

T.C.
MİLLÎ EĞİTİM BAKANLIĞI



MEGEP

(MESLEKÎ EĞİTİM VE ÖĞRETİM SİSTEMİNİN
GÜÇLENDİRİLMESİ PROJESİ)

ENDÜSTRİYEL OTOMASYON
TEKNOLOJİLERİ

BİLGİSAYARLI KONTROL-5

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 ulaşılabilirler.
- Basılmış modüller, eğitim kurumlarında öğrencilere ücretsiz olarak dağıtılır.
- Modüller hiçbir şekilde ticari amaçla kullanılamaz ve ücret karşılığında satılamaz.

İÇİNDEKİLER

| | |
|---|----|
| AÇIKLAMALAR | ii |
| GİRİŞ | 1 |
| ÖĞRENME FAALİYETİ-1 | 3 |
| 1. PARALEL PORTUN TANIMI..... | 3 |
| 1.1. Giriş..... | 3 |
| 1.1.1. Paralel Haberleşme | 4 |
| 1.2. Paralel Port Modları..... | 5 |
| 1.3. Paralel Port Yazmaçları | 5 |
| 1.3.1. Veri Yazmacı | 8 |
| 1.3.2. Durum Yazmacı | 8 |
| 1.3.3. Denetim Yazmacı..... | 8 |
| UYGULAMA FAALİYETİ | 9 |
| ÖLÇME VE DEĞERLENDİRME | 12 |
| ÖĞRENME FAALİYETİ-2 | 13 |
| 2. PARALEL PORTTAN ÇIKIŞ ALMAK..... | 13 |
| 2.1. Visual Basic Port Denetimi..... | 13 |
| 2.1.1. Paralel Portun Veri Yazmacından Çıkış Almak | 14 |
| 2.1.2. Paralel Portun Denetim Yazmacından Çıkış Almak | 22 |
| UYGULAMA FAALİYETİ | 24 |
| ÖLÇME VE DEĞERLENDİRME | 32 |
| ÖĞRENME FAALİYETİ-3 | 33 |
| 3. PARALEL PORTTAN GİRİŞ ALMAK..... | 33 |
| 3.1. Durum Yazmacından Giriş Almak | 33 |
| 3.1. Veri Yazmacından Giriş Almak..... | 37 |
| UYGULAMA FAALİYETİ | 40 |
| ÖLÇME VE DEĞERLENDİRME | 42 |
| MODÜL DEĞERLENDİRME | 43 |
| CEVAP ANAHTARLARI..... | 44 |
| KAYNAKÇA | 45 |

AÇIKLAMALAR

| | |
|--|--|
| KOD | 523EO0316 |
| ALAN | Endüstriyel Otomasyon Teknolojileri |
| DAL/MESLEK | Alan Ortak |
| MODÜLÜN ADI | Bilgisayarlı Kontrol-5 |
| MODÜLÜN TANIMI | Paralel portun yapısı ve visual basic programlama dilinde paralel port üzerinden veri alış verişi işlemlerini yapabilmeye yeterliliğinin kazandırıldığı modüldür. |
| SÜRE | 40/32 |
| ÖN KOŞUL | Bilgisayarlı Kontrol-4 modülünü almış olmak. |
| YETERLİK | Paralel port kontrolü yapmak. |
| MODÜLÜN AMACI | Genel Amaç: Paralel port ile giriş ve çıkış işlemlerini doğru olarak yapabileceksiniz. Amaçlar 1. Paralel port kontrol devresini devre şemasına ve baskı devre tekniklerine uygun olarak yapabileceksiniz. 2. Paralel port donanımı yoluyla sayısal çıkış değerlerini kontrol edebileceksiniz. 3. Paralel port donanımı yoluyla sayısal giriş değerlerini kontrol edebileceksiniz. |
| EĞİTİM ÖĞRETİM ORTAMLARI VE DONANIMLARI | Ortam: Bilgisayar Laboratuvarı, Elektrik-Elektronik Laboratuvarı Donanım: Visual Basic 6.0 çalıştırabilen bilgisayar, baskı devre araç gereçleri, lehimleme araç gereçleri, multimedre. Yazılım: Visual Basic 6.0. programlama dili |
| ÖLÇME VE DEĞERLENDİRME | Her faaliyetin sonunda ölçme soruları ile öğrenme düzeyinizi ölçeceksiniz. Araştırmalarla, grup çalışmaları ve bireysel çalışmaları öğretmen rehberliğinde ölçme ve değerlendirmeyi gerçekleştirebileceksiniz. |

GİRİŞ

Sevgili öğrenci,

Bir kumanda işlemini bilgisayarla yapmaya karar veren bir teknisyen için paralel port iyi bir seçenek olabilir. Paralel porta hükmedebilmek için ise onun yapısının iyi bilinmesi gerekmektedir. Esasında yapısı oldukça basit olan paralel port iyi bilindiğinde herhangi bir programla dili kullanılarak etkin uygulamalar geliştirilebilir. Elbette tercih edilecek programlama dilinin paralel port kontrol ve erişim komutlarının iyi bilinmesi de gerekmektedir.

Bu modülün birinci bölümünde yazıcı bağlantı noktası olarak geliştirilen paralel port etraflıca incelenerek bilinmesi gereken temel hususlara değinilmiştir. İkinci bölümde ise Visual Basic programlama dili kullanılarak paralel port üzerinden bilgilerin nasıl dış ortama gönderileceği konusu işlenmiştir. Son bölüm ise paralel port kullanılarak dış ortamdan gönderilen işaretlerin alınması konusuna ayrılmıştır.

Her bölümün sonunda uygulama faaliyetleri verilmiştir. Bu uygulama faaliyetlerinde program kodları sizleri kendi algoritmalarınızı serbestçe düşünmeye ve oluşturmaya sevk etmek için özellikle verilmemiştir.

Bu üç bölümde verilen örnek uygulamaların eksiksiz yapıldığı takdirde paralel portla bilgi alışverişi yönteminin zihinlerinizde aydınlanacağına inanıyoruz.

ÖĞRENME FAALİYETİ-1

AMAÇ

Paralel port kontrol devresini devre şemasına va baskı devre tekniklerine uygun olarak yapabileceksiniz.

ARAŞTIRMA

- Endüstride, okulunuzdaki atölye ve laboratuarlarda paralel port kullanarak haberleşme yapan cihaz yada sistemleri araştırınız.

1. PARALEL PORTUN TANIMI

1.1. Giriş

Bilgisayar denilince insanlar genellikle hızla hesap yapan, bilgileri tutan ve gösteren , internete girilebilen bir büro cihazı düşünür. Bu düşünce doğru olmakla birlikte bilgisayarların işlevlerinin bir kısmını oluşturur. Bilgisayarlar aynı zamanda farklı ekipmanlar ve makinelerle de irtibat kurabilen, haberleşebilen ve onları kontrol edebilen cihazlardır. Bilgisayarların dış ortamla kurdukları irtibat noktalarına port adı verilir. Port terimi Türkçe “liman” anlamına gelmektedir. Nasıl ki limanlar bir ülkelin dışarıyla mal alışverişinin sağlandığı alanları ise portlar da bilgisayarların yazılımlar vasıtasıyla dış dünyaya bağlandığı donanımlardır.

Liman örneğinden devam edecek olursak hava limanları, deniz limanları gibi temel işlevi aynı yöntem ve teknikleri farklı limanlar bulunduğu gibi bilgisayar üzerinde seri ,paralel , usb, PS/2, IrDA gibi farklı portlar mevcuttur.

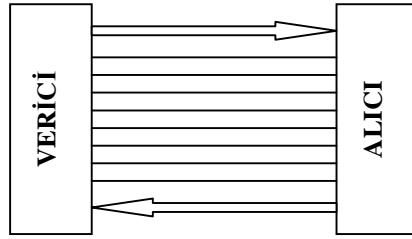


Şekil 1.1: Bilgisayar portları

Portların temel görevi yazıcıyla, modemle, ekranla, klavye ile yada sistemin dışındaki herhangi bir aygıt yada birimle iletişimdir.

1.1.1. Paralel Haberleşme

İki farklı noktada bulunan alıcı ve verici cihazların birbirlerine veri aktarımını sırasında verinin her bir biti için ayrı bir hat kullanılıyorsa bu tür iletişime paralel iletişim adı verilir. Aşağıdaki şekilde verici cihaz ile alıcı cihaz arasında kurulan paralel bağlantının basit gösterimi vardır. Verici cihaz elindeki 8 bitlik (1 byte) veriyi göndermeden önce alıcıyı haberdar eder ve “ben sana bilgi gönderiyorum” anlamına gelen bir sinyal gönderir. Eğer alıcı meşgul değil ise yada veri kabul edebilecek durumda ise “hazırım ” anlamında bir işareti vericiye gönderir. Bundan sonra verici her bir biti ayrı bir hat vasıtasıyla 8 bitlik veriyi gönderir. Bu işlem bir sonraki gönderilecek byte’lar içinde tekrarlanarak aktarım dosya aktarımı sonuna kadar sürdürülür.



Şekil 1.2: Paralel haberleşme

Bu modülde hemen her bilgisayarda şu an için oldukça yaygın ve etkin olarak kullanılan paralel portlar incelenecektir. Paralel port adından da anlaşılabilir gibi aynı anda birden fazla veri bitini birlikte aktarabilen bir port türüdür. Bunu örneğin 8 şeritli bir otoyola benzetebiliriz. Böyle bir otoyolda 8 adet araç arka arkaya gitmek yerine farklı şeritlerden aynı anda aynı hedefe ilerleyebilir.

Bilgisayarlar insan yaşamına girdiğinde, üretilen dokümanların ve yapılan işlerin sanal ortamdan gerçek ortama yani yazıcılar vasıtasıyla kağıtlara basılması fikri ortaya çıkmıştır. Bunun sonucu olarak tasarlanan ve yazıcı adı verilen cihaza bilgisayardan verilerin doğrudan kablo bağlantısıyla gönderilebilmesi için paralel port teknolojisi geliştirilmiştir. Bu sebeple her ne kadar sonradan farklı amaçlara kullanımlardaki performansından dolayı oldukça popüler ve tercih edilse de paralel portun temel görevi yazıcı bağlantısıdır.

Paralel port donanım ve yazılım elemanlarının müşterek çalışması sonucu işlev kazanır. Bu anlamda;

Donanım olarak paralel port 12 çıkış, 5 giriş ve 8 toprak hattı olan 25 D-tipi dişi bir bağlaçtır. Centronics firması tarafından geliştirilmesi dolayısıyla bu adla da anılır.



Şekil 1.3: Paralel port konnektörü (dişi)

Yazılım için paralel port, bilgisayarın giriş/çıkış (I/O map) haritasında ardışık üç adresi işgal eden 8 bitlik üç yazmaçtır.

Paralel portlar ortalama 6-7 metreye kadar ki kısa mesafe iletişimlerde ideal haberleşme imkanı sağlarlar. Ayrıca seri portlarda karşılaştırıldığında her bit ayrı hattan gönderildiği için daha yüksek bir aktarım hızına sahiptir.

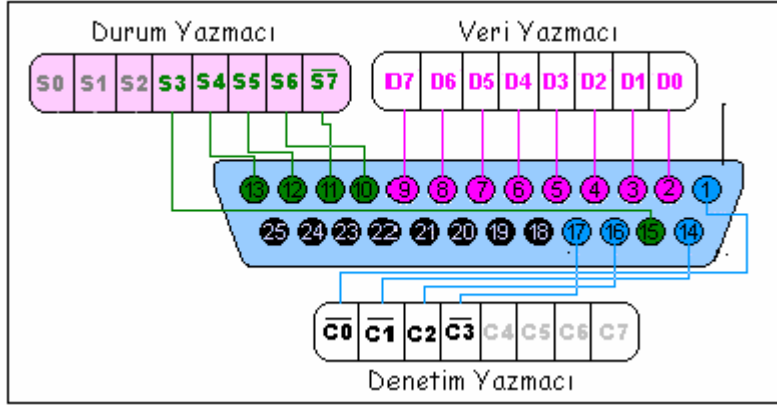
1.2. Paralel Port Modları

İlk tasarlanan paralel portlardan sonra zaman içerisinde de farklı ve daha gelişmiş port yapıları ortaya çıkmıştır. Her yeni çıkan yapı kendisinden önceki yöntemleri desteklemektedir. Bu farklı çalışma yapıları “mod” olarak isimlendirilmiştir.

- **SPP (Standard Parallel Port):** Yazıcı bağlantısını gerçekleştirmek amacıyla ilk olarak geliştirilmiştir. SPP modunda paralel portun data yazmacı yalnızca çıkış olarak kullanılmaktadır.
- **Bi-Directional:** Yazıcı ile çift yönlü haberleşmeye imkan verecek şekilde tasarlanan portlardır. Yani data hattı hem çıkış hem de giriş olarak kullanılmaktadır.
- **EPP (Enhanced Parallel Port):** Geliştirilmiş paralel port anlamına gelen EPP modunda bilgisayar ile yazıcı önceki modlarda programcılar tarafından yazılım yoluyla yaptırılan el sıkışma (handshaking) adı verilen işlemi donanım olarak gerçekleştirebilmektedir. El sıkışma işleminde kablo bağlantısı kurulduğunda her iki taraftaki cihazlar kendi durumları ile ilgili bilgileri birbirlerine göndererek haberleşmenin doğru ve güvenilir bir şekilde başlamasını sağlarlar. Ayrıca EPP modu daha hızlı haberleşmenin de yolunu açmıştır.
- **ECP (Extended Capability Port):** Geliştirilmiş kapasiteli port anlamına gelen ECP modunda EPP modunun getirmiş olduğu yeniliklerin yanı sıra tek gövde içerisinde birden fazla barındıran büro cihazlarıyla haberleşme imkanı sunar.

1.3. Paralel Port Yazmaçları

Paralel port üç adet yazmaç (register) tarafından kontrol edilir. Bu yazmaçların her birinin kendine ait bir adresi vardır. Bu adresler vasıtasıyla programcı yazmaçlara ve dolayısıyla paralel porta ulaşabilir.

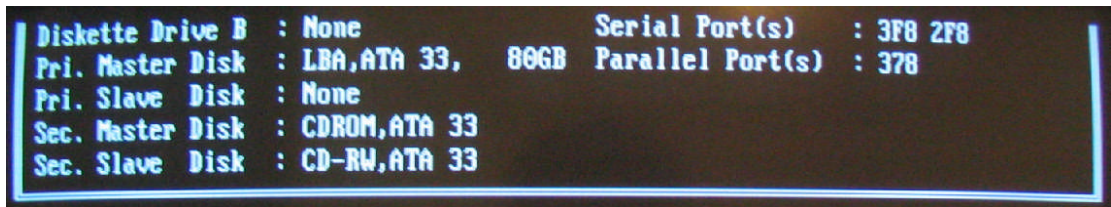


Şekil 1.4: Paralel port yazmaçları

Bilgisayarlar genellikle 3 adet paralel portu destekleyecek şekilde tasarlanmışlardır. Bu yüzden I/O haritasında 3 adet paralel port adresi ayrılmıştır. Bu adreslere aynı zamanda “taban adresi (base address)” adı verilir. Her bir paralel porta ait üç yazmacın adresleri bu taban adresinden itibaren sıralanır. Buna ilişkin verilen aşağıdaki tabloyu inceleyelim.

| | DATA YAZMACI | DURUM YAZMACI | DENETİM YAZMACI |
|--------------|-----------------|-----------------|-----------------|
| Genel | Taban adresi +0 | Taban adresi +1 | Taban adresi +2 |
| LPT1 | 0378H | 0379H | 037AH |
| LPT2 | 0278H | 0279H | 027AH |
| LPT3 | 03BCH | 03BDH | 03BEH |

Bilgisayarın I/O haritasında yer alan paralel port, seri port benzeri port adresleri ile takılı olan kartların adresleri BIOS tarafından okunarak bir tablo haline getirilir. Bu tablo her bilgisayar açılışında karşımıza çıkmaktadır.



Şekil 1.5: Bilgisayar açılış ekranı

BIOS tablosunda tutulan adreslerin yerleri standarttır. Dolayısıyla bu tablonun 0x400 :0008 adresinden itibaren LPT1 , LPT2 ve LPT3 portlarından var olanlarının adresleri tutulur. Bunu görmek için Windows komut istemini çalıştırarak “Debug” işlemiyle görebiliriz.

Bunun için komut istemine şu komutları uygulayalım.

C:\debug

- d0000:0400

Bu işlemler sonucunda aşağıdaki şekilde görüldüğü gibi karşımıza onaltılık sayı tabanında adresleme tablosu gelir. Farklı bir bilgisayarda aynı tabloyu görmeye çalıştığımızda adres sıralarının boş yada sırasının değişik olduğunu görebiliriz. Diğer bir noktada bilgisayarımızda tek bir tane paralel port olmasına karşın bu tabloda üç tane adresin var olduğunu görebiliriz. Bu durumlar BIOS versiyonlarından kaynaklanır.

```
          LPT3   LPT1   LPT2
          |     |     |
          v     v     v
Komut İstemi - debug
Microsoft Windows XP [Sürüm 5.1.2600]
(C) Telif Hakkı 1985-2001 Microsoft Corp.
C:\Documents and Settings\usevin>debug
-d0000:0400
0000:0400  F8 03 F8 02 E8 03 E8 02 BC 03 78 03 73 02 00 9F
0000:0410  23 C8 20 80 02 89 00 20 00 00 31 00 34 00 64 20
0000:0420  30 0B 30 0B 30 0B 30 0B 3A 35 30 52 34 4B 30 52
0000:0430  30 52 0D 1C 00 00 00 00 00 00 00 00 00 00 00 00
0000:0440  19 00 C3 00 00 00 00 02 02 03 50 00 80 11 00 00
0000:0450  00 0A 00 00 00 00 00 00 00 00 00 00 00 00 00 00
0000:0460  0D 0B 00 D4 03 29 30 00 00 00 00 00 A3 3B 10 00
0000:0470  00 00 00 00 00 00 08 00 14 14 14 14 01 01 01 01
```

Şekil 1.6: Debug işlemi ve adresler

Paralel port konnektöründe bulunan iğnelerin görevleri şu şekildedir.

| NO | GÖREVİ | AÇIKLAMASI |
|---------------|--------------------------------|--|
| 1.iğne | Metronom(strobe) hattı. | |
| 2-9.iğneler | Veri hattı | |
| 10.iğne | Acknowledge (hazır) hattı. | Yazıcının hazır olduğu ve bilgilerin gönderebileceğini belirtir. |
| 11.iğne | Busy (Meşgul) hattı | |
| 12.iğne | No paper (Kağıt bitti) hattı | |
| 13.iğne | Selected (seçme) hattı | |
| 14.iğne | AutoLineFeed (Satırbaşı) hattı | |
| 15.iğne | Error (Hata) hattı | |
| 16.iğne | Initialize (Açma/Kapama) hattı | |
| 17.iğne | Select (Hazır / Değil) hattı | |
| 18-25.iğneler | Ground (Toprak) hattı | |

1.3.1. Veri Yazmacı

Veri yazmacı konnektör üzerinden gönderilecek veya alınacak 8 adet veri bitini (D0 – D7) tutar. Yazmaca yeni bir bilgi yazılıncaya kadar bu tutma işlemini yapar. Bu yazmaç bilgisayardan yazıcıya gönderilen belgelerin aktarılması işinde kullanılır. Veri yazmacının bitleri konnektörün 2. ile 9.iğneleri arasında fiziksel karşılığını bulur. Herhangi bir bilgisayarlı kontrol sisteminde genellikle dışarıya veri gönderme işlemlerinde tercih edilir.

1.3.2. Durum Yazmacı

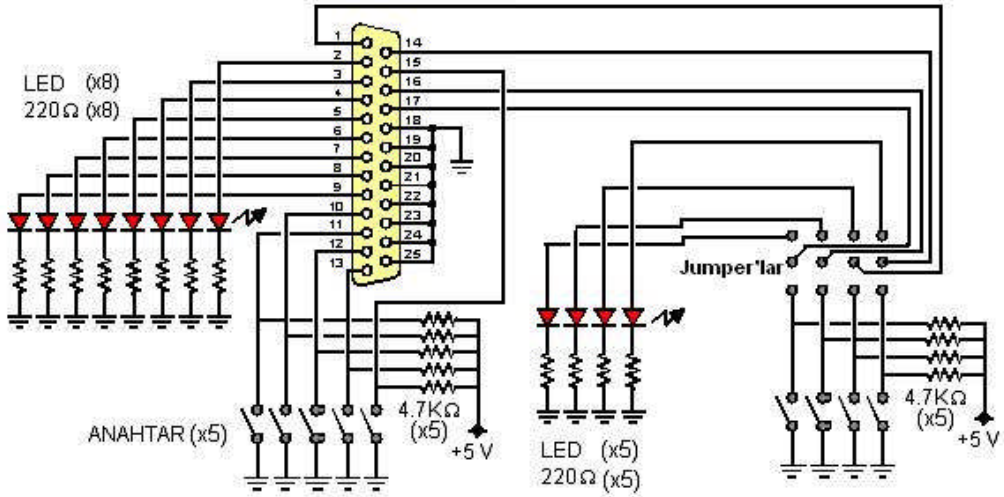
Esas görevi haberleşme esnasında yazıcının durumuna ilişkin gelen sinyalleri tutmak olan durum portu aynı zamanda harici ortamdan giriş sinyali almak içinde kullanılabilir. 10,11,12,13,14 ve 15 nolu iğnelere bağlı 5 biti kullanılan bu yazmacın diğer 3 biti kullanılmamaktadır. 11 nolu iğneye bağlı S7 biti terlenmiş durumdadır. Kullanılmayan 3 bit ileride geliştirilebilecek yöntem ve tekniklerde kullanılmak üzere ayrılmıştır.

1.3.3. Denetim Yazmacı

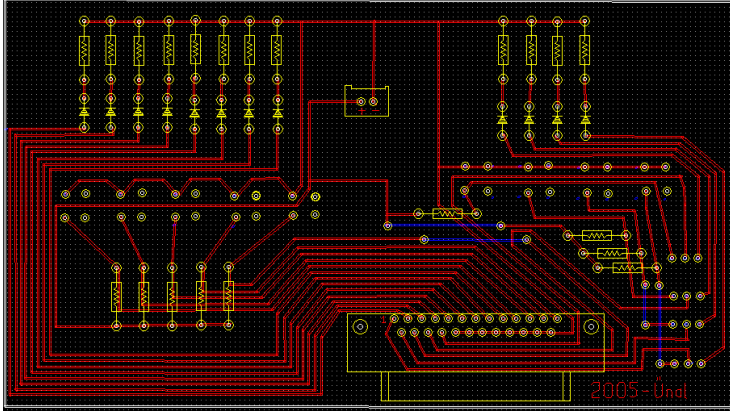
Bilgisayarın yazıcıyı kontrol edebilmesi için gerekli sinyalleri tutan bu yazmaç hem giriş hem de çıkış olarak harici ortamdan sinyal alışverişinde kullanılır. Port konnektörü üzerindeki 4 adet iğne bu yazmacın 4 bitine bağlanmıştır. Bu bitlerden C0,C1 ve C3 bitleri terslenmiştir. Diğer 4 bit ise durum portunda olduğu gibi ileride geliştirecek uygulamalarda kullanılmak üzere ayrılmıştır.

UYGULAMA FAALİYETİ

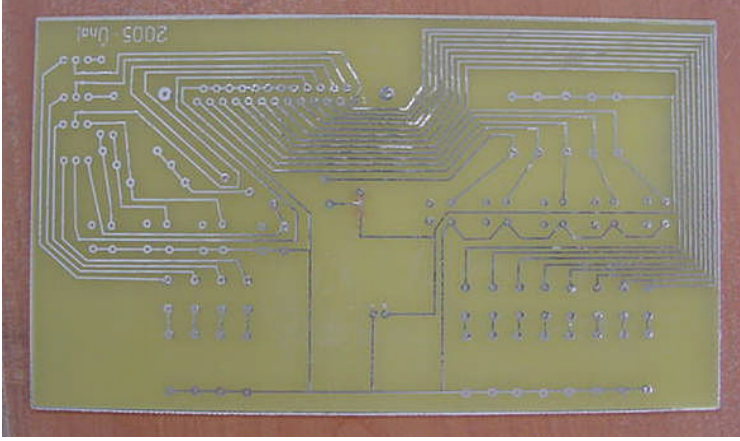
Aşağıdaki işlem basamaklarına göre uygulama faaliyetini yapınız.



Şekil 1.7: Denei kartı devresi

| İşlem Basamakları | Öneriler |
|---|----------|
| <p>➤ Yukarıda verilen devrenin baskı devre şemasını , herhangi bir baskı devre çizim programı yardımıyla çiziniz.</p>  | |

- Çizdiğiniz baskı devre şemasını pozitif20 yada ipek baskı gibi seçeceğimiz bir yöntem ile bakır plakete üzerine aktarınız.



- Hazır olan bakır plaketimizi çözelti içerisinde atarak baskı devre kartımızı elde ediniz.

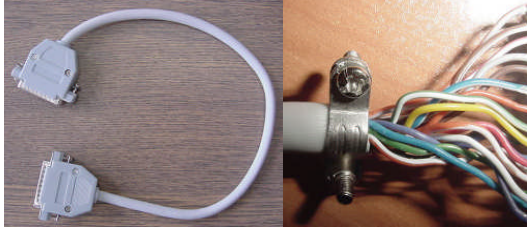
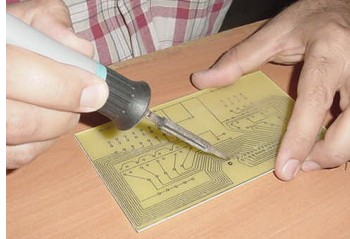


- Elemanların montajı için gerekli delikleri matkap yardımıyla açınız.



- 0,8mm'lik matkap ucu kullanınız.

| | |
|--|---|
| <p>➤ Lehim kullanarak tüm elemanları kart üzerine monte ediniz.</p> | |
| <p>➤ Deney kartı üzerine elemanların montajının tamamlanmasının ardından pleksiglas malzemeden altlık keserek katın tabanını ve ayaklarını monte ederek işlemi tamamlayınız.</p> | |
| <p>➤ Bağlantı kablosu için yeterli uzunlukta kablo kesiniz ve kablo uçlarınızı hazırlayınız.</p> | |
| <p>➤ Kablonun her iki tarafına 25. iğneli konnektörlerin montajını yaparak işlemi sonlandırınız.</p> | <p>➤ Kabloların montajında telleri birbir lehimleme yapınız. Örneğin A konnektörünün 2.iğnesindeki teli B konnektörünün 2.iğnesiyle lehimleyiniz.</p> |



ÖLÇME VE DEĞERLENDİRME

Aşağıdaki soruları cevaplayarak bu faaliyette kazandığınız bilgileri ölçünüz.

OBJEKTİF TESTLER (ÖLÇME SORULARI)

1. Aşağıdakilerden hangisi bir port değildir.

- A) IrDA B)Seri C)Status D)PS/2

2. Paralel port yapısından kaç tane yazmaç bulunur?

- A) 1 B)2 C) 3 D)4

3. Geliştirilmiş kapasiteli port tanımını aşağıdakilerden hangisine uyar?

- A) PPP B)SPP C) EPP D)ECP

4. Durum yazmacının görevi nedir?

- A) Gönderilecek bilgileri denetler
B)Yazıcının durumuna ilişkin gelen sinyalleri tutar
C) İletişim esnasında bilgisayarın durumuna ilişkin verileri tutar
D)Bilgisayarın yazıcıyı kontrol edebilmesi için gerekli sinyalleri tutar.

5. Paralel port konnektöründeki 15 nolu iğnenin görevi nedir.

- A) Error (Hata) B)Busy (Meşgul) C) Ground (Toprak) D)Select (Seçme)

6. Aşağıdakilerden hangisi paralel port adresi değildir?

- A) 378H B)278H C) 3F8H D)03BCH

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

Paralel port donanımı yoluyla sayısal çıkış değerlerini kontrol edebileceksiniz.

ARAŞTIRMA

- Farklı programlama dillerinde paralel port üzerinden dış ortama veri gönderimi nasıl yapılmaktadır?

2. PARALEL PORTTAN ÇIKIŞ ALMAK

Bir bilgisayarın paralel portunu kullanarak dışarıya bilgi göndermek kolay sayılabilecek bir işlemdir. Bunu herhangi bir programlama dili kullanarak yapabiliriz. Ancak biz bu bölümde Visual Basic programlama dili ile gerçekleştireceğiz.

2.1. Visual Basic Port Denetimi

Dll dosyaları programların dinamik olarak bağlantı kurabileceği fonksiyonların bir kütüphanesidir. Visual Basic programlama diline ait “inpout32.dll” kütüphanesi de içerisinde paralel porta erişim ve denetim için gerekli fonksiyonları içerisinde barındırır. Böylelikle programcıların fazladan gereksiz kod yazımı yapmasına gerek kalmamış olur. Inpout32.dll kütüphanesi programcının kendisi tarafından “c” yada “Delphi” benzeri bir programla yazılabildiği gibi internet benzeri bir kaynaktan hazır olarak temin edilebilir.

Visual Basic kodları ile paralel porta erişmek ve denetimini yapmak için öncelikle “inpout32.dll” dosyasının bilgisayarınızın “c:\windows\system” klasörü içerisinde bulunuyor olması gerekmektedir. Eğer bilgisayarınızda bu Visual Basic eklenti dosyası mevcut değil ise aşağıdaki internet adresinden bu dosyayı edinerek ilgili klasöre kopyalamamız gerekir.

[http:// www.programmersheaven.com/download/28409/download.aspx](http://www.programmersheaven.com/download/28409/download.aspx)

Paralel port yazmaçlarından bir değer göndermek için “inpout32.dll” kütüphanesine ait “OUT()” komutunu kullanmak yeterlidir. Bu komutun kullanım formatı aşağıdaki gibidir.

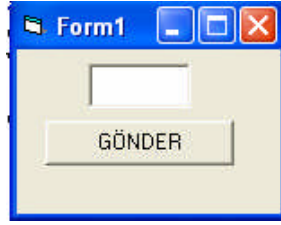
OUT Val (&adres) , Gönderilecek Veri

2.1.1. Paralel Portun Veri Yazmacından Çıkış Almak

Basit düzeyden karmaşığa doğru gidecek şekilde aşağıdaki örnekler yardımıyla paralel porttan çıkış alma işlemini öğrenelim

Örnek 2.1:

Bir metin kutusuna girdiğimiz 0-255 arası bir sayıyı paralel port üzerinden deney kartımıza gönderebiliriz. Gönderdiğimiz verinin ikilik düzendeki karşılığını kart üzerinde bulunan veri hatlarına bağlı ledler üzerinde görebiliriz. Bunun için aşağıda gösterilene benzer bir form tasarlayalım. Ardından kullandığımız nesnelere özelliklerini değiştirelim.



Şekil 2.1: Örnek form

| Nesne | Name | Caption |
|--------------|--------|---------|
| Buton | gonder | GÖNDER |
| Metin Kutusu | veri | |

Form tasarımını bitirdikten sonra yeni bir modül açarak , inpout32.dll kütüphanesini kullanabilmemiz için gerekli tanımlamaları aşağıdaki gibi yazalım. Daha sonra Module1 olarak kaydedelim.

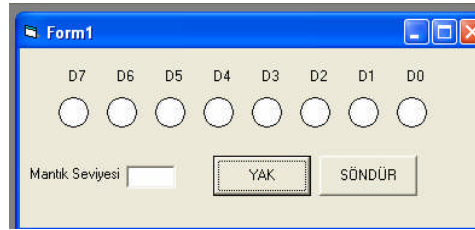
```
Public Declare Function Inp Lib "inpout32.dll" _
Alias "Inp32" (ByVal PortAddress As Integer) As Integer
Public Declare Sub Out Lib "inpout32.dll" _
Alias "Out32" (ByVal PortAddress As Integer, ByVal Value As Integer)
```

Program kodumuz oldukça kısa ve basittir. If şart yapısı ile 0-255 arası sayı girilip girilmediği denetlenmektedir. Eğer 255'ten daha büyük bir sayı girilmiş ise MsgBox fonksiyonu ile kullanıcıya bir uyarı mesajı verilmektedir. 255 veya daha küçük bir rakam girilmiş ise Out Val komutu ile 0378H adresindeki paralel porta metin kutusuna girilen değer gönderilmektedir. Sonuç olarak da deney kartı üzerinde bulunan 8 adet led vasıtasıyla gönderdiğimiz sayının ikili düzende karşılığını görebiliriz.

Örnek 2.2:

Deney kartımızın üzerinde bulunan veri hatlarına bağlı led'lerden birincisini yakan ve söndüren bir program yapalım. Bunun için aşağıda gösterilene benzer bir form tasarlayalım.

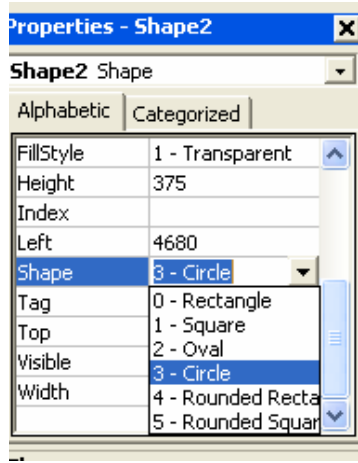
```
Private Sub gonder_Click()  
If Val(veri.Text) <= 255 Then  
    Out Val("&H378"), Val(veri.Text)  
Else  
    MsgBox "LÜTFEN 0-255 ARASI BİR RAKAM GİRİNİZ"  
End If
```



Şekil 2.2: Örnek form

Yukarıdaki form üzerine nesnelere yerleştirdikten sonra özellikler penceresinden aşağıdaki ayarları yapalım. Ledleri simgeleyen daire şekillerini çizmek için shape nesnesi açılır liste kutusundan “3-Circle” seçimini yapmalısınız.

| Nesne | Name | Caption |
|---|--------|-----------------|
| Buton | yak | YAK |
| Buton | sön | SÖNDÜR |
| Metin Kutusu | Text2 | |
| Etiket | Label1 | Mantık Seviyesi |
| Şekil nesnesi BackStye= 1-Opaque BackColor =Beyaz , Shape=3-Circle olarak ayarlayınız | Shape1 | |
| | Shape2 | |
| | Shape3 | |
| | Shape4 | |
| | Shape5 | |
| | Shape6 | |
| | Shape7 | |
| | Shape7 | |
| Etiket | Label2 | D0 |
| Etiket | Label3 | D1 |
| Etiket | Label4 | D2 |
| Etiket | Label5 | D3 |
| Etiket | Label6 | D4 |
| Etiket | Label7 | D5 |
| Etiket | Label8 | D6 |
| Etiket | Label9 | D7 |



Şekil 2.3: Özellikler penceresi

Form tasarımını bitirdikten sonra yeni bir modül açarak , inpout32.dll kütüphanesini kullanabilmemiz için gerekli tanımlamaları aşağıdaki gibi yazalım. Daha sonra Module1 olarak kaydedelim.

```
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)
```

Aşağıdaki programı incelediğimizde “veri_portu” isimli bir alt yordamın hazırlanmış olduğunu görebiliriz.

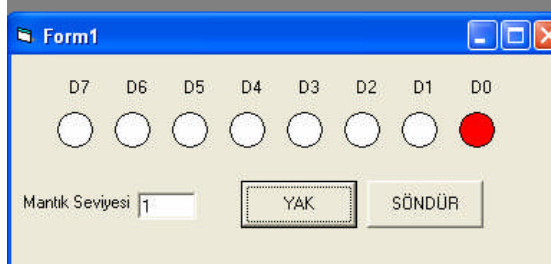
- Yak ve söndür butonlarından birisi tıklandığına ilgili butona ait olan kodlar çalışır. Yak butonu tıklandığında led1 değişkeninin içeriği 1 olur.
- Daha sonra veri_portu alt yordamı çağrılır.
- Altyordam içerisinde eğer led1 değişkeninin içeriği 1 ise text2 değişkenine 1 yazdırılır. Son olarak Out Val komutu ile 378H adresindeki paralel porta led1 içeriği olan 1 değeri (binary karşılığı 00000001 ‘dir) gönderilir.
- Bu ise D0 ucuna bağlı olan deney kartındaki 1 nolu led’e mantık 1 değerinin (+5V) gönderilmesi anlamına gelir ve bunun sonucunda 1 nolu led yanar.
- Bu aşamadan sonra alt yordam sonlanır ve program akışı tekrar yak butonu yordamına geri döner.
- Yak butonu yordamının son satırında form üzerinde D0 olarak temsil edilen led şeklinin arka planı kırmızı olur ve program sonlanır.
- Benzer şekilde söndür butonu tıklandığında ise led1 değişkenine 0 bilgisi atanarak veri_portu alt yordamı çağrılır. Led1 içeriği text2 metin kutusuna yazdırıldıktan sonra led1 bilgisi paralel port üzerinden deney kartına gönderilir.
- Led1 değişkeninin içeriği 0 olduğu için denet kartı üzerindeki 1.nolu led bu defa mantık 0 (0V) bilgisi olarak söner.

```

Dim led1 As Integer 'led1 deęişkenini tüm alt yordamlarda kullanılabilir şekilde tanımlanır.
Public Sub Veri_Portu()
    Dim veri As Integer
    veri=0
    If led1 = 1 Then veri = 1 'Eđer led1 içerięi 1 iser veri deęişkeninide 1 yap
    Text2.Text = veri 'veri deęişkeninin içerięini metin kutusuna yazdır
    Out Val(&H378), Val(led1) '378H adresindeki paralel porta led1 içerięini gönder
End Sub
Private Sub sön_Click()
    led1 = 0 ' led1 deęişkeninin içerięini 0 yap.
    Call Veri_Portu 'veri_portu alt programı çağır
    Shape2.BackColor = &HFFFFFF ' Form üzerindeki D0 ledini beyaz renk yapar.
End Sub
Private Sub yak_Click()
    led1 = 1 ' led1 deęişkeninin içerięini 1 yap.
    Call Veri_Portu 'veri_portu alt programı çağır
    Shape2.BackColor = &HFF& ' Form üzerindeki D0 ledini kırmızı renk yapar.
End Sub

```

Programımızın kod yazımını tamamladıktan sonra çalıştırarak deney kartı üzerindeki 1 nolu ledimizi bilgisayar üzerinden kontrol edebildiğimizi görebiliriz.

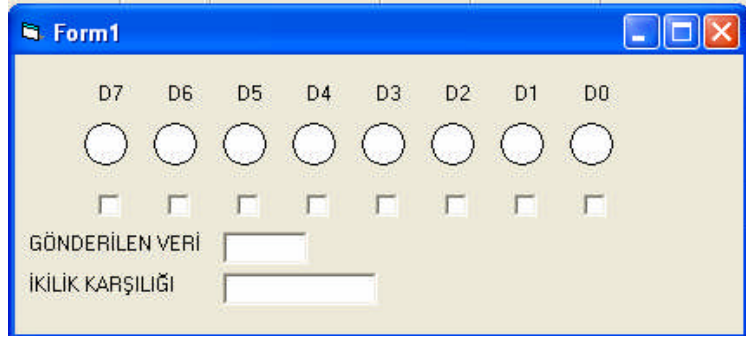


Şekil 2.4. Programın çalıştırılması

Deney kartımız ve kullandığımız led temsili bir anlam taşımaktadır.

Örnek 2.3:

Bir önceki örnekte sadece bir ledi yakıp söndürebiliyorduk. Bu örneğimizde ise tüm ledleri denetleyelim. Her zamanki gibi Visual Basic dilinin görsel dünyasından faydalanmak için aşağıdaki gibi bir formu tasarlayalım.



Şekil 2.5: Örnek form

Form üzerine yerleştirdiğimiz nesnelere ait özellikleri de ayarlayarak tasarım işlemini sonlandırabiliriz.

| Nesne | Name | Caption |
|---|--------|------------------|
| Etiket | Label0 | GÖNDERİLEN VERİ |
| Etiket | Label1 | İKİLİK KARŞILIĞI |
| Metin Kutusu | Text1 | |
| Metin kutusu | Text2 | |
| Şekil nesnesi BackStye= 1-Opaque BackColor =Beyaz , Shape=3-Circle olarak ayarlayınız | Shape0 | |
| | Shape1 | |
| | Shape2 | |
| | Shape3 | |
| | Shape4 | |
| | Shape5 | |
| | Shape6 | |
| | Shape7 | |
| Etiket | Label2 | D0 |
| Etiket | Label3 | D1 |
| Etiket | Label4 | D2 |
| Etiket | Label5 | D3 |
| Etiket | Label6 | D4 |
| Etiket | Label7 | D5 |
| Etiket | Label8 | D6 |
| Etiket | Label9 | D7 |
| İşaret kutusu | d0 | |
| İşaret kutusu | d1 | |
| İşaret kutusu | d2 | |
| İşaret kutusu | d3 | |
| İşaret kutusu | d4 | |
| İşaret kutusu | d5 | |
| İşaret kutusu | d6 | |
| İşaret kutusu | d7 | |

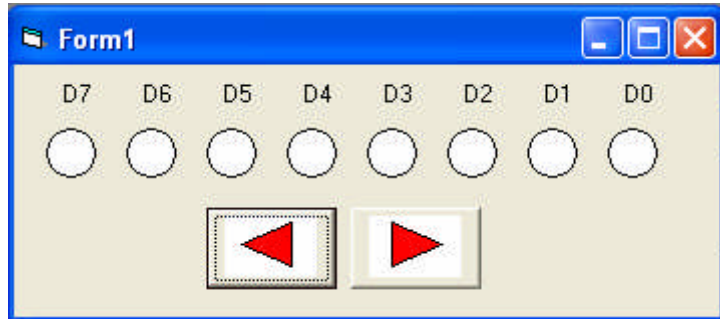
Bir önceki örneğimizde hazırladığımız modülün aynısını hazırlayıp kaydedelim. Daha sonra kod yazımına geçelim.

```
Public Sub Veri_Portu()  
Dim topla As String  
Dim veri As Integer  
If d7.Value = 1 Then veri = veri + 128  
If d6.Value = 1 Then veri = veri + 64  
If d5.Value = 1 Then veri = veri + 32  
If d4.Value = 1 Then veri = veri + 16  
If d3.Value = 1 Then veri = veri + 8  
If d2.Value = 1 Then veri = veri + 4  
If d1.Value = 1 Then veri = veri + 2  
If d0.Value = 1 Then veri = veri + 1  
topla = d7.Value & d6.Value & d5.Value & d4.Value & d3.Value & d2.Value & d1.Value & d0.Value  
Text1.Text = topla  
Text2.Text = veri  
Out Val(&H378), Val(veri)  
End Sub  
Private Sub d0_Click()  
Call Veri_Portu  
If d0.Value = 1 Then Shape0.BackColor = &HFF& Else Shape0.BackColor = &HFFFFFF  
End Sub  
Private Sub d1_Click()  
Call Veri_Portu  
If d1.Value = 1 Then Shape1.BackColor = &HFF& Else Shape1.BackColor = &HFFFFFF  
End Sub  
Private Sub d2_Click()  
Call Veri_Portu  
If d2.Value = 1 Then Shape2.BackColor = &HFF& Else Shape2.BackColor = &HFFFFFF  
End Sub  
Private Sub d3_Click()  
Call Veri_Portu  
If d3.Value = 1 Then Shape3.BackColor = &HFF& Else Shape3.BackColor = &HFFFFFF  
End Sub  
Private Sub d4_Click()  
Call Veri_Portu  
If d4.Value = 1 Then Shape4.BackColor = &HFF& Else Shape4.BackColor = &HFFFFFF  
End Sub  
Private Sub d5_Click()  
Call Veri_Portu  
If d5.Value = 1 Then Shape5.BackColor = &HFF& Else Shape5.BackColor = &HFFFFFF  
End Sub  
Private Sub d6_Click()  
Call Veri_Portu  
If d6.Value = 1 Then Shape6.BackColor = &HFF& Else Shape6.BackColor = &HFFFFFF  
End Sub  
Private Sub d7_Click()  
Call Veri_Portu  
If d7.Value = 1 Then Shape7.BackColor = &HFF& Else Shape7.BackColor = &HFFFFFF  
End Sub
```

Biraz uzun gibi görünmesine rağmen yukarıdaki programın çalışması oldukça basittir. Form üzerindeki işaret kutularından herhangi biri tıklandığında buna ilişkin kod parçası çalışır ve veri_portu alt yordamı çağrılır. Veri_portu alt yordamında veri ve topla değişkenlerine tıklanan uçların onluk ve ikilik değeri hesaplandıktan sonra Out Val komutu ile 378H adresi üzerinden paralel porta çıkış olarak gönderilir. Bunun sonucunda deney kartı üzerindeki bağlı olan led yanar. Program tekrar tıklanan işaret kutusuna ait kod parçasına geri döner ve ilgili Shape nesnesinin zemin rengi kırmızı yapılarak form üzerindeki lede yanma efekti verilir. Son olarak program akışı yeni bir tıklama olayını beklemeye koyulur.

Örnek 2.4:

Bu defa deney kartımızda bulunan ledlerimizi sağa ve sola kaydıralım. Bunun için yeni bir Visual Basic projesi açarak daha önceki örnekler için hazırladığımız gibi modülümüzü yazalım. Bundan sonra aşağıdaki gibi bir form tasarımını gerçekleştirelim.



Şekil 2.6. Örnek form

Yukarıdaki form tasarımında bu defa görsel zenginlik katmak amacıyla butonların üzerine şekiller yerleştirdik. Bu işlem için Paint yada benzeri herhangi bir çizim programında sağ ve sol ok şekilleri çizerek resim formatlarının herhangi birisinde (bmp, jpeg vs.) kaydettik. Daha sonra butonlarımızın özellikler penceresinden gerekli değişiklikleri aşağıdaki gibi yapalım.

| | | |
|---------|-------------|---------------------------------------|
| Shape | 1-Graphical | |
| Picture | (Bitmap) | Bu özelliğe çizdiğiniz resmi atayınız |

Buton üzerine resim yerleştirmeye ilgili işlemi bitirdikten sonra form üzerindeki nesnelimizin diğer özelliklerini aşağıdaki tabloya göre ayarlayalım.


```

Dim yön, basla, bitis, y As Integer
Public Sub Veri_Portu()
veri = 1
For i = basla To bitis Step yön
veri = 2 ^ i ' her dafasında ikinin karesi alınır
Out Val(&H378), Val(veri) ' porta gönder
Call geciktir 'geciktime yordamını çağır
Next
End Sub
Private Sub sag_Click() 'sağ buton yordamı
y = -1
yön = 1
basla = 0
bitis = 7
For k = 0 To 7
Shape(k).BackColor = &HFFFFFF ' dairelerin zeminini beyaz yap
Next
Call Veri_Portu
End Sub
Private Sub sol_Click() 'sol buton yordamı
y = 8
yön = -1
basla = 7
bitis = 0
For k = 0 To 7
Shape(k).BackColor = &HFFFFFF ' dairelerin zeminini beyaz yap
Next
Call Veri_Portu
End Sub
Public Sub geciktir() 'geciktirme yordamı
If yön = 1 Then
y = y + 1
Shape(y).BackColor = &HFF& ' dairelerin zeminini kırmızı yap
DoEvents
For j = 1 To 25600000
Next
Shape(y).BackColor = &HFFFFFF ' dairelerin zeminini beyaz yap
Else
y = y - 1
Shape(y).BackColor = &HFF& ' dairelerin zeminini kırmızı yap
DoEvents '
For j = 1 To 25600000
Next
Shape(y).BackColor = &HFFFFFF ' dairelerin zeminini beyaz yap
End If
End Sub

```

2.1.2. Paralel Portun Denetim Yazmacından Çıkış Almak

Paralel portun denetim yazmacı hem okunabilir hem de yazılabilir bir yazmaçtır. Dört bitinden üçü terslenmiştir. Bu bacakları “mantık 1” yapmak, ilgili bacakta 0V olarak görülür.

Denetim yazmacı normal durumda çıkış konumundadır.

Örnek 2.5:

Denetim yazmacına bağlı ledleri metin kutusundan gireceğimiz sayı kadar yakıp söndüren bir program yaparak denetim yazmacının nasıl çıkış olarak kullandığımızı öğrenebiliriz. İsteğimize uygun olarak formumuzu tasarlayalım.



Şekil 2.7: Örnek form

Form üzerine yerleştirdiğimiz nesnelerin özelliklerini ayarlayalım.

| Nesne | Name | Caption |
|---|----------|--------------------|
| Etiket | başlık | DENETİM YAZMACI |
| Etiket | yazı | YANIP SÖNME SAYISI |
| Metin Kutusu | Text1 | |
| Şekil nesnesi BackStye= 1-Opaque BackColor =Beyaz , Shape=3-Circle olarak ayarlayınız | Shape(0) | |
| | Shape(1) | |
| | Shape(2) | |
| | Shape(3) | |
| Etiket | Label2 | C0 |
| Etiket | Label3 | C1 |
| Etiket | Label4 | C2 |
| Etiket | Label5 | C3 |
| Buton | Command1 | BAŞLA |

Önceki örneklerde kullandığımız modülümüzü projemize ekledikten sonra kod yazımına geçelim.

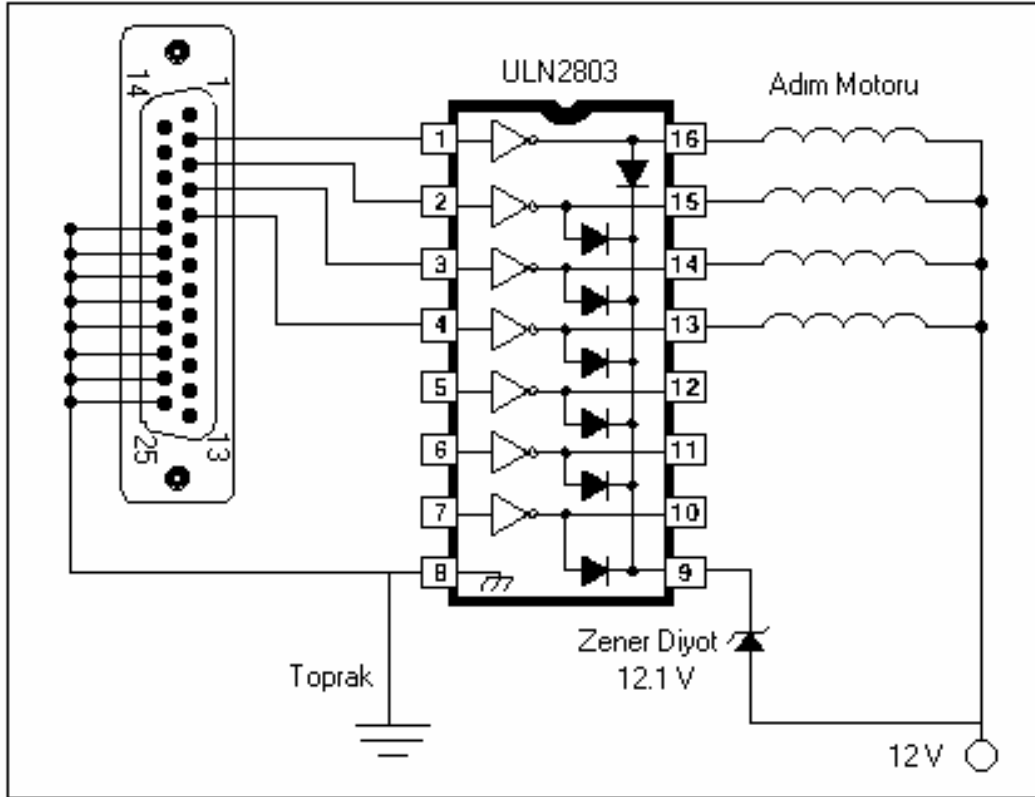
```
Public Sub denetim_Portu()
    tüm_yak = 4 'c0,c1,c3 uçları terslendiği için 0100 bilgisi gönderilir.
    tüm_sön = 11 'c0,c1,c3 uçları terslendiği için 1011 bilgisi gönderilir.
    For i = 1 To Val(Text1.Text) ' for döngüsü 1'den metin kutusuna girilen değer kadardır.
        Shape(0).BackColor = &HFF&
        Shape(1).BackColor = &HFF&
        Shape(2).BackColor = &HFF&
        Shape(3).BackColor = &HFF&
        DoEvents
        Out Val(&H37A), tüm_yak 'denetim yazmacının adresi 37A'ya 0100 gönderir.
        Call geciktir
        Out Val(&H37A), tüm_sön 'denetim yazmacının adresi 37A'ya 1011 gönderir.
        Shape(0).BackColor = &HFFFFFF
        Shape(1).BackColor = &HFFFFFF
        Shape(2).BackColor = &HFFFFFF
        Shape(3).BackColor = &HFFFFFF
        DoEvents
        Call geciktir
    Next
End Sub
Private Sub Command1_Click()
    Call denetim_Portu 'alt yordamı çağırır.
End Sub
Public Sub geciktir() 'geciktirme yordamıdır
    For j = 1 To 45000000
        Next
    End Sub
```

Programı çalıştırmadan önce deney kartımızın üzerinde bulunan 4 adet köprüyü (jumper) uygun konuma getirmemiz gerekir.

Bu işlemden sonra programımızı çalıştırdığımızda deney kartı üzerinde denetim yazmacına bağlı olan ledlerin yanıp söndüğünü hem kart üzerinde hem de ekran üzerinde görebiliriz.

UYGULAMA FAALİYETİ

Aşağıdaki işlem basamaklarına göre uygulama faaliyetini yapınız.



Şekil 2.8: Örnek form

Burada adım motorunu sürececek uçlar veri hattının 0-3 nolu (2-5 nolu iğneler) uçlarıdır. ULN2803 Darlington bağlantılı bir sürücü olup 500 mA akım tedarik edebilmektedir. Besleme hattının toprağı ile bilgisayarın toprak hattının birleştirildiğine dikkat ediniz.

GEREKLİ BİLGİ

Adım Motoru :

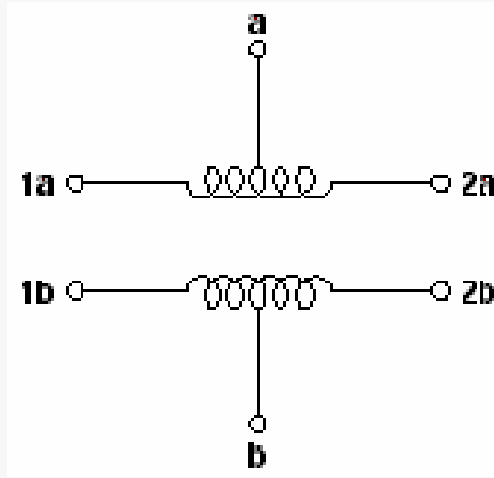
Adım motoru, sayısal darbe dizilerini açısal dönme hareketine çeviren elektrik motorlarıdır.

Adım motorlarının, motor cinsine ve uygulama alanlarına göre çeşitli çalışma usulleri vardır. Bunlar:

- Normal Sürüm (yalnızca 1 faz uyarımda)
- Tam Adım Sürüm (2 faz aynı anda uyarımda)
- Yarım Adım Sürüm (önce bir sonra iki faz uyarımda)

Normal Sürüm

Saat yönünde dönme için sargılar ardışık olarak 1a-1b-2a-2b sırasına göre uyarılmalıdır. Saat yönünün tersinde bir dönme için 2b-2a-1b-1a sırasına göre sargılar uyarılmalıdır.



GEREKLİ BİLGİ

| Adım | 1a | 1b | 2a | 2b | Onluk Sayı |
|------|----|----|----|----|------------|
| 1 | 1 | 0 | 0 | 0 | 8 |
| 2 | 0 | 1 | 0 | 0 | 4 |
| 3 | 0 | 0 | 1 | 0 | 2 |
| 4 | 0 | 0 | 0 | 1 | 1 |

Tam Adım Sürüm

Adım motorunu sürerken iki fazın daima uyarımda olmasına tam adım sürüm yada iki fazlı uyarım denir.

| Adım | 1a | 1b | 2a | 2b | Onluk Sayı |
|------|----|----|----|----|------------|
| 1 | 1 | 1 | 0 | 0 | 12 |
| 2 | 0 | 1 | 1 | 0 | 6 |
| 3 | 0 | 0 | 1 | 1 | 3 |
| 4 | 1 | 0 | 0 | 1 | 9 |

İki fazlı uyarımda tek fazlı uyarımda olduğu gibi rotor, bir adım miktarı dönecektir. Fakat bu dönme sonucunda gelinen denge konumunda, stator ve rotorun dişleri, tek fazlı uyarımda olduğu gibi aynı hizada değildir.

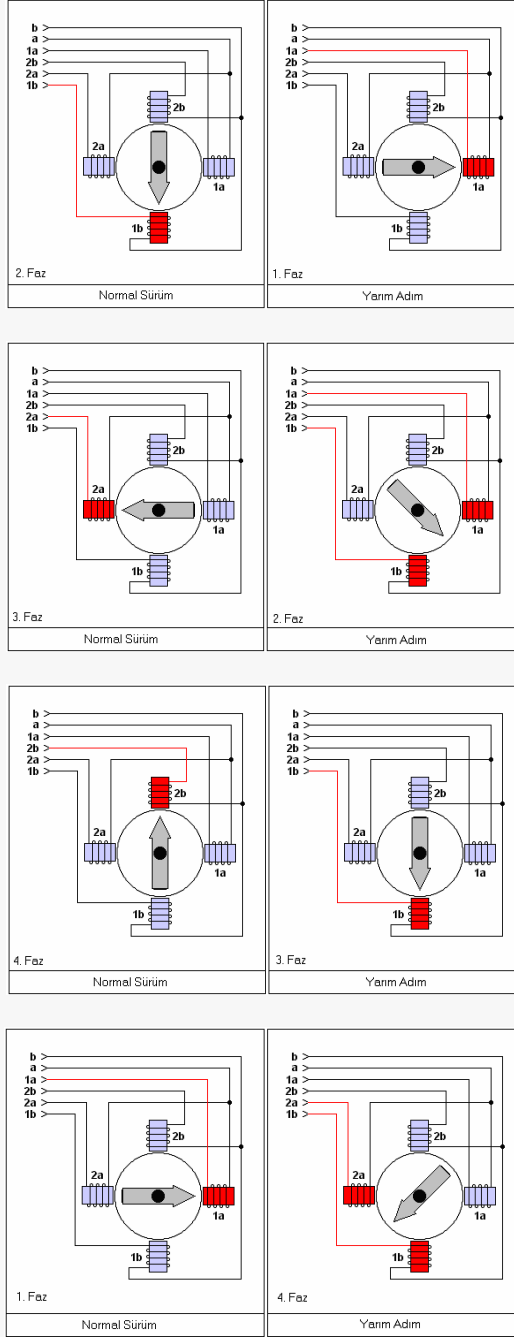
Yarım Adım Sürüm

Tek fazlı uyarım ile iki fazlı uyarımın ardışık olarak tekrarlanması ile elde edilir. Bu sebepten bir devirdeki adım sayısı da ikiye katlanmaktadır.

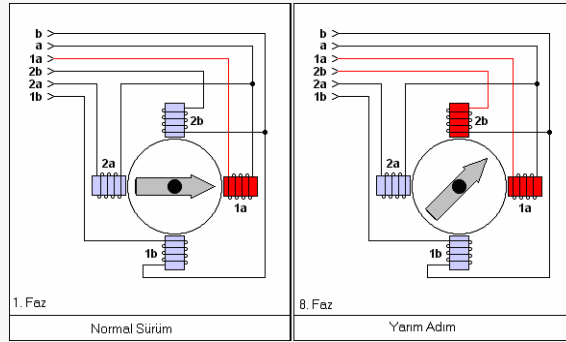
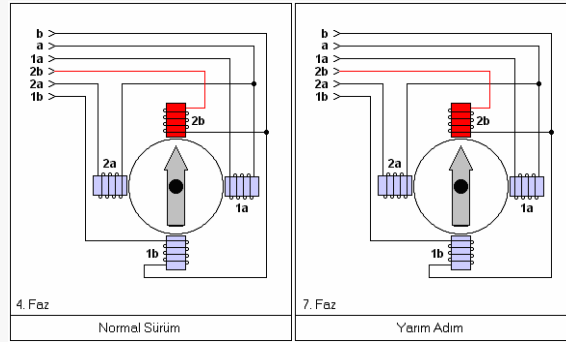
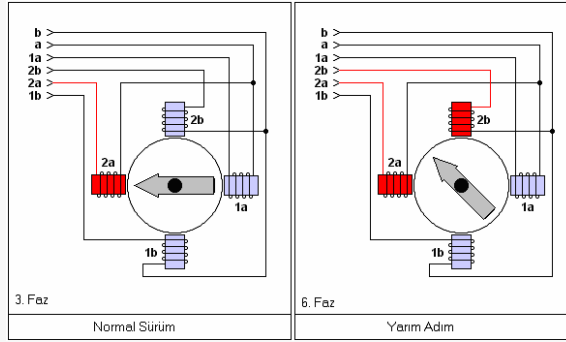
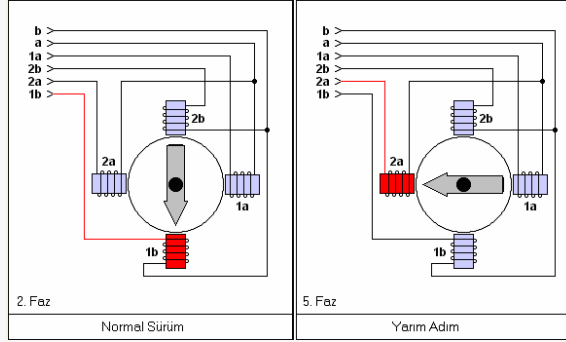
| Adım | 1a | 1b | 2a | 2b | Onluk Sayı |
|------|----|----|----|----|------------|
| 1 | 1 | 0 | 0 | 0 | 8 |
| 2 | 1 | 1 | 0 | 0 | 12 |
| 3 | 0 | 1 | 0 | 0 | 4 |
| 4 | 0 | 1 | 1 | 0 | 6 |
| 5 | 0 | 0 | 1 | 0 | 2 |
| 6 | 0 | 0 | 1 | 1 | 3 |
| 7 | 0 | 0 | 0 | 1 | 1 |
| 8 | 1 | 0 | 0 | 1 | 9 |

Aşağıdaki şekillerde normal adım sürüm ve yarım adım sürümün karşılaştırılması görülmektedir.

GEREKLİ BİLGİ



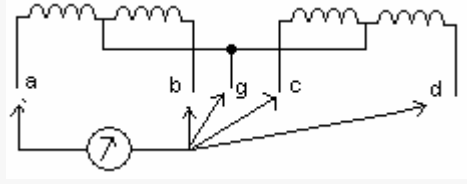
GEREKLİ BİLGİ



GEREKLİ BİLGİ

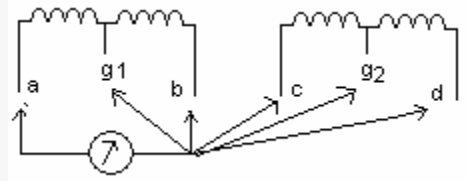
Motor şase ucunun tespiti : Ölçü aletinin probu bir uçta sabit iken diğer uçlar sırayla dokundurularak her defasında not alınır. 6 uçlu adım motorlarında eğer sonsuz direnç görülürse bunun anlamı dokundurulan ucun farklı diğer sargıya ait olduğu anlamını taşır. Büyük direnç ölçüldüğünde uçların sargının her iki ucuna ait olduğunu gösterir. Küçük direnç okunduğunda ise uçlardan birinin şase olduğuna kanaat getirilir. Daha sonra diğer uçlara da benzer işlem yapılarak sağlaması yapılır ve kesin kanaate varılır.

5 uçlu adım motorunda :



| | | |
|-----------|--------|--|
| a-b arası | 70 ohm | |
| a-g arası | 35 ohm | Bu ölçümde uçlardan birinin şase olduğu tespit edilir. |
| a-c arası | 70 ohm | |
| a-d arası | 70 ohm | |
| b-a arası | 70 ohm | |
| b-g arası | 35 ohm | Bu ölçümde g ucunun şase olduğu kesinleşir. |

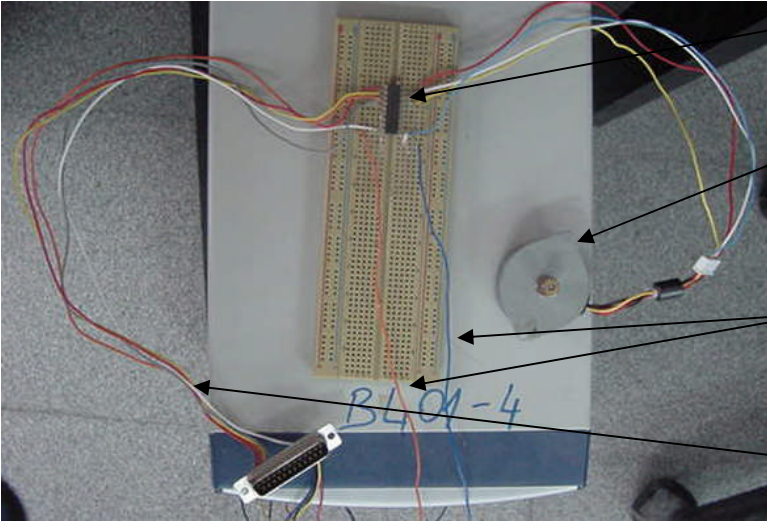
6 uçlu adım motorunda : Bu tip motorlarda sargılar ayrıdır ve iki adet şase vardır.



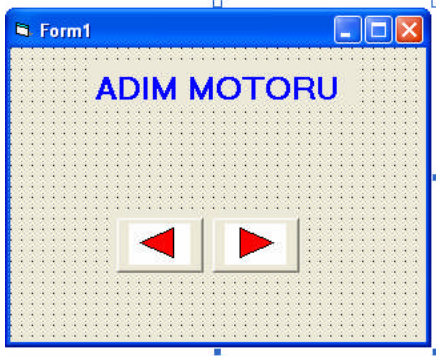
| | | |
|------------|--------|--|
| a-g1 arası | 35 ohm | Bu ölçümde uçlardan birinin şase olduğu tespit edilir. |
| a-b arası | 70 ohm | |
| a-c arası | Açık | |
| a-g2 arası | Açık | |
| a-d arası | Açık | |
| g1-a arası | 35 ohm | |
| g1-b arası | 35 ohm | Bu ölçümle g1 ucunun şase olduğu kesinleşir |

Bu aşamadan sonra diğer uçlarda kendi aralarında aynı işleme tabi tutularak g2 bulunur.

(Not: Yukarıdaki aşamalar örnektir. İşleminiz ölçmeye başladığınız uçlara göre yukarıdaki verilen aşamalardan daha fazla olabilir. Direnç değerleri de rastgele verilmiştir)

| İşlem Basamakları | Öneriler |
|--|---|
| <ul style="list-style-type: none"> ➤ Kullanacağınız adım motorunun adımlama sırasını tespit ediniz. | <ul style="list-style-type: none"> ➤ Öncelikle sargı direncini ölçerek motorun şase'sini bulunuz. Daha sonra kablolarına sırayla +5V vererek motorun aynı yönde dönmesini sağlayınız. Motorun dönmesini görmek için miline herhangi bir renkli bant parçası yapıştırabilirsiniz. |
| <ul style="list-style-type: none"> ➤ Bilgisayarın paralel portundan breadboard üzerine bağlantı yapmak için bir ucu paralel port konnektörüyle lehimlenmiş şekilde 50 – 100 cm arası bir kablo hazırlayınız. | |
| <ul style="list-style-type: none"> ➤ ULN2803 sürücü entegresinin kullanım kılavuzunu temin ederek detaylıca inceleyiniz. | <ul style="list-style-type: none"> ➤ Bu entegre temin edilmesi çok kolay ve yaygın bir elemandır. Kullanım katalogu ise internet ortamından bulunabilir. |
| <ul style="list-style-type: none"> ➤ Breadboard üzerine devreyi kurunuz. Bunun için aşağıdaki fotoğraf size yardımcı olabilir. | |
| <div style="display: flex; align-items: center;">  <div style="margin-left: 20px;"> <p>ULN2803 Entegresi</p> <p>Adım motoru</p> <p>Besleme uçları</p> <p>Konnektörlü Paralel port kablosu</p> </div> </div> | |
| <ul style="list-style-type: none"> ➤ Programınızın algoritmasını düşününüz ve çiziniz. | <ul style="list-style-type: none"> ➤ Tasarladığınız formunuzu öğretmeninize ve arkadaşlarınıza göstererek fikirlerini alabilirsiniz. Farklı algoritmalar üzerinde de düşünebilirsiniz. |

UYGULAMA FAALİYETİ

| İşlem Basamakları | Öneriler |
|--|---|
| <p>➤ Programınız için form tasarımı yapınız ve kullandığınız nesnelerin özelliklerini ayarlayınız.</p> | <p>➤ Form boyutlarınızı program çıktısını öngörerek ayarlayınız. Tasarladığınız formunuzu öğretmeninize göstererek fikrini alınız.. Örnek form :</p>  |
| <p>➤ Program kodlarınızı yazınız.</p> | <p>➤ Kod yazım kurallarını hatırlayınız.</p> |
| <p>➤ Programınızı çalıştırarak deneyiniz.</p> | |

ÖLÇME VE DEĞERLENDİRME

Aşağıdaki soruları cevaplayarak bu faaliyette kazandığınız bilgileri ölçünüz.

OBJEKTİF TESTLER (ÖLÇME SORULARI)

1. Visual Basic programlama dilinde paralel port uygulamalarını yapabilmek için hangi dosyaya ihtiyaç vardır?

- A) parallel.ocx B) inpout32.dll C) port32.dll D) inpout.dll

2. Aşağıdaki komutlardan hangisi doğrudur?

- A) Out val(378) , veri
B) Outport(H256), 255
C) Out Val (&H379), 255
D) Out (H378), veri

3. Visual Basic'te paralel portu kullanmak için gerekli .dll dosyası hangi bölümde koda eklenir?

- A) Form B) Özellikler C) Modül D) General

4. Paralel portun çıkış iğnelerinden alınan gerilim kaç volt'tur?

- A) +5V B) -12V C) +12V D) -5 V

5. Aşağıdaki programın çıktısı nedir?

```
Sub Command1_click()  
    Veri=&HAA  
    Out Val (H378) , veri  
End Sub
```

- A) 10101011 B) 01010101 C) 10100101 D) 10101010

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

Paralel port donanımı yoluyla sayısal giriş değerlerini kontrol edebileceksiniz.

ARAŞTIRMA

- Farklı programlama dillerinde paralel port kullanarak dış ortamdaki verilerin alımı nasıl yapılmaktadır?

3. PARALEL PORTTAN GİRİŞ ALMAK

Paralel portu kullanarak dışarıya veri gönderebildiğimiz gibi, dış ortamdaki herhangi bir veriyi de paralel port üzerinden alabiliriz. Durum ve denetim yazmaçları üzerinden bitler halinde veri okumak mümkündür. Harici ortamdan almak istediğimiz veriler bit şeklinde olabileceği gibi 1 bayt'lık düzende de olabilir. Eğer 1 bayt'lık bir veri okunmasına ihtiyaç duyuluyorsa üç yolla alabiliriz.

- Durum portundan ardı ardına 4 bit okuyup bunları birleştirerek
- Durum portundan 4 bit, denetim portundan 4 bit okuyarak birleştirmek
- Çift yönlü , EPP ve ECP port modlarını kullanarak veri hattından okuyarak.

3.1. Durum Yazmacından Giriş Almak

Durum yazmacının esas işlevi yazıcının bilgisayara kendi durumu ile ilgili bilgileri aktarmasına aracılık etmektir. Yazıcının “kağıt yok” bilgisini göndermesi yada o anda başka bir yazdırma görevi yapıyor ise “meşgul” bilgisini bilgisayara bildirmesi bunlara örnek olarak verilebilir. Aşağıdaki tablo durum yazmacıyla ilgili bir fikir vermektedir.

| Temsil | İsim (görev) | İğne no | Okunan Değer | İkilik karşılığı |
|----------|--------------------|---------|--------------|------------------|
| S7 | Busy (Meşgul) | 11 | 255 | 11111111 |
| S6 | Ack (Hazır) | 10 | 63 | 00111111 |
| S5 | NoPaper(Kağıt yok) | 12 | 95 | 01011111 |
| S4 | Selected (Seçili) | 13 | 111 | 01101111 |
| S3 | Error (Hata) | 15 | 119 | 01110111 |
| S2-S1-S0 | Tanımsız | - | - | - |

Şekil 3.1: Durum yazmacı

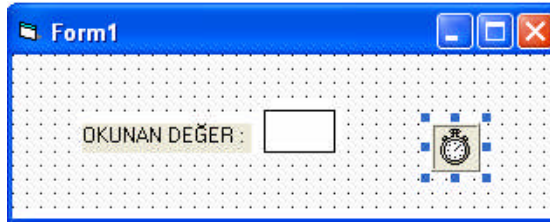
Hiçbir giriş yapılmadığı takdirde okunan değer 127 (01111111)'dir. Çünkü deney kartı üzerinde tüm girişler (girişler) 4,7K değerinde dirençlerle 5V gerilime bağlanmıştır. "Meşgul" biti (11.nolu uç) terslendiği için bu bit 0 olarak okunur. Paralel port bağlacında temsil edilmeyen ilk 3 bit donanım olarak 1 görülür. Dolayısıyla girişe bağlanan 5 butona sırayla basıldığında yukarıdaki tabloda verilen değerler okunacaktır.

Not: Bazı bilgisayarlarda okunan değerler farklılık gösterebilir. Örneğin 119 yerine 118 okunabilir. Bunu nedeni tanımsız olan ilk 3 bitin durumudur. EPP modunda ilk bit zaman aşımı biti olarak atanır ve 0 değerini alır. Buna göre değerler farklı çıkar.

Örnek 3.1:

Deney kartımıza bulunan ve durum portu uçlarına bağlı olan butonlara bastığımızda ekranda bu giriş değeri gösteren bir program yazalım.

Port üzerinden okuma işlemi yaptırılırken sürekli olarak yazmaç iğnelerinin dinlenmesi gerekir. Bunun için bir döngü kurabilirsiniz. Ancak en sağlıklı yöntem "Timer" nesnesinin kullanılmasıdır.



Şekil 3.2: Örnek form

Formun üzerine timer nesnesi yerleştirdikten sonra interval özelliğini 100 olarak belirleyelim. Bunun anlamı her 100 milisaniye aralıklarla timer nesnesi çalışacaktır.

Tüm örneklerimizde olduğu gibi modülümü aynı şekilde projemize ekledikten sonra kod yazımına geçelim. Kodlarımızı timer nesnesine alt yordam olarak ekleyelim. Böylelikle her 100 milisaniyede bir kodlarımız çalışacak ve durum yazmacındaki veriler alınacaktır.

Paralel porttan bilgi okumak için "inpout32.dll" kütüphanesinin "INP ()" komutu kullanılır. Okunan değer herhangi bir değişkene aktarılarak program içerisinde kullanılabilir. Bu komutun kullanım formatı aşağıdaki gibidir.

Değişken adı=INP (&adres)

Bu örneğimizde durum portundan okuma işlemi yaptığımızdan dolayı durum yazmacının adresini INP komutu içerisinde yazacağız..

```

Private Sub Timer1_Timer()
oku = Inp(&H379) ' 379H durum yazmacı adresinden okuma işlemi yap ve oku değişkenine aktar.
Text1.Text = oku 'okunan değeri metin kutusuna yazdır.
End Sub

```

Yukarıdaki kod satırlarını yazdıktan sonra programımızı çalıştıralım ve deney kartı üzerindeki butonlara sırayla basarak aşağıdaki tabloyu doldurunuz.

| Temsil | İğne no | Okunan Değer | İkilik karşılığı |
|--------|---------|--------------|------------------|
| S7 | 11 | | |
| S6 | 10 | | |
| S5 | 12 | | |
| S4 | 13 | | |
| S3 | 15 | | |

Örnek 3.2:

Bir önceki örneğimize buton resimleri ve basma efektini de ekranda gösterecek şekilde görsel olarak zenginleştirelim.

Bunun için öncelikle Paint benzeri bir çizim programı açarak aşağıdaki gibi buton resimlerini çizelim. Çizdiğimiz bu resimleri proje klasörümüzün içerisine kaydedelim.



butonbırak.bmp



butonbas.bmp

Boş formumuzun üzerine aşağıdaki şekilde gösterilen nesneleri yerleştirelim. Buton efekti vermek için 5 tane image nesnesini yerleştirelim. Bunun için ilk image nesnesini yerleştirdikten sonra kopyala – yapıştır yöntemi ile dizi olarak çoğaltalım.

Şekil 3.3: Örnek form tasarımı

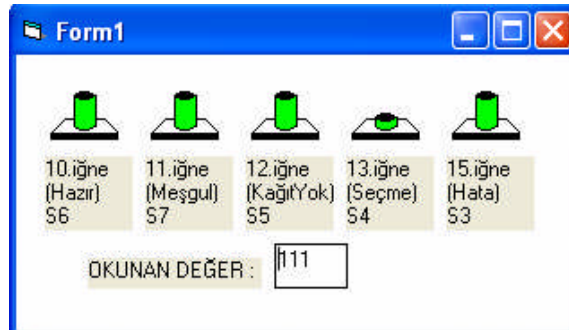
Form tasarımında bu defa zemini beyaz yapalım. Bunun için formun "backcolor" özelliğini beyaz renk seçelim. Bu durumda yerleştirdiğimiz metin kutusunun da daha görünür olması bakımından "Appearance" özelliğini "0-flat" yapalım.

Son olarak form üzerine yerleştirdiğimiz etiketlerin "Caption" özelliklerini kullanarak şekilde görünen ifadeleri yazarak tasarımımızı sonlandıralım.

Aşağıdaki görülen kodların çoğu görsel zenginlik katmak için eklediğimiz butona basılma ve bırakma efektlerini içermektedir. Temel okuma işlemini içerek satırlar oldukça basittir.

```
Private Sub Timer1_Timer()  
oku = Inp(&H379) ' 379H durum yazmacı adresinden okuma işlemi yap ve oku değişkenine aktar.  
If oku = "255" Then Image1(1).Picture = LoadPicture(App.Path & "\butonbas.bmp")  
If oku = "63" Then Image1(0).Picture = LoadPicture(App.Path & "\butonbas.bmp")  
If oku = "95" Then Image1(2).Picture = LoadPicture(App.Path & "\butonbas.bmp")  
If oku = "111" Then Image1(3).Picture = LoadPicture(App.Path & "\butonbas.bmp")  
If oku = "119" Then Image1(4).Picture = LoadPicture(App.Path & "\butonbas.bmp")  
If oku = 127 Then 'Eğer hiçbir butona basılmamış ise butonlara basılı değil efekti ver.  
Image1(0).Picture = LoadPicture(App.Path & "\butonbirak.bmp")  
Image1(1).Picture = LoadPicture(App.Path & "\butonbirak.bmp")  
Image1(2).Picture = LoadPicture(App.Path & "\butonbirak.bmp")  
Image1(3).Picture = LoadPicture(App.Path & "\butonbirak.bmp")  
Image1(4).Picture = LoadPicture(App.Path & "\butonbirak.bmp")  
End If  
Text1.Text = oku 'okunan değeri metin kutusuna yazdır.  
End Sub
```

Programımızı çalıştırdığımızda ve örneğin 13 nolu iğnenin bağlı olduğu S4 butonuna bastığımızda aşağıdaki gibi bir form çıktısı elde ederiz.



Şekil 3.4: Programın çalıştırılması

Dikkat edilecek olursa S4 nolu butona basıldığında, durum yazmacından 111 değerinin okunmakta ve aynı zamanda butona basılma efektinin de ekran üzerinde görülmektedir.

3.1. Veri Yazmacından Giriş Almak

Veri yazmacı normalde yazıcıya veri gönderilmesi için çıkış amaçlı olarak kullanılır. Bazı durumlarda dış ortamdan 8 bitlik veri alınması gerekebilir. Eski tip paralel portlar bu gibi durumlar için veri portundan veri girişine imkan tanımaz iken sonradan geliştirilen ve günümüz ana kartlarında kullanılan paralel portlar ile bu işlem gerçekleştirilebilir.

Günümüz ana kartlarındaki paralel portlarda 1.öğrenme faaliyetinde “1.2.Paralel Port Modları” başlığı altında bahsedilen “çift yönlü (bi-directional)” adı verilen çalışma modu bulunmaktadır. Bilgisayarımızın paralel portu genellikle fabrika ayarları olarak “normal” mod olarak ayarlanmıştır. Veri yazmacından giriş almak için öncelikle yapılması gereken işlem bilgisayarın BIOS programından paralel port modunun çift yönlü (bi-directional) duruma getirilmesi gerekmektedir. Bazı BIOS programlarında çift yön (bi-directional) modu ayrı bir seçenek halinde sunulmamıştır. Bu durumda EPP yada ECP modlarından birisi seçilmelidir.

Aşağıdaki şekilde örnek bir BIOS ekranında paralel port ayarları görülmektedir.



Şekil 3.5: Bios ekranı

Çift yönlü portlarda denetim yazmacının 5.biti (C4 – enable) yön biti olarak kullanılmaktadır. Hatırlayacağınız gibi denetim yazmacının ilk 4 biti konnektör pinlerine bağlı olarak kullanılmaktadır. C4 bitinin konnektör üzerinde fiziki bir karşılığı bulunmamaktadır. Dolayısıyla bu bitin kullanımı program üzerinden olmaktadır. C4 biti mantık 1 yapılırsa veri yazmacı giriş , mantık 0 yapılırsa çıkış olarak ayarlanır.

Örnek 3.3:

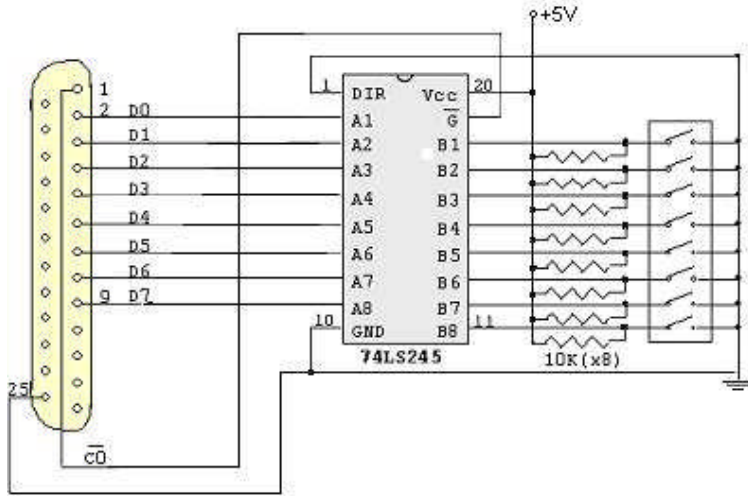
Bu örneğimizde paralel portumuzun veri yazmacına bağladığımız bir buton yardımı ile giriş alalım ve değeri formumuzda görelim.

Bu örneği gerçekleştirmek için öncelikle aşağıdaki devreyi breadboard üzerinde kuralım. Devrede kullanılan 74LS245 entegresi üç durumlu iki yönlü sekizli (octal) tampon olarak görev yapmaktadır. 74LS245 entegresi aşağıdaki doğruluk tablosuna göre işlem yapmaktadır.

| GİRİŞ | | ÇIKIŞ |
|-------|-----|--------------------|
| G | DIR | |
| L | L | B'den A'ya aktarım |
| L | H | A'dan B'ye aktarım |
| H | X | Yalıtımda |

Şekil 3.6: 74LS245 doğruluk tablosu

Doğruluk tablosundan da anlaşılacağı üzere 74LS245 entegresinin B 'den A'ya aktarım yapabilmesi için G ve DIR iğnelerinin her ikisinin de L (low-düşük) yapılması gerekmektedir. DIR (direction) ucu toprak hattına verilerek L (low-düşük) yapılır. G ise paralel port konnektörün de 1 nolu iğneye bağlanmıştır. 1 nolu iğne denetim yazmacında C0 bitinin karşılığıdır ve terslenmiştir.



Şekil 3.7: Buton devresi

Bu işlemden sonra bilgisayarımızın BIOS' una girip paralel portumuzu "çift yönlü (bi-directional)" moda ayarlayalım.

Daha sonra aşağıdaki gibi basit bir form tasarlayarak program yazımına geçebiliriz.



Şekil 3.8: Örnek form tasarımı

| Nesne | Name | Caption |
|--------------|----------|-------------------------------|
| Etiket | Label1 | Veri yazmacından okunan değer |
| Metin Kutusu | Text1 | |
| Buton | Command1 | OKU |

Programımızda 74LS245 entegresinin G ucunu L (low) yapmak için C0 bitini , veri yazmacını giriş olarak ayarlamak için ise C4 bitini mantık 1 yapmalıyız.

| C0 | C1 | C2 | C3 | C4 | C5 | C6 | C7 | |
|----|----|----|----|----|----|----|----|-----|
| 1 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | =17 |

Şekil 3.9: Durum yazmacının bitlerinin aldıkları değerler

Bunun için elde ettiğimiz 17 bilgisini form Load (yükle) kısmında denetim yazmacına gönderelim. Form Load kısmına yazarak programımız çalıştırıldığında bu ayarın yapılmasını sağlamaktır.

```
Private Sub Form_Load()
    Out Val("&H37A"), Val(17) 'denetim yazmacının 5.biti mantık 1 yapılır.
End Sub
Private Sub Command1_Click()
    oku = Inp(&H378) 'veri yazmacı okunur
    Text1.Text = oku 'okunan değer metin kutusuna yazdırılır
End Sub
```

Butonumuzun kod sayfasına ise veri yazmacından değer okuyarak metin kutusuna yazılmasını sağlayan kod satırlarımızı ekleyerek kod yazımını sonlandıralım.

Son olarak programı çalıştıralım ve breadboard üzerinden buton (yada butonlarımıza) basarak metin kutusunda okunan değer doğruluğunu kontrol edelim.

UYGULAMA FAALİYETİ

Aşağıdaki işlem basamaklarına göre uygulama faaliyetini yapınız.

Aşağıdaki form üzerinde bir kimyasal sıvının endüstriyel ortamda işlenmesi görülmektedir. Bu işlem şu şekilde çalışmaktadır.

- Dolum vanası butonuna basılarak işlem başlatılır ve tanka kimyasal sıvı girişi yapılır.
- Tank dolduğunda vana otomatik olarak kapanır ve ısıtıcı çalışır.
- Sıcaklık sensöründen kaynama bilgisi geldiğinde ısıtıcı kapanır.
- Boşaltım vanası otomatik olarak açılarak tank içindeki kimyasal sıvı boşaltılır.

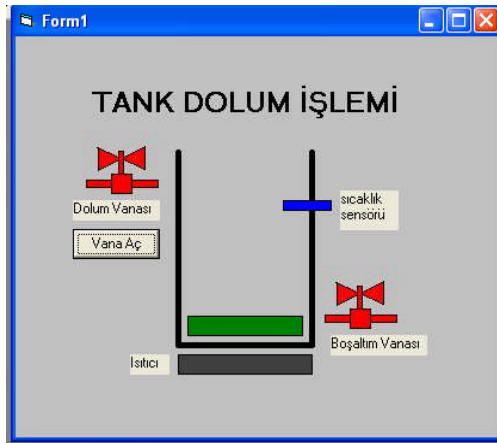
Bu uygulama için deney kartından yararlanılacaktır. Buna göre;

Sıcaklık sensörü: Deney kartındaki durum portu butonlarından birisi seçilecektir. Sıcaklık sensörünün göndereceği kaynama bilgisi temsili olarak butona basılarak verilecektir.


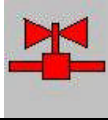
Dolum vanası: Deney kartındaki 1. veri yazmacı ledi seçilecektir. Vana form üzerinden açıldığında yeşile dönecek ve led1 yakılarak temsili olarak vananın açıldığını gösterilecektir. Tank dolduğunda led1 sönecek ve vana tekrar kırmızıya dönecektir.

Boşaltım vanası: Deney kartındaki 2. veri yazmacı ledi seçilecektir. Vana form üzerinden açıldığında yeşile dönecek ve led2 yakılarak temsili olarak vananın açıldığını gösterilecektir. Tank dolduğunda led2 sönecek ve vana tekrar kırmızıya dönecektir.

Isıtıcı: Deney kartındaki 3. veri yazmacı ledi seçilecektir. Isıtıcı sıcaklık sensöründen kaynama bilgisi alınarak (butona basıldığında) çalıştığında kırmızıya dönecek ve led3 yakılarak temsili olarak ısıtıcının çalıştığı gösterilecektir. Isıtıcı durduğunda led3 sönecek ve ısıtıcı tekrar siyaha dönecektir.



Şekil 3.10. Örnek form tasarımı

| İşlem Basamakları | Öneriler |
|---|--|
| <p>➤ Öncelikle form tasarımında kullanacağınız vana şekillerini (kırmızı ve yeşil olarak) Paint programında çizerek kaydediniz.</p> <div style="display: flex; justify-content: center; gap: 20px;">   </div> | <p>➤ Vanaların zemin rengi ile tasarlayacağınız formunuzun zemin rengini aynı yapınız. Böylece form görünümümüz de estetiği yakalayabilirsiniz.</p> |
| <p>➤ Programınız için form tasarımı yapınız.</p> | <p>➤ Şekiller için image nesnesinden faydalanınız. Tank içindeki sıvının temsili için dikdörtgen kullanınız..</p> <p>➤ Tank içindeki sıvı ilk çalıştırıldığında minimum seviyede olacağı için dikdörtgeni çizerken yüksekliğini en düşük seviyede tutunuz..</p> <p>➤ Tasarladığınız formunuzu öğretmeninize göstererek fikrini alınız.</p> |
| <p>➤ Form üzerindeki nesne özellikleri ayarlayınız.</p> | <p>➤ Image nesneleri için “backstyle” özelliğinin “opak” yapılacağını unutmayınız.. Renklendirmeleri yapınız.</p> |
| <p>➤ Programın algoritmasını geliştiriniz.</p> | <p>➤ Algoritmanızı uygulama faaliyetinin başında verilen işlem sırasına göre alarak geliştiriniz.</p> <p>➤ Tank içindeki sıvıya dolun ve boşaltım efekti vermek için sıvıyı temsil eden image nesnesinin “Top” ve “Height” özelliklerini kod içerisinde bir döngü ile değiştirerek yapabilirsiniz.</p> <p>➤ Vanaların renk değişimleri için 3.öğrenme faaliyetindeki 3.1.konusunda verilen örnekteki kodları uyarlayabilirsiniz.</p> |
| <p>➤ Program kodlarınızı yazınız.</p> | <p>➤ Bir modül açarak Inpout32.dll dosyasının tanımlama kodlarını yazmayı unutmayınız.</p> <p>➤ Kod yazım kurallarını hatırlayınız</p> |
| <p>➤ Programınızı çalıştırarak deneyiniz.</p> | |

ÖLÇME VE DEĞERLENDİRME

Aşağıdaki soruları cevaplayarak bu faaliyette kazandığınız bilgileri ölçünüz.

OBJEKTİF TESTLER (ÖLÇME SORULARI)

1. Veri girişi yapmak için hangi komut kullanılır?

- A)Inpout() B)Inp() C)DataInput () D)In()

2. Aşağıdaki ifadelerden hangisi yanlıştır?

- A) Visual Basic'te paralel port kullanmak için inpout32.dll dosyası gereklidir.
B) Veri girişi yaparken tampon (buffer) entegresi kullanmak portu korur.
C) Normal modda veri yazmacından giriş yapılamaz.
D) Visual Basic'te paralel port kontrolünde denetim yazmacından giriş yapılamaz.

3. Durum yazmacının hangi biti terslenmiştir?

- A) 0 B)2 C)5 D)7

4. Aşağıdaki seçeneklerde verilen satırlardan hangisi doğrudur?

- A) Z =Inp(oku, &378)
B) Oku ::Inp (&H38A)
C) Oku= Inp (&H378)
D) Oku= In(378), 8

5. Aşağıdakilerden hangisi paralel port modlarından birisi değildir?

- A)Standart B)EPP C)EPD D)ECP

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

PERFORMANS TESTİ (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 | HAYIR |
| Programlarınızı yazmadan önce algoritma çıkardınız mı? | | |
| Aynı uygulamayı farklı algoritmalar geliştirerek yapmayı denediniz mi? | | |
| Tasarladığınız form programınızı kullanacak kişi açısından kullanışlı oldu mu? | | |
| Yazdığınız kodlarda gereksiz satırlardan kaçındınız mı? | | |
| Programlarınız doğru ve hatasız çalıştı mı? | | |

DEĞERLENDİRME

Yukarıdaki değerlendirme sorularında hayır cevaplarınız var ise ilgili uygulama faaliyetini tekrar ediniz. Cevaplarınızın tümü evet ise bir sonraki modüle geçebilirsiniz.

Modülü tamamladıktan sonra öğretmeninizin danışmanlığında bir sonraki modüle geçebilirsiniz. Bu modül konularının bundan sonraki modüllerde verilen uygulamalarda kullanacağınızı göz önüne alarak kazandığınız bilgi ve becerilerinizi sürekli geliştirmeniz iyi bir programcı olmanızı sağlayacaktır.

CEVAP ANAHTARLARI

ÖĞRENME FAALİYETİ-1'İN CEVAP ANAHTARI

| Soru | Cevap |
|------|-------|
| 1. | C |
| 2. | C |
| 3. | D |
| 4. | B |
| 5. | A |
| 6. | C |

ÖĞRENME FAALİYETİ-2'NİN CEVAP ANAHTARI

| Soru | Cevap |
|------|-------|
| 1. | B |
| 2. | C |
| 3. | C |
| 4. | A |
| 5. | D |

ÖĞRENME FAALİYETİ-3'ÜN CEVAP ANAHTARI

| Soru | Cevap |
|------|-------|
| 1. | B |
| 2. | D |
| 3. | A |
| 4. | C |
| 5. | C |

KAYNAKÇA

- AXELSON Jan, **Her Yönüyle Paralel Port**, Bileşim Yayıncılık & Era Bilgi Sistemleri ve Yayıncılık, Mart - 2000
- ÖZDEVECİ Murat, **Bilgisayar ile Kumanda Teknikleri**, M.E.B. Hizmet İçi Eğitim Seminer Notları, İzmir - 2005.