GİKAL TEKNOLOJİ

KASIM 2022
Sayı 1

ARDUINO NEDİR?
PROJELERLE ARDUINO

YAYIN EKİBİ
RAMAZAN ÇOBAN
ÖZLEM MENEKŞE
ETEM DURSUN

01

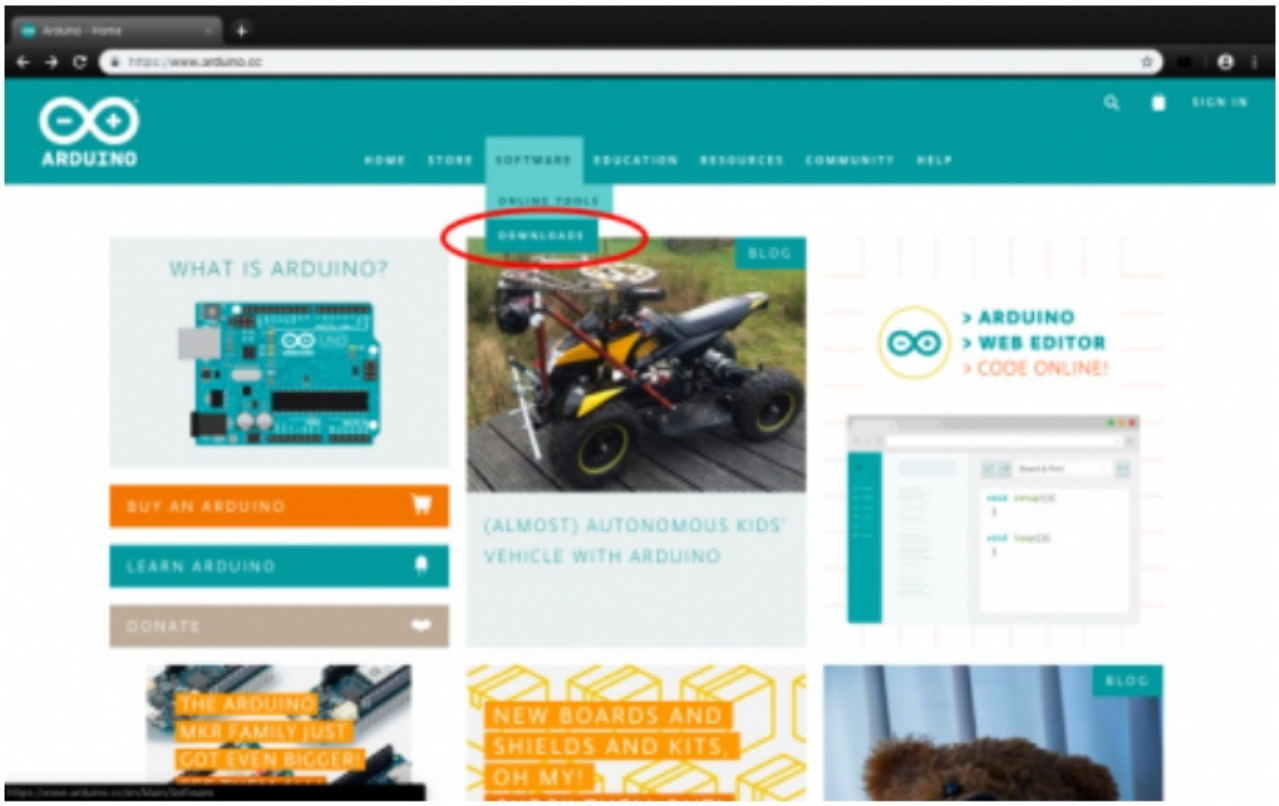
Arduino Nedir? Nasıl Kurulur ve Neler Yapılabilir?

Arduino Yazılım Kurulumu

Bu kitap ile birlikte Arduino dünyasına giriş yapıp, ileri seviye uygulamalara kadar gideceğiz. Uygulamalara başlamadan önce bilgisayarımızda Arduino sürücülerinin ve yazılımının kurulmuş olması gerekiyor. Benim tavsiyem, Arduino kartını bilgisayarımıza USB kablosu ile takmadan önce yazılımı yüklemeniz. Bu sayede, yazılımla birlikte gelen Arduino sürücüleri bilgisayarımıza kurulmuş oluyor ve böylece kartımızı kolaylıkla tanıtıp hemen kullanmaya başlayabiliyoruz.

Arduino Yazılımının İndirilmesi

Arduino yazılımını indirmek için www.arduino.cc adresinden **"Downloads"** sekmesine gidiyoruz.

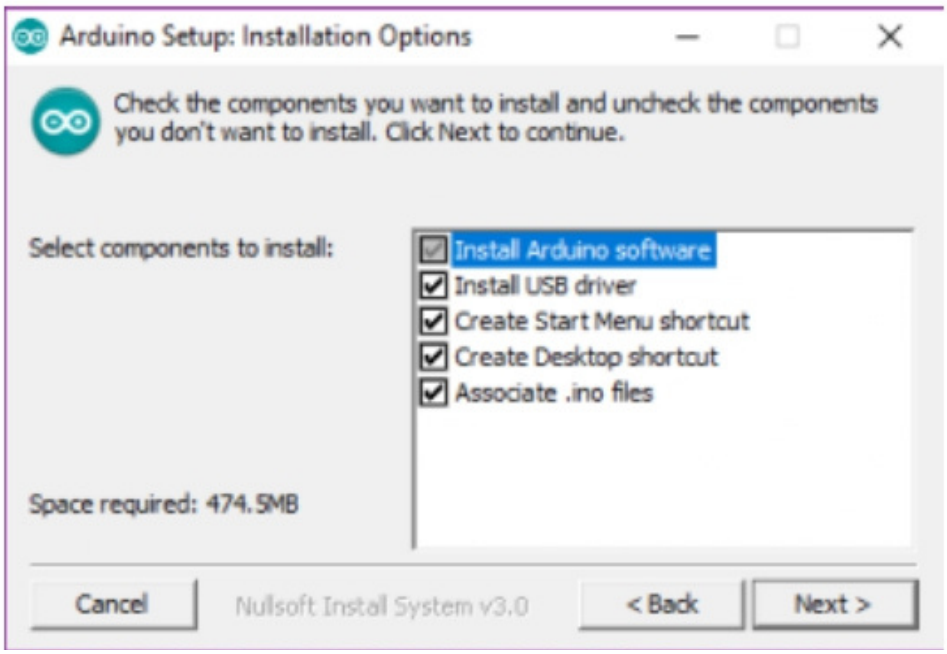


Downloads sekmesini tıkladıktan sonra karşımıza işletim sistemimize göre olan dosyayı indireceğimiz ekran çıkıyor. Bu yazıyı hazırladığım sırada Arduino yazılımının en güncel sürümü 1.8.7 idi. Windows kullananlar **"Windows Installer"** seçeneğini tıklayabilirler. Diğer işletim sistemlerinin de yükleme dosyaları aşağıda bulunmaktadır.

Daha sonra bize, bağış yapmamızı rica eden bir sayfa açılıyor. Tercihimize göre bağış yapabiliriz ya da **"Just Download"** seçeneği ile bağış yapmadan yazılımı indirebiliriz.

Arduino Sürücülerinin Yüklenmesi

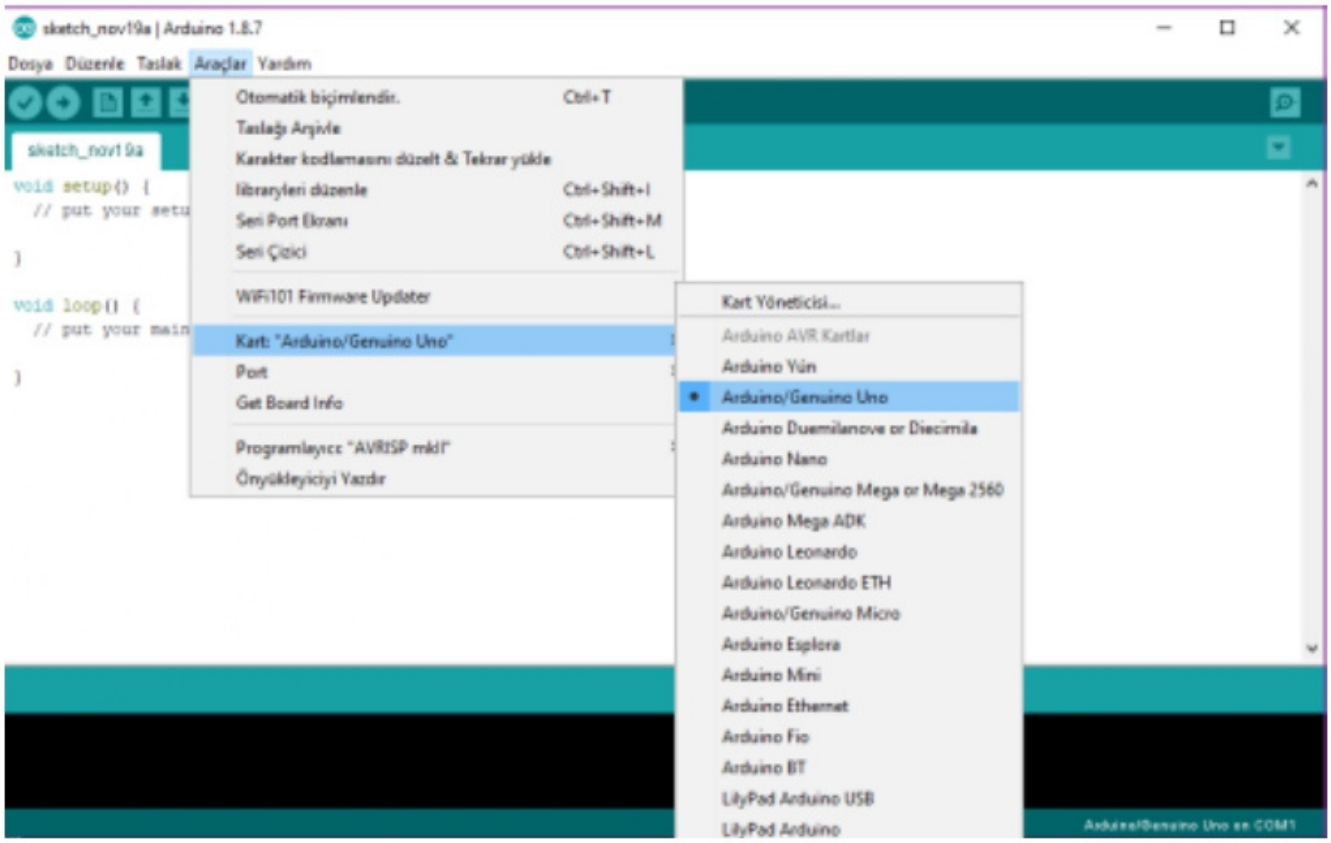
Bundan sonra yazılım kurulum dosyamız inmeye başlıyor. İndirme işlemi bittikten sonra dosyayı açarak kurulum işlemini başlatıyoruz. Kurulum sırasında çıkan "Install USB driver" seçeneğinin seçili olduğundan emin oluyoruz.



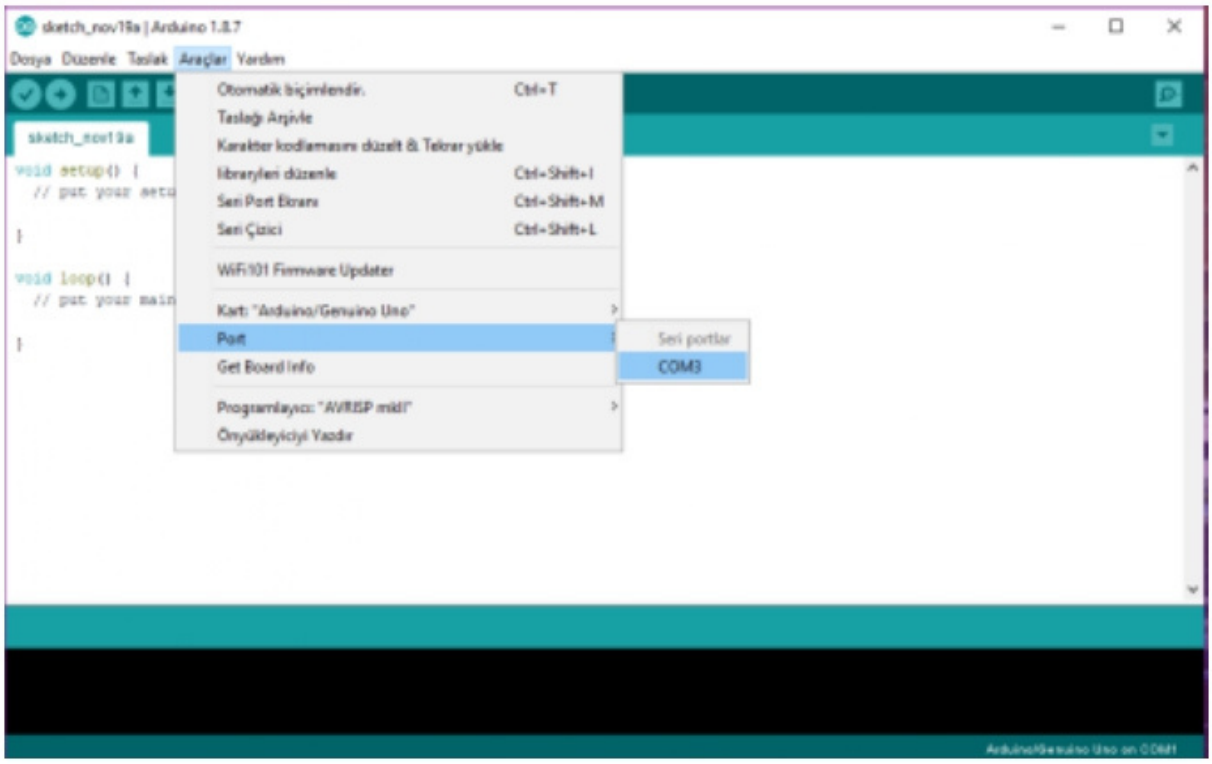
Kurulum işlemi bittikten sonra, kartımızı USB kablomuzla bilgisayarımıza bağlıyoruz. Bilgisayarımızda "Yeni donanım bulundu" penceresi açılıyor. Eğer sürücüler yazılımla birlikte kurulduysa, otomatik yükleme seçeneği Arduino'muzun sürücülerini otomatik olarak yükleyecektir.

Arduino Programının Bilgisayarımızda İlk Çalıştırılması

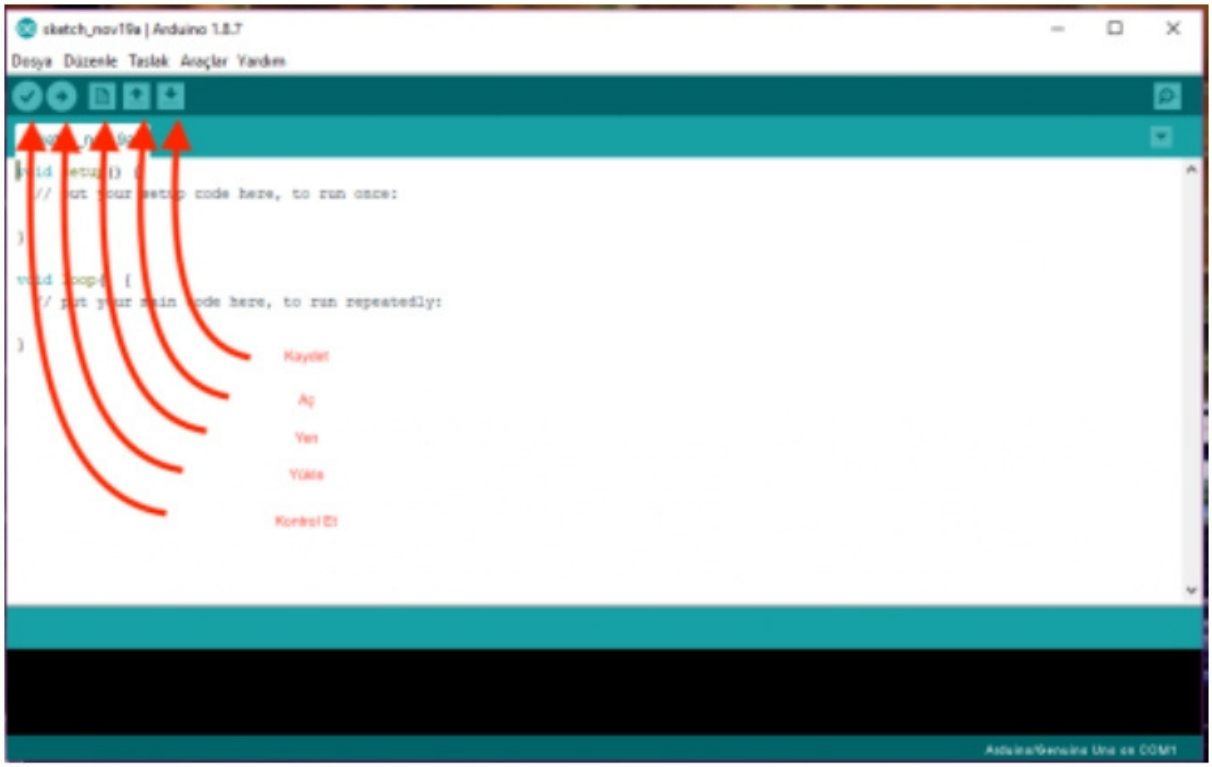
Artık Arduino programımızı açabiliriz. Programımızı açtıktan sonra ilk yapmamız gereken şey, programın Arduino UNO kartımızla çalışacak şekilde ayarlanmasıdır. **Araçlar > Kart** menüsünden Arduino UNO seçeneğini tıklıyoruz.



Daha sonra, yine "Araçlar" menüsünden "Port" alt menüsü altında Arduino'muzun bağlı göründüğü portu seçiyoruz. Bu port numarası, her bilgisayarda farklı olabilmektedir.

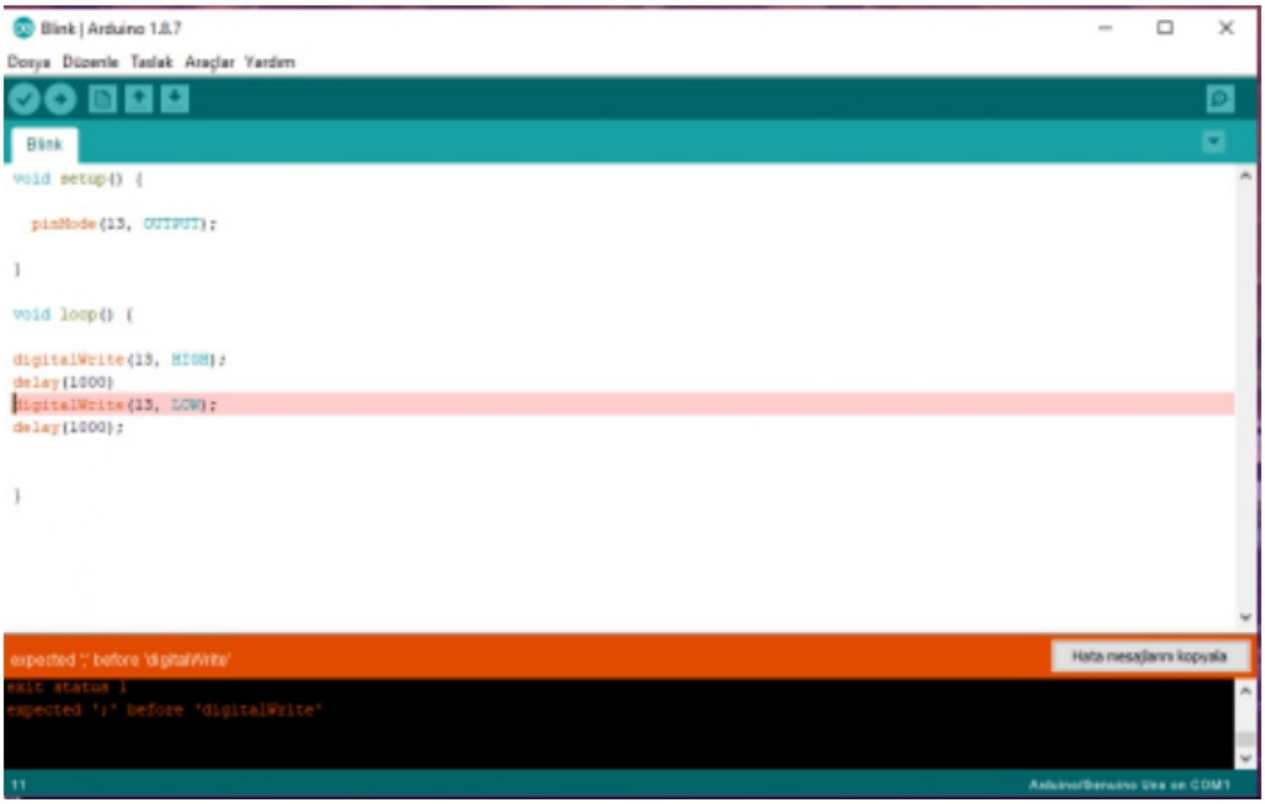


Artık her şeyiyle kullanıma hazır bir Arduino programımız var.



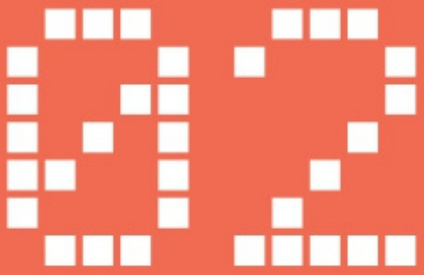
Programda "void setup()" kısmına yazacağımız fonksiyonlar, kart ilk enerji alıp çalıştığında sadece bir kere çalışır. Kullanacağımız giriş/çıkış pinlerini, seri port konfigürasyonunu vb. ayarları bu kısımda yapıyoruz. "void loop()" kısmında ise, "void setup()" fonksiyonundaki komutlar çalıştıktan sonra kartın enerjisi kesilene kadar sürekli çalışacak olan fonksiyonları barındırır.

Programımızı yazdıktan sonra kartımıza yüklemek istediğimizde, öncelikle "Kontrol Et" seçeneğine tıklıyoruz. Program, yazdığımız kodu öncelikle bilgisayarımızda bir klasöre kaydetmemizi istiyor, daha sonra da yazdığımız kodu derleyerek herhangi bir hata varsa bu hatayı bize bildiriyor.



Örneğin, bu kodda **"digitalWrite"** fonksiyonundan bir önceki komut olan **"delay"** komutunu yazdıktan sonra noktalı virgül (;) koymayı unuttuğumuz için bize bu satırla ilgili bir hata mesajı görüyoruz.

Eğer yazdığımız kodda bir hata yoksa ve Arduino kartımız bilgisayarımıza USB ile bağlıysa, **"Yükle"** seçeceğine tıklayarak kodumuzu kartımıza yükleyebiliyoruz.



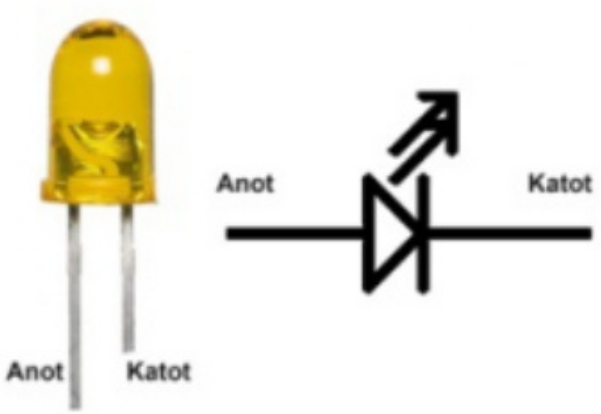
Arduino ile LED Yakma
Blink Uygulaması

Gerekli Malzemeler:

- Arduino Uno
- Breadboard
- Kırmızı LED
- 330 Ohm Direnç (Turuncu - Turuncu - Kahverengi)
- 2 Adet Erkek-Erkek Jumper Kablo

LED Nedir?

LED, ışık yayan diyot anlamına gelen "Light Emitting Diode" sözcüğünün baş harflerinden oluşan bir kısaltmadır. Alışık olduğumuz ve çoğu projemizde kullandığımız 6V ile çalışan ufak ampullerin aksine LED'lerin anot ve katot olmak üzere iki farklı bacağı vardır. Bunlardan anodu pozitif gerilime yani "+" uca, katot ise negatif gerilime yani "-" uca ya da toprak hattına (GND, Ground) bağlanmalıdır.



Gerilim, Akım ve Ohm Yasası

Çeşitli devre elemanlarının farklı gerilim yani voltajlarda çalıştığını biliyoruz. Arduino kartımız ise 5V gerilimle çalışmaktadır. LED'imiz için ise bu durum biraz farklıdır. LED'in üzerinden geçecek maksimum akımın 15 mA (miliamper = amperin 1000'de 1'i) değerini geçmemesi gereklidir. Arduino'muz 5V ile çalışıyor demiştik. 5V değeri bize kartın çıkış gerilimini ifade etmektedir. Fakat LED 15 mA akıma ihtiyaç duymakta. Sanırım işler biraz karışmaya başladı. Korkmaya gerek yok! Her şeyin bir çözümü var.

Arduino ile LED Yakma

LED, ışık yayan diyot anlamına gelen Light Emitting Diode sözcüğünün baş harflerinden oluşan bir kısaltmadır. Alışık olduğumuz ve çoğu projemizde kullandığımız 6V ile çalışan ufak ampullerin aksine LED'lerin anot ve katot olmak üzere iki farklı bacağı vardır. Bunlardan anodu pozitif gerilime yani + uca, katot ise negatif gerilime yani - uca ya da toprak hattına (GND, Ground) bağlanmalıdır.

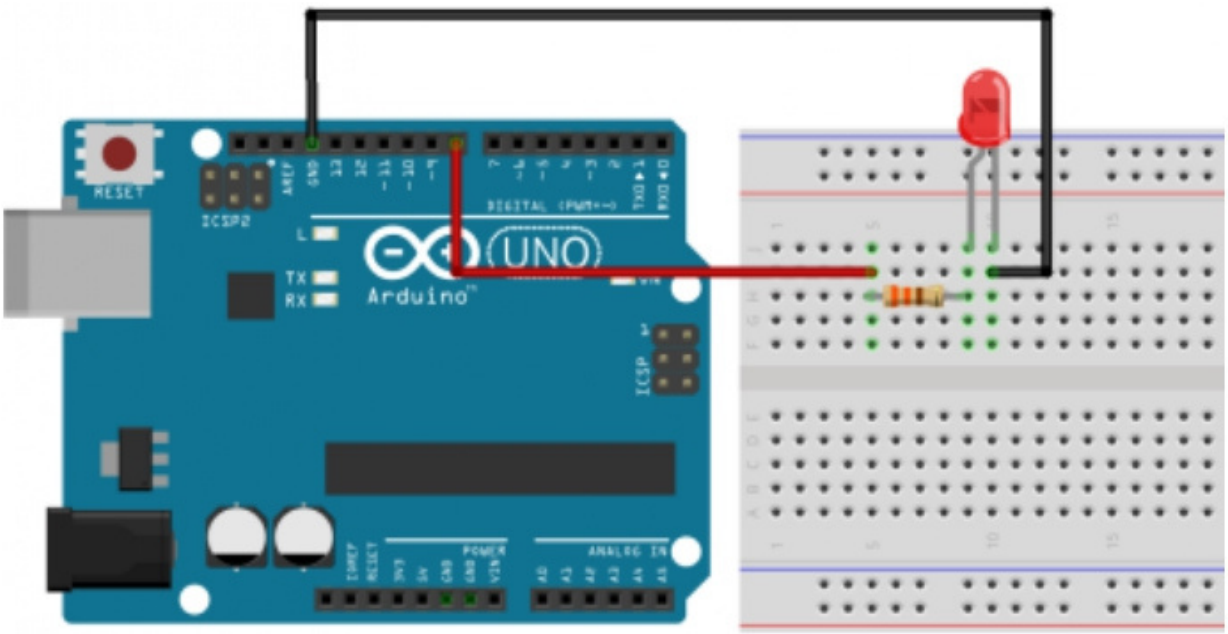
$$V = i \times R$$

Bu denklemde "V" bize gerilimi, "i" akımı ve "R" ise direnci temsil ediyor. Eğer 15 mA akıma ihtiyaç duyan LED'i, Arduino'muzun 5V çıkış sağlayan pinlerinden birine bağlayacak olursak;

$$5V = 0{,}015A \times R$$

Denklemini elde etmiş oluruz. Bu denklemden "R"yi çekecek olursak sonucu 333 buluruz. Bu demek oluyor ki LED'imizi 5V gerilimle kullanmak için 333 Ω (ohm) değerinde bir dirence ihtiyacımız var. Tam değeri doğru tutturmamız çok önemli değil, elimizde mevcut olan 330 Ω'luk direnci kullanabiliriz.

Hemen devremizi kuralım ve sonrasında proje kodumuzu yazmaya başlayalım.



Devre kısmını kurduktan sonra kodu yazmak için Arduino IDE'yi açarak, yukarıdaki sekmelerden "Dosya" sonrada "Yeni" seçeneğini seçerek yeni bir program sayfası açalım. Açılan sayfada "//" ve sonrasında yorum yazan satırları silebilirsiniz.

Bu sayfada ekleyeceğimiz kodları "void setup" ve "void loop" ile başlayan alanlara süslü parantezler"{}" içerisine yazacağız.

Arduino

```
1 void setup() {
2   pinMode(8,OUTPUT);
3 }
4
5 void loop() {
6   digitalWrite(8,HIGH);
7   delay(500);
8   digitalWrite(8,LOW);
9   delay(500);
10 }
```

Setup kısmında kart üzerindeki 8 numaralı pini çıkış verecek şekilde ayarlıyor. Kullanacağımız pin çıkış veya giriş olarak belirlenmez ise programın devamında yazacağımız giriş veya çıkış fonksiyonları, o pini kullanamaz. Pin için ayar yaptığımıza göre artık LED yakıp söndürme için gerekli olan kodu yazalım.

"loop" kısmında ise öncelikle 8 numaralı pine HIGH lojik seviyesine, yani 5V'a ayarlıyor, 500 milisaniye (yarım saniyeye eşittir) hiçbir işlem yapmadan bekliyor ve bu sefer 8 numaralı pini lojik LOW yani 0V veya toprak hattı seviyesine ayarlıyor. Bu işlemi yaptıktan sonra mikokontrolcü, "delay" fonksiyonu sayesinde tekrardan yarım saniye hiçbir işlem yapmadan bekliyor.

Bu koddaki "delay" komutlarının sürelerini değiştirerek LED'in açık ve kapalı kaldığı süreleri değiştirebiliriz. Eğer başka bir pin kullanmak istersek tek yapmamız gereken "pinMode" ve "digitalWrite" fonksiyonlarında bulunan pin numarasını kullanmak istediğimiz pin numarası ile değiştirmek. LED'imize 330 Ω'luk bir direnci seri bağlamayı unutmuyoruz!

03

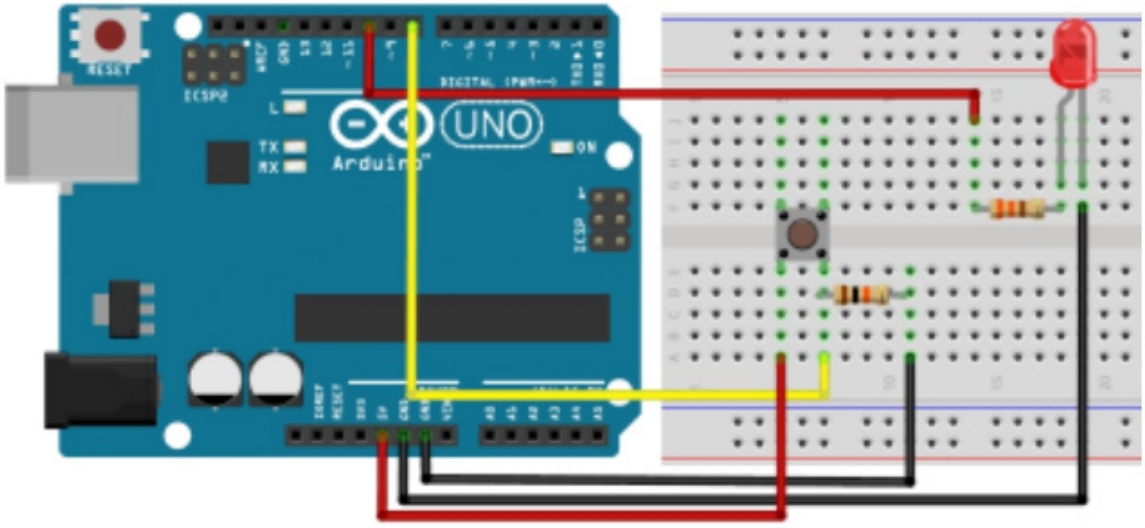
Buton ile LED Yakma Blink Uygulaması

Buton ile LED Yakma

Gerekli Malzemeler:

- Arduino Uno
- Breadboard
- Kırmızı LED
- Push Buton
- 330 Ohm Direnç (Turuncu - Turuncu - Kahverengi)
- 10k Ohm Direnç (Kahverengi - Siyah - Turuncu)
- 5 Adet Erkek-Erkek Jumper Kablo

Bu uygulamamızda Arduino üzerindeki pinleri giriş olarak kullanmayı öğreneceğiz. Bu sayede dışarıdan bir butona basıldığında Arduino içerisinde haberimizin olmasını sağlayacağız. LED devremiz bir önceki uygulama ile aynı olabilir. Sadece LED'in bağlı olduğu bacak bu uygulamada 10 numara olacak. Şimdi devremizi kurup daha sonra kod kısmına geçelim.



Butondan sağlıklı veri okuyabilmek için 10k Ohm direnç ile birlikte kullanmamız gerekiyor. Butona basılı değilken pin üzerinde oluşabilecek parazitleri ve bu parazitlerden kaynaklanan yanlış sinyal algılamalarını engellemek için "pull-up" veya "pull-down" direnci kullanmamız gerekiyor. Biz bu uygulamamız da "pull-down" direnci kullanacağız. Bu projemizde buton basılı değilken pinden okunan değer 0V yani lojik LOW seviyesindedir. "Pull-down" direnci, buton basılıp değer HIGH'a çekilmediği sürece bu pindeki gerilimi 0V'ta sabit kalmasını sağlar. Devre kısmındaki mantığı öğrendiğimize göre artık kod kısmına geçelim.

```arduino
#define Buton 8
#define Led 10

int buton_durumu = 0;

void setup() {
  pinMode(Buton, INPUT);
  pinMode(Led, OUTPUT);
}

void loop() {
  buton_durumu = digitalRead(Buton);
  if(buton_durumu == 1){
  digitalWrite(Led,HIGH);
  }
  else{
  digitalWrite(Led,LOW);
  }
}
```

"#define" satırı ile 8 numaralı pine "Buton" ismini veriyoruz bu sayede gerekli yerlerde 8 yazmak yerine "Buton" yazarak çok daha hatırlanabilir ve kolay kod yazabiliriz. LED içinde benzer tanımlamayı 10 numaralı pin için yapıyoruz.Yazılım içerisinde okuduğumuz verileri veya saklamak istediğimiz bilgileri, sonradan tekrar erişebilmek için değişkenleri kullanırız. Değişkenler tiplerine göre içerisinde barındırdıkları veriler farklılık gösterir. En çok karşımıza çıkacak olan ve sıkça kullanacağımız "int" değişkeni "integer"ın kısaltmasıdır. Bu değişken içerisinde -32767'den 32767 ye kadar sayıları tutabilir. Bu sayılar tam sayı olmalıdır. Virgüllü bir sayı içerisine atamak isterseniz yuvarlayarak tam sayıya dönüştürecektir. Bu kodda "buton_durumu" değişkeni tanımlayıp, ilk değerini sıfır olarak atıyoruz. İlk değerin sıfır atanması şart değil ancak "integer" tanımladığınızda değişken içerisinde rasgele bir sayı olabilir. Yazılım kısmında her ihtimale karşı sıkıntı oluşturmaması için ilk değerini sıfır olarak atıyoruz.

"pinMode" komutu ile "Buton" pinini (8 numara olarak belirledik) giriş olarak ayarlıyoruz. Bir alt satırda ise LED pinini (10 numara olarak belirledik) giriş olarak ayarlıyoruz.

Buton ile LED Yakma

Giriş-çıkış ayarlarken giriş yapmak istediğimiz butonlara "INPUT", çıkış yapmak istediğimiz pinlerede "OUTPUT" yazmamız yeterli. Giriş-çıkış olarak kullanacağınız pinleri tanımlamadığınız taktirde, bu pinler istediğiniz gibi veya stabil çalışmayacaktır."loop" kısmına geçtiğimizde butondan gelen veriyi okuyup, bu veriyi "if-else" komutu ile değerlendireceğiz. Değerlendirme sonucunda gelen verinin "1" veya "0" oluşuna göre karar vererek LED'i yakıp söndüreceğiz.

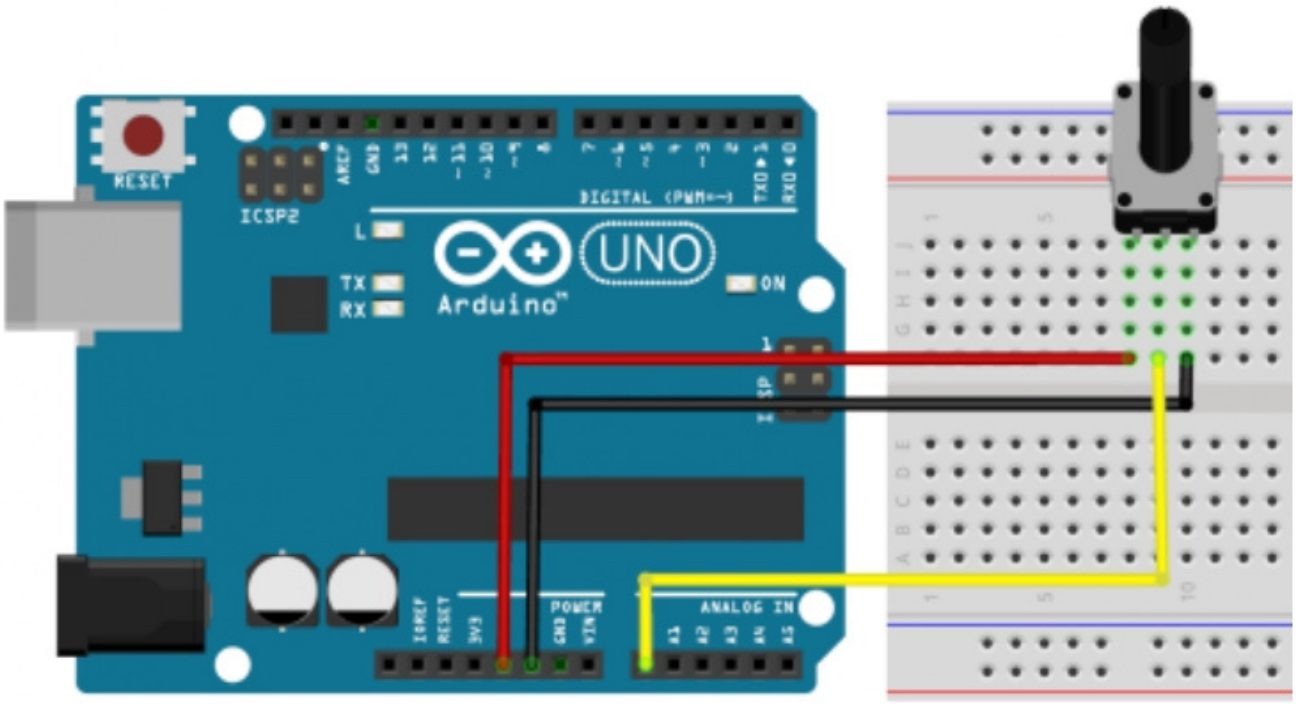
04

Arduino ile Analog Okuma ve Seri Haberleşme

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
1 #define potpin A0
2
3 int deger=0;
4
5 void setup() {
6   Serial.begin(9600);
7   Serial.println("Pot Değer Okuma");
8 }
9
10 void loop() {
11   deger = analogRead(potpin);
12   Serial.println(deger);
13   delay(300);
14 }
```

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.