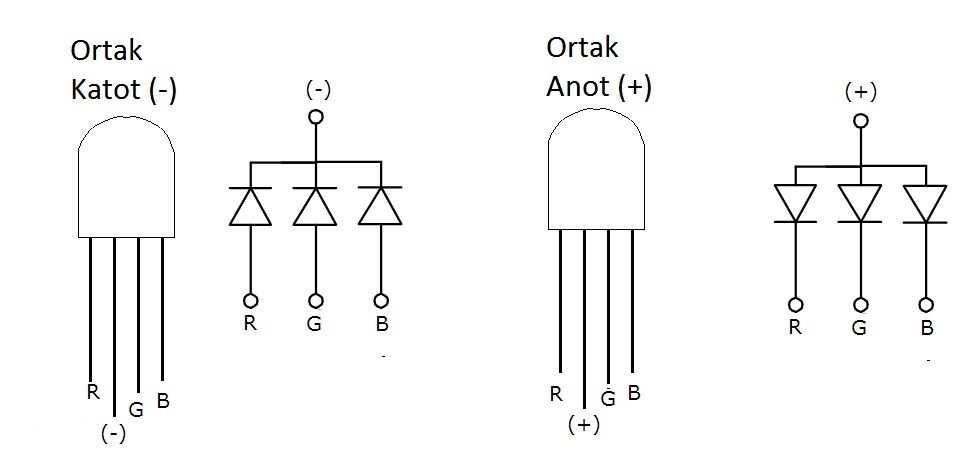
Merhaba arkadaşlar. Bugün sizlerle beraber Arduino ile RGB LED kontrolü yapacağız.

Gerekli malzemelerimiz:

* Arduino UNO
* Breadboard
* 1 adet ortak anot veya ortak katot RGB LED
* 3 adet 220 Ω direnç
* İki ucu erkek jumper kablo

RGB LED’ler, normal LED’lerden farklı olarak tek paket içerisinde 3 farklı renk (kırmızı, yeşil ve mavi) LED’i bir arada bulundurur. LED yakıp söndürme dersimizden hatırlayacak olursak LED’lerin anot ve katot uçları bulunuyordu. RGB LED’lerde ise LED’in üretim şekline göre anot veya katot bağlantıları ortak olarak bulunmaktadır.



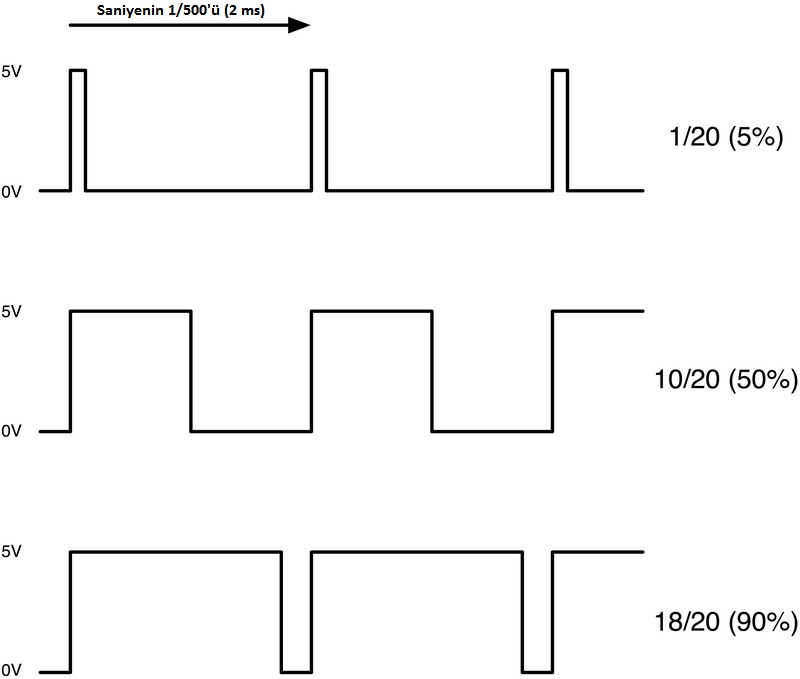
Bizim kullandığımız kitte bulunan RGB LED’imiz ortak anoda sahip. Bu yüzden Arduino için hazırlayacağımız kod ortak anoda göre çalışacak. Eğer ortak katot bir LED kullanmak istersek, kodumuzda ufak bir değişiklik yapmamız gerekecek.

**PWM ve Dijitalden Analog’a Dönüşüm**

Bildiğimiz gibi Arduino’muzun giriş/çıkış pinlerinde kullanılan voltaj 5V seviyesinde. Bir önceki dersimizde LED’imizi 5V gerilimde 20 mA akım çekecek şekilde kartımıza bağlamıştık. Bu şekilde bağladığımızda LED’imiz olabilecek en parlak şekilde yanmaktaydı. Peki parlaklığı değiştirmek istersek ne yapmamız gerekir?

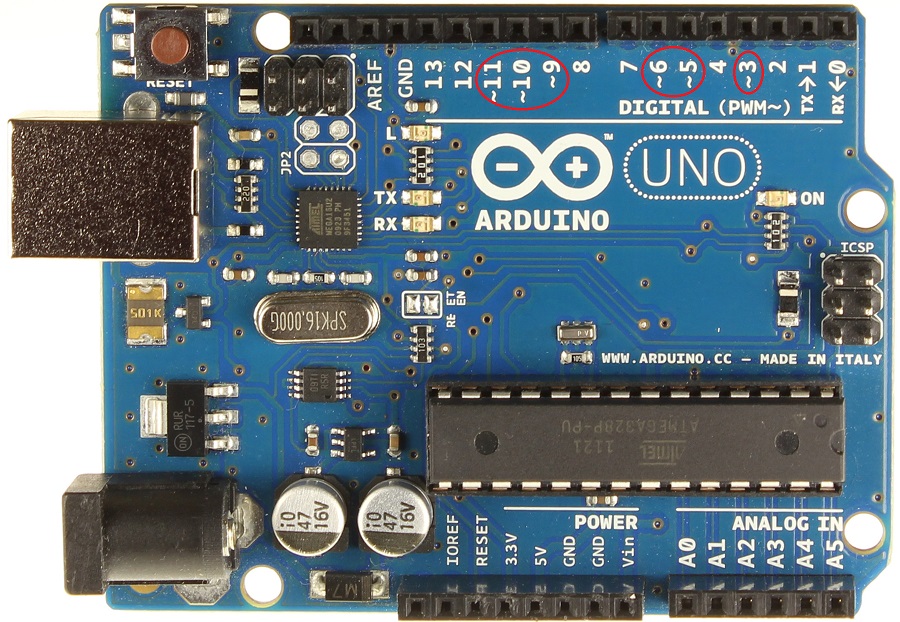
Sorunun cevabı aslında basit: gerilimi düşürmek. Eğer 5V ile çalışan LED’imizi daha düşük bir gerilimle, örneğin 3V ile çalıştırırsak parlaklığı azalacaktır. Fakat bu sefer de şunu sorabilirsiniz: Arduino çıkış gerilimi 5V değil miydi? Nasıl 3V çıkış alabiliriz?

Burada darbe genişliği modülasyonu (PWM – pulse width modulation) tekniğini kullanmamız gerekiyor. Bu yazıda darbe genişliği modülasyonu yerine kısaca PWM olarak bahsedeceğim. PWM, çıkışta aldığımız 5V gerilimi belirli bir zaman aralığında (genellikle saniyenin 1/500’ü) açıp kapatarak 0 ile 5V arasında istediğimiz gerilimi Arduino’nun çıkış pininden almamızı sağlar. Şu şekilde düşünelim: eğer bir LED çok hızlı bir şekilde yanıp sönerse gözümüz bu yanıp sönmeyi tam olarak algılayamaz ve parlaklığı daha düşük olarak algılar.



Yukarıdaki görselde gördüğümüz üzere, 5V’u 2 milisaniyelik sürenin sadece %5’lik kısmında açık olacak şekilde verirsek, elde ettiğimiz değer 5V’un %5’i; yani 0,25V olacaktır. Aynı şekilde eğer 2 ms sürenin yarısında (%50) açık kalacak şekilde ayarlarsak 2,5V elde ederiz.

Arduino UNO kartımızın tüm pinleri PWM çıkış yeteneğine sahip değildir. Kartımızın dijital pinlerinde pin numarasının önünde ~ işareti olan bazı pinler mevcuttur. Eğer PWM çıkış almak istiyorsak, bu pinleri kullanmak zorundayız. Bu pinler Arduino UNO için 3, 5, 6, 9, 10 ve 11 numaralı pinlerdir.



İlk dersimizde Arduino yazılımı içinde mevcut olan örnek kodu değiştirmiştik. Bu sefer kodumuzu kendimiz yazıyoruz:

int kirmiziPin = 9;

int yesilPin = 10;

int maviPin = 11;

void setup()

{

pinMode(kirmiziPin, OUTPUT);

pinMode(yesilPin, OUTPUT);

pinMode(maviPin, OUTPUT);

}

void loop()

{

renkAyarla(255, 0, 0); // kirmizi

delay(1500);

renkAyarla(0, 255, 0); // yesil

delay(1500);

renkAyarla(0, 0, 255); // mavi

delay(1500);

renkAyarla(255, 255, 0); // sari

delay(1500);

renkAyarla(80, 0, 80); // mor

delay(1500);

renkAyarla(0, 255, 255); // acik mavi

delay(1500);

renkAyarla(255, 255, 255); // beyaz

delay(1500);

}

void renkAyarla(int kirmizi, int yesil, int mavi)

{

kirmizi = 255 - kirmizi;

yesil = 255 - yesil;

mavi = 255 - mavi;

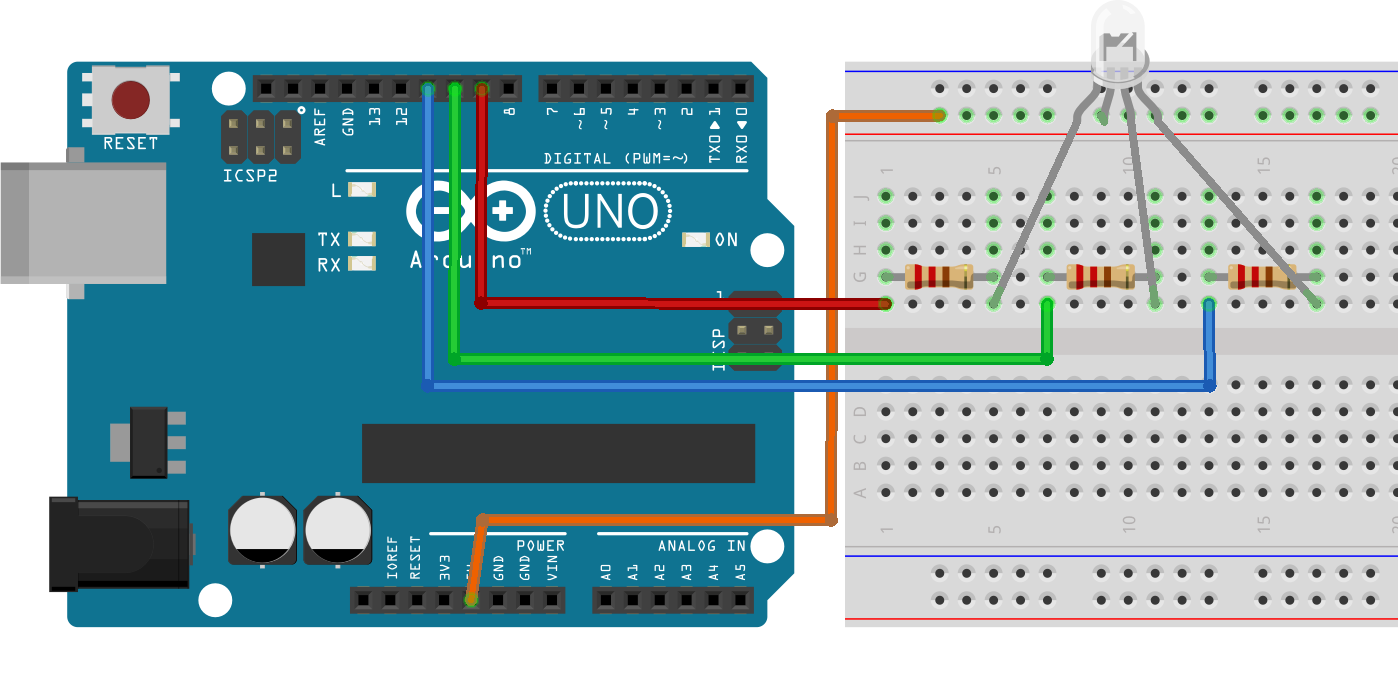
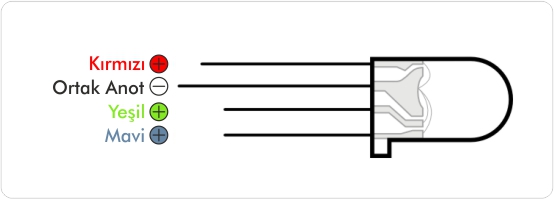
analogWrite(kirmiziPin, kirmizi);

analogWrite(yesilPin, yesil);

analogWrite(maviPin, mavi);

}

RGB LED’imizin kırmızı bacağını 9 numaralı pine, yeşil bacağını 10 numaralı pine ve mavi pinini 11 numaralı pine 220 Ω’luk dirençler ile bağlıyoruz. Kullandığımız LED ortak anot yapıya sahip olduğu için anot bacağını da kartımızın 5V pinine bağlıyoruz.



Kodumuzun setup fonksiyonunda kullanacağımız pinleri çıkış olarak tanımlıyoruz. Ayrıca yazdığımız renkAyarla isimli fonksiyondaki analogWrite komutu, bize her bir PWM çıkış pininden alacağımız voltajın yüksekliğini ayarlamamızı sağlıyor. analogWrite komutu şu şekilde kullanılıyor:

analogWrite(PWM çıkış pin numarası, 0-255 arası sayısal değer);

analogWrite komutunda 255 değeri maksimum çıkış voltajını yani 5V’u temsil ediyor. 0 ile 255 arası tüm değerler 0 - 5V arası voltaj değerlerine denk düşüyor. Örneğin analogWrite(9 , 80) komutu, 9 numaralı pinden **5V x (80/255) = 1,57V** geriliminde çıkış almamızı sağlıyor. Farklı parlaklıklardaki kırmızı mavi ve yeşil ışığı karıştırarak istediğimiz renkte ışık elde etmemiz bu sayede mümkün oluyor.

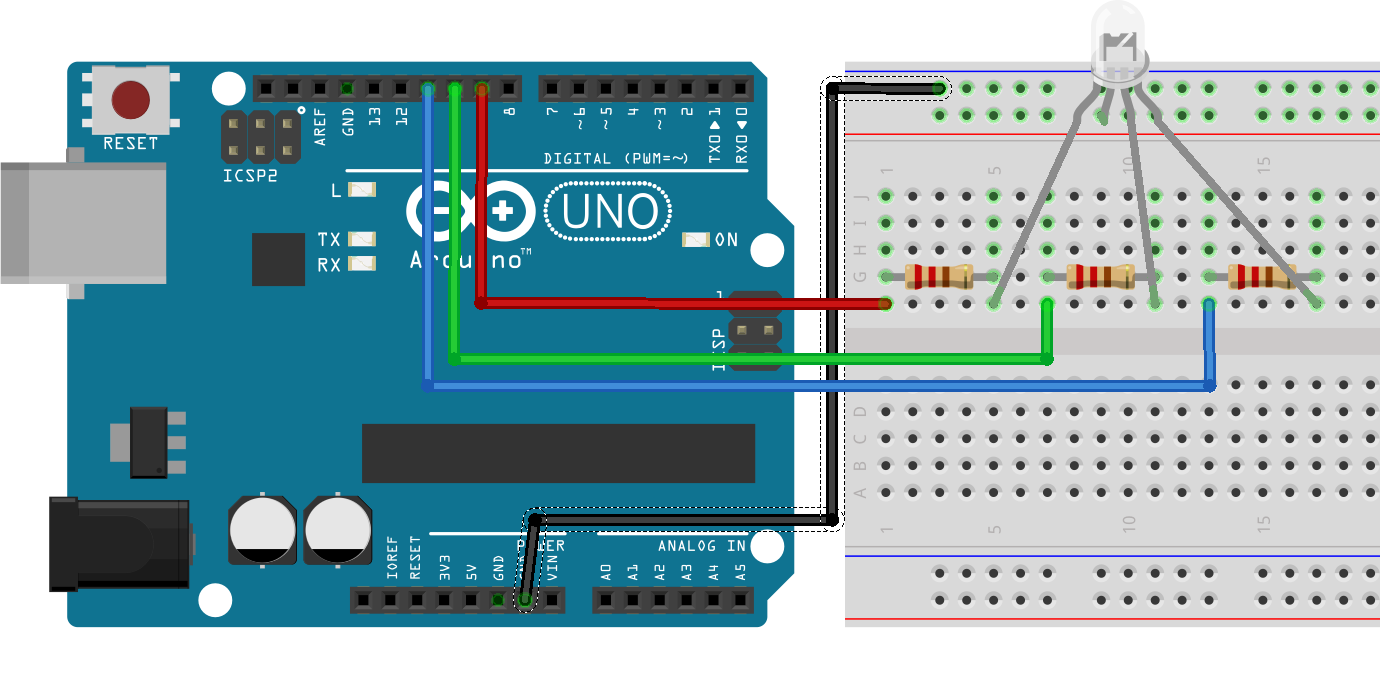
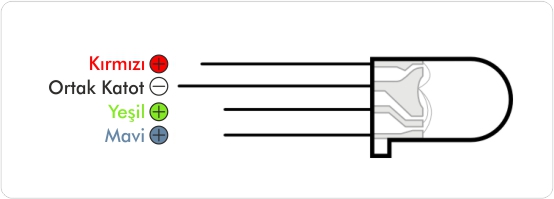
Eğer kullandığımız LED ortak anot yerine ortak katot yapıya sahipse, ortak bacağı +5V yerine GND pinlerine bağlamamız ve koddaki renkAyarla fonksiyonundaki

kirmizi = 255 - kirmizi;

yesil = 255 - yesil;

mavi = 255 - mavi;

kısmını silmemiz gerekiyor.



loop fonksiyonunda ise kartımız, oluşturduğumuz renkAyarla kodunu tekrar tekrar çağırarak çıkışlara istediğimiz değerleri yazmamızı sağlıyor. delay fonksiyonu bildiğimiz üzere her komut arasında kartımızın işlem yapmadan beklemesini sağlıyor. Bu fonksiyonun değerini değiştirerek istediğimiz hızda renkler arası geçiş sağlayabiliriz. Sizler de farklı parlaklık değerleriyle oynayarak farklı renkler elde edebilirsiniz.