



Kendimiz Yapalım

Yavuz Erol*

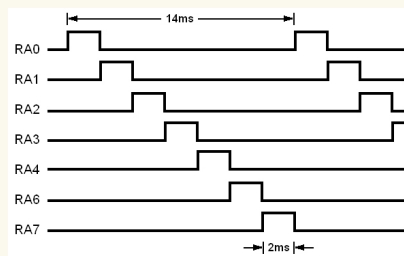
Mikro Denetleyici Kontrollü Display Sürücü

Mikro denetleyici kullanarak elektronik devre tasarlamak çoğu zaman donanım bileşenlerinin sayısını en aza indirir. İşin önemli bir kısmını yazılım üstlendiği için tasarlanan sistem daha küçük boyutlu ve daha düşük maliyetli olur. Mikro denetleyicinin sağladığı bu avantaj, programlama aşamasında yüksek seviye bir dil kullanılmasıyla daha da belirginleşir. Bu yazıda böyle bir sistemin nasıl tasarlandığı hakkında ayrıntılı bilgiler yer alıyor. Gerçekleştirilen proje ile kronometre, 7 dijitalik sayıcı veya genel amaçlı gösterge gibi uygulamalar yapılabilir.

Display sürücü devresinde kontrol işlemlerini yürütmek üzere PIC16F628A mikro denetleyicisi bulunuyor. Bu entegre A ve B olmak üzere iki adet giriş-çıkış portuna sahip. PIC16F84A'dan farklı olarak dâhili osilatör ile çalıştırılabilir. Ayrıca 15 ve 16 nolu bacaklarını giriş-çıkış olarak kullanmak da mümkün. İstenirse 4 nolu RA5/MCLR ucu giriş olarak kullanılabilir. Dış ortamdan bilgi girişi sağlamak için giriş ucuna uygun bir sensör ya da buton bağlamak gerekiyor.

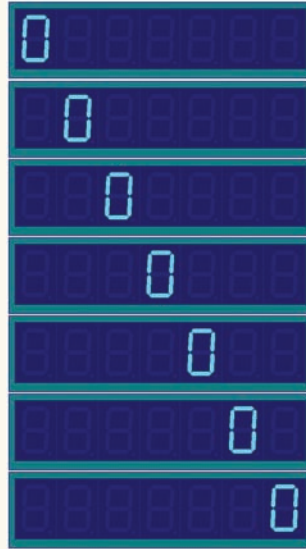
Display sürücü uygulamasına ait elektronik devre şeması şekil 1'de görülmekte. Devrede PIC16F628A mikro denetleyicisi, ULN2003 ve UDN2981A entegreleri bulunuyor. Gösterge kısmında ise 38mm boyutunda 7 adet ortak katotlu display bulunuyor.

Devredeki 7 adet display tarama yöntemine göre sürülüyor. Bu sayede display sürme devresi oldukça basit olmaktadır. Tarama tekniği ile display sürmek için şekil 2'deki zamanlama diyagramını dikkate almak gerekiyor.



Şekil 2: Zamanlama

Şekilden görüldüğü gibi her bir display sadece 2ms süresince seçili vaziyette. Bu durumda toplam tarama süresi 14ms oluyor. Bu da tarama frekansının yaklaşık 70Hz olması anlamına geliyor. Yüksek tarama hızı sayesinde görüntüde herhangi bir titreşim/kırışma etkisi oluşmuyor. Örneğin göstergede yan yana 7 adet sıfır görüntülemek için şekil 3'deki gibi sırayla display'leri seçmek ve seçilen display'e 0 değerini yazdırmak gerekiyor.

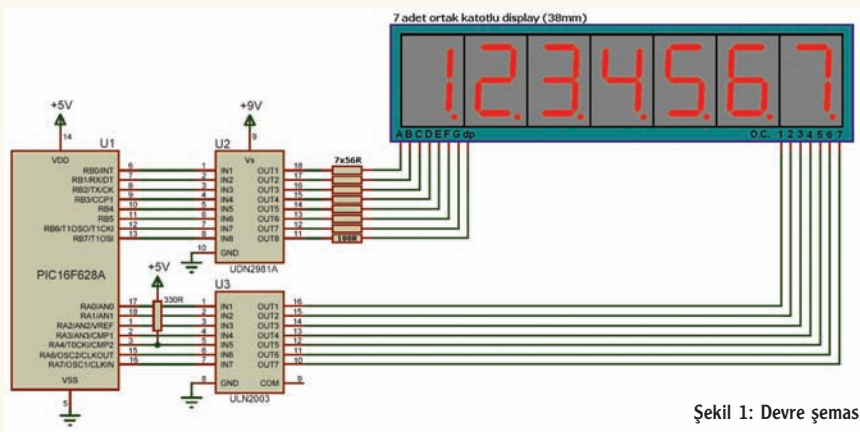


Şekil 3: Tarama tekniği

Bu işlem saniyede 70 kez tekrarlandığında gözümüz tarama etkisini fark edemiyor ve göstergedeki değeri şekil 4'deki gibi sabit bir görüntü olarak algılıyor.



Şekil 4: Algılanan görüntü



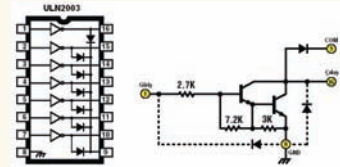
Şekil 1: Devre şeması

Devre şemasına ait açıklamalar şöyle: Mikro denetleyicinin A portuna bağlı ULN2003 entegresi dijital sürücü olarak çalışıyor ve 7 adet display'i sırayla seçme işini yürütüyor. Bu amaçla PortA'nın RA5 dışındaki bütün pinlerini çıkış olarak seçmek gerekiyor. PortB'ye bağlı UDN2981A entegresi ise display'lere veri göndermek için segment sürücüsü olarak çalışıyor. Devredeki 7 adet display'in aynı numaralı segmentleri birbirine paralel bağlı durumda. a,b,c,d,e,f,g adlı bu segmentler 56 ohm'luk birer direnç üzerinden UDN2981A entegresinin çıkışına bağlı. Ondalık noktayı gösteren dp ucuna ise 100 ohm'luk bir direnç bağlı. Display'lerin ortak katot (O.C.) uçları doğrudan ULN2003 entegresinin çıkışlarına bağlı halde.

Devrede kullanılan entegrelerin ve display'lerin iç yapıları aşağıda görülmekte.

ULN2003

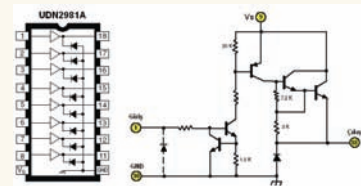
Şekil 5'den görüldüğü gibi ULN2003 entegresinin iç yapısında darlington bağlı transistörler bulunuyor. Transistörlerin ilettime girebilmesi için entegrenin giriş uçlarına 5V'luk gerilim uygulamak yeterli. Giriş uçlarına seri bağlı dâhili dirençler bulunduğu için giriş akımını sınırlamak üzere harici dirençler kullanmaya gerek yok. Bundan dolayı, ULN2003 entegresi mikro denetleyici çıkışına doğrudan bağlanabiliyor. Her bir kanalın çıkış akımı 500mA seviyesinde.



Şekil 5: ULN2003

UDN2981A

UDN2981A entegresi iç yapısında 8 adet transistörlü sürücü devre içermekte. ULN2003 entegresine benzer şekilde 5V'luk giriş gerilimi ile çalışıyor. Her bir kanal 350mA akım sağlayabilecek özellikte.



Şekil 6: UDN2981A

Ortak katotlu display

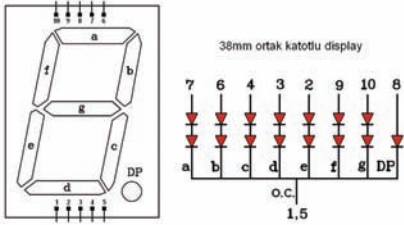
Piyasada farklı boyutlarda display çeşitleri bulunmakta. Rakam yükseklikleri 38mm, 45mm, 57mm şeklinde ifade ediliyor.

Kendimiz Yapalım



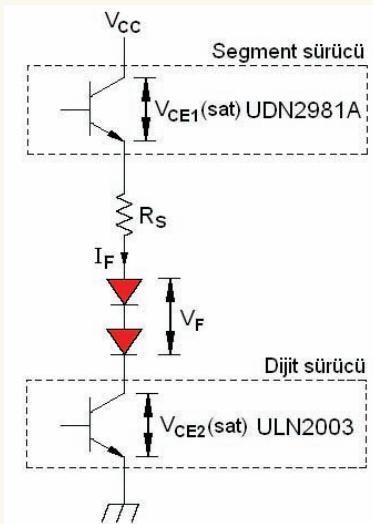
Şekil 7: Display çeşitleri

Bu projede kullanılan 38mm ortak katotlu display'in iç yapısı şekil 8'de görülüyor. Display'in her bir segmentinde seri bağlı iki adet LED bulunuyor. Ondalık nokta ise tek bir LED'den oluşuyor.



Şekil 8: 38mm 7 segment display

Tarama tekniği ile display'leri sürerken segment akımlarını uygun şekilde belirlemek gerekiyor. 20mA'lık akım seviyesi yeterli parlaklığı sağlayamadığından akım değeri daha yüksek olmalı. Çünkü ortalama LED akımı 2.85mA olmakta (20mA/7). Akım değerinin üst sınırını belirleyen sürücü entegrelerinin özelliklerini dikkate almak gerekiyor. Örneğin, toplam display akımı (yani segment akımının 7 katı değerindeki akım) ULN2003 entegresinin çıkış ucundan toprağa doğru akacağından, bu değer 500mA'ı geçemez. Bu da display'in her bir segmentindeki LED'lerden en fazla 71.5mA'lık akım geçmesi anlamına gelir. Bu akımı sağlamak için devrede uygun değerde bir direnç kullanmak gerekiyor. Direnç değeri hesaplanırken, şekil 9'daki eşdeğer devre dikkate alınmalı.



Şekil 9: Direnç hesabı için eşdeğer devre

Şekil 9'dan görüldüğü gibi, LED akımının aktığı kol üzerinde gerilim düşümüne neden olan elemanlar bulunmaktadır. Bunlar, UDN ve ULN kodlu entegrelerin VCE(sat) saturasyon gerilimleri ile

LED'lerin ileri yön (VF) gerilimi. Bu değerleri üretici firma kataloglarından okumak veya ölçüm yoluyla elde etmek gerekiyor. Aşağıda 50mA'lık LED akımı için ölçüm sonuçları görülmekte.

$$\begin{aligned} V_{CC} &= 9V \\ V_{CE1(sat)} &= 1.45V \\ V_{CE2(sat)} &= 0.75V \\ V_F &= 3.9V \end{aligned}$$

Bu durumda, Rs direncini belirlemek için çevre denklemi aşağıdaki gibi yazılır.

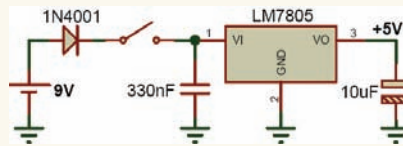
$$V_{CE1(sat)} + I_F \cdot R_S + V_F + V_{CE2(sat)} = V_{CC}$$

Bilinen değerler denklemde yerine yazılırsa RS değeri 58 ohm bulunur.

$$R_S = \frac{9 - 1.45 - 0.75 - 3.9}{50 \cdot 10^{-3}} = 58\Omega$$

Pratikte 56 ohm'lık bir direnç kullanılabilir. Direnç değeri, display'in sadece bir segmentinin ışık yayması durumu için hesaplanmıştır. Eğer bütün segmentler ışık yayıyorsa, toplam akım artacağından VCE2(sat) değeri 0.75V'dan büyük olur. 56 ohm'lık direnç değeri için LED akımının yeni değeri hesaplanırsa yaklaşık 45mA elde edilir. Kısaca, display'in kaç segmentinin ışık yaydığına bağlı olarak akım değeri 45 ile 50mA arasında değişir. Akım değişimi %10 ile sınırlı kaldığından, LED parlaklığındaki küçük değişimi göz zaten fark edemez.

Devre için gereken 5V'luk gerilim şekil 10'daki regülatör devresi ile sağlanabilir. Display parlaklığını olumsuz etkilememesi için 9V'luk güç kaynağının mutlaka regüleli olması gerekir.



Şekil 10: 5V'luk regülatör

Malzeme listesi aşağıdaki gibi.

Malzeme Listesi	
PIC16F628A	1 adet
ULN2003 entegresi	1 adet
UDN2981A entegresi	1 adet
38 mm ortak katotlu display	7 adet
56Ω direnç (0.25W)	7 adet
330Ω direnç (0.25W)	1 adet
100Ω direnç (0.25W)	1 adet
18'li entegre soketi	1 adet
1N4001 diyot	1 adet
330nF kutupsuz kondansatör	1 adet
10uF/16V elektrolitik kond.	1 adet
LM7805 regülatör	1 adet
Aç/kapa anahtar	1 adet
9V regüleli güç kaynağı	1 adet

PIC programında dijit ve segment sürücüyü gönderilecek sinyaller uygun şekilde ayarlanarak display'lerde istenen görüntüyü oluşturmak mümkün. Şekil 11'de Hi-Tech PIC C derleyicisinin yazılan örnek programın sonucu görülüyor.

Program, ilk olarak göstergede ieee.org yazısını yazıyor, ardından 9.999.999'a kadar sayma işlemine geçiyor. Programın ayrıntıları ve hex dosyası kendimiz yapalım köşesine ait web sayfasından indirilebilir.



Şekil 11: Baskı devre kartı

PIC C Programı

```
#include <pic.h>
#include <delay.c>
#include <led.c>
#include <stdio.h>
__CONFIG(WDTDIS&PWRRTEN&LVPDIS&INTIO&
MCLRDIS);
unsigned char i,tekrar,say[8];
unsigned const char
ieeee[]={0x06,0x79,0x79,0x9F,0x5C,0x50,0x6F};
unsigned const char
rakam[]={0x3F,0x06,0x5B,0x4F,0x66,0x6D,
0x7D,0x07,0x7F,0x6F};
unsigned char dijiti[8]={1,2,4,8,16,64,128};
main(void)
{
TRISA=0x00; TRISB=0x00; CMCON=0x07;
PORTB=0; PORTA=0;

// 2s boyunca ieee.org yazısını yazdır
for(tekrar=1;tekrar<150;tekrar++){
for(i=0;i<7;i++){
PORTA=dijiti[i];
PORTB=ieeee[i];
DelayMs(2);
PORTB=0;
}
}
//sayıları sıfırla
for(i=0;i<7;i++)say[i]=0;
// 0 ile 9.999.999 arasında say
for(;;){
if(say[0]==10){say[1]++;say[0]=0;}
if(say[1]==10){say[2]++;say[1]=0;say[0]=0;}
if(say[2]==10){say[3]++;say[2]=0;say[1]=0;say[0]=0;}
if(say[3]==10){say[4]++;say[3]=0;say[2]=0;
say[1]=0;say[0]=0;}
if(say[4]==10){say[5]++;say[4]=0;say[3]=0;say[2]=0;
say[1]=0;say[0]=0;}
if(say[5]==10){say[6]++;say[5]=0;say[4]=0;say[3]=0;
say[2]=0;say[1]=0;say[0]=0;}
if(say[6]==10){say[7]++;say[6]=0;say[5]=0;say[4]=0;
say[3]=0;say[2]=0;say[1]=0;say[0]=0;}

// 7 dijitali sayıcı değerini 0.2s süresince görüntüle
for(tekrar=1;tekrar<15;tekrar++){
for(i=0;i<7;i++){
PORTA=dijiti[i];
PORTB=rakam[say[6-i]];
DelayMs(2);
PORTB=0;
}
}
say[0]++; // Birler basamağını 1 artırır
}
} // Programın sonu
```



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