



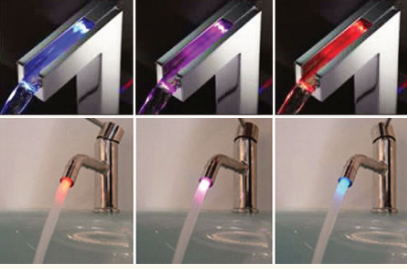
Kendimiz Yapalım

Yavuz Erol*

Renk Değiştiren Termometre

Bu ayki proje, ortam sıcaklığına bağlı olarak ışık rengini değiştiren bir termometre yapımı ile ilgili. Gerçekleştirilen elektronik devre ile çok farklı uygulamalar yapılabilir. Örnek uygulamalardan birkaçı aşağıda görülüyor.

Şekil 1'de, musluktan akan suyun sıcaklığına göre LED'in ışık rengini değiştiren bir uygulama görülüyor. Su soğuk iken ışık rengi mavi, su sıcak iken ışık rengi kırmızı olmaktadır. Böylece suya temas etmeden su sıcaklığını öğrenmek mümkün oluyor.



Şekil 1: Musluk uygulaması

Benzer bir uygulama duş başlığı için kullanılmıştır. Su ısındıkça ışık rengi maviden kırmızıya doğru değişir. Bu sistemde, LED'lerin çalışması için gereken enerji, duş başlığı içine yerleştirilen düşük güçlü bir jeneratör ile sağlanır.



Şekil 2: Duş başlığı uygulaması

Şekil 3'de, ayarlanan sıcaklığa göre gövde rengi değişen bir ütü modeli görülüyor. Böylece ütü sıcaklığını uzaktan bakarak anlamak mümkün oluyor.



Şekil 3: Ütü uygulaması

Şekil 4'de oda sıcaklığını ölçen bir cihaz görülmektedir. Sıcaklık değeri dijital göstergeden okunabildiği gibi, gövde rengine bakılarak da anlaşılabilir.



Şekil 4: Sıcaklık ölçümü uygulaması

Işık rengini değiştirmek için sıcaklık bilgisine gerek duymayan uygulama örnekleri de var. Örneğin şekil 5 görülen cihaz, telefon hattına bağlanıyor ve arayan numarayı gösterme özelliğine sahip. Cihazın yaydığı ışığın rengi, rehberde kayıtlı kişilere göre önceden ayarlanıyor. Telefon çaldığında, ışığın rengine bakılarak kimin aradığını bilmek mümkün oluyor.



Şekil 5: Caller ID uygulaması

Şekil 6'da ise her saat başı rengini otomatik olarak değiştiren dekoratif bir masa saati görülüyor.



Şekil 6: Masa saati uygulaması

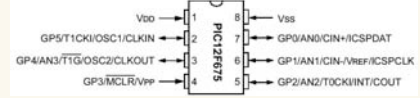
Ortam sıcaklığına göre ışık rengini değiştiren uygulamanın ayrıntılarını yazının devamında bulabilirsiniz. Renk değiştiren termometre projesinde sıcaklık bilgisi LM35 sıcaklık sensörü ile elde ediliyor. Bu sensör 10 mV/°C eğime sahip. Yani santigrad derece başına 10 mV gerilim üretiyor. Çıkış gerilimi sıcaklıkla doğru orantılı olduğundan herhangi bir ayarlama gerek kalmıyor. Fiyatının ucuz olması ve kolaylıkla temin edilebilmesi nedeniyle termometre devresinde LM35DZ adlı model kullanıldı. Sıcaklık sensörünün dış görünüşü şekil 7'de görülüyor. 3 bacaklı olan LM35DZ'nin kullanımı oldukça basit. 1 ve 3 nolu uçlara besleme gerilimi uygulanıyor, 2 nolu uçtan da sıcaklıkla doğru orantılı değişen gerilim ediliyor.

Şekil 7: LM35DZ sıcaklık sensörü



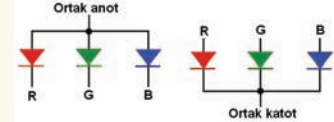
Şekil 7: LM35DZ sıcaklık sensörü

LM35DZ'nin ürettiği analog gerilimi sayısal değere dönüştürmek için PIC12F675 mikro denetleyicisi kullanıldı. 8 bacaklı bu mikro denetleyici, düşük fiyatı, küçük boyutu ve 4 adet analog-sayısal dönüştürücüye sahip oluşu nedeniyle tercih edildi. PIC12F675 entegresi uygun şekilde programlanarak RGB LED'in yaydığı ışığın rengi ayarlanacak.



Şekil 8: PIC12F675'in bacak bağlantısı

RGB LED'lerin bağlantı şekli ortak anotlu veya ortak katotlu olabilmektedir. Tek bir kılıf içerisinde 3 adet LED çipi barındıran RGB LED'in iç yapısı şekil 9'da görülüyor.



Şekil 9: RGB LED'in iç yapısı

Piyasada RGB LED'lerin çeşitli tipleri satılıyor. Bu LED'ler 5mm çaplı normal LED görünümünde olabildiği gibi, flux LED veya power LED türünde de olabiliyor. Şekil 10'da RGB LED çeşitleri görülüyor.



Şekil 10: RGB LED'ler (5mm, flux ve power)

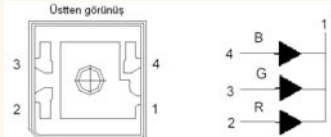
Bu projede kullanılan flux RGB LED'in ürün kodu 913PRGB2C, görüş açısı ise 130 derece. Bu LED (veya eşdeğer özelliklerde bir benzeri) ülkemizde kolayca bulunabiliyor. RGB LED'i temin etmek için internette arama yapmak yeterli. Bu LED ile renk spektrumundaki bütün renkleri elde etmek mümkün. Oluşturulan birkaç renk tonu şekil 11'de görülmekte.

Kendimiz Yapalım



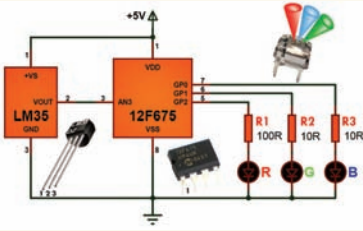
Şekil 11: Renk tonları

4 bacaklı flux RGB LED'in bacak bağlantısı şekil 12'de gibi.



Şekil 12: Flux RGB LED bacak bağlantısı

Projeye ait elektronik devre şeması şekil 13'de görülmüyor. Devrede PIC12F675 mikro denetleyicisi, LM35 sıcaklık sensörü, 3 adet direnç ve bir adet flux RGB LED bulunuyor. Besleme gerilimi 5V. Mikro denetleyici 4MHz'lik dâhili osilatör ile çalışıyor. Yazılımda uygun konfigürasyon ayarları yapıldığından, MCLR ucu ile VDD arasına bağlanması gereken dirence gerek yok.



Şekil 13: Devre şeması

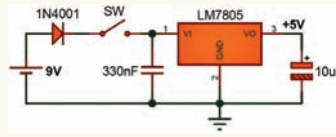
Devredeki R1, R2, R3 dirençleri, her bir LED'den 15-20mA akım geçecek şekilde seçilmeli. Direnç değerini hesaplarken mikro denetleyicinin port çıkış gerilimini dikkate almak gerekir. Çünkü porttan çekilen akım arttıkça çıkış gerilim seviyesi önemli ölçüde düşer. Örneğin, porttan hiç akım çekilmezken çıkış gerilimi 5V olduğu halde, 20mA akım çekilirken 3.7V'a kadar düşer. Aşağıdaki tabloda port akımı ile port gerilimi arasındaki ilişki görülmüyor.

I_{port} (mA)	V_{port} (V)
0	5
5	4.7
10	4.4
15	4
18	3.8
20	3.7

Devredeki direnç değeri ($V_{port}-V_f$)/ILED formülü ile hesaplandığından, LED'in ileri yön gerilimini de bilmek gerekiyor. Bu projede kullanılan flux RGB LED'in ileri yön gerilimi kırmızı için 1.9V; yeşil için 3.55V; mavi için 3.6V olarak ölçüldü. Buna göre, port çıkış geriliminin 3.8V olduğu dikkate alınırsa R1 direnci 100 ohm, R2 ve R3 dirençleri 10 ohm seçilebilir. Bu durumda kırmızı, yeşil ve mavi LED'lerin her birinden yaklaşık 19mA akım geçer. Bu da yeterli miktarda parlaklık sağlar.

Eğer istenirse yeşil ve mavi LED'ler direnç kullanmaya gerek olmaksızın port çıkışına doğrudan bağlanabilir. Bu durumda da LED akımları 20mA olur. Fakat akımı bir direnç ile sınırlamak daha uygun bir yöntem.

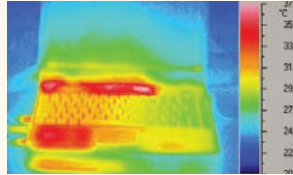
Elektronik devrenin çalışması için gereken 5V'luk gerilim, şekil 14'deki regülatör devresi ile sağlanabilir. Uzun süreli bir çalışma için 9V'luk pilin alkalın tipte olması gerekir.



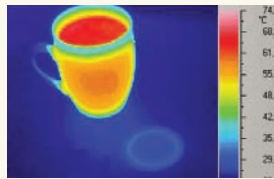
Şekil 14: Regülatör devresi

PIC programı PIC C Lite derleyicisi kullanılarak yazıldı. Hi-Tech firması tarafından üretilen programın demo sürümü (V8.05PL2), 12F675 mikro denetleyicisini tanımakta. Programın demo sürümünü "kendimiz yapalım" köşesine ait web sayfasından indirebilirsiniz.

PIC C programını yazmadan önce belirli bir sıcaklık aralığı için LED'in hangi renkte ışık yatacağını belirlemek gerekiyor. Bunun için termografik sistemleri incelemek faydalı olur. Şekil 15 ve 16'da görülen renk-sıcaklık grafikleri dizüstü bilgisayara ve sıcak sıvı dolu bir bardağa ait. Şekle bakıldığında hangi bölgelerin daha sıcak, hangi bölgelerin daha soğuk olduğu kolayca görülebiliyor. Şeklin sağında görülen renk skalası ise uygulamaya uygun şekilde seçilmiş. İlk uygulamada alt sıcaklık değeri 20°C, üst sıcaklık değeri 37°C seçilmiş. İkinci uygulamada ise bu değerler 22°C ve 75°C olarak alınmış.



Şekil 15: Dizüstü bilgisayar için test



Şekil 16: Sıcak sıvı dolu bardak için test

Bu projedeki alt sıcaklık değeri 20°C, üst sıcaklık değeri ise 45°C alındı. Bu sıcaklık aralığı birer derecelik 26 adıma bölündü. Her bir adım için şekil 17'deki gibi renk seçimi yapıldı. Böylece sıcaklık 20°C iken renk pembe olacak, sıcaklık arttıkça ışık rengi mavi, turuncu, yeşil, sarı, turuncu, kırmızı şeklinde yavaşça değişecek. Sıcaklık 45°C ve üzerinde iken renk kırmızı olacak.



Şekil 17: Renk skalası

Sıcaklığa göre renk değiştirme algoritmasına ait C programı aşağıdaki gibi.

```
#include <pic.h>
#include <delay.c>
#include <stdio.h>

#define BLEED GPIO0
#define GLED GPIO1
#define RLED GPIO2

// Dahili osilatör, MCLR direnci yok
_CONFIG(MCLRDIS&WDTDIS&PWRRTN&INTIO);

// Global değişkenler
const float lsb=5.0/1023.0;
float deger;sonuc;
unsigned char sıcaklik;
unsigned char (R=0,G=0,B=0);

// A/D donusum alt programi
void donusum_yap(void)
{
  ADCON0=0x8F; // A/D donusumu baslat
  while(GODONE!=0); // Donusumun bitmesini bekle

  deger=256*ADRESH+ADRESL;
  sonuc=deger/lsb*100.0;
  sıcaklik=(int)sonuc; // sıcaklik degeri
}

// Renk tonu degistirme alt programi
void RGB_ON(unsigned char (R,unsigned char (G, unsigned char (B);
unsigned int i;

for(a=0;a<25;a++){
  for(i=1;i<=255;i++){ // 255 adimda 3 ayri PWM isaret uret
    i(i<=R)RLED=1;
    i(i>R)RLED=0;

    i(i<=G)GLED=1;
    i(i>G)GLED=0;

    i(i<=B)BLED=1;
    i(i>B)BLED=0;

    DelayUs(10);
  }
}

// Sıcakliga gore renk tonunu belirleyen alt program
void bak_renk_tablosu(void)
{
  if(sıcaklik<20){(R=255; (G=0; (B=255);
  if(sıcaklik==20 && sıcaklik<21){(R=255; (G=0; (B=255); // pembe
  if(sıcaklik==21 && sıcaklik<22){(R=204; (G=0; (B=255);
  if(sıcaklik==22 && sıcaklik<23){(R=153; (G=0; (B=255);
  if(sıcaklik==23 && sıcaklik<24){(R=102; (G=0; (B=255);
  if(sıcaklik==24 && sıcaklik<25){(R=51; (G=0; (B=255);
  if(sıcaklik==25 && sıcaklik<26){(R=0; (G=0; (B=255); // mavi
  if(sıcaklik==26 && sıcaklik<27){(R=0; (G=51; (B=255);
  if(sıcaklik==27 && sıcaklik<28){(R=0; (G=102; (B=255);
  if(sıcaklik==28 && sıcaklik<29){(R=0; (G=153; (B=255);
  if(sıcaklik==29 && sıcaklik<30){(R=0; (G=204; (B=255);
  if(sıcaklik==30 && sıcaklik<31){(R=0; (G=255; (B=204); // turkuaz
  if(sıcaklik==31 && sıcaklik<32){(R=0; (G=255; (B=153);
  if(sıcaklik==32 && sıcaklik<33){(R=0; (G=255; (B=102);
  if(sıcaklik==33 && sıcaklik<34){(R=0; (G=255; (B=51);
  if(sıcaklik==34 && sıcaklik<35){(R=0; (G=255; (B=0);
  if(sıcaklik==35 && sıcaklik<36){(R=0; (G=255; (B=0); // yeşil
  if(sıcaklik==36 && sıcaklik<37){(R=51; (G=255; (B=0);
  if(sıcaklik==37 && sıcaklik<38){(R=102; (G=255; (B=0);
  if(sıcaklik==38 && sıcaklik<39){(R=153; (G=255; (B=0);
  if(sıcaklik==39 && sıcaklik<40){(R=204; (G=255; (B=0);
  if(sıcaklik==40 && sıcaklik<41){(R=255; (G=255; (B=0); // sarı
  if(sıcaklik==41 && sıcaklik<42){(R=255; (G=204; (B=0);
  if(sıcaklik==42 && sıcaklik<43){(R=255; (G=153; (B=0);
  if(sıcaklik==43 && sıcaklik<44){(R=255; (G=102; (B=0);
  if(sıcaklik==44 && sıcaklik<45){(R=255; (G=51; (B=0);
  if(sıcaklik==45 && sıcaklik<46){(R=255; (G=0; (B=0); // kırmızı
  if(sıcaklik==46){(R=255; (G=0; (B=0);
}

// ANA PROGRAM
main(void)
{
  CMCON=0x07; // GPIO portlari sayisal I/O secildi
  ANSEL=0x18; // AN3 analog giris secildi
  TRISIO=0x10; // GP0,GP1,GP2 cikis
  GPIO=0; // Baslangicta cikistari lojik 0
  ADCON0=0x8D; // A/D ayarlarini yap

  for(;;){
    donusum_yap();
    bak_renk_tablosu();
    RGB_ON((R,(G,(B);
  }
} // Program sonu
```

C programı derlendikten sonra hex dosya oluşturulur. Uygun bir programlama kartı ile bu hex dosya PIC mikro denetleyiciye yüklenir. Konfigürasyon ayarları C kodu içerisinde verildiğinden PIC programlama aşamasında herhangi bir ayar yapmaya gerek yok. Yani osilatör türü yanlışlıkla XT olarak seçilmemeli.

Projenin diğer ayrıntılarını ve video görüntülerini www.biltek.tubitak.gov.tr sayfasındaki "kendimiz yapalım" köşesinde bulabilirsiniz.