



MBLOCK İLE ARDUINO UYGULAMALARI EL KİTAPÇIĞI

KODLUYORUZ DERNEĐİ

İÇİNDEKİLER

1. LED(Elektronik, Blink, Buton ile) Yakma	2
2. Trafik Işıđı Yapımı	7
3. Potansiyometre ile LED Kontrolü	10
4. Buzzer ile Ses Çıkartma.....	13
5. LDR ile Gece Lambası Yapımı	16
6. Mesafe Ölçer Yapımı	19
7. Park Sensörü Yapımı.....	22
8. 8x8 Dot Ekran Kullanımı	25
9. LCD Ekran Kullanımı	28
10. Yağmur sensörlü Cam Sileceđi Yapımı	31

LED(Elektronik, Blink, Buton ile) Yakma

Öğrenilen Kavramlar

- Arduino kart
- Breadboard
- LED
- Direnç
- Buton
- Olayı başlatma
- Koşullar
- Döngüler

Gerekli Malzemeler

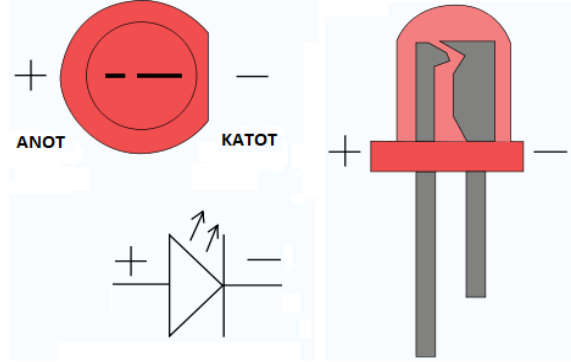
- Bilgisayar
- mBlock 3 programı
- Buton
- 10K veya 220 ohm direnç
- LED
- Arduino UNO
- Breadboard
- Jumper kablolar
- USB kablo

Amaç

- LED ve direnç bağlayarak devre kurmayı öğrenmek
- Kod kullanmadan ve kod kullanarak devreyi çalıştırmayı görmek

Kazanım

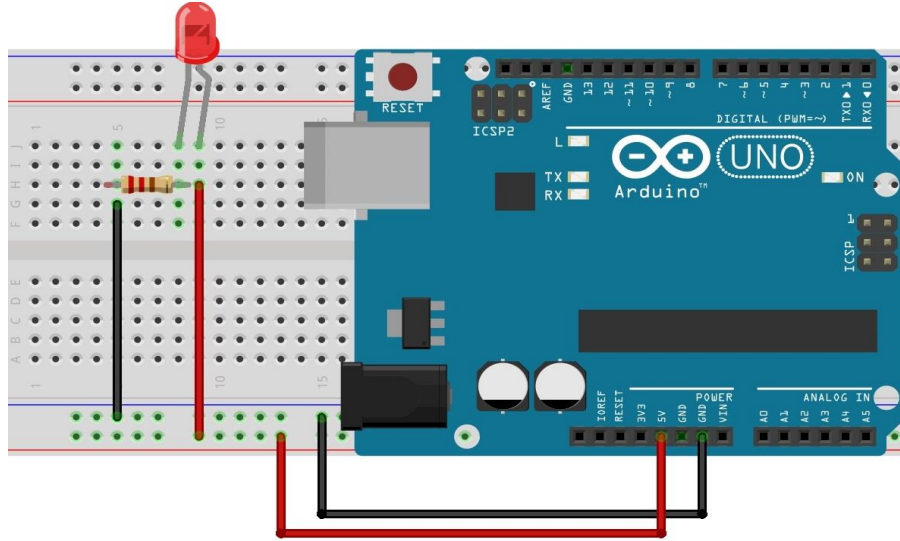
- Temel elektronik devre elemanlarını tanır.
- Elektronik devre elemanlarının çalışma mantığını kavrar.
- Temel düzeyde breadboard kullanır.



LED, ışık yayan diyot anlamına gelen Light Emitting Diode sözcüğünün baş harflerinden oluşan bir kısaltmadır. LED'lerin uzun bacak +(anot) ve kısa bacak -(katot) şeklinde 2 bacağı bulunur. Uzun bacak, Arduino kart üzerindeki dijital sinyal pinine, kısa bacak ise yine kart üzerindeki GND pinine bağlanmalıdır. Ancak dikkat etmemiz gereken bir nokta var. Arduino pinlere 5V enerji gönderir, LEDler ise 3V ile çalışmaktadır. İşte aradaki 2V kadar enerjiyi direnç ile dengeliyoruz.

1. Devre Şeması:

Öncelikle aşağıda gösterilen devre şemasına uygun olarak devremizin kurulumunu yapıyoruz.

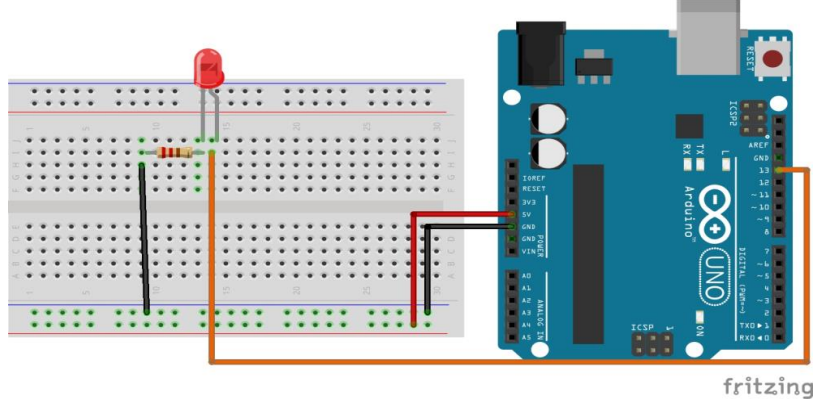


Uygulama:

- İlk uygulamamız elektronik olarak LED yakma olacak. Bunun için öncelikle yukarıda bulunan devre şemasının kurulumunu yapıyoruz.
- Arduino'yu USB kablo ile bilgisayara bağlıyoruz.
- Bu uygulama ile öncelikle bir güç kaynağından aldığımız güç ile LED'imizin direkt olarak çalışmasını sağlamış oluruz.
- Böylece LED'i dışarıdan bir müdahale olmadan, direkt fişe takıp çalıştırmış gibi kullanmış oluyoruz. Buna dışarıdan müdahale edebilmek için fişe takıp, çıkarmamız gerekecektir. Bu nedenle 2. aşamada programlama ile LED'imize müdahale etmeye geçeceğiz.

2. Devre Şeması:

Öncelikle aşağıda gösterilen devre şemasına uygun olarak devremizin kurulumunu yapıyoruz.



Uygulama:

- İlk uygulamamızda elektronik olarak LED yakmayı gördük ama bu şekilde dışarıdan müdahale sağlamak zor olacağından şimdi dijital bir pine bağlayarak program ile kontrollerini sağlayacağız. Bunun için öncelikle yukarıda bulunan devre şemasının kurulumunu yapıyoruz.
- Arduino'yu USB kablo ile bilgisayara bağlıyoruz ve "Kartlar" sekmesinden "Arduino Uno", "Uzantılar" sekmesinden ise "Arduino" seçeneğini seçiyoruz. "Bağlan" sekmesinden "Seri Port" kısmına tıklıyoruz ve açılan kısımdan seri portumuzu seçiyoruz. Son olarak da "Bağlan" sekmesinden "Aygıt Yazılımı Güncellemesi" kısmına tıklayarak Arduino'yu kullanıma hazırlamış oluyoruz.
- Daha sonra "Olaylar" dizisinden "_ tıkladığında" bloğunu alarak kodumuza bir başlangıç noktası belirliyoruz. Daha sonra LED'imizin çalıştığı sürece sürekli yanıp sönmelerini sağlayacak olan "Kontrol" dizisinden "sürekli tekrarla" kod bloğunu alıyoruz.
- Bu kod bloğunun içerisine "Robotlar" dizisinden "9 sayısal pini YÜKSEK yap" bloğunu alarak LED'i hangi dijital pine taktıysak o pini belirliyoruz.
- Daha sonra ne kadar süre yanacağını belirlemek için "Kontrol" dizisinden "1 saniye bekle" kod bloğunu ekliyoruz.
- LED'imizin sönmesi için de yine "Robotlar" dizisinden "9 sayısal pini YÜKSEK yap" bloğunu alarak LED'imizi hangi dijital pine taktıysak o pini belirliyoruz ve "DÜŞÜK" yap olarak değiştiriyoruz ve son olarak bir bekleme süresi ekliyoruz.

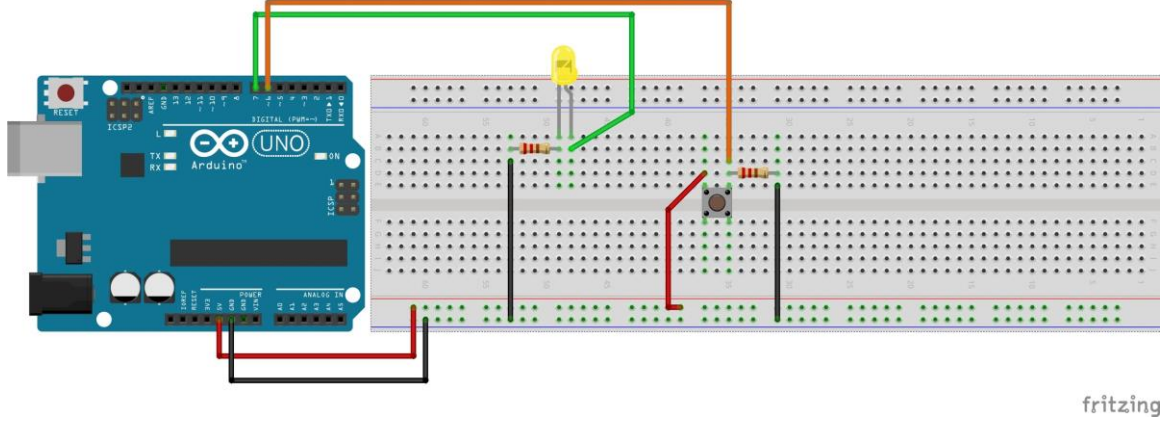
Gerekli kod bloklarını aşağıda detaylı olarak görebilirsiniz.



- Böylece LED gücü kesilene kadar 1 saniye aralıklarla yanıp sönecektir.

3. Devre Şeması:

Öncelikle aşağıda gösterilen devre şemasına uygun olarak devremizin kurulumunu yapıyoruz.



Uygulama:

- Devre kurulumu bittikten sonra mBlock programını açabiliriz. Eğer program açıksa üst kısımda bulunan “Dosya” sekmesinden “Yeni” kısmına tıklayarak yeni bir proje açabiliriz.
- Arduino’yu USB kablo ile bilgisayara bağlıyoruz ve “Kartlar” sekmesinden “Arduino Uno”, “Uzantılar” sekmesinden ise “Arduino” seçeneğini seçiyoruz. “Bağlan” sekmesinden “Seri Port” kısmına tıklıyoruz ve açılan kısımdan seri portumuzu seçiyoruz. Son olarak da “Bağlan” sekmesinden “Aygıt Yazılımı Güncellemesi” kısmına tıklayarak Arduino’yu kullanıma hazırlamış oluyoruz.
- Daha sonra butona bastığımızda, LED’ in yanıp sönmelerini sağlayacak kodları yazmaya başlayabiliriz.
- Öncelikle butona kaç kez basıldığını hesaplayabilmek için “Diziler” deki “Veri&Blok” bölümünde bulunan “Bir Değişken Oluştur” butonuna basarak “sayaç” isminde yeni bir değişken oluşturuyoruz.
- Oluşturduğumuz “sayaç” değişkeni ile butona her bastığımızda artacak şekilde gereken koşul ifadesini oluşturuyoruz. Sayacımız en başta “0” olarak başlıyor ve butonun bağlı olduğu pin de 0 ise sayacımızın 1 artacak şekilde ayarlıyoruz.

- Butonun basılı olup olmadığını programın anlayabilmesi için basılı iken 0, basılı değilken 1 konumunda olduğunu unutmamalıyız.

Gerekli kod bloklarını aşağıda detaylı olarak görebilirsiniz.



- Sayaç kodumuzu tamamladıktan sonra ise butona bastıkça yanıp ve butondan elinizi çeker çekmez sönen bir LED için gerekli kodları yazıyoruz.
- Bu kısım içinde butonunun değeri "0" olduğunda LED'imiz yansın, eğer "0" değilse sönsün şeklinde kod bloklarını yazıyoruz.

Gerekli kod bloklarını aşağıda detaylı olarak görebilirsiniz.

```

Arduino Programı
sürekli tekrarla
eğer 6 sayısal pini oku = 0 ise
  7 sayısal pini YÜKSEK yap
değilse
  7 sayısal pini DÜŞÜK yap

```

- Eğer programımızı daha da geliştirmek istersek butona bir kez basınca LED'in yandığı ve bir daha bastığımızda söndüğü bir sistem yapabiliriz.
- Bunun için de LED'in yanıp yanmadığını yönetebilmek için "Diziler" deki "Veri&Blok" bölümünden "Bir Değişken Oluştur" butonuna basarak "açık" isminde bir değişken oluşturuyoruz. Daha sonra yukarıdaki kodlara benzer olarak aşağıda bulunan kodlarımızı yazıyoruz.

Gerekli kod bloklarını aşağıda detaylı olarak görebilirsiniz.

```

Arduino Programı
açık , 0 olsun
sürekli tekrarla
eğer 6 sayısal pini oku = 0 ise
  eğer açık = 0 ise
    açık , 1 olsun
  değilse
    açık , 0 olsun
  6 sayısal pini oku olana kadar bekle
  6 pwm pini açık yap

```

- Son kodumuz ile birlikte butona basıldığında yanan, tekrar basıldığında ise sönen bir LED'imiz oldu. Buda bize ileride geliştirdiğimiz projelerin ihtiyaçlarına göre kullanabileceğimiz bir sisteme sahip olmamızı sağladı.

Trafik Işıđı Yapımı

Öđrenilen Kavramlar

- Arduino kart
- Breadboard
- LED
- Direnç
- Olayı başlatma
- Koşullar
- Döngüler

Gerekli Malzemeler

- Bilgisayar
- mBlock 3 programı
- Arduino UNO
- 1 adet kırmızı LED
- 1 adet yeşil LED
- 1 adet sarı LED
- 3 adet 220 ohm direnç
- Breadboard
- Jumper kablolar
- USB kablo

Amaç

- Birden fazla LED ve direnç kullanarak devre kurmayı öğrenmek
- mBlock programını kullanarak belli bir algoritma ile devreyi çalıştırmayı öğrenmek

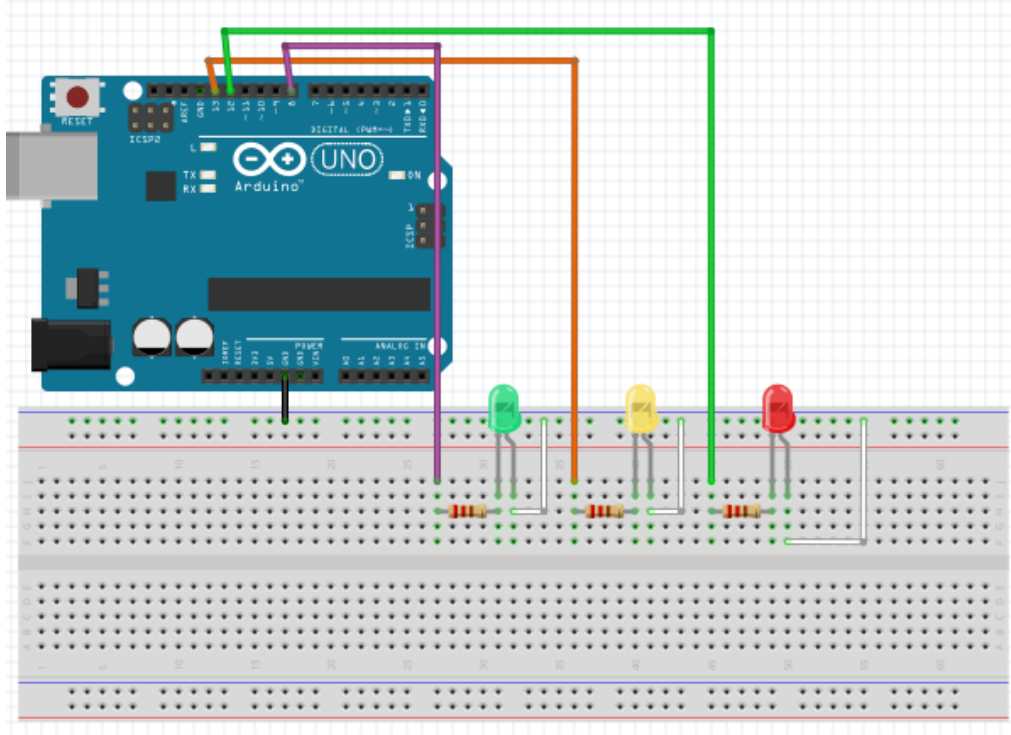
Kazanım

- Kodlama yardımıyla devreyi çalıştırmayı öğrenilir.
- Kod döngülerini ve koşul ifadelerini kullanmayı öğrenilir.
- Birden fazla devre elemanını beraber çalıştırmayı öğrenilir.
- Günlük hayat ile bağlantı kurar.

Bu uygulama ile günlük hayatta karşımıza çıkan trafik ışıklarının çalışma prensibini ve nasıl kurgulandığını öğreneceğiz. Bununla birlikte birden fazla LED'in nasıl kontrol edilebileceğini de öğrenmiş olacağız.

Devre Şeması:

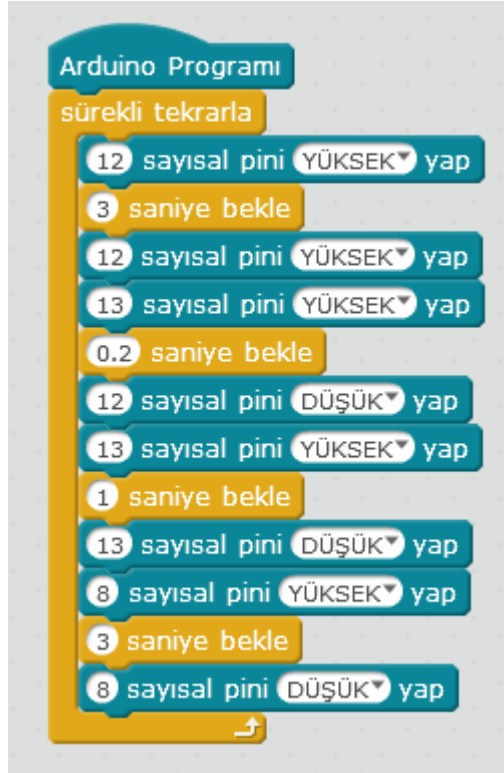
- Aşağıda gösterilen devre şemasına uygun olarak devremizin kurulumunu yapıyoruz.



Uygulama:

- Arduino'yu USB kablo ile bilgisayara bağlıyoruz ve "Kartlar" sekmesinden "Arduino Uno", "Uzantılar" sekmesinden ise "Arduino" seçeneğini seçiyoruz. "Bağlan" sekmesinden "Seri Port" kısmına tıklıyoruz ve açılan kısımdan seri portumuzu seçiyoruz. Son olarak da "Bağlan" sekmesinden "Aygıt Yazılımı Güncellemesi" kısmına tıklayarak Arduino'yu kullanıma hazırlamış oluyoruz.
- Böylece kodlarımızı yazmaya başlayabiliriz.
- Öncelikle tıkladığımız zaman programımızın çalışmaya başlayabilmesi için "Robotlar" dizisinden "Arduino Programı" bloğunu alıyoruz. Daha sonra kodumuzun sürekli olarak çalışabilmesi için "Kontrol" dizisinden "sürekli tekrarlar" komutu olarak "Arduino Programı" bloğunun altına yapıştırıyoruz.
- Sırası ile "Robotlar" dizisinden aldığımız "_ sayısal pini YÜKSEK yap" komutunu alarak, kırmızı, sarı ve yeşil LED i bağladığımız pinlere göre düzenliyoruz. Bu komutu eğer "YÜKSEK yap" dersek LED'imiz yanacak, "DÜŞÜK yap" olarak değiştirdiğimizde de sönecektir. Buna göre sırası ile önce kırmızı, sonra sarı ve sonra yeşil LED yanacak şekilde sırası ile "Kontrol" dizisinden "1 saniye bekle" kod bloğunu alarak belirlediğimiz bekleme sürelerini de ekleyerek kodumuzu yazabiliriz.

Gerekli kod bloklarını aşağıda detaylı olarak görebilirsiniz.



```
Arduino Programı
sürekli tekrarla
12 sayısal pini YÜKSEK yap
3 saniye bekle
12 sayısal pini YÜKSEK yap
13 sayısal pini YÜKSEK yap
0.2 saniye bekle
12 sayısal pini DÜŞÜK yap
13 sayısal pini YÜKSEK yap
1 saniye bekle
13 sayısal pini DÜŞÜK yap
8 sayısal pini YÜKSEK yap
3 saniye bekle
8 sayısal pini DÜŞÜK yap
```

- Kodlarımızı tamamladıktan sonra “Arduino Programı” bloğuna tıkladığımızda “Kırmızı LED” imiz 3 saniye yanacak, 3 saniye sonra “Sarı LED” imiz yanacak ve 0,2 saniye bekledikten sonra “Kırmızı LED” imiz sönecek. Daha sonra “Sarı LED” imiz tek başına 1 saniye daha yanacak ve sonra sönecek. “Sarı LED” söndükten sonra da “Yeşil LED” imiz 3 saniye yanacak ve sonra sönecek, bunlar bir döngü içerisinde programımız çalıştığı sürece uygulanacak.
- Trafik ışıklarında olduğu gibi her bir ışık belirli bir aralıkta yakılır ve daha sonra söndürülür.

Not: Bu projemize yeni güncelleştirmeler ya da gelişmeler eklemek istersek eğer, yaya geçitlerinin olduğu trafik ışıklarında bulunan butonların mantığı ile yeşil LED yanar iken, butona basıldığında belirlenen süre geçtikten sonra kırmızı LED’ in yanmasını sağlayan bir kod da ekleyebiliriz.

Potansiyometre ile LED Kontrolü

Öğrenilen Kavramlar

- Arduino kart
- Breadboard
- Diziler
- Bloklar
- Döngüler
- Mantıksal operatörler
- Analog pin
- Potansiyometre
- LED
- Direnç

Gerekli Malzemeler

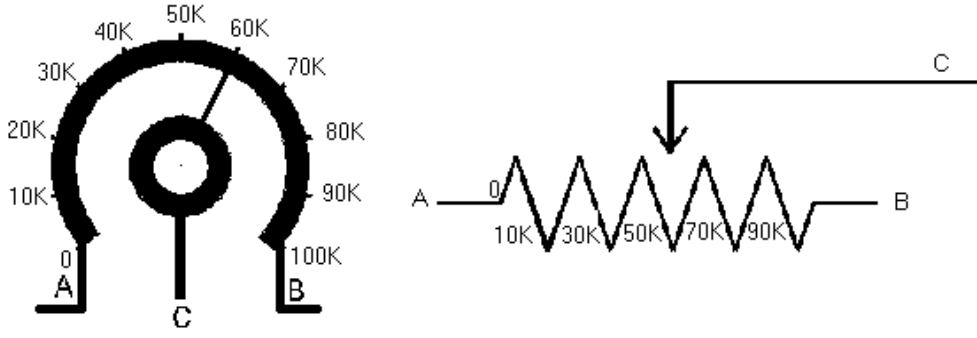
- Arduino UNO
- Breadboard
- USB kablo
- LED
- Direnç
- Potansiyometre
- Jumper Kablo

Amaç

- Potansiyometre ile LED parlaklığını kontrol etmek

Kazanım

- Kodlama yardımıyla devreyi çalıştırmayı öğrenilir.
- Kod döngülerini kullanmayı öğrenilir.
- Potansiyometre kavramını öğrenilir.



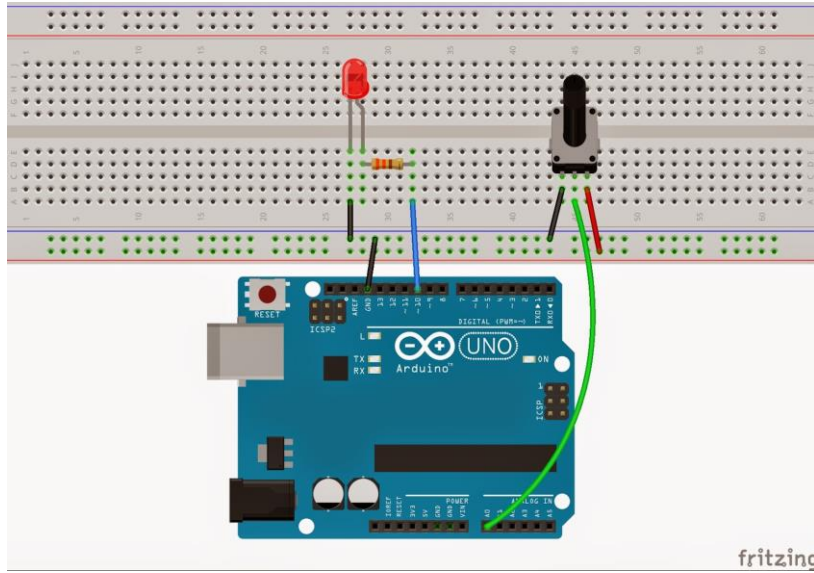
Potansiyometre ayarlı direnç ya da sürgülü reosta olarak bilinen bir devre elemanıdır. İç yapısında bir direnç malzemesi bulunur. Bu sayede istenilen seviyeye getirilebilir ve bize sinyal değerleri verebilir.

Bağlantısı; A Bacağı 5V, B Bacağı GND, C Bacağı Analog Sinyal

Analog pinlere bağlanan C Bacağından 0-1023 arasında değerler elde edilir.

Devre Şeması:

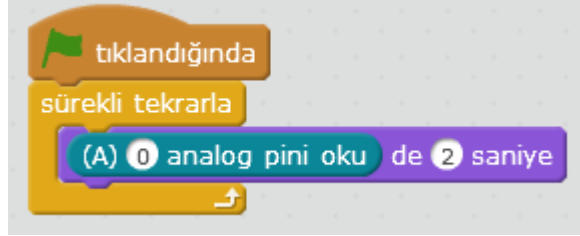
Öncelikle aşağıda gösterilen devre şemasına uygun olarak devremizin kurulumunu yapıyoruz.



Uygulama:

- Devre kurulumu bittikten sonra mBlock programını açabiliriz. Eğer program açıksa üst kısımda bulunan “Dosya” sekmesinden “Yeni” kısmına tıklayarak yeni bir proje açabiliriz.
- Arduino’yu USB kablo ile bilgisayara bağlıyoruz ve “Kartlar” sekmesinden “Arduino Uno”, “Uzantılar” sekmesinden ise “Arduino” seçeneğini seçiyoruz. “Bağlan” sekmesinden “Seri Port” kısmına tıklıyoruz ve açılan kısımdan seri portumuzu seçiyoruz. Son olarak da “Bağlan” sekmesinden “Aygıt Yazılımı Güncellemesi” kısmına tıklayarak Arduino’yu kullanıma hazırlamış oluyoruz.
- Öncelikle potansiyometreden gelen değerleri bize gösteren kodları yazmaya başlayabiliriz.
- İlk olarak potansiyometrenin değerlerini ekrana yazdırabilmek için “Görünüm” dizisinden “merhaba de 2 saniye” kod bloğunu alarak merhaba yerine, “Robotlar” dizisinden “A0 analog pini oku” bloğunu koyuyoruz. Böylece ekranımızda potansiyometreden gelen değerleri görebiliriz.

Gerekli kod bloklarını aşağıda detaylı olarak görebilirsiniz.



- Daha sonra potansiyometre ile LED in parlaklığını ayarlayacağımız kod bloklarını yazmaya geçebiliriz.
- İlk olarak potansiyometreden gelen değerleri tutması için bir değişken oluşturacağız. Bunun için "Veri&Blok" dizisinde bulunan "Bir Değişken Oluştur" butonuna basarak "pot" isminde yeni bir değişken oluşturuyoruz.
- Bu değişkeninin içerisine potansiyometreden gelen değerleri aktarıyoruz. Ama öncelikle bir işlem içerisine sokmamız gerekiyor. Potansiyometre bize normal hali ile 0 ile 1023 arasında değerler verebilmektedir. Fakat LED' imiz "pwm" pine bağlı olduğu için 0 ile 255 arasında değerler alabilmektedir. Bu nedenle potansiyometrenin değer aralığını LED' imiz ile aynı seviyeye çekebilmek için, potansiyometreden gelen değeri "4" e bölüyoruz ve bu şekilde değişkenimizin içerisine koyuyoruz.
- Son olarak da LED' i taktığımız pwm pini çalıştırmak için "Robotlar" dizisinden "5 pwm pini 0 yap" bloğunu alarak, öncelikle taktığımız pini "10" değerimizi de potansiyometreden gelen değerleri alabilmesi için "0" yerine "pot" değişkenini yerleştiriyoruz.

Gerekli kod bloklarını aşağıda detaylı olarak görebilirsiniz.



- Artık projemizi çalıştırabiliriz. Bu şekilde potansiyometre ile ledin parlaklığının ayarını yapmış oluyoruz.

Buzzer ile Ses Çıkartma

Öğrenilen Kavramlar

- Arduino kart
- Breadboard
- Olayı başlatma
- Buzzer
- Döngüler

Gerekli Malzemeler

- Bilgisayar
- mBlock 3 programı
- Arduino UNO
- Breadboard
- USB kablo
- Buzzer
- Direnç
- Jumper Kablo

Amaç

- Kodları kullanarak günlük hayatta kullanılan sirenlerin bir benzerini yapmak

Kazanım

- Günlük hayattaki kullanılan sistemlerin içeriğini görür.
- Kod kullanarak hazırlanan devreyi çalıştırmayı öğrenir.

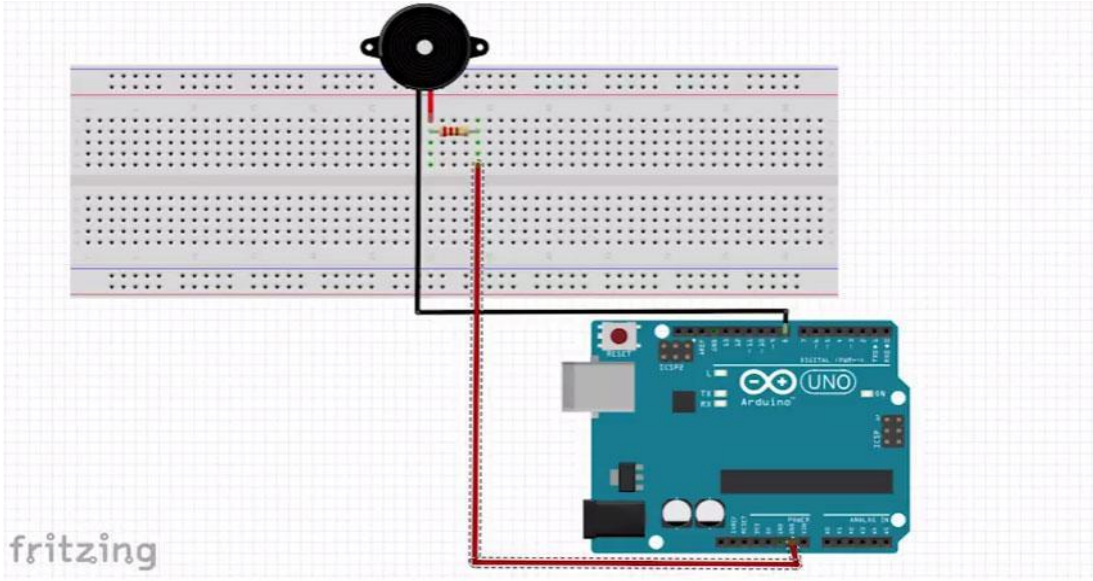


Buzzer; projelerimizde uyarı sesleri çıkarabilmek amacı ile kullanılan mini hoparlördür. 2 çeşit buzzer vardır. Bunlar, aktif ve pasif buzzerlardır. Aktif buzzer istenilen notaya göre ses çıkarabilir ve programla çeşitli müziklerin yapılması mümkün olan buzzerlardır. Pasif buzzer ise, sadece tek bir tonda ses çıkabilen ve sadece uyarı amacı ile kullanılan buzzerlardır.

Biz bu projemizde farklı notalar kullanacağımız için buzzerin aktif buzzer olmasına dikkat etmeliyiz.

Devre Şeması:

Öncelikle aşağıda gösterilen devre şemasına uygun olarak devremizin kurulumunu yapıyoruz.



Uygulama:

- Devre kurulumu bittikten sonra mBlock programını açabiliriz. Eğer program açıksa üst kısımda bulunan "Dosya" sekmesinden "Yeni" kısmına tıklayarak yeni bir proje açabiliriz.
- Arduino'yu USB kablo ile bilgisayara bağlıyoruz ve "Kartlar" sekmesinden "Arduino Uno", "Uzantılar" sekmesinden ise "Arduino" seçeneğini seçiyoruz. "Bağlan" sekmesinden "Seri Port" kısmına tıklıyoruz ve açılan kısımdan seri portumuzu seçiyoruz. Son olarak da "Bağlan" sekmesinden "Aygıt Yazılımı Güncellemesi" kısmına tıklayarak Arduino'yu kullanıma hazırlamış oluyoruz.
- Öncelikle buzzerden bir ses çıkmayı sağlıyoruz.

mBlock programımızda bulunan notalarımız harfler ile belirtilmektedir. Bunların dönüşümleri aşağıda gösterilmektedir.

Türk Müziğinde	La	Si	Do	Re	Mi	Fa	Sol
Avrupa müziğinde	A	B	C	D	E	F	G

- İlk ses çıkartma denemesi için “Robotlar” dizisinden “9 ses tonu pini C4 notasında Yarım vuruş çal” şeklindeki kod bloğunu alıyoruz. Buradaki “9” buzzer takmış olduğumuz pin “C4” nota ve “Yarım” da ne kadar süre ses çıkaracağı şeklindedir. İlk olarak bir ses çıkarıyoruz. Gerekli kod bloklarını aşağıda detaylı olarak görebilirsiniz.

```
Arduino Programı
sürekli tekrarla
9 ses tonu pini D2 notasında Yarım vuruş çal
1 saniye bekle
9 ses tonu pini D2 notasında Sifir vuruş çal
1 saniye bekle
```

- Daha sonra bütün notaların seslerini duyacağımız bir uygulama oluşturuyoruz. Gerekli kod bloklarını aşağıda detaylı olarak görebilirsiniz.

```
Arduino Programı
sürekli tekrarla
9 ses tonu pini C2 notasında Yarım vuruş çal
0.5 saniye bekle
9 ses tonu pini D2 notasında Yarım vuruş çal
0.5 saniye bekle
9 ses tonu pini E2 notasında Yarım vuruş çal
0.5 saniye bekle
9 ses tonu pini F2 notasında Yarım vuruş çal
0.5 saniye bekle
9 ses tonu pini G2 notasında Yarım vuruş çal
0.5 saniye bekle
9 ses tonu pini A2 notasında Yarım vuruş çal
0.5 saniye bekle
9 ses tonu pini B2 notasında Yarım vuruş çal
0.5 saniye bekle
```

- Artık istediğimiz müzikleri oluşturmanız mümkün :)

LDR ile Gece Lambası Yapımı

Öğrenilen Kavramlar

- Arduino kart
- Breadboard
- Diziler
- Bloklar
- LDR
- Döngüler
- Koşullar
- Değişken
- Analog pin

Gerekli Malzemeler

- Bilgisayar
- mBlock 3 programı
- Arduino UNO
- Breadboard
- USB kablo
- LED
- LDR LED
- Direnç
- Jumper Kablo

Amaç

- LDR ışık şiddetini ölçme işlemini öğrenmek
- Sokak lambalarının çalışma mantığını öğrenmek

Kazanım

- LDR kavramını, döngüler, koşullar ve değişken konularını öğrenir.
- Işık şiddetine göre LED'i yakıp söndürür.
- Günlük hayat ile bağlantı kurar.

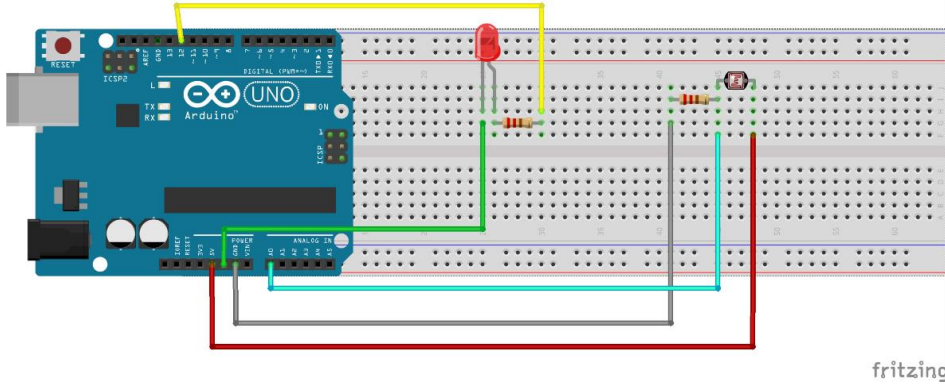


Işığa bağlı olarak değerleri değişen dirençlere fotodirençler denilmektedir. LDR(Light Dependent Resistor)'nin üzerine düşen ışık şiddeti ile biz verdiği direnç değeri ters orantılıdır. Yani ışık şiddeti arttığında direnç düşer, azaldığında ise direnç artar.

LDR, ışık ile kontrol gerektiren projelerde ve sistemlerde basitçe kullanılabilen bir sensördür.

Devre Şeması:

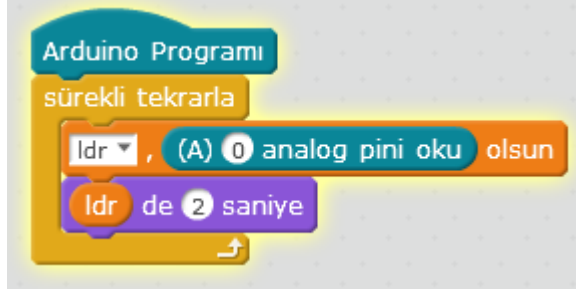
Öncelikle aşağıda gösterilen devre şemasına uygun olarak devremizin kurulmasını yapıyoruz.



Uygulama:

- Devre kurulumu bittikten sonra mBlock programını açabiliriz. Eğer program açıksa üst kısımda bulunan "Dosya" sekmesinden "Yeni" kısmına tıklayarak yeni bir proje açabiliriz.
- Arduino'yu USB kablo ile bilgisayara bağlıyoruz ve "Kartlar" sekmesinden "Arduino Uno", "Uzantılar" sekmesinden ise "Arduino" seçeneğini seçiyoruz. "Bağlan" sekmesinden "Seri Port" kısmına tıklıyoruz ve açılan kısımdan seri portumuzu seçiyoruz. Son olarak da "Bağlan" sekmesinden "Aygıt Yazılımı Güncellemesi" kısmına tıklayarak Arduino'yu kullanıma hazırlamış oluyoruz.
- Öncelikle sensörden gelen değerleri ekrana yazdıracakız. Sensörden gelen değerleri tutması için bir değişken oluşturacağız. Bunun için "Veri&Blok" dizisinde bulunan "Bir Değişken Oluştur" butonuna basarak "ldr" isminde yeni bir değişken oluşturuyoruz.
- Daha sonra, bu değişkenin içerisine LDR'den gelen değerleri "Robotlar" dizisinden "A0 analog pini oku" bloğunu koyarak aktarıyoruz. "Görünüm" dizisinden "merhaba de 2 saniye" kod bloğunu alarak merhaba yerine, oluşturduğumuz "ldr" değişkenini koyuyoruz ve böylece ekranımızda sensörden gelen değerleri görebiliriz.

Gerekli kod bloklarını aşağıda detaylı olarak görebilirsiniz.



- Bu kodumuz ile sensörün karanlıkta ve aydınlıkta verdiği değerleri belirleyebiliriz.
- Değer aralıklarını belirledikten sonra bu değerlere göre bir şart oluşturuyoruz. Karanlıkta yani sensörün üzeri kapalı iken verdiği değer bizim sensörümüz için bu 100 den küçük değerler arasında değişiyordu. Bu nedenle gelen değer 100 'den küçükse ($ldr < 100$) ışığı yani LED i yakacağız.
- Kodumuzu oluşturmak için öncelikle "Kontrol" dizisinden "eğer ise _ değilse" kod bloğunu alarak içerisine, "İşlemler" dizisinden "<" (küçüktür) bloğu koyuyoruz ve şartımızı oluşturuyoruz. Böylece şartımız doğru olduğunda ışığımızın yanması için "Robotlar" dizisinden "9 sayısal pini YÜKSEK" yap kod bloğunu alarak, "9" yazan yere LED imizi taktığımız pini yazıyoruz. Değilse kısmında ise aynı kod bloğunu "DÜŞÜK" olarak değiştiriyoruz.

Gerekli kod bloklarını aşağıda detaylı olarak görebilirsiniz.



- Böylece sokak lambalarında da olduğu gibi karanlıkta otomatik olarak yanan ve ışıklı ortamda sönen bir ışık sistemi oluşturmuş oluruz.

Mesafe Ölçer Yapımı

Öğrenilen Kavramlar

- Arduino kart
- Breadboard
- Mesafe sensörü
- Döngüler
- Değişken
- Mantıksal operatörler

Gerekli Malzemeler

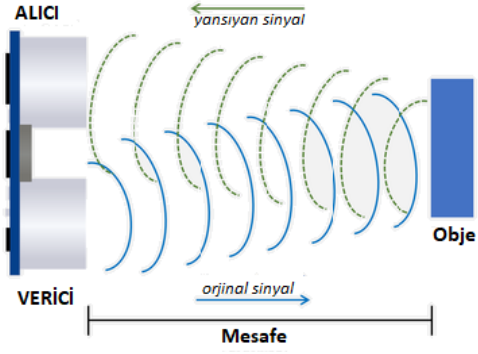
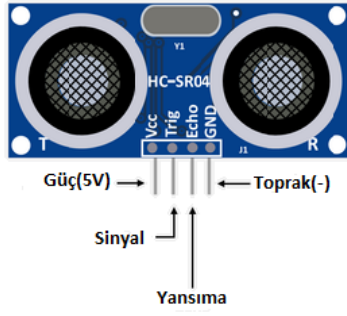
- Bilgisayar
- mBlock 3 programı
- Ultrasonik Sensör
- Arduino UNO
- Breadboard
- Jumper kablolar
- USB kablo

Amaç

- Sensör kullanımını öğrenmek
- Sensör yardımı ile nesnelere arasındaki mesafeyi ölçmek

Kazanım

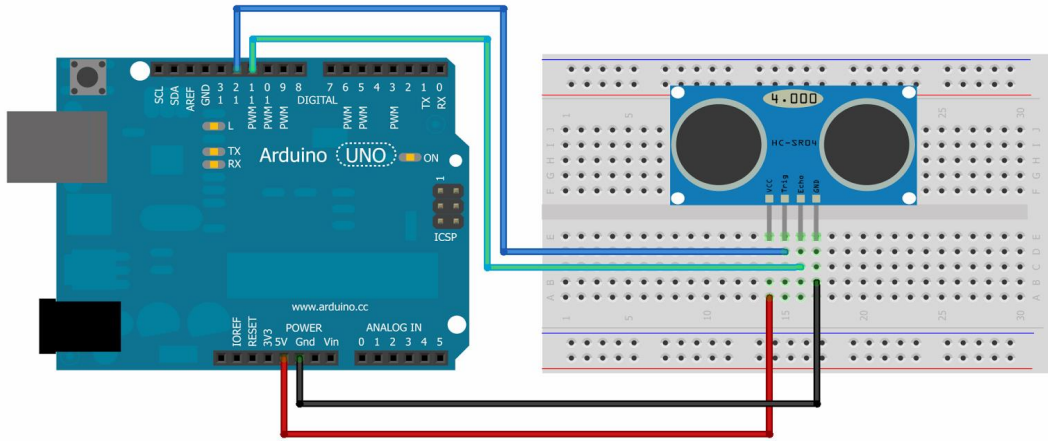
- Teknoloji sayesinde nesnelere arası mesafeyi nasıl ölçebileceğini öğrenir.
- Dijital bir metre yapmış olur.
- Sensör kavramını öğrenir.



Ultrasonik ses dalgaları, insan kulağının duymayacağı frekans bandına sahiptir. Bu ses dalgalarını kullanan sensör ile nesnelere herhangi bir temas sağlamadan mesafe ölçümleri yapabiliriz. Bunun için, hareket problemlerinde olduğu gibi “ $X(\text{yol})=V(\text{hız}) \cdot t(\text{zaman})$ ” formülüne göre çalışmaktadır. Ses dalgalarının belirli koşullardaki hızının biliniyor olmasından faydalanarak, ses dalgasının gidip, daha sonra geri gelmesi arasında geçen sürenin yarısı ile hızının çarpımı sonucu bize mesafeyi vermektedir.

Devre Şeması:

Öncelikle aşağıda gösterilen devre şemasına uygun olarak devremizin kurulmasını yapıyoruz.



Uygulama:

- Devre kurulumu bittikten sonra mBlock programını açabiliriz. Eğer program açıksa üst kısımda bulunan “Dosya” sekmesinden “Yeni” kısmına tıklayarak yeni bir proje açabiliriz.
- Arduino’yu USB kablo ile bilgisayara bağlıyoruz ve “Kartlar” sekmesinden “Arduino Uno”, “Uzantılar” sekmesinden ise “Arduino” seçeneğini seçiyoruz. “Bağlan” sekmesinden “Seri Port” kısmına tıklıyoruz ve açılan kısımdan seri portumuzu seçiyoruz. Son olarak da “Bağlan” sekmesinden “Aygıt Yazılımı Güncellemesi” kısmına tıklayarak Arduino’yu kullanıma hazırlamış oluyoruz.
- Böylece kodlarımızı yazmaya başlayabiliriz.
- Öncelikle mesafe sensöründen gelen değerleri okuyabilmek için gereken kod bloklarını yazarak başlıyoruz. “Kontrol” dizisinden “sürekli tekrarla” içerisine, “Görünüm” dizisinden “merhaba de 2 saniye” alıyoruz. “merhaba de 2 saniye” içerisindeki “merhaba” kısmına “Robotlar” dizisinden “ultrasonik _ tetik pini _ okuma pini” (tetik pini - trig, okuma pini - echo pinlerini ifade etmektedir.) kod bloğunu ekliyoruz. Böylece kodumuzu çalıştırarak sensörden gelen değeri artık mBlock ekranımızda görebiliriz.

Gerekli kod bloklarını aşağıda detaylı olarak görebilirsiniz.



- Daha sonra sensörden gelen değerimizi bir değişkene aktaracağız. Bunun içinde “Veri&Blok” dizisi bölümünde bulunan “Bir Değişken Oluştur” butonuna basarak “mesafe” isiminde yeni bir değişken oluşturuyoruz. Bu alanda oluşan kod bloklarından “mesafe 0 olsun” u alarak “0” içerisine “İşlemler” dizisinden “_ i yuvarla” kod bloğu alıyoruz ve son olarak bunun içerisine “Robotlar” dizisinden “ultrasonik _ tetik pini _ okuma pini” kod bloğunu ekliyoruz. Böylece sensörden gelen değeri “mesafe” değişkenine yuvarlayarak yani tam sayılar şeklinde görülebilecek şekilde atamış oluyoruz.

Gerekli kod bloklarını aşağıda detaylı olarak görebilirsiniz.



- Son olarak da mesafe azaldığında çalacak bir alarm sistemi ekleyebiliriz.
- Ayrıca alarm haberi geldiğinde arka plan rengini değiştirerek farklı geliştirmeler ekleyebiliriz. Bunun için öncelikle “işlemler” dizisinden “<” (küçüktür) simgesini alarak bir tarafına “Veri&Blok” dizisinden “mesafe” değişkeni koyuyoruz ve diğer tarafına da mesafemiz kaçtan küçük olduğunda alarmın çalmasını istiyor isek o değeri yazıyoruz. Bu oluşturduğumuz bloğu bir şart “Eğer” kod bloğuna yerleştirerek eğer mesafemiz belirttiğimiz değerden küçük ise alarm çalsın şeklinde uygulamalarımızı yapıyoruz.

Gerekli kod bloklarını aşağıda detaylı olarak görebilirsiniz.



- Bu kısmı da tamamladıktan sonra mesafeyi ölçerek, engel ya da sensörün önüne denk gelen belirttiğimiz değerden yakın herhangi bir şey olduğunda alarm çalan bir sisteme sahip olduk.

Park Sensörü Yapımı

Öğrenilen Kavramlar

- Arduino kart
- Breadboard
- Mesafe sensörü
- Buzzer
- Döngüler
- Değişken
- Mantıksal operatörler
- Koşullar

Gerekli Malzemeler

- Bilgisayar
- mBlock 3 programı
- Ultrasonik Sensör
- Buzzer
- Arduino UNO
- Breadbard
- Jumper kablolar
- USB kablo

Amaç

- Sensörleri ve kodları kullanarak günlük hayatta kullanılan araçlardaki park sensörünün bir benzerini yapmak

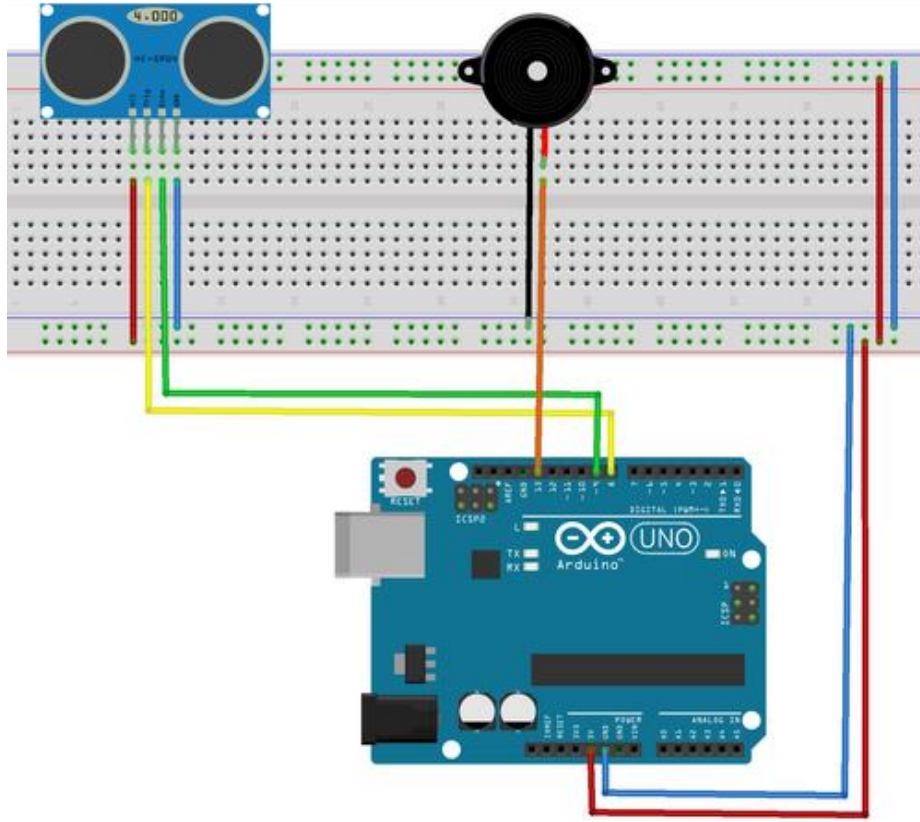
Kazanım

- Kodlama yardımıyla devreyi çalıştırmayı öğrenir.
- Kod döngülerini ve koşul ifadelerini kullanmayı öğrenir.
- Birden fazla devre elemanını beraber çalıştırmayı öğrenir.
- Sensör ve buzzer kavramları öğrenilir.
- Günlük hayattan bir örnek olduğundan elektronik kartları ve kodları günlük hayatımızda nasıl kullandığımız kavramı somutlaşmış olur.

Not: Park Sensörü uygulamasını yapmadan önce Mesafe Ölçer ve Buzzer uygulamalarını yapmanız konunun anlaşılması adına daha iyi olabilir.

Devre Şeması:

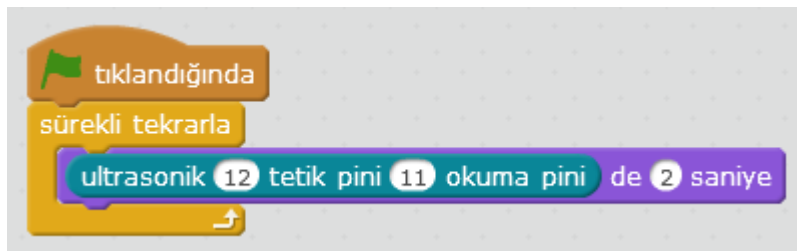
Öncelikle aşağıda gösterilen devre şemasına uygun olarak devremizin kurulumunu yapıyoruz.



Uygulama:

- Arduino'yu USB kablo ile bilgisayara bağlıyoruz ve "Kartlar" sekmesinden "Arduino Uno", "Uzantılar" sekmesinden ise "Arduino" seçeneğini seçiyoruz. "Bağlan" sekmesinden "Seri Port" kısmına tıklıyoruz ve açılan kısımdan seri portumuzu seçiyoruz. Son olarak da "Bağlan" sekmesinden "Aygıt Yazılımı Güncellemesi" kısmına tıklayarak Arduino'yu kullanıma hazırlamış oluyoruz.
- Böylece kodlarımızı yazmaya başlayabiliriz.
- Öncelikle mesafe sensöründen gelen değerleri okuyabilmek için gereken kod bloklarını yazarak başlıyoruz. "Kontrol" dizisinden "sürekli tekrarlar" içerisine, "Görünüm" dizisinden "merhaba de 2 saniye" alıyoruz. "merhaba de 2 saniye" içerisindeki "merhaba" kısmına "Robotlar" dizisinden "ultrasonik _ tetik pini _ okuma pini" (tetik pini - trig, okuma pini - echo pinlerini ifade etmektedir.) kod bloğunu ekliyoruz. Böylece kodumuzu çalıştırarak sensörümüzden gelen değeri artık mBlock ekranımızda görebiliriz.

Gerekli kod bloklarını aşağıda detaylı olarak görebilirsiniz.



- Daha sonra sensörden gelen değerimizi bir değişkene aktaracağız. Bunun içinde “Veri&Blok” dizisi bölümünde bulunan “Bir Değişken Oluştur” butonuna basarak “mesafe” isminde yeni bir değişken oluşturuyoruz. Bu alanda oluşan kod bloklarından “mesafe 0 olsun” u alarak “0” içerisine “İşlemler” dizisinden “_ i yuvarla” kod bloğu alıyoruz ve son olarak bunun içerisine “Robotlar” dizisinden “ultrasonik _ tetik pini _ okuma pini” kod bloğunu ekliyoruz. Böylece sensörden gelen değeri “mesafe” değişkenine yuvarlayarak yani tam sayılar şeklinde görülebilecek şekilde atamış oluyoruz.
- Son olarak da engel ile aramızda bulunan mesafe azaldığında çalacak alarm sistemimizi ekliyoruz. Bunun için öncelikle “işlemler” dizisinden “<” (küçüktür) simgesini alarak bir tarafına “Veri&Blok” dizisinden “mesafe” değişkeni koyuyoruz ve diğer tarafına da mesafemiz kaçtan küçük olduğunda alarmın çalmasını istiyor isek o değeri yazıyoruz. Bu oluşturduğumuz bloğu bir şart “Eğer” kod bloğuna yerleştirerek eğer mesafemiz belirttiğimiz değerden küçük ise buzzerdan alarm çalan uygulamalarımızı yapıyoruz.

Gerekli kod bloklarını aşağıda detaylı olarak görebilirsiniz.



- Projemizi farklı mesafe ve aralıklarda, farklı sesler olacak şekilde de geliştirmemiz mümkün. Bunun için “_ ses tonu pini _ notasında _ vuruş çal” bloğundan notayı değiştirebiliriz.
- Tüm adımları bitirdiğimizde park sensörümüz denemek için hazır.

8x8 Dot Ekran Kullanımı

Öğrenilen Kavramlar

- Arduino kart
- Diziler
- Bloklar
- Döngüler
- Değişkenler
- LED ekran
- Fonksiyonlar
- Arduino IDE Kütüphanesi

Gerekli Malzemeler

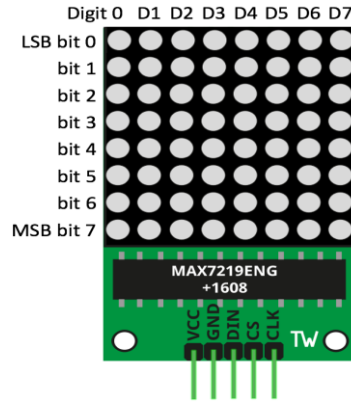
- Bilgisayar
- mBlock 3 programı
- 8x8 LED ekran
- Arduino UNO
- Breadboard
- Jumper kablolar
- USB kablo

Amaç

- LED ekranı çalıştırma

Kazanım

- LED ekran kavramını ve fonksiyonlar konusunu öğrenir.
- Kod döngülerini kullanmayı ve değişken oluşturmayı öğrenir.
- Arduino IDE' ye kütüphane yüklemeyi öğrenir.

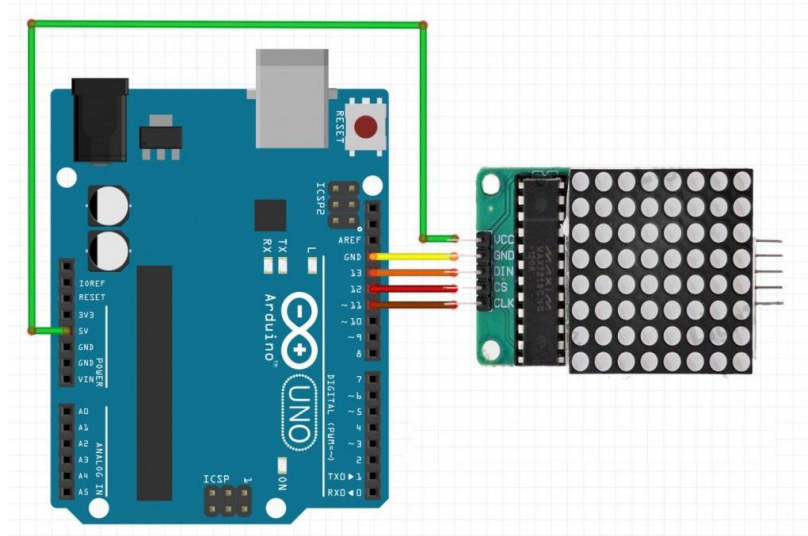


LED matrix içerisinde tek renkli ya da RGB LED ler kullanılarak 4x4 veya 8x8 şeklinde versiyonları vardır. Ayrıca birbirlerine bağlanarak daha büyük boyutları elde edilebilir. 8x8 ifadesi dikeyde 8 LED, düşeyde 8 LED şeklinde toplam 64 adet LED den oluşur.

LED matrix, kayan yazı devrelerinde ya da LED lerin entegreler ve ya mikro denetleyiciler ile sürülerek şekiller ve ifadeler oluşturulabilir.

Devre Şeması:

Öncelikle aşağıda gösterilen devre şemasına uygun olarak devremizin kurulumunu yapıyoruz.



Uygulama:

- Devre kurulumu bittikten sonra mBlock programını açabiliriz. Eğer program açıksa üst kısımda bulunan “Dosya” sekmesinden “Yeni” kısmına tıklayarak yeni bir proje açabiliriz.
- Arduino’yu USB kablo ile bilgisayara bağlıyoruz ve “Kartlar” sekmesinden “Arduino Uno”, “Uzantılar” sekmesinden ise “Arduino” seçeneğini seçiyoruz. “Bağlan” sekmesinden “Seri Port” kısmına tıklıyoruz ve açılan kısımdan seri portumuzu seçiyoruz. Son olarak da “Bağlan” sekmesinden “Aygıt Yazılımı Güncellemesi” kısmına tıklayarak Arduino’yu kullanıma hazırlamış oluyoruz.

- Böylece kodlarımızı yazmaya başlayabiliriz.
- İlk olarak yapılması gereken önemli kısım, LED ekranın çalışabilmesi için “Uzantılar” sekmesine tıklayarak “Uzantıları Yönet” kısmına girilir ve arama bölümüne “Dotmatrix TR” yazılır ve ekrana gelen uzantı indirilir. Daha sonra bu uzantı uzantılar sekmesinin altına görünür hale gelmiş olur ve buradan blokları görmek için tıklayarak görünür olmasını sağlayabiliriz.
- “Veri&Blok” dizisi bölümünde bulunan “Bir Değişken Oluştur” butonuna basarak “v” ve “h” isiminde yeni değişken oluşturuyoruz. Farklı olarak bu kısımda, “Veri&Blok” dizisi bölümünde bulunan “Bir Blok Oluştur” butonuna basarak “değer” isiminde bir yeni blok oluşturuyoruz.
- Oluşturduğumuz “değer” bloğu içerisinde oluşturduğumuz değişkenleri “0” olarak tanımlıyoruz.
- Sırası ile başlangıçta LED ekranımızı bağladığımız pinleri belirliyoruz, ekran parlaklığını ayarlıyoruz ve başlangıçta temizlenmesini sağlıyoruz.
- Daha sonra “Kontrol” dizisinden “sürekli tekrarlar” bloğunu alarak içerisinde sırası ile bütün 8x8 dot ekran üzerinde ışıkların tek tek sırası ile yanıp, daha sonra tek tek sırası ile sönmesini sağlayan kod bloklarını ekliyoruz.

Gerekli kod bloklarını aşağıda detaylı olarak görebilirsiniz.

```

Arduino Programı
LED panel pinleri: DIN 7 , CS 6 , CLK 5
LED Ekran parlaklığını ayarla 0.8
LED Ekranı Temizle
sürekli tekrarlar
değer
8 defa tekrarlar
8 defa tekrarlar
LED Ekran aç LED Satır h Sütun v
h 'i 1 arttır
1 saniye bekle
v 'i 1 arttır
h , 0 olsun
değer
8 defa tekrarlar
8 defa tekrarlar
LED Ekran kapat LED Satır h Sütun v
h 'i 1 arttır
1 saniye bekle
v 'i 1 arttır
h , 0 olsun

```

LCD Ekran Kullanımı

Öğrenilen Kavramlar

- Arduino kart
- Breadboard
- Diziler
- Bloklar
- Olayı başlatma
- Döngüler
- Değişken
- LCD ekran
- Potansiyometre

Gerekli Malzemeler

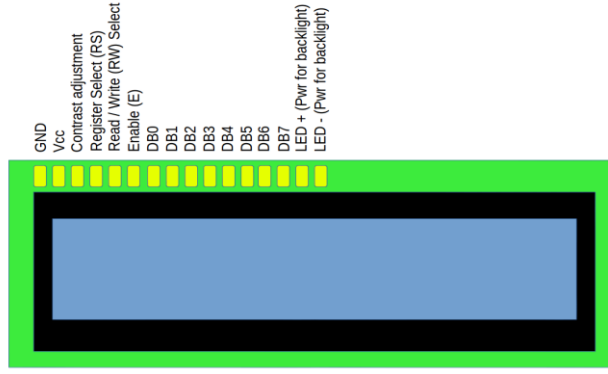
- Bilgisayar
- mBlock 3 programı
- Arduino UNO
- Breadboard
- LCD ekran
- Potansiyometre
- Jumper kablolar
- USB kablo

Amaç

- LCD ekranı çalıştırma
- Potansiyometre ile ekran parlaklığını arttırıp azaltma

Kazanım

- Elektronik devre elemanlarının çalışma mantığını kavrar.
- LCD ekran ve potansiyometre kullanmayı öğrenir.
- Kodlama yardımıyla devreyi çalıştırmayı öğrenir.
- Kod döngülerini kullanmayı öğrenir.
- Değişken oluşturmayı öğrenir.



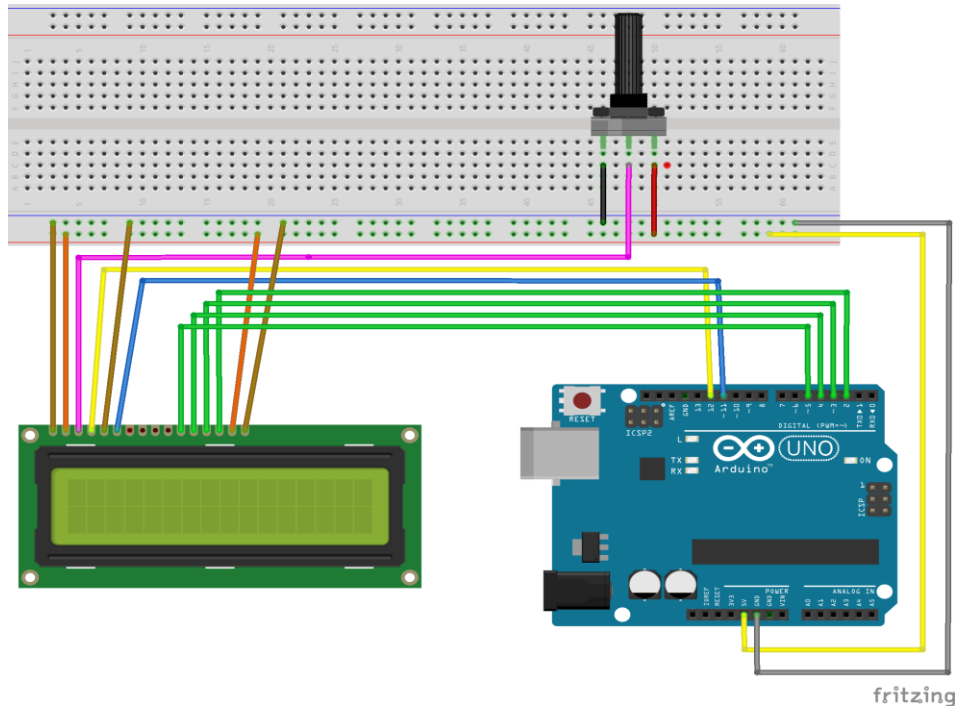
LCD ekranlar genellikle robotik projelerde ya da otomasyon projelerinde kullanılmak için Arduino ve birçok mikro denetleyici platformu ile kontrol edilebilmektedir. LCD paneller piyasada satır ve sütun sayılarına göre 1x8, 2x8, 1x16, 2x16, 1x20, 2x20, 1x40 ve 2x40 gibi farklı boyutlarda bulunmaktadır. Bunlardan genel olarak projelerinde 2x16 boyutlarındaki LCD paneller kullanılmaktadır. LCD ekranların çoğunda tek sıra halinde 16 pin bulunmaktadır. Bu pinlerden ilk 14 tanesi kontrol için, son 2 tanesi ise arka ışık için kullanılır.

Pinlerin sırası ile fonksiyonları;

- Vss : Toprak(Ground)
- Vcc : +5V
- VEE : Kontrast, girişine bağlanan direnç ile LCD panelin kontrastı ayarlanabilir.
- RS : Register Select, LCD ekrana komut mu yoksa veri mi gönderileceğini belirler.
- RW : Read/Write, LCD ekrana okuma mı yoksa yazmamı yapılacağını belirler.
- E : Enable, LCD ve pinler arasındaki gerçek veri alışverişini sağlayan
- D0 - D7 : Veri Girişleri
- BL+ : Arka panel ışığının pozitif ucu
- BL- : Arka panel ışığının negatif ucu

Devre Şeması:

Öncelikle aşağıda gösterilen devre şemasına uygun olarak devremizin kurulumunu yapıyoruz.



Uygulama:

- Devre kurulumu bittikten sonra mBlock programını açabiliriz. Eğer program açıksa üst kısımda bulunan "Dosya" sekmesinden "Yeni" kısmına tıklayarak yeni bir proje açabiliriz.
- Arduino'yu USB kablo ile bilgisayara bağlıyoruz ve "Kartlar" sekmesinden "Arduino Uno", "Uzantılar" sekmesinden ise "Arduino" seçeneğini seçiyoruz. "Bağlan" sekmesinden "Seri Port" kısmına tıklıyoruz ve açılan kısımdan seri portumuzu seçiyoruz. Son olarak da "Bağlan" sekmesinden "Aygıt Yazılımı Güncellemesi" kısmına tıklayarak Arduino'yu kullanıma hazırlamış oluyoruz.
- İlk olarak yapılması gereken önemli kısım, LED ekranın çalışabilmesi için "Uzantılar" sekmesine tıklayarak "Uzantıları Yönet" kısmına girilir ve arama bölümüne "LCD1602IIC" yazılır ve ekrana gelen uzantı indirilir. Daha sonra bu uzantı uzantılar sekmesinin altına görünür hale gelmiş olur ve buradan blokları görmek için tıklayarak görünür olmasını sağlayabiliriz.
- "Veri&Blok" dizisi bölümünde bulunan "Bir Değişken Oluştur" butonuna basarak "x" isiminde yeni bir değişken oluşturuyoruz.
- Oluşturduğumuz "x" değişkenini başlangıçta "0" olarak tanımlıyoruz.
- Sırası ile başlangıçta LCD ekranının boyutunu belirliyoruz, temizlenmesini sağlıyoruz, ilk satır ve sütuna "x=" yazmasını sağlıyoruz ve son olarak da 1.satır 5.sütun a "x" değişkeninden gelen değerleri yazmasını sağlıyoruz.
- "Kontrol" dizisinde bulunan "10 defa tekrarla" bloğunu alarak içerisine her defasında "x" değişkenin 1 artmasını sağlıyor ve ekranı silerek yeni değeri ekrana yazdırıyoruz. Böylece sırası ile 1 ile 10 arasında, 1 saniye aralıklarla ekrana sayılar yazmasını sağlamış oluruz.

Gerekli kod blokları aşağıda detaylı olarak gösterilmiştir.

```
Arduino Programı
x, 0 olsun
Clear LCD(16X2)
Clear LCD(16X2) and Line 1 Col 1 Show x=
Clear LCD(16X2) and Line 1 Col 5 Show x
1 saniye bekle
10 defa tekrarla
  x'i 1 arttır
  Clear LCD(16X2)
  Clear LCD(16X2) and Line 1 Col 1 Show x=
  Clear LCD(16X2) and Line 1 Col 5 Show x
  1 saniye bekle
```

Yağmur sensörlü Cam Sileceği Yapımı

Öğrenilen Kavramlar

- Arduino kart
- Breadboard
- Yağmur sensörü
- Servo motor
- Döngüler
- Koşullar
- Değişken

Gerekli Malzemeler

- Bilgisayar
- mBlock 3 programı
- Arduino UNO
- Servo motor
- Yağmur sensörü
- Jumper kablolar
- USB kablo

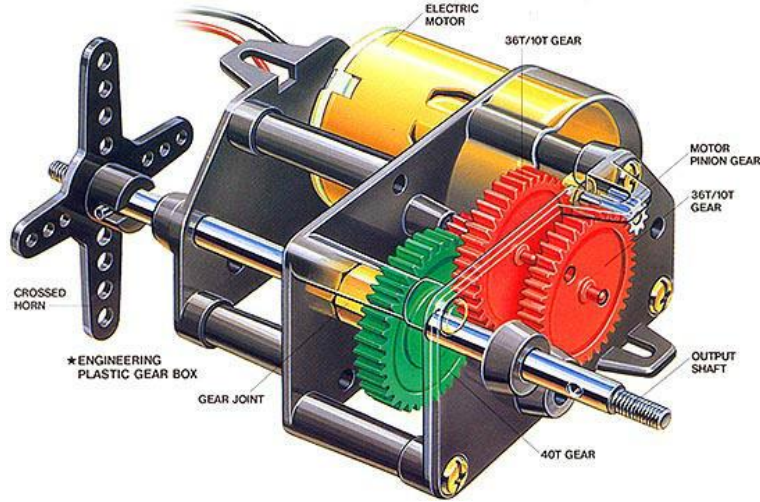
Amaç

- Araba sileceklerinin çalışma mantığını öğrenmek

Kazanım

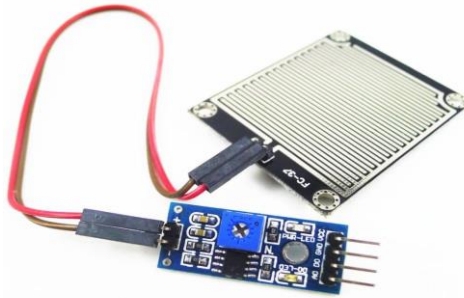
- Yeni sensörlerle tanışır.
- Servo motor kavramını öğrenir.
- Günlük hayat ile bağlantı kurar.

Bu uygulama ile arabalarda bulunan cam sileceklerinin çalışma mantığını anlayacağız ve bir otomatik cam sileceği nasıl olur, nasıl yapılır bunları göreceğiz. Cam silecekleri otomatik olarak yağmur yağdığında çalışmaya başlayacak, bunun için kullanacağımız yağmur sensörü ve sileceğimiz olarak çalışacak olan servo motorun kullanımını da öğrenmiş olacağız.



Servo motorların millerine bağlı olan bir enkoder bulunmaktadır. Bu enkoderler servo motorların içerisinde, istenilen yöne doğru hareket etmesini ya da belirli amaçlar ile motorun değişik yönlerde çalışmasını sağlayan ve sistemde bulunan verilerin düzenli şekilde çalışması için sinyal verilerini göndererek geri besleme işlemleri yapabilir.

Değişik türlerde üretimleri yapılmakta olan servo motorların genel olarak en çok tercih edilen, 0-180 derece açılımları arasında hareket sağlayan modellerdir.

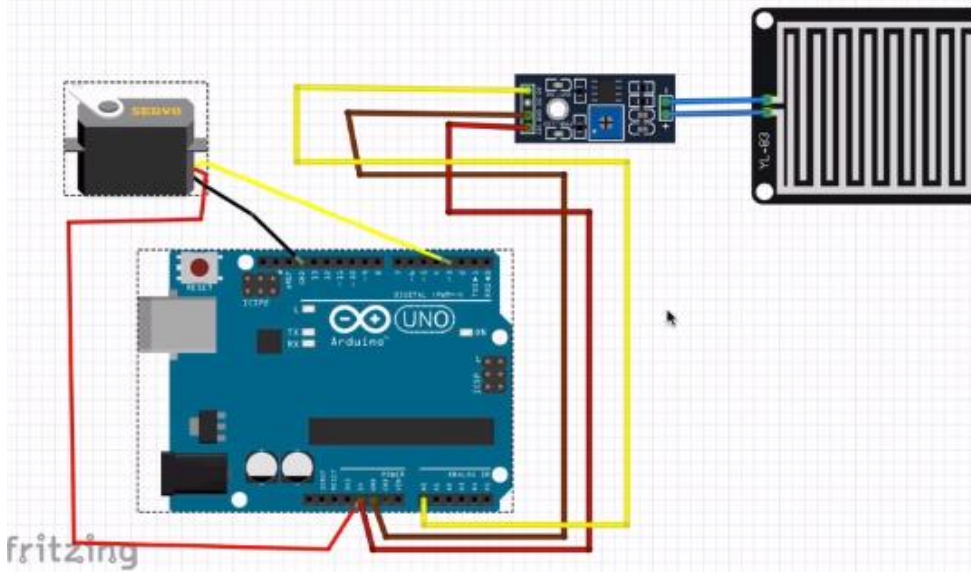


Birbirine paralel olarak çekilmiş olan iletken hatların su ile teması sonucunda sensörün çıkış pininden analog bir değer okuyabiliriz. Arduino ve birçok mikro denetleyici platformu ile kullanılabilir.

Sensörün kullanımı besleme voltajı ve toprak bağlantısı yapılarak sağlanır. Ayrıca sensör çıkış bacağından okuma yapılabilir. Hem analog, hem de dijital çıkışı bulunduğu için, farklı sistemlerde rahatlıkla kullanılabilir. Sensör üzerinde bulunan potansiyometre ile hassasiyet ayarları yapılabilir.

Devre Şeması:

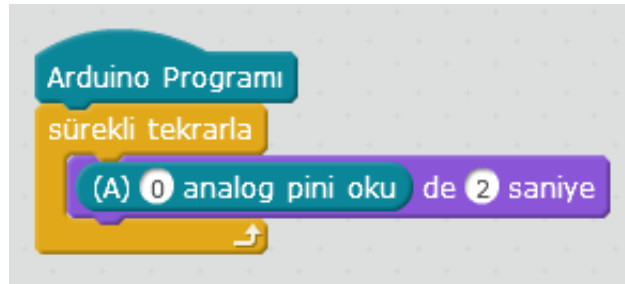
Öncelikle aşağıda gösterilen devre şemasına uygun olarak devremizin kurulumunu yapıyoruz.



Uygulama:

- Devre kurulumu bittikten sonra mBlock programını açabiliriz. Eğer program açıksa üst kısımda bulunan "Dosya" sekmesinden "Yeni" kısmına tıklayarak yeni bir proje açabiliriz.
- Arduino'yu USB kablo ile bilgisayara bağlıyoruz ve "Kartlar" sekmesinden "Arduino Uno", "Uzantılar" sekmesinden ise "Arduino" seçeneğini seçiyoruz. "Bağlan" sekmesinden "Seri Port" kısmına tıklıyoruz ve açılan kısımdan seri portumuzu seçiyoruz. Son olarak da "Bağlan" sekmesinden "Aygıt Yazılımı Güncellemesi" kısmına tıklayarak Arduino'yu kullanıma hazırlamış oluyoruz.
- Daha sonra yağmur sensörünün suyu algılayarak servo motoru çalıştırmasını sağlayacak kodları yazmaya başlayabiliriz.
- İlk olarak sensörden ıslak ve kuru ortamda gelen değerleri okumak için kısa bir kod yazıyoruz. Bu kod ile sahnedeki kuklanın olduğu kısımda konuşma balonunda değerler karşımıza çıkıyor olacak. Buradan ıslak ve kuru ortamdaki değerleri belirliyoruz.

Gerekli kod blokları aşağıda detaylı olarak gösterilmiştir.



- Sensörden gelen değerlerimizi belirledikten sonra üst kısımdaki kodu siliyoruz ve servo motorumuzu sensörden gelen değerlere göre hareket ettirebileceğimiz kodlarımızı yazıyoruz.
 - Kodumuz için öncelikle “Diziler”de bulunan “Veri&Blok” kısmından “yagmur” adında bir değişken oluşturuyoruz ve sensörden gelen değerleri bu değişkenin içerisine aktarıyoruz. Daha sonra sensörden gelen değerlere göre servo motorun hareketlerini belirleyecek olan koşul(eğer) komutlarımızı oluşturarak kodumuzu tamamlıyoruz.
 - Değişken oluştururken Türkçe karakter kullanılamaz.
- Gerekli kod blokları aşağıda detaylı olarak gösterilmiştir.

```

Arduino Programı
sürekli tekrarla
yagmur , (A) 0 analog pini oku olsun
eğer yagmur > 900 ise
  9 servo pini açısını 0 yap
eğer yagmur > 400 ise
  9 servo pini açısını 180 yap
  2 saniye bekle
eğer yagmur > 200 ise
  9 servo pini açısını 0 yap
  0.8 saniye bekle
  9 servo pini açısını 180 yap
  0.8 saniye bekle

```

- Kod yazma işlemi tamamladıktan sonra, sensörün üzerine su damlatarak kodların çalışıp çalışmadığını test edebiliriz.
 - Burada bilgisayar ve diğer elektronik materyallere su gelmemesi için dikkatli olunmalıdır.
- Test işlemlerini bitirdikten sonra, artık cam sileceğimiz kullanıma hazır hale gelmiş olur.