



Kendimiz Yapalım

Yavuz Erol*

16 Sütunlu Kayan Yazı

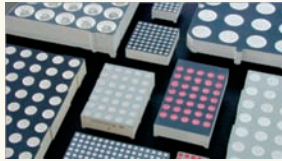
Bu yazıda 8 satır, 16 sütundan oluşan LED'li kayan yazı projesi anlatılıyor. Projenin en önemli özelliği gerek donanım gerekse yazılım olarak basit olması. Çok az sayıda donanım bileşeni gerektirdiğinden elektronik devrenin yapımı oldukça kolay. 8 satır, 16 sütundan oluşan gösterge için 128 adet LED kullanılabileceği gibi hazır modüller de kullanılabilir. Montaj kolaylığı sağladığı için bu projede 8x8 dot matrix display (nokta matris gösterge) tercih edildi. Her bir matris display içerisinde 64 adet LED bulunduğundan 2 adet display kullanılarak 8 satır 16 sütunlu gösterge kolayca oluşturulabilir.

Kayan yazı projesi için gerekli malzemeler şunlar:

Malzeme Listesi	
PIC16F628A	1 adet
8x8 dot matris display (0. katotlu)	2 adet
74HC154 entegre	1 adet
330nF kondansatör	2 adet
4.7kΩ direnç (0.25W)	1 adet
560Ω direnç (0.25W)	8 adet
5V'luk güç kaynağı	1 adet

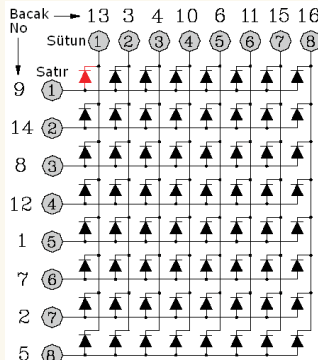
Matris display

Elektronik sektöründe matris display kullanımı oldukça yaygın. Bu tür göstergelere genellikle reklam panolarında, mağaza vitrinlerinde ve asansör kabinlerinde rastlanıyor (Şekil 1). Montajı kolay, birim maliyeti düşük olduğu için bilgilendirme amaçlı görsel uygulamalarda tercih ediliyor.



Şekil 1: Matris display çeşitleri

Piyasada satır ve sütun sayısı farklı pek çok matris display türü bulunmaktadır. Gerçekleştirilen uygulamaya göre istenen boyutta display seçmek de mümkündür. Bu projede kullanılan 8x8 ortak katotlu matris displayin iç yapısı Şekil 2'de görülmekte.



Şekil 2: Matris displayin iç yapısı

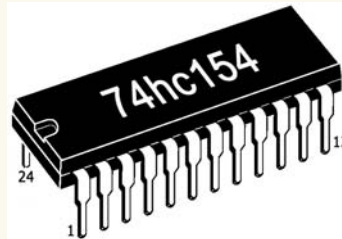
Çizimden görüldüğü gibi displayin 16 adet bacağı mevcut. Bunlardan 8 tanesi satır ucu, 8 tanesi ise sütun ucu. Matris display üzerindeki herhangi bir LED'i yakmanın yolu, uygun satır ve sütun uçlarından akım geçirmek. Örneğin, sol üst köşedeki LED'in ışık yayabilmesi için 9 nolu bacadan 13 nolu bacağı doğru 5-20mA seviyesinde bir akımın geçmesi gerekiyor. Bunu sağlayabilmek için bu bacaklara uygulanan gerilimin, LED'in ileri yön geriliminden daha büyük olması gerekli. Aynı zamanda LED akımını sınırlandırmak için seri bir direnç de kullanılmalı. Display türü ortak katotlu olduğu için, her bir sütundaki LED'lerin katot uçları birbirine bağlı durumda. Böylece, hangi sütun lojik 0'a bağlı ise, o sütundaki 8 LED'in biri ya da tamamı ışık yayabilir. LED'lerin kaç tanesinin ışık yayacağı satır uçlarının gerilim seviyesine göre belirlenir. Örneğin, bütün satır uçları birer direnç üzerinden +5V'a bağlanırsa, seçili olan sütundaki LED'lerin tamamı ışık yayar.

Çalışma mantığı

Kayan yazı devresi, tarama (scanning) yöntemine göre çalışır. Bu yöntemde, göstergedeki LED'lerin tamamı aynı anda ışık yaymaz. Bunun yerine, sadece seçili olan sütundaki LED'ler ışık yayar. Sütun seçme işlemi yeterli kadar yüksek hızda yapıldığında göz, tarama olayını fark edemez. Örneğin, her bir sütundaki LED'ler 1ms süreyle ışık yayacak olursa, 16 sütundan oluşan göstergenin tarama süresi 16ms olur. Bu da saniyede 62 tekrar yapıldığı anlamına gelir. Tekrarlama frekansı gözün algılaya sınırının üzerinde olduğu için sanki bütün LED'ler aynı anda yanıyormuş izlenimi verir. Oysa, her bir sütundaki LED'ler 1ms süresince yanık, 15ms süresince sönmük halde kalır.

Sütun sürücü

16 adet sütunu sırasıyla seçmek için en uygun yöntem 16 çıkışa sahip 74HC154 entegresini kullanmak. 4'den 16'ya kod çözücü olarak çalışan bu entegre, Şekil 3'den görüldüğü gibi 24 bacaklı.



Şekil 3: Sütun sürücü entegresi

Entegrenin 4 adet seçme ucu ve 16 adet çıkış ucu bulunmaktadır. 18 ve 19 nolu bacaklar lojik 0 seviyesinde tutulduğunda, ABCD girişlerinin ikili (binary) durumuna göre, istenen çıkış ucunu aktif etmek mümkün. Tablodan görüldüğü gibi ABCD girişlerinin lojik durumuna göre, 16 adet çıkıştan sadece bir tanesi lojik 0 seviyesinde. Burada, H lojik 1; L lo-

jik 0 anlamında. Aktif olan çıkış ucunun lojik 0 seviyesinde olması, ortak katotlu display sürmede kolaylık sağlar.

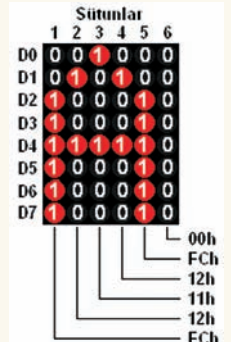
GİRİŞLER	ÇIKIŞLAR															
A B C D	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
0000	L	H	H	H	H	H	H	H	H	H	H	H	H	H	H	H
0001	H	L	H	H	H	H	H	H	H	H	H	H	H	H	H	H
0010	H	H	L	H	H	H	H	H	H	H	H	H	H	H	H	H
0011	H	H	H	L	H	H	H	H	H	H	H	H	H	H	H	H
0100	H	H	H	H	L	H	H	H	H	H	H	H	H	H	H	H
0101	H	H	H	H	H	L	H	H	H	H	H	H	H	H	H	H
0110	H	H	H	H	H	H	L	H	H	H	H	H	H	H	H	H
0111	H	H	H	H	H	H	H	L	H	H	H	H	H	H	H	H
1000	H	H	H	H	H	H	H	H	L	H	H	H	H	H	H	H
1001	H	H	H	H	H	H	H	H	H	L	H	H	H	H	H	H
1010	H	H	H	H	H	H	H	H	H	H	L	H	H	H	H	H
1011	H	H	H	H	H	H	H	H	H	H	H	L	H	H	H	H
1100	H	H	H	H	H	H	H	H	H	H	H	H	L	H	H	H
1101	H	H	H	H	H	H	H	H	H	H	H	H	H	L	H	H
1110	H	H	H	H	H	H	H	H	H	H	H	H	H	H	L	H
1111	H	H	H	H	H	H	H	H	H	H	H	H	H	H	H	L

Tablo: 74HC154'ün çalışma şekli

Kayan yazı devresinde sütun seçme ve ilgili sütuna ait satır verilerini gönderme işlemi için bir mikro denetleyici gerekli. Piyasada çok çeşitli türde mikro denetleyiciler bulmak mümkün. Dahili osilatör kullanma fırsatı verdiği için bu projede PIC16F628A tercih edildi. Böylece, kristal ve kondansatör bağlantısı olmadan mikro denetleyici 4MHz frekansta çalıştırılabilir. 16F628'in fiyatının 16F84'e göre daha düşük olması ve program hafızasının daha geniş olması da diğer tercih nedenleri arasında.

Karakter oluşturma

Matris display üzerinde harf, rakam ve diğer karakterleri görüntüleyebilmek için öncelikle bir karakter tablosu oluşturmak gerekir. Bu tabloda, karakterin kaç satırdan ve kaç sütundan oluşacağı, hangi LED'lerin ışık yayacağı belirlenir. Şekil 4'de A harfi için karakter oluşturma mantığı görülmüyor. Kırmızı renkli noktalara 1, diğerlerine 0 yazılarak karaktere ait satır verileri elde edilir. Örneğin, A harfi 6 sütun, 8 satırdan oluşur ve matris displayde bu karakteri görüntüleyebilmek için sırasıyla Fch, 12h, 11h, 12h, FCh, 00h verilerinin satırlara iletilmesi gerekir. 16'lık tabanda (hexadesimal) yazılmış bu verileri matris displayin satırlarına iletmenin en uygun yolu bir mikro denetleyici kullanmak. Mikro denetleyicinin çıkış portuna ait pinler, birer direnç üzerinden matris displayin satır uçlarına bağlanırsa, karakter tablosundaki hexadesimal değerlere göre LED'leri yakmak mümkün olur.



Şekil 4: A harfine ait satır verileri

Kendimiz Yapalım

Devre şeması

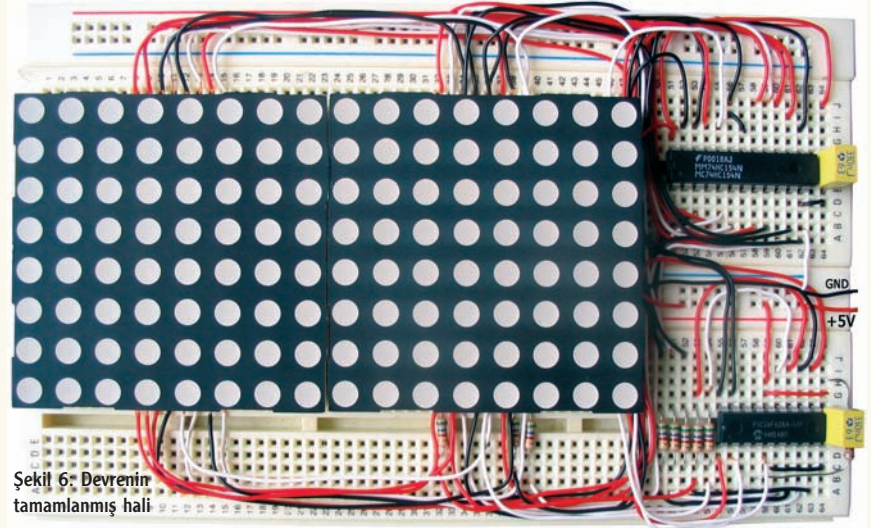
Şekil 5'de kayan yazı devresi görülüyor. Devre şemasından görüldüğü gibi sütun seçme işlemi 74HC154 entegresi yürütüyor. Entegrenin ABCD seçme girişleri mikro denetleyicinin PORTA uçlarına bağlı. Satır verileri ise PORTB uçlarından gönderiliyor. LED akımını sınırlandırmak için 560 ohm'lık dirençler bulunmakta. Devrenin doğru olarak çalışabilmesi için matris displaylerin aynı numaralı satır uçları birbirine bağlanmalı. Yani, her bir displayin 9, 14, 8, 12, 1, 7, 2 ve 5 nolu uçları birbirine bağlı olmalı. Katalog verilerine göre, 74HC154'ün çıkışı lojik 0 seviyesinde iken toprağa akabilecek akım 25mA civarında. Pratikte, akım değeri 35mA seviyesine kadar çıkabiliyor. Bu akım, toplam LED akımına eşit olduğu için tasarımda dikkate alınması gerekiyor. Matris displayin bir sütunundaki toplam LED akımının 35mA olması, her bir LED'den yaklaşık 5mA akım geçmesi anlamına geliyor. Bu tasarım şekli ile LED akımını daha yüksek tutmak mümkün değil. Fakat, pnp transistörler kullanarak sütun akımı, dolayısıyla LED akımı kolayca artırılabilir. Montaj kolaylığı sağladığı için bu projede sadece 74HC154 entegresi kullanıldı.

Şekil 6'da elektronik devrenin tamamlanmış hali görülüyor. Bu devre ile çeşitli uygulamalar yapılabilir. Matris display üzerinde sabit bir mesaj görüntülenebileceği gibi animasyon şeklinde hareketli görüntüler de oluşturulabilir. Uzun bir metni görüntülemenin yolu ise kayan yazı mantığını uygulamak. Bu mantığa göre, yazıya ait karakterler soldan sağa doğru belirli bir hızda geçiş yapar. Böylece, çok uzun bir metnin gösterilmesi mümkün olur.

Aşağıda PIC C programı ile yazılmış çeşitli uygulama örnekleri bulunuyor. İlk iki program, matris displayde sabit görüntü oluşturuyor. Son program ise kayan yazı uygulamasına ait.

Örnek 1

Şekil 7'deki yukarı-aşağı okları matris displayde oluşturmak için aşağıdaki C programını yazmak gerekiyor. Program yazımı için PICC Lite programı kullanılmalı. PICC Lite programında derleme esnasında PIC16F628A mikro denetleyicisi yerine PIC16F627 de seçilebilir. Derleme işleminin ayrıntıları piyasadaki PIC C ile ilgili kitaplardan öğrenilebilir. Derleme işlemi sonucunda elde edilen hex uzantılı dosya uy-



Şekil 6: Devrenin tamamlanmış hali

gun bir programlama kartı ve ICPROG yazılımı yardımıyla mikro denetleyiciye yüklenirse matris displayde istenen görüntü ortaya çıkar.



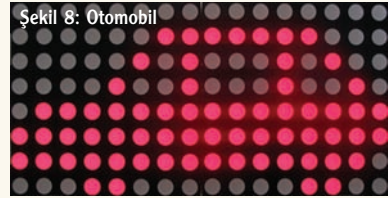
Şekil 7: Yukarı-aşağı oklar

```
#include <pic.h>
#include <delay.c>
main(void)
{
    // Değişken tanımlamaları
    unsigned int i;
    unsigned const char oklar[]={
        0x08,0x0c,0xfe,0xff,0xfe,0x0c,0x08,0x00,
        0x00,0x10,0x30,0x7f,0xfF,0x7f,0x30,0x10};
    // Port ayarlama işlemleri
    TRISB=0; // PortB'nin hepsi çıkış
    TRISA=0; // PortA'nın hepsi çıkış
    CMCON=0x07; // PORTA sayısal giriş/çıkış
    PORTB=0x00; // Başlangıçta LED'ler sönmük
    // 16 adet satır verisini sırayla PORT'a gönder
    for(;;){
        for(i=0;i<=15;i++){
            PORTB=oklar[i]; // Verileri PortB'ye gönder
            PORTA=i; // ilgili sütunu seç
            DelayMs(1); // 1ms bekle
        }
    } // Program sonu
```

Örnek 1'e ait C kodları

Örnek 2

Displayde otomobil resmi oluşturmak için örnek 1'deki programda küçük bir değişiklik yapmak gerekir. 16 elemanlı dizi, aşağıdaki gibi değiştirilirse matris displayde şekil 8'deki gibi otomobil resmi görülür.



Şekil 8: Otomobil

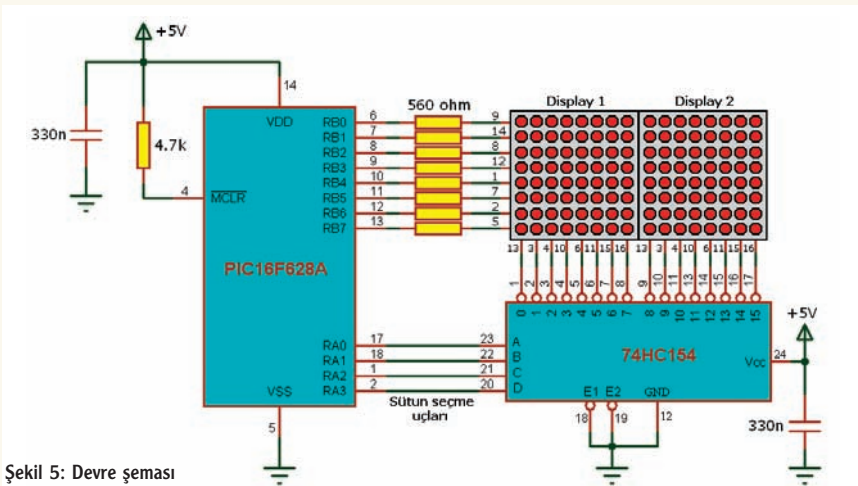
```
#include <pic.h>
#include <delay.c>
main(void)
{
    // Değişken tanımlamaları
    unsigned int i;
    unsigned const char araba[]={
        0x60,0x70,0x70,0xf0,0xf8,0x74,0x72,0x7e,
        0x72,0x72,0x72,0x7e,0xf2,0xf4,0x78,0x30};
    // Port ayarlama işlemleri
    TRISB=0; // PortB'nin hepsi çıkış
    TRISA=0; // PortA'nın hepsi çıkış
    CMCON=0x07; // PORTA sayısal giriş/çıkış
    PORTB=0x00; // Başlangıçta LED'ler sönmük
    // 16 adet satır verisini sırayla PORT'a gönder
    for(;;){
        for(i=0;i<=15;i++){
            PORTB=araba[i]; // Verileri PortB'ye gönder
            PORTA=i; // ilgili sütunu seç
            DelayMs(1); // 1ms bekle
        }
    } // Program sonu
```

Örnek 2'ye ait C kodu

Matris displayde hareketli animasyon oluşturmak için verileri belirli aralıklarla güncellemek gerekir. Örneğin, hareket eden otomobil görüntüsü elde etmek için, otomobilin anlık durumuna ait satır verileri, ayrı dizilere yüklenir ve sırayla dizi elemanları port'a iletilir. Böylece basit animasyonlar oluşturulabilir.

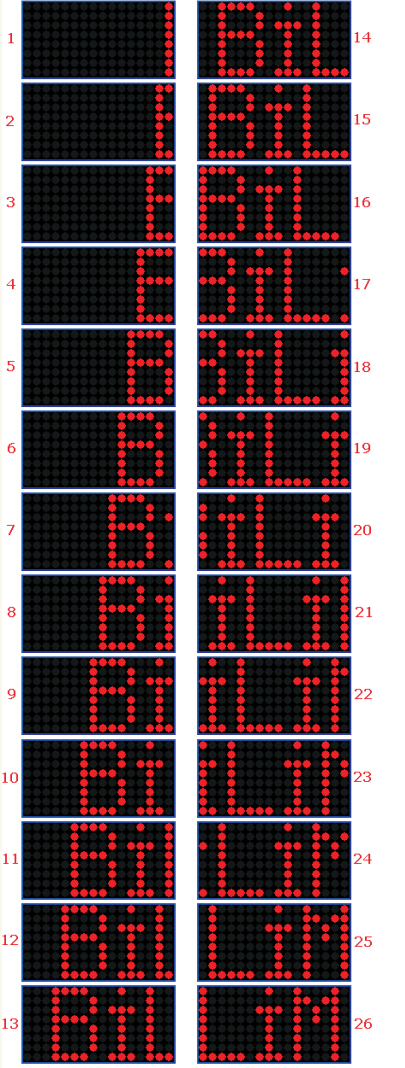
Örnek 3

Uzun bir metnin matris displayde gösterilmesi için şekil 9'daki gibi kayan yazı mantığı kullanılır. Şekillerden görüldüğü gibi her bir kaydırma işlemi sonrasında displaydeki görüntü bir kez sola ötelenir. Bu esnada en sağdaki sütundan metnin devamına ait veriler giriş yapar.



Şekil 5: Devre şeması

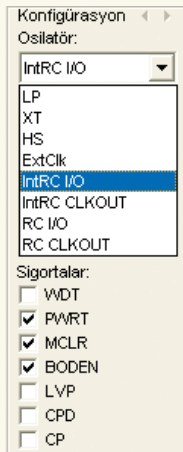
Kendimiz Yapalım



Şekil 9: Kayan yazı mantığı

Kayan yazı uygulamasına ait C programı aşağıda görülmüyor.

Harici osilatör devresi olmadan mikro denetleyiciyi çalıştırabilmek için ICPROG programında konfigürasyon ayarları şekil 10'daki gibi yapılmalı. Osilatör olarak IntRC I/O seçildiğinde dahili osilatörün 4MHz frekansta çalışması sağlanmış olur.



Şekil 10: ICPROG ayarları

```
#include <pic.h>
#include <delay.c>
main(void)
{
    unsigned char gecici_dizi[16];
    unsigned char i,a,toplam_sutun;
    signed int kayma,deger;
    unsigned const char metin[]={
        0xFF,0x89,0x89,0x89,0x76,0x00, // B
        0x84,0xFD,0x84,0x00, // İ
        0xFF,0x80,0x80,0x80,0x80,0x00, // L
        0x84,0xFD,0x84,0x00, // İ
        0xFF,0x02,0x0C,0x02,0xFF,0x00, // M
        0x00,0x00,0x00,0x00, // Boşluk
        0x38,0x40,0x80,0x40,0x38,0x00, // v
        0x70,0xA8,0xA8,0xA8,0xB0,0x00, // e
        0x00,0x00,0x00,0x00, // Boşluk
        0x01,0x01,0xFF,0x01,0x01,0x00, // T
        0xFF,0x89,0x89,0x89,0x81,0x00, // E
        0xFF,0x18,0x24,0x42,0x81,0x00, // K
        0xFF,0x04,0x08,0x10,0xFF,0x00, // N
        0x84,0xFD,0x84,0x00, // İ
        0xFF,0x18,0x24,0x42,0x81,0x00; // K
    // Metindeki sütunların sayısını hesapla
    toplam_sutun=80; //10x6+5x4=60+20=80
    // Port ayarlama işlemleri
    TRISB=0;
    TRISA=0;
    CMCON=0x07;
    PORTB=0x00;
    for(;;){ //Ana döngü
        // Geçici diziyi sıfırla
        for(i=0;i<=15;i++){
            gecici_dizi[i]=0; //Dizi elemanları başlangıçta 0
        }
        // Kaydırma işlemleri
        for(kayma=-14;kayma<=toplam_sutun;kayma++){
            // Metni 16 sütunluk parçalara böl
            for(i=0;i<=15;i++){
                deger=i+kayma;
                if(deger<0)gecici_dizi[i]=0; //metin girişi
                if(deger>=0&&deger<=toplam_sutun-1)
                    gecici_dizi[i]=metin[deger];
                if(deger>toplam_sutun)gecici_dizi[i]=0; //metin çıkışı
            }
            // Tarama işlemleri
            for(a=0;a<40;a++){ // Aynı görüntüyü 40 kez tekrarla
                for(i=0;i<=15;i++){ // Geçici diziyi görüntüle
                    PORTB=gecici_dizi[i]; // Veriyi PortB'ye gönder
                    PORTA=i; // ilgili sütun'u seç
                    DelayUs(200); // 200 mikrosaniye bekle
                }
            }
            // İşlemleri tekrarla
            // Program sonu
        }
    }
}
```

Örnek 3'e ait C kodları

Günümüzde matris display içeren pek çok yeni cihaz ve aksesuar tasarlanmaktadır. Şekil 11-16'da bunlardan birkaçı görülmüyor. Kol saatleri, kolyeler, kemer tokaları ve yaka kartları ilginç ürünler arasında. Bu yazıda verilen düşük maliyetli ve esnek tasarım, benzer projeleri geliştirmekte kullanılabilir.



Şekil 11



Şekil 12



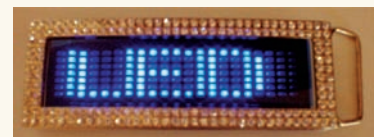
Şekil 13



Şekil 14



Şekil 15



Şekil 16

*Fırat Üniv. Elek-Elektronik Müh. Bölümü
yerol@firat.edu.tr