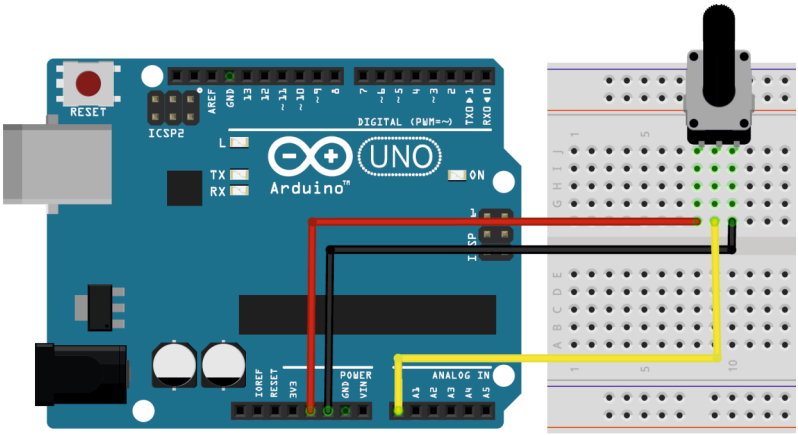
Gerekli malzemeler:

- Arduino Uno
- Breadboard
- 10k Ohm Potansiyometre
- 3 Adet Erkek-Erkek Jumper Kablo

Arduino kartının üzerine baktığınızda "Analog Input" pinlerini göreceksiniz. Bu pinleri kullanarak dijitalden analog sinyale dönüşüm yaparak bu pindeki voltajı okumamız mümkün. Arduino dijital okuma yaparken 0V (sıfır) ve 5V okuyabilmektedir. Bu iki uç değer arasında ara değerler gelirse bunu algılayamamakta ve gelen voltajı eşik değerine göre 0V veya 5V olarak kabul etmektedir. Analog pinler sayesinde 0V'dan 5V'ta kadar ara gerilim değerlerini de algılayarak dijitale çevirebiliyoruz. Ara değerlerdeki sinyalleri elde edebilmek için ayarlı direnç (potansiyometre) kullanacağız. Uygulamamızda analog giriş pininden gelen gerilimin sayısal karışılığını seri porttan okuma işlemini sağlayacağız. Devremizi kurup kod kısmına geçelim.



```arduino
#define potpin A0

int deger=0;

void setup() {
  Serial.begin(9600);
  Serial.println("Pot Değer Okuma");
}

void loop() {
  deger = analogRead(potpin);
  Serial.println(deger);
  delay(300);
}
```

Önceki kodlarımızda da yaptığımız gibi define işlemi ile "A0" pinine "potpin" ismini veriyoruz. Sonraki satırda analog pinden okuduğumuz değerleri saklamak için "integer" türünde ve değer isminde değişken tanımlıyoruz.

"setup" kısmında önceki yazılımlarımızda dijital giriş-çıkış kullandığımızdan dolayı, pinleri ne olarak kullanacağımıza göre ayarlıyorduk. Analog okuma yaparken giriş çıkış tanımlamamıza gerek yok bu sebepten dolayı bu yazılımda "pinMode" komutunu kullanmıyoruz.
Okuduğumuz verileri bilgisayara göndermek için seri haberleşme başlatmamız gerekiyor. Bu seri haberleşme sayesinde Arduino ile bilgisayar USB bağlantı üzerinden haberleşecek ve istediğimiz verileri bilgisayara aktarabileceğiz.

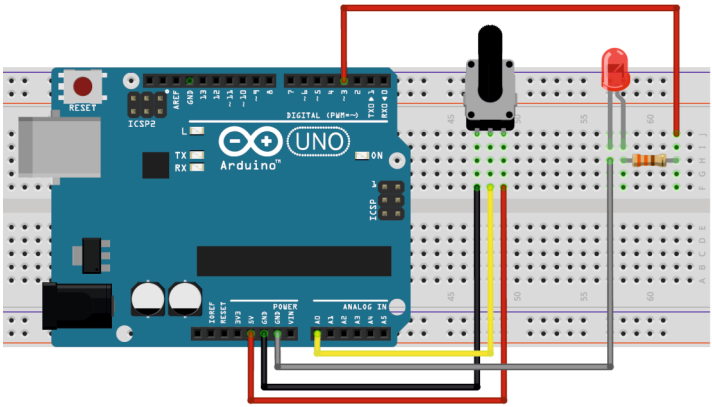
"Serial.begin(9600);" satırı ile bu haberleşmeyi başlatıyoruz. Arduino kodu çalıştırmaya başladığında ilk olarak bilgisayar ile haberleşmeyi başlatacaktır. Haberleşme başladıktan sonra "Serial.println("Pot Deger Okuma");" satırı ile "Pot Deger Okuma" yazısı bilgisayarda seri monitöre yazdırılacaktır. "Serial.print " ve "serial.println" komutlarını aşağıda detaylı açıklayacağız.

05 Potansiyometre ile LED Yakma

Gerekli malzemeler:

- Arduino UNO
- Breadboard
- 10k Ohm potansiyometre
- Kırmızı LED
- 330 Ohm Direnç (Turuncu - Turuncu - Kahverengi)
- 5 Adet Erkek-Erkek Jumper Kablo

Önceki uygulamamızda analog pinden gerilim değerini okumuştuk bu uygulamamızda yine analog pinden gelen değere göre LED'in parlaklığını kontrol edeceğiz. İlk led yakma uygulamamızda dijital çıkış kullandığımızdan dolayı LED'e 0V veya 5V gönderebiliyorduk. Bu yüzden led ya sönüyordu yada yanıyordu. Arduino'nun yeni bir özeliğini kullanarak LED'e 0-5V aralığında ara değerlerde gerilimde gönderebileceğiz. Bu gerilim kontrolü sayesinde LED'in parlaklığını ayarlayabileceğiz. Bu uygulamaya kadar dijital giriş-çıkış ve analog girişi öğrendik. Bu uygulamayla birlikte analog çıkış yani PWM özelliğini öğreneceğiz. Şimdi devremizi kurup hemen kod kısmına geçelim.



PWM (Pulse with Modulation) , sinyal genişlik modülasyonun kısaltmasıdır.Bu özellik Arduino Uno üzerine 6 pinde mevcut yaklaşık işareti (~) bulunan pinlerden (3,5,6,9,10 ve 11. Pinler) PWM ile ilgili detaylı bilgiler için uygulama sonundaki kare kodu taratarak bu uygulamaya ait olan videoyu izleyebilirsiniz. Devremizi kurduktan sonra hemen kod kısmına geçelim.

```
#define led 3
#define pot A0

void setup() {
}

void loop() {
  int deger = analogRead(pot);
  deger = map(deger,0,1023,0,255);
  analogWrite(led,deger);
}
```

Bu uygulamamız da dijital giriş-çıkış kullanmadığımızdan dolayı yine "setup" kısmında bir ayarlama yapmıyoruz.

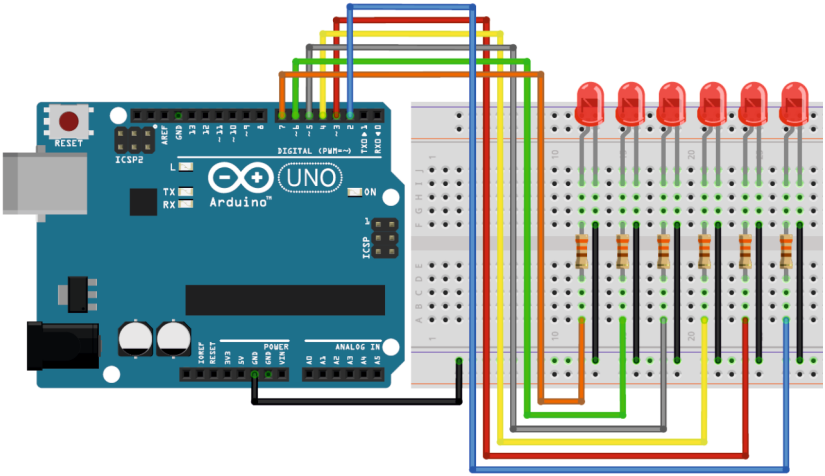
Ana program döngümüzde POT'tan veriyi okuyup bu veriyi LED'e göndermek istiyoruz. İlk olarak "analogRead" komutu ile potansiyometreden veriyi okuyoruz. Okuduğumuz değeri "deger" değişkenine yazıyoruz. İkinci satırda ise "map" komutunu kullanarak 0 ile 1023 arasında gelen değeri 0 ile 255 arasına oranlıyoruz.

Analog okumayı 10 bit (2^10 = 1024) çözünürlükte yaparken, analog yazmayı 8 bit (2^8 = 256) çözünürlükte yapabiliyoruz. Okuduğumuz veriyi, çıkışa göre oranlayıp yazdırmamız gerekiyor. "map" komutu yerine dilerseniz direk olarak 4'e de bölebilirsiniz. Oranlama işleminden sonra "analogWrite" komutu ile PWM pinlerinden çıkış verebiliriz.

Gerekli malzemeler:

- Arduino Uno
- Breadboard
- 13 Adet Erkek-Erkek Jumper kablo
- 6 Adet LED
- 6 Adet 330 Ohm Direnç (Turuncu - Turuncu - Kahverengi)

Bu uygulamamızda "for" döngüsünü kullanımını göreceğiz. "for" döngüsü ardı sıra yapılması gereken işlemlerde kullanılabilir. Uygulamamızda 6 adet LED'i yakmak için hepsini çıkış vermemiz ve sırasıyla yakmamız gerekecek. Bunu normal öğrendiğimiz çıkış tanımlama ve LED yakma söndürme komutları ile rahatlıkla uygulayabilmekteyiz. "for" döngüsü kullanmadan yazılan kodda değişiklik yapmak istediğinizde her satırda tekrar tekrar değişiklik yapmanız gerekecek, "for" döngüsünde ise hem kodu anlamak hem yazmak hem de değişiklik yapmak istediğiniz de çok daha hızlı şekilde ilerleyebileceksiniz. Devremizi kurduktan sonra hemen kod kısmına geçelim.



```
int ledler[] = {2,3,4,5,6,7};

void setup() {
  for(int i=0; i<6; i++){
  pinMode(ledler[i], OUTPUT);
  }
}

void loop() {
  for(int i=0; i<6; i++){
    digitalWrite(ledler[i], HIGH);
    delay(80);
    digitalWrite(ledler[i], LOW);
  }

  for(int j=5; j>-1; j--){
    digitalWrite(ledler[j], HIGH);
    delay(80);
    digitalWrite(ledler[j], LOW);
  }

}
```

Dijital pinleri tek tek tanımlarken "int" veya "#define" ile çıkışlara isim tanımlaması yapıyorduk. Bu sefer 6 çıkış birden kullacağımızdan dolayı dizi kullanarak ilerleyeceğiz. Dizileri, değişkenleri barındıran bir küme olarak düşünebilirsiniz. Kodun üst kısmında içerisindeki değişken türleri "int" (tamsayı) olan ve ismi "ledler" olan dizi tanımlıyoruz. Dizi elemanlarını ise içerisine virgüller ile ayırarak yazıyoruz. Dijital çıkışlardan 2 numaranda 7 numaraya kadar kullanacağımız için elamanlarımızı bu şekilde belirledik.

Dizinin elemanlarını çoğaltadabilirsiniz. Dizi tanımlama yaparken eğer istersek "ledler[]" ifadesi içerisine dizinin kaç elemanlı olacağını yazabiliriz. Örneğin "int ledler[6] = {2,3,4,5,6,7};" gibi Diziden eleman çağırırken ilk elemanın numarası 0'dan (sıfırdan) başlar. İstediğiniz elemanı çağırmak içinde "setup" veya "loop" içerisinde "ledler[*dizi elemanı sıra numarası*]" ifadesini kullanabilirsiniz. Yani eğer sıfırıncı dizi elemanını çağırmak için "ledler[0]" yazarsanız bu 2'ye eşit olacaktır. 5. Dizi elemanını çağırmak için "ledler[5]" yazarsanız buda 7'ye eşit olacaktır.

Uygulamamızda 6 adet LED'i yakmak istiyoruz bu durumda 6 adet dijital pinlerden çıkış belirlememiz gerekiyor. "for" döngüsünün parantez içerisinde ilk noktalı virgüle kadarki kısım döngü için kullanılacak koşulun değişkeni olarak tanımlanıyor.

Bu yazılımda sadece "for" döngüsü için kullanılacağından dolayı parantez içerisinde tanımladık. İsterseniz hali hazırda farklı bir değişkeninizi veya değişkeni yazılımın üstünde tanımlayıp sonradan burada kullanabilirsiniz.

"İ<6" ifadesi ise "for" döngüsünün koşulunu belirliyor. "i" değişkeni 6'dan küçük olduğu sürece "for" döngüsü içerisindeki kod satırlarını tekrarlayacak. Eğer "i" değişkeni 6'ya eşit veya büyük olursa artır "for" döngüsüne yapmayıp döngünün bittiği yerden kodu işletmeye devam edecek.

"i++" ifadesi ile "for" döngüsünü her yaptığımızda "i" değişkeninin değerini 1 arttırmasını istiyoruz. Böylelikle "i" değişkeninin ilk değeri 0(sıfır) oluyor. Dijital çıkış verdiğimiz komutlar içerisinde de "i" değişkenini yerine yazdığınızda diziden elemanı çağırıyor. Yani "pinMode(ledler[0], OUTPUT)" yazdığınızda yazılım "ledler" dizisinden sıfırıncı elemanı çağırarak "pinMode(2, OUTPUT)" olarak algılıyor. Bu sayede 2. pini çıkış olarak tanımlıyoruz. "for" döngüsü 0'dan 5'e kadar bunu yapacağı için 6 adet pini çıkış olarak tek satır ile tanımlamış oluyoruz.

"loop" kısmında da "setup" kısmındaki gibi "for" döngüsü kullanımı aynı mantıkla ilerliyor. Bu seferde çıkış tanımlamak yerine "digitalWrite" komutu ile sırasıyla 6 adet LED'i yakıp söndürüyoruz. "for" döngüsü ile ilgili detaylı video anlatımına aşağıdaki kare kodu taratarak izleyebilirsiniz.