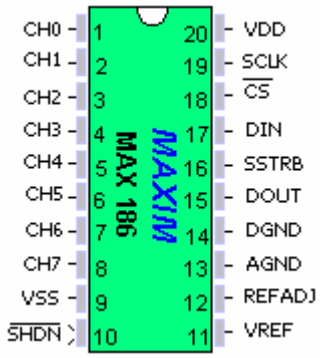
1111 1111 1111

12 Bit 0 ~ 4095 (4096 Desen)

Hassasiyet:

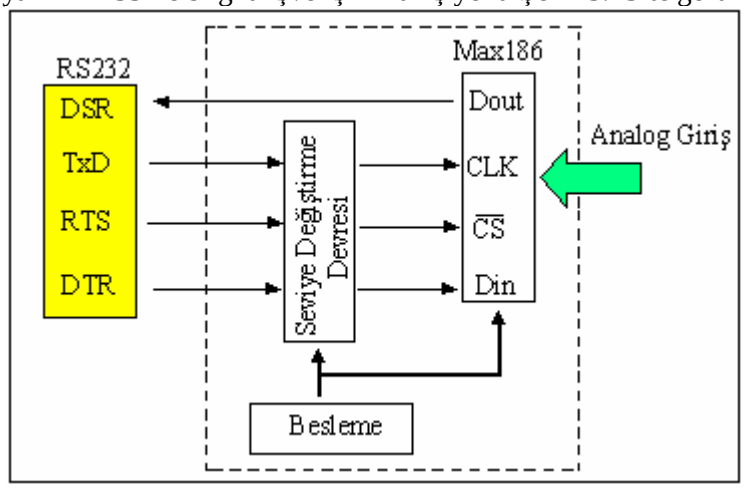
4.096 V / 4096 = 1 mV/ bit'dir.

Yonganın bacak yapısı Şekil 3.22'de görülmektedir.



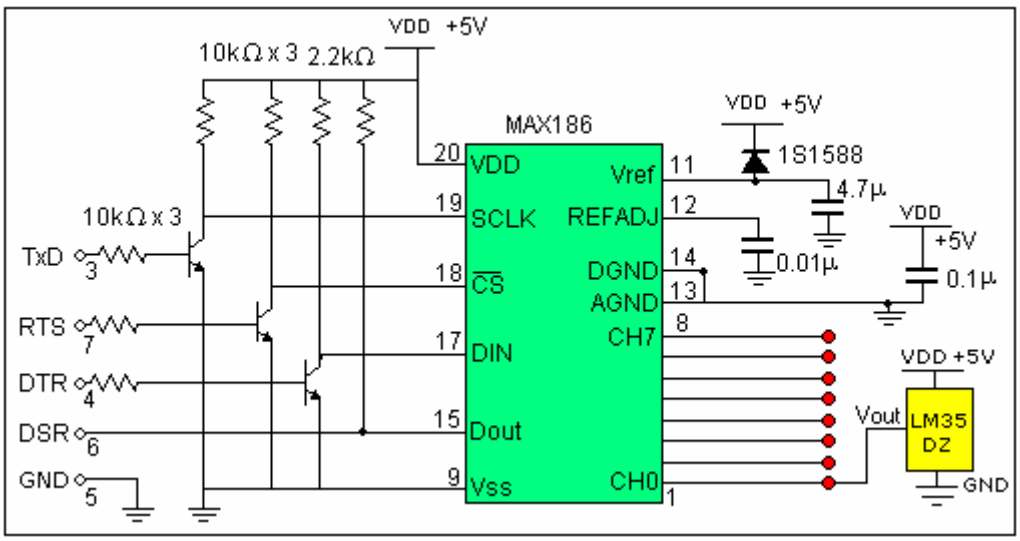
Şekil 3.23: MAX 186 yongası

Bilgisayarın LM35 ile bilgi alışverişinin akış yönü Şekil 3.23'te görülmektedir.



Şekil 3.24: Seri Port- Max 186 ilişkisi

Sıcaklık ölçümü için kullanılacak devrenin şeması Şekil 3.24'te verilmiştir.



Şekil 3.25: Devre şeması

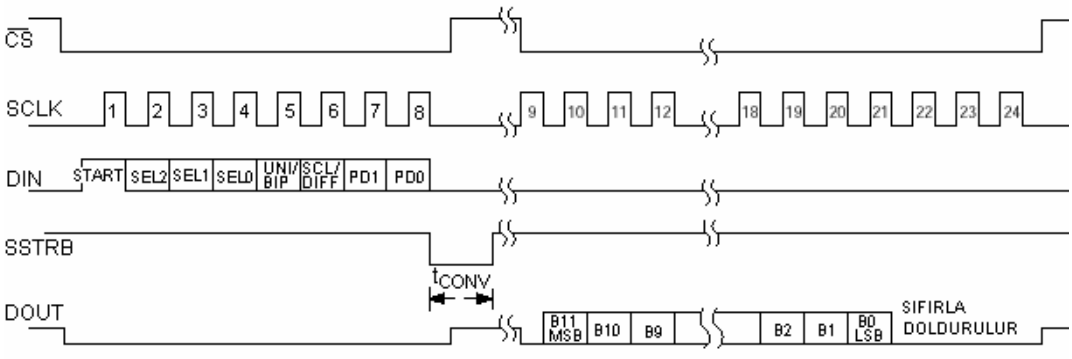
RS232 sinyal seviyesi 12V olduğundan MAX186 ile uyumlu olması için 5V'a düşürülmelidir. Burada transistör kullanılarak bu seviye yakalanmıştır. 3V üzerindeki sinyal seviyesi seri port tarafından algılandığından MAX186 tarafından DSR'ye uygulanan 5V'luk giriş değişmeden kullanılır.

MAX186 seri olarak çalışan A/D çevirici olduğundan bilgiler seri olarak alınır. Yongayı kullanabilmek için öncelikle ayar bilgilerinin program tarafından seri olarak yongaya yüklenmesi gerekir. Yüklenecek 1 baytlık bilginin her bitin anlamı aşağıdaki tabloda görülmektedir.

Bu ayar bilgileri eş zamanlı (synchronous) olarak yongaya yüklenir. Şekil 3.25'te takip edilmesi gereken sinyal durumları görülmektedir. SCLK bacağından saat darbeleri gönderilirken DIN bacağından ayar bilgilerinin girilmesi, bu arada CS bacağının da 0V seviyesinde olması gerekir. Ayar bilgileri girildikten sonra yükseğe çekilir. Analog bilginin sayısala çevrilmesi için kısa bir zaman beklendikten sonra tekrar alçağa çekilerek sayısal bilgi DSR bacağından seri porta aktarılır.

Bit		İşlem Muhtevası
7	START	“1” Dönüşümü Başlatır
6 5 4	SEL2 SEL1 SEL0	Bir Giriş Kanalı Seçer (1-8 arası giriş uçlarından biri)
3	UNI / BIP	“1” Unipolar Ölçüm Aralığı: 0~4.095 **à** 0 ~ V_{REF} “0” Bipolar Ölçüm Aralığı: -2.042~2.042 **à** - $V_{REF/2}$ ~ $V_{REF/2}$
2	SCL/ DIF	“1” Tek Giriş (Toprak hattına göre giriş kanallarından biri) “0” Fark Girişi (İki giriş hatlarıarasındaki gerilim ölçülür)
1 0	PD1 PD0	Saat Kipini(Modunu) Seçer PD1 PD0 0 0 0 1 1 0 **à** Genellikle bu kullanılır 1 1

Tablo 3.4: MAX 186 ayar bilgileri



Şekil 3.26: Zamanlama diyagramı

MAX186’nın 8 giriş kanalından herhangi birisi giriş olarak SEL2..SEL0 bacakları ile seçilebilir. Aşağıdaki tabloda bu seçim görülmektedir.

SEL2	SEL1	SEL0	Ch0	Ch1	Ch2	Ch3	Ch4	Ch5	Ch6	Ch7
0	0	0	+							
1	0	0		+						
0	0	1			+					
1	0	1				+				
0	1	0					+			
1	1	0						+		
0	1	1							+	
1	1	1								+

Tablo 3.5: Kanal seçme

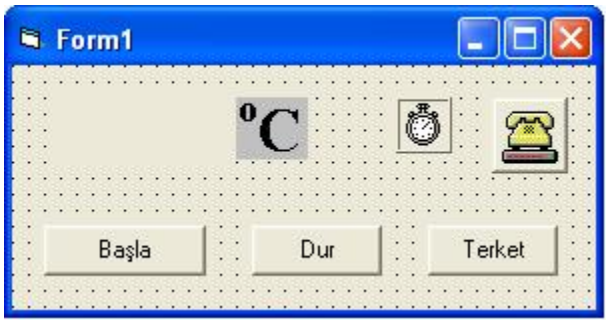
Daha önce de kullanılan Tablo 3.6'da çıkış ve giriş uçlarının adresleri görülmektedir.

Yazmaç	Ekleme Değeri	Bit 7 128	Bit 6 64	Bit 5 32	Bit 4 16	Bit 3 8	Bit 2 4	Bit 1 2	Bit 0 1
MCR	4							RTS	DTR
MSR	6	DCD	RI	DSR	CTS				
LSR	3		TD						

Tablo 3.6: MSR yazmacı

Program:

Form tasarımı Şekil 3.26'da görüldüğü gibi yapalım Timer'ın Interval özelliğini 500 olarak ayarlayalım.



Şekil 3.27: Form tasarımı

Projenin modülü:

```
Global MCR
Global LCR2
Global MSR
Global READDATA

Sub data_read()
MCR = &H3FC
LCR2 = &H3FB
MSR = &H3FE

Dim indata, i, j, a
GoSub odt001    'İlk ayar
GoSub odt000
GoSub odt100

GoSub odt110    'CLK
GoSub odt000
```

```
GoSub odt010    'SEL2
GoSub odt000

GoSub odt010    'SEL1
GoSub odt000

GoSub odt010    'SEL0
GoSub odt100

GoSub odt110    'UNI
GoSub odt100

GoSub odt110    'SGL
GoSub odt100

GoSub odt110    'PD1
GoSub odt000

GoSub odt010    'PD0
GoSub odt001
GoSub odt000

indata = 0
j = &H800
For i = 12 To 1 Step -1
GoSub odt010
GoSub odt000
a = Inp(MSR)
If (a And &H20) <> 0 Then
indata = indata + j
End If
j = j / 2
Next i
GoSub odt010
GoSub odt000

GoSub odt010
GoSub odt000

GoSub odt010
GoSub odt000

READDATA = indata
GoTo Hitam

odt000:
```

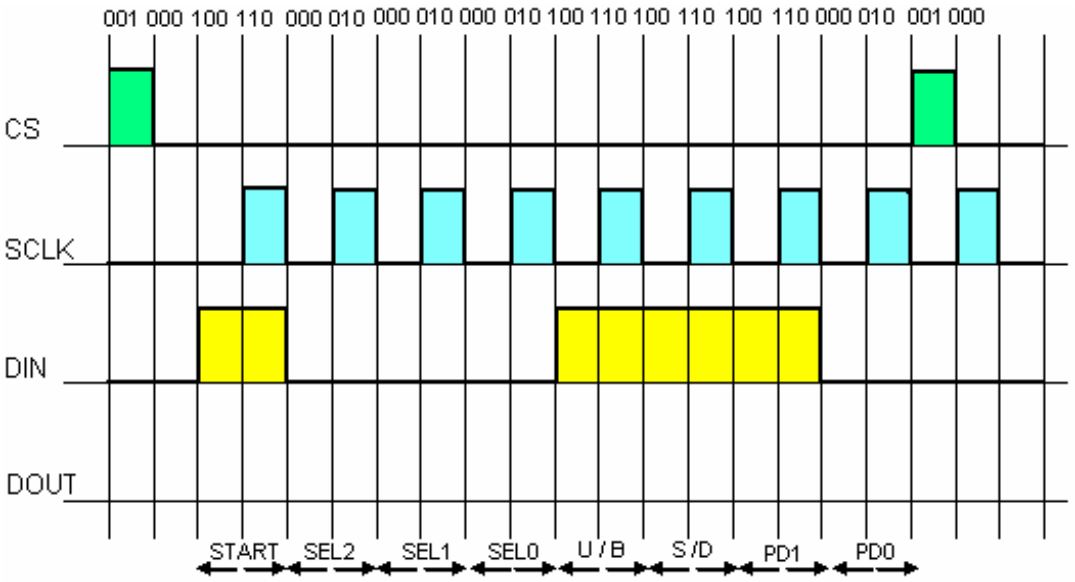
```
Call Out(LCR2, 64)
Call Out(MCR, 3)
Return

odt001:
Call Out(LCR2, 64)
Call Out(MCR, 1)
Return

odt010:
Call Out(LCR2, 0)
Call Out(MCR, 3)
Return

odt011:
Call Out(LCR2, 0)
Call Out(MCR, 1)
Return

odt100:
Call Out(LCR2, 64)
Call Out(MCR, 2)
Return

odt101:
Call Out(LCR2, 64)
Call Out(MCR, 0)
Return

odt110:
Call Out(LCR2, 0)
Call Out(MCR, 2)
Return
odt111:
Call Out(LCR2, 0)
Call Out(MCR, 0)
Return
Hitam:
End Sub
```

Form ait kodlar:

```
Private Sub Command1_Click()
MSComm1.PortOpen = True
Timer1.Enabled = True
Image1.Visible = True
End Sub
```

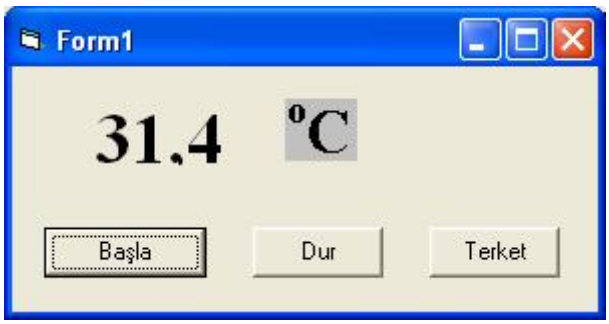
```
Private Sub Command2_Click()
Timer1.Enabled = False
Label1.Caption = ""
Image1.Visible = False
MSComm1.PortOpen = False
End Sub

Private Sub Command3_Click()
End
End Sub

Private Sub Form_Load()
MSComm1.CommPort = 1
End Sub

Private Sub Timer1_Timer()
Call data_read
Label1.Caption = READDATA / 10
End Sub
```

GoSub ile gidilen etiketlerle A/D'ye ayar bitleri yüklenmektedir. Şekil 3.27, etiketlere göre sinyal seviyelerini göstermektedir.



Şekil 3.28: Zamanlama sinyal seviyeleri

Programın çalışmasına ait ekran görüntüsü:



Şekil 3.29: Ekran görüntüsü

UYGULAMA FAALİYETİ

Aşağıdaki sorulara ilişkin uygulama faaliyetini yapınız.

- Örnek 3.2'te yapılan uygulamaya RTS-CTS el sıkışma sinyallerini ekleyiniz.
- Dll kullanrak basit bir mesaj gönderme alma programı yapınız.
- Aynı dll programını ReadFile ve WriteFile API fonksiyonları ile yapınız.
- Winsock ile sıcaklık ölçümün internet üzerinden yapınız.

İşlem Basamakları	Öneriler
Ø Form tasarımını yapınız. Ø Bileşenlere uygun isimler veriniz. Ø Programı yazınız. Ø Yazdığınız programı derleyiniz. Ø Programda hata var ise bunları düzeltiniz. Ø Ekran görüntüsünü kontrol ediniz.	Ø Programda kullanacağınız değişkenlerin tipini belirleyiniz. Ø Değişken isimlendirme kurallarına dikkat ediniz. Ø Program satırlarının düzenli olmasına özen gösteriniz. Ø Karar ifadelerinin belirlenen şartlara uygun olmasına dikkat ediniz.

ÖLÇME VE DEĞERLENDİRME

OBJEKTİF TESTLER (ÖLÇME SORULARI)

1. Aşağıdaki ifadelerden hangisi bir el sıkışma bağlantısı değildir?

A) CTS-RTS
B) DSR-DTR
C) DSR-DTR-DCD
D) CTS-DTR

2. Aşağıdakilerden hangisi bir yazılım el sıkışma anlaşmasıdır?

A) XON-XOFF
B) DSR-DTR
C) DSR-DTR
D) CTS-DTR

3. Aşağıdakilerden hangisi standart bir baud oranı değildir?

A) 19200
B) 110
C) 600
D) 3800

4. Veri bitini ayarlarken aşağıdakilerden hangisinin belirtilme zorunluluğu yoktur?

A) Stop biti
B) Eşlik biti
C) Başlangıç biti
D) Veri biti

DEĞERLENDİRME

Cevaplarınızı cevap anahtarı ile karşılaştırınız. Doğru cevap sayınızı belirleyerek kendinizi değerlendiriniz. Yanlış cevap verdiğiniz ya da cevap verirken tereddüt yaşadığınız sorularla ilgili konuları faaliyete geri dönerek tekrar inceleyiniz.

MODÜL DEĞERLENDİRME

YETERLİK ÖLÇME

Modülde yaptığınız uygulamaları tekrar yapınız. Yaptığınız bu uygulamaları aşağıdaki tabloya göre değerlendiriniz.

AÇIKLAMA: Aşağıda listelenen kriterleri uyguladıysanız EVET sütununa, uygulamadıysanız HAYIR sütununa X işareti yazınız.		
Değerlendirme Ölçütleri	**Evet**	**Hayır**
Ø Bileşenlere doğru isimler verdiniz mi?		
Ø Değişken tanımlamalarını tam yaptınız mı?		
Ø Olayları yerinde seçtiniz mi?		
Ø Bileşen özelliklerini uygun atamalar yaptınız mı?		
Ø For döngüsünü kullandınız mı?		
Ø Şartlı ifadeler kullandınız mı?		
Ø Dll dosyasını tanımladınız mı?		
Ø API tanımlamalarını yaptınız mı?		
Ø Program yazım işlemlerinin doğruluğunu kontrol ettiniz mi?		

DEĞERLENDİRME

Hayır cevaplarınız var ise ilgili uygulama faaliyetini tekrar ediniz. Cevaplarınızın tümü evet ise bir sonraki modüle geçebilirsiniz.

CEVAP ANAHTARLARI

ÖĞRENME FAALİYETİ -1 CEVAP ANAHTARI

1	**C**
2	**A**
3	**A**
4	**B**
5	**D**

ÖĞRENME FAALİYETİ -2 CEVAP ANAHTARI

1	**D**
2	**A**
3	**B**
4	**C**
5	**B**

ÖĞRENME FAALİYETİ -3 CEVAP ANAHTARI

1	**D**
2	**A**
3	**C**
4	**B**

KAYNAKÇA

- David I. Schneider, **Computer Programming Concepts and Visual Basic,** Pearson Custom Publishing, 1999.
- Peter Norton, **Peter Norton's Guide to Visual Basic 6,** Macmillan Computer Publishing, 1998.
- Steven Holzner The Coriolis Group, **Visual Basic 6 Black Book,** 1998.
- Zeydin PALA, İhsan KARAGÜLLE, **Visual Basic 6.0 Pro,** Türkmen Yayınevi, İstanbul, 2002.
- Jan Axelson, **Her Yönüyle Seri Port,** Bileşim Yayıncılık, İstanbul, 2000.
- http://vb-helper.com/
- http://www.vbitalia.it
- http://www.thaiio.com
- http://www.vincenzov.net
- http://www.bitwisemag.com
- http://www.rs485.com