



Mekteb-i Lisey Kompleksi

Fatih MTAL

Mehmet Tuza Pakpen MTAL

Fatma Hacı Hüseyin Akgül MTAL

Gazi Anadolu Lisesi

Atakent Şehit Selçuk Paker Anadolu Lisesi

İçindekiler	
ARDUİNO VE TARİHÇESİ	4
NEDENARDUİNO?	5
EN ÇOK KULLANILAN ARDUİNO TÜRLERİNELERDİR?	5
▪ Arduino UNO	5
▪ Arduino NANO	5
▪ Arduino MEGA	6
BAŞLANGIÇ SEVİYESİ İÇİN ARDUİNO YARDIMCI EKİPMANLARI NELERDİR?	6
▪ Direnç	6
▪ Potansiyometre	6
▪ Jumper Kablo	6
▪ Kondansatör	6
▪ Led	6
▪ Role Modülü	6
▪ Buzzer	6
▪ 7 Segment Display	6
▪ Yağmur Sensörü	6
▪ 8*8 Matris Ekran	7
▪ Ateş Sensörü	7
▪ LDR Işık Sensörü	7
▪ Ultrasonik Mesafe Sensörü (HC-SR04)	7
▪ Gaz Sensörü	7
▪ Servo Motor	7
▪ DC Motor	7
▪ Bluetooth Modülü	7
▪ Wireless/Wifi Modülü	7
ARDUİNO TEMEL GİRİŞ/ÇIKIŞ FONKSİYONLARI	8
Pinleri Giriş Olarak Ayarlama (INPUT)	8
Dahili Pull-up Direnç Kullanılarak Ayarlanmış Pinler	8
Pinleri ÇIKIŞ Olarak Ayarlama (OUTPUT)	9
pinMode() Fonksiyonu Kullanımı	9
ARDUİNO'NUN TEXT TABANLI KODLANMASI	9
A. Blok ve Metin Tabanlı Kodlama	9
1. Blok Tabanlı Kodlama Nedir?	9
2. Metin Tabanlı Kodlama Nedir?	10
B. Arduino Temel Komutlar	11
1. void setup()	11
2. void loop()	11
3. pinMode(pin, mode)	11
4. digitalWrite(pin, value)	12
5. digitalRead(pin)	13
6. Arduino Gecikme Komutları	13
7. Serial.begin	14
8. Serial.print	14
9. Serial.read	14
10. Arduino'da if else kullanımı	14
11. Arduino'da for kullanımı	14
12. Random	15
13. While döngüsü	15

14. Arduino'nun Kurulumu ve Arduino Programı	16
Tools menüsü:	16
Program compile.....	16
Program Compile & Upload.....	16
Serial Monitor.....	16
Tinkercad Circuits (Devreler)	17
Sisteme Üye Olma ve Giriş.....	17
Kişisel Hesap Oluştur.....	19
Kayıt Aşaması.....	19
TINKERCAD İLE ADIM ADIM DEVRE OLUŞTURMA	21
ARDUINO/TINKERCAD UYGULAMA OLUŞTURMA	25
ARDUINO ile TINKERCAD PROJEMİZDE YAPTIĞIMIZ UYGULAMALARIMIZ	26
▪ BUTON LED UYGULAMASI	26
▪ TRAFİK IŞIKLARI UYGULAMASI	26
▪ KARAŞİMŞEK UYGULAMASI	27
▪ LDR ile LED YAKMA:	29
▪ POTANSİYOMETRE İLE SERVO MOTOR KONTROLÜ	30
▪ POTANSİYOMETRE/SERİ MONİTÖRDE DEĞER OKUMA	30
▪ L293D MOTOR SÜRÜCÜ ENTEGRESİ KULLANMA	31
▪ POTANSİYOMETRE İLE DC MOTOR HIZ KONTROLÜ-1	32
▪ POTANSİYOMETRE İLE DC MOTOR HIZ KONTROLÜ-2	33
▪ LDR İLE BUZZER UYGULAMASI	34
▪ ULTRASONİK MESAFE SENSÖRÜ VE SESLİ UYARILI PARK SİSTEMİ	34
ÖRNEK UYGULAMALAR	35
1. Basit Radar Yapımı	35
2. Uzunlukölçer Yapımı	37
3. Hırsız Alarmı	39
KAYNAKÇA	41
PROJE TAKIMLARI GÖREV DAĞILIMI	41

ARDUINO VE TARİHÇESİ



Arduino'yu icat eden kurucu 4 ortak tarafından kurulmuştur. Bu ortaklar ise başta **Massimo Banzi** olmak üzere **David Cuartielles**, **Tom Igoe** ve **David Mellis**'tir. Arduino'nun gelişimi İtalya'da bulunan **Etkileşim Tasarımı Enstitüsü İvrea**' da başlamıştır. **Arduino** günümüzde sıkça tasarlanan teknolojik aletlerin (El Yapımı Robot,Uçak ve Dron) tasarımların da beyin görevi üstlenir.

Arduino Ne Demek ve Ne Anlama Gelir? diye merak ediyorsanız; **Arduino** Türkçe kelime anlamı, manası ise **“Güvenilir Dost, Arkadaş”** anlamına gelmektedir. Güvене biliceğiniz bir tasarıma sahip olmasından dolayı arduino, yaptırmak ve uygulamak istediğiniz bütün komutları eksiksik bir şekilde yerine getirecektir. **Arduino** ile birçok proje yapılabilir. Tek ihtiyacımız olan kodlama ve programlamadır. Arduino sayesinde insanların yararına kullanılan birbirinden güzel teknolojik aletler insan yaşamını rahatlatmaktadır.

Arduino, elektronik projeleri oluşturmak için kullanılan açık kaynaklı bir platformdur. Arduino, hem fiziksel olarak programlanabilir bir devre kartından (genellikle mikrodenetleyici olarak anılır) hem de bir yazılım parçasından veya bilgisayarınızda çalışan, bilgisayar kodunu fiziksel karta yazmak ve yüklemek için kullanılan IDE'den (Integrated Development Environment – Entegre Geliştirme Ortamı) oluşur.

Arduino platformu, yeni başlayanlar arasında oldukça popülerdir. Çünkü önceki programlanabilir devre kartlarının çoğundan farklı olarak, Arduino'nun karta yeni kod yüklemek için ayrı bir donanım parçasına (programlayıcı adı verilir) ihtiyacı yoktur.



NEDEN ARDUINO?

Basit ve erişilebilir kullanıcı deneyimi sayesinde binlerce farklı proje ve uygulamada kullanılan Arduino'nun, yeni başlayanlar için kullanımı kolaydır, ancak ileri düzey kullanıcıların da yararlanabileceği kadar esnektir. Mac, Windows ve Linux üzerinde çalışır. Öğretmenler ve öğrenciler Arduino'yu düşük maliyetli bilimsel araçlar oluşturmak, kimya ve fizik ilkelerini kanıtlamak veya programlama ve robotik ile başlamak için kullanır.

Arduino, yeni şeyler öğrenmek için önemli bir araçtır. Herkes (çocuklar, sanatçılar, programcılar) kitin adım adım talimatlarını izleyerek veya Arduino topluluğunun diğer üyeleriyle çevrimiçi olarak fikirlerini paylaşarak Arduino'yu kullanmaya başlayabilir.

Arduino mikrodenetleyicilerle çalışma sürecini basitleştirirken; öğretmenler, öğrenciler ve ilgili amatörler için diğer sistemlere göre bazı avantajlar sağlar. Bunları inceleyecek olursak:

1. Ucuzdur

Arduino kartları, diğer mikrodenetleyici platformlarına kıyasla nispeten ucuzdur.

2. Çapraz Platform

Arduino Yazılımı (IDE), Windows, Macintosh OSX ve Linux işletim sistemlerinde çalışır.

3. Basit,Net Programlama Ortamı

Arduino Yazılımı'nın (IDE), yeni başlayanlar için kullanımı kolaydır, ancak ileri düzey kullanıcıların da yararlanabileceği kadar esnektir. Öğretmenler için, uygun bir şekilde Processing programlama ortamına dayanır. Bu nedenle bu ortamda programlamayı öğrenen öğrenciler, Arduino IDE'nin nasıl çalıştığına aşina olur.

4. Açık Kaynak ve Genişletilebilir Yazılım

Arduino yazılımı, deneyimli programcılar tarafından genişletilebilen açık kaynaklı araçlar olarak yayınlanır.

5. Açık Kaynak ve Genişletilebilir Donanım

Arduino kartlarının planları Creative Commons lisansı altında yayınlanır, böylece deneyimli devre tasarımcıları modülün kendi versiyonlarını oluşturarak onu genişletebilir ve geliştirebilir.

EN ÇOK KULLANILAN ARDUINO TÜRLERİ NELERDİR?

▪ Arduino UNO

Program kodlarının çoğunu saklamak için 32Kb hafızanın yanı sıra 16 tip dijital giriş ve çıkışa sahiptir.

▪ Arduino NANO

UNO'nun en küçük versiyonu olarak kabul edilir. Sınırlı alana sahip projeler için mükemmeldir.

- **Arduino MEGA**

UNO'nun bir üst modeli olarak bilinir, çünkü onunla tamamen uyumludur. Kalkanları veya genişletmeleri bağlamak için pimler aynı konumuna sahiptir.

BAŞLANGIÇ SEVİYESİ İÇİN ARDUINO YARDIMCI EKİPMANLARI NELERDİR?

- **Direnç:**

Devreden geçen akımı belli bir seviyeye ayarlar sınırlar ve sabit tutar. Besleme gerilimi küçültüp ayırarak diğer devre elemanlarında beslemeyi kolaylaştırır. Hassas devre elemanlarını düzenli çalışmasını sağlar. Yük görevi yapar. Isı enerjisi üretimi içinde kullanılabilir.

- **Potansiyometre:**

Potansiyometre, dışardan fiziksel bir etkiyle iç direncini arttırıp azaltabilen devre elemanıdır. Sadece Arduino da değil çoğu elektronik uygulamada analog okuma, pwm gibi işlemlerde sensörlerden sonra en çok kullanılan elektronik malzemedir.

- **Jumper Kablo:**

Bu tarz bağlantı kabloları, devre tahtası ve Arduino gibi geliştirme kartlarının bir arada kullanıldığı devreler için oldukça uygundur. Uçlarında dişi ve erkek girişlerin olduğu üç çeşidi bulunmaktadır. Erkek-erkek Erkek-dişi ve Dişi-dişi Projelerinizin boyutuna bağlı olarak birbirlerine bağlayıp bağlantı kablonuzun boyunu uzatabilirsiniz.

- **Kondansatör:**

Kondansatör doğru akımı (DC) iletmeyip, alternatif akımı (AC) iletme özelliğine sahiptir. Bu özellikleri sayesinde çoğu devrede farklı amaçlar ile kullanılırlar. Güç kaynağı devrelerinde filtrelemede, rezonans devrelerinde istenilen frekansı üretmede ve güç aktarım hatlarında gerilim düzenlenmesi ve güç akışının kontrolünde kullanılırlar.

- **Led:**

LED, ışık yayan diyottur. İngilizce ismi "Light Emitting Diode" olan LED'in ismi bu kelimelerin baş harflerinden oluşur. Diyot, akımın yalnızca tek yönden geçmesini sağlayan iki bacaklı yarı-iletken bir devre elemanıdır.

Başta olmak üzere çeşitli yardımcı ekipmanlar kullanılmaktadır.

- **Role Modülü:**

Arduino modül çeşidi olan röle modülü, düşük akımlar kullanarak yüksek akım çeken cihazları anahtarlamada kullanılan devre elemanıdır. Oldukça basit bir çalışma prensibine sahiptirler: rölenin bobinine enerji verildiğinde mıknatıslanan bobin bir armatürü hareket ettirerek kontakların birbirine temasını sağlar ve devrede iletim sağlanmış olur.

- **Buzzer:**

Arduino modülleri içerisinde buzzer verilen voltaja göre farklı ses sinyalleri sağlayan bir cihazdır. Maliyetleri az, üretimi basit ve çok hafif yapıda olmalarından dolayı kullanım alanı çok geniştir. Hırsız alarmları, araçlarda uyarı veren çoğu sistem, bazı zil sesleri, kısaca uyarı almak, korunmak, ayırım yapmak amaçlı her yerde kullanılabilecek bir cihazdır.

- **7 Segment Display:**

İçerisinde 7 adet LED bulunduran ve 0 ile 9 arasındaki rakamları üzerinde görüntüleyebilen arduino modülü 7 segment display, sayısal gösterge elemanıdır. 7 segment ekran; klima, duvar saati, sıcaklık göstergesi gibi birçok alanda kullanılır.

- **Yağmur Sensörü:**

Birbirine paralel olarak çekilmiş iletken hatların su ile teması sonucu sensör çıkış pininde bir değer alınmaktadır. Besleme voltajı ve toprak bağlantısı yapılarak, sensör çıkış bacağından okuma yapılabilmektedir. Dijital ve analog çıkış verdiğinden dolayı, farklı sistemlere rahatlıkla uyarlanabilir. Sensör üzerindeki pot yardımı ile de sensör hassasiyeti ayarlanabilmektedir.

- **8*8 Matris Ekran:**

Arduino modülü Led Matris çok kullanışlı bir çıkış elemanıdır, standart bildiğimiz 7 segment ekran gibi kısıtlı değil daha geniş bir görüntüleme alanı sağlar bu sayede isterseniz rakam göstermek için isterseniz de tabela yapmak için kullanabilirsiniz.

- **Ateş Sensörü:**

Ateş algılayıcı sensör kartı 760 nm - 1100 nm arasındaki dalga boyuna sahip ateşi tespit etmek için kullanılan bir sensör kartıdır. Üzerinde bir adet kızılaltı ışın alıcı barındırır. Yangın söndüren robotlarda ateş algılama sensörü olarak kullanabilmektedir. Üzerindeki trimpot ile hassasiyet ayarı yapılabilmekte ve hem analog hem de dijital şekilde çıkış alınabilmektedir.

- **LDR Işık Sensörü:**

Foto direnç, üzerine düşen ışığın şiddetine bağlı olarak değişen dirence sahip bir elemandır. Direnci, üzerine düşen ışık miktarıyla ters orantılı olarak değişir. Gündelik kullandığımız çoğu elektronik alette “fotosel” ismiyle yaygın olarak kullanılır.

- **Ultrasonik Mesafe Sensörü (HC-SR04):**

Hc-sr04 ultrasonik sensör sonar (Sound Navigation and Ranging) kullanarak karşısındaki nesneye olan mesafesini hesaplayan bir giriş kaynağıdır. Sonar dediğimiz sistem ses dalgalarını kullanarak cismin uzaklığını boyutunu elde etmemizi sağlar.

- **Gaz Sensörü:**

MQ-2 Gaz Sensör Modülü ev içinde ve endüstride gaz kaçağını belirlemede faydalıdır. LPG,i-bütan, propan, metan, alkol, hidrojen ve sigarayı tespit edebilir. Bazı modüller sensör hassasiyetini ayarlamak için değişken dirençli bir yapıya sahip olurlar.

- **Servo Motor:**

Servo motor 0 ila 180 derece arasında 1 derece hassasiyetle dönebilen motor çeşididir. Tam tur atamaz. Genellikle robot kol gibi tam tur dönmesine gerek olmayan, hassas açılı yerlerde kullanılır. Servo motor içerisinde bir adet DC motor bulunur.

- **DC Motor:**

Robotikte en sık tercih edilen motor tipi DC motorlardır. Bu motorlar 360 derece dönebilirler. DC motorlar ucuz, küçük ve etkilidir. Ayrıca boyut, şekil ve güç bakımından çok çeşitli olmaları da DC motorların sık kullanılmalarının bir diğer sebebidir. DC motorlar robotlarda veya herhangi bir sistemde direkt ya da dişli kutularıyla birlikte kullanılabilirler.

▪ Bluetooth Modülü:

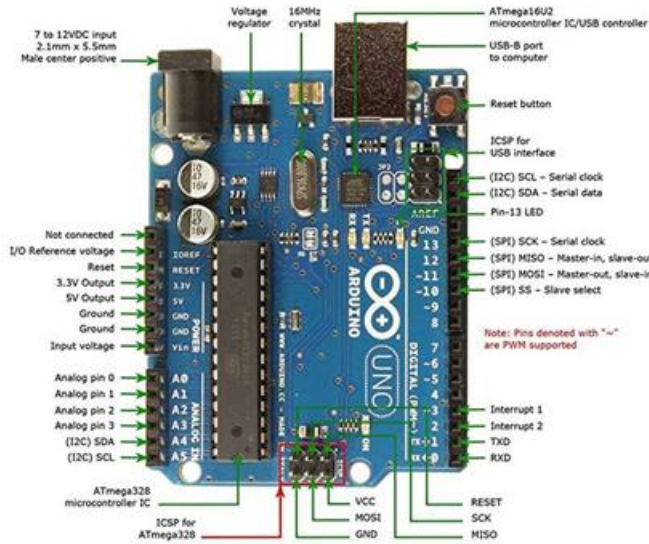
Çalışma mantığı ise seri haberleşme esasına dayanır. Tx ucu verici ucudur, Rx ise alıcı ucudur. 3.3V ve 5V arasındaki gerilimlerde çalışabilir.

▪ Wireless/Wifi Modülü:

Arduino projelerinin çoğunda kablosuz olarak bir yerden, başka bir yere veri aktarmak isteriz. Bu veriler kimi zaman diğer Arduino'ya bağlı parçaları kontrol etmek için, kimi zaman da karşı taraftaki Arduino'dan sensör verilerini almak için kullanılır. Arduino, tek başına kablosuz veri aktarımı için yeterli değildir. Arduino, kablosuz veri aktarımı için kablosuz haberleşme modüllerine ihtiyaç duymaktadır.

ARDUİNO TEMEL GİRİŞ/ÇIKIŞ FONKSİYONLARI

Arduino kartındaki pinler, giriş veya çıkış olarak yapılandırılabilir. Bu modlardaki pinlerin işleyişini açıklayacağız. Arduino analog pinlerinin çoğunun, dijital pinlerle tam olarak aynı şekilde yapılandırılabilirliğini ve kullanılabilirliğini unutmamak önemlidir.



Pinleri Giriş Olarak Ayarlama (INPUT):

Arduino pinleri varsayılan INPUT olarak yapılandırılmıştır, bu nedenle giriş olarak kullanırken pinMode() ile açık olarak bildirilmeleri gerekmez. Bu şekilde yapılandırılan pinlerin, yüksek empedanslı bir durumda olduğu söylenir. Giriş pinleri, örnekleme yaptıkları devrede, pinin önündeki 100 megaohm'lık bir dizi rezistansına eşdeğer oldukça küçük taleplerde bulunurlar. Bu, giriş pinini bir durumdan diğerine geçirmek için çok az akımalı anlamına gelir. Bu, pinleri, kapasitif bir dokunma algılayıcısının uygulanması veya bir fotodiyot olarak bir LED'in okunması gibi görevler için yararlı hale getirir. pinMode (pin, INPUT) olarak ayarlanmış pinler, bunlara bağlı hiçbirşey olmadan veya diğer devrelere bağlı olmayan tellerle bağlantılı olarak, pin durumundaki rastgele değişimler rapor eder, çevreden gelen elektriksel gürültüyü toplar.

Dahili Pull-up Direnç Kullanılarak Ayarlanmış Pinler:

Yazılımdan erişilebilen Atmega yongasına yerleşik 20.000 çekme direnci vardır. Bu dahili pull-up dirençleri, pinMode() 'yi INPUT_PULLUP olarak ayarlayarak erişilebilir. Bu, HIGH sensörün OFF olduğu ve LOW sensörün ON olduğu anlamına gelen INPUT modunun davranışını etkili bir şekilde tersine çevirir. Bu çekmenin değeri kullanılan mikrodenetleyiciye bağlıdır. Çoğu AVR tabanlı kartta, değerin 20kΩ ile 50kΩ arasında olması garanti edilir. Arduino Due, 50kΩ ile 150kΩ arasındadır. Tam değer için, kartınızdaki mikro denetleyicinin veri sayfasına bakın. Bir sensörü INPUT_PULLUP ile yapılandırılmış bir pine bağlarken, diğer ucun toprağa bağlanması gerekir. Basit bir anahtar durumunda, anahtar açık olduğunda pin HIGH okumaya ve anahtara basıldığında LOW'a neden olur. Çekme dirençleri, giriş olarak yapılandırılmış bir pine bağlı olan bir LED'i yakmak için yeterli akım sağlar. Bir projedeki LED'ler çalışıyor gibi gözüküyorsa, ancak çok zayıfsa, bunun olması muhtemeldir.

Bir pinin HIGH veya LOW yukarı çekme dirençlerini kontrol edip etmediğini kontrol eden aynı yazmaçlar (dahili çip bellek konumları). Sonuç olarak, pin INPUT mode olduğunda, yukarı-çekme dirençlerine sahip olacak şekilde konfigüre edilen bir pin, eğer pin daha sonra pinMode() ile birOUTPUTmoduna getirilirse, pin HIGH olarak konfigüre edilir. Bu, diğer yönde de çalışır ve HIGH durumda kalan bir çıkış pini, pinMode () ile bir girişe geçildiğinde çekme direncine sahip olacaktır.

```
void setup() {  
  pinMode(7,INPUT_PULLUP); // 7. pin giriş ve pull up olarak ayarlandı.  
  pinMode(13,OUTPUT); // 13. pin çıkış olarak ayarlandı  
}
```

Pinleri ÇIKIŞ Olarak Ayarlama (OUTPUT):

PinMode() ile OUTPUT olarak yapılandırılan pinlerin düşük empedanslı bir durumda olduğu söylenir. Bu, diğer devrelere önemli miktarda akım sağlayabildikleri anlamına gelir. Atmega pinleri kaynağı (pozitif akım sağlar) veya diğer cihazlara / devrelere akımın 40 mA (miliamper) değerine kadar (negatif akım sağlar) kaynak yapabilir. Bu, bir LED'i parlak bir şekilde yakmak için yeterlidir (serirezistansını unutmayınız), ya da birçok sensörü çalıştırır, ancak röleleri, solenoidleri veya motorları çalıştırmak için yeterli akım yoktur. Çıkış pinlerinden yüksek akım cihazlarını çalıştırmaya çalışmak, pin içindeki çıkış transistörlerine zarar verebilir veya yok edebilir veya tüm Atmega çipine zarar verebilir. Çoğunlukla, bir mikrodenetleyicide bir "ölü" pin ile sonuçlanır, ancak kalan yongalar hala yeterli şekilde çalışır. Bu nedenle, belirli bir uygulama için pinlerden maksimum akım çekilmesi gerekmediği sürece, OUTPUT pinlerini diğer cihazlara 470Ω veya 1k dirençler üzerinden bağlamak iyi bir fikirdir.

pinMode() Fonksiyonu Kullanımı:

PinMode() fonksiyonu, bir girdi veya çıktı olarak davranması için belirli bir pin yapılandırmak için kullanılır. Dahili çekme dirençlerini INPUT_PULLUP modu ile etkinleştirmek mümkündür. Ayrıca, INPUT modu dahili çekimleri açığa devre dışı bırakır.

```
void setup() {  
  pinMode(13, OUTPUT); // 13. pini çıkış için ayarla  
}
```

ARDUİNO'NUN TEXT TABANLI KODLANMASI

C. Blok ve Metin Tabanlı Kodlama

Kodlamayı iki ana başlıkta ayırmak gerekirse, bunları blok ve metin tabanlı kodlama olarak isimlendirebiliriz.

3. Blok Tabanlı Kodlama Nedir?

Blok (görsel) tabanlı kodlama robot programlama için oluşturulmuş sürükle bırak yöntemi ile çalışan ortamlardır.

- Uygulamaların çoğu görsel ara yüz bakımından eğlenceli ve dikkat çekici olduğu için ve aynı zamanda sürükle – bırak tekniği ile kullanım kolaylığı sağladığı için çocuklar kodlamaya başlarken ilk adım niteliğinde önemli bir başlangıçtır.
- Bilgisayar programlama dilleri bir yabancı dil gibi öğrenilmesi zaman alan ve ilk bakışta karmaşık dillerdir. Bu dillerin daha kolay öğrenilmesini ve yaratıcı şekilde kullanımını desteklemek amacıyla çeşitli blok tabanlı programlama araçları geliştirilmiştir.
- Blok tabanlı programlama araçları belirli bir program dilinde kullanılan komutu işlevini anlatan bloklara çevirmektedir. Bu şekilde iletişim kolaylaşır. Blok tabanlı kodlama ile hiçbir kod kullanmadan sürükle bırak veya yapboz oynar gibi programlar oluşturmak olanaklı hale gelmiştir.

Blok tabanlı programlar çocuklara kodlamanın mantığını en sade ve basit şekilde kavramasını sağlarken aynı zamanda eğlenceli görsellerle çocukların dikkatini çekmektedir. Blok tabanlı kodlamada ki yapılar metin tabanlı kodlamada olan komutların birebir blok versiyonudur. Blok tabanlı kodlama şekiller, yapılar anlaşılabilirliği üst düzeyde tutularak hazırlanmış ve kodlamaya yeni başlayanlar için geliştirilmiştir. Blok tabanlı kodlama çocukların kodlama öğrenmesi için önemli bir araçtır. Blok tabanlı kodlama sayesinde çocuklar kodlama ve kodlamanın temeli olan algoritmayı kolaylıkla kavrayabilir.



Blok tabanlı kodlama ile metin tabanlı kodlamanın arasındaki en önemli fark anlaşılabilirlik ve zamandır. Metin tabanlı kodlama ile hazırlanan bir kodu daha kısa sürede ve kodlama mantığını daha kolay kavrayarak blok tabanlı kodlama ile yapılabilir. Blok tabanlı programlamanın dezavantajı da ne kadar projeler geliştirilse de sınırlı kalacaktır. Metin tabanlı programlamada geliştirebileceğimiz proje sayısı sınırsızdır.

4. Metin Tabanlı Kodlama Nedir?

Metin tabanlı programlama ise kelimelerle ve komutlarla yapılan kodlama yapısıdır. Programlama dilinin söz dizim kurallarına uygun olarak metin tabanlı karakterlerin kullanılması ile yapılan kodlamadır.

```
int buzzer= 3; //Ledi 3.pine bađladıđımızı tanımladık.

void setup() {
  pinMode (buzzer, OUTPUT); //Led modülü çıkış pinidir.
}

void loop() {
  digitalWrite (buzzer, HIGH); //Ledi yakıyoruz.
  delay (1000); //1 saniye boyunca
  digitalWrite (buzzer, LOW); //Ledi söndürüyoruz.
  delay (1000); //1 saniye boyunca
}
```

Blok tabanlı kodlama uygulamalarına örnek: **Scratch, Mblock, MakeCode**

Metin tabanlı kodlama uygulamalarına örnek: **Arduino IDE, C#, C++, Java, Python, Php**

D. Arduino Temel Komutlar

1. void setup()

```
{
```

// komutlar. Bu cihaz her açıldığında bir kere çalışacak komutları içerir. Genellikle ayar komutlarıdır.

```
}
```

Setup() fonksiyonu, Arduino'ya yüklenmiş olan .ino uzantılı kod parçasının, Arduino başlatıldığında veya yeniden başlatıldığında ilk çalıştırılan kısmıdır. Setup() fonksiyonu, çalışma ortamını başlangıç için bizlere hazırlar ve görevini tamamladıktan sonra bir yeniden başlatmaya kadar tekrar çalıştırılmaz.

2. void loop()

```
{
```

//komutlar. program boyunca sürekli çalıştırılacak komutlar buraya yazılır. }

Loop() fonksiyonu, setup fonksiyonu çalıştırdıktan sonra çalıştırılır ve bir sonsuz döngü işlevi görür. Loop fonksiyonun bu sonsuz döngü özelliği kullanılarak sürekli tekrar edecek olan işlemlerimizin gerçekleştirilmesini sağlar. Örneğin; Arduino'nun belkide en temel örneği olan Blink örneğinde olduğu gibi 1'er saniye aralıkla bir ledi yakmak ve söndürmek gibi tekrar eden bir işlemlerde kullanılabilir.

3. pinMode(pin, mode)

Dijital giriş/çıkış pinlerinden herhangi birini giriş ya da çıkış olarak tanımlamak için kullanılır. Pin denilen alana dijital pinlerden hangisini kullanacaksak onun numarası yazılır. Mode ile ifade edilen alana ise giriş mi? Çıkış mı? Olacağı yazılır. Ancak İngilizce olarak yazılmalıdır. Giriş ise INPUT, çıkış ise OUTPUT yazılır.

Kullanım şekli:

pinMode(13, OUTPUT);

Açıklama:

- pinMode(pin, mode) komutu yukarıda anlatılan program bloklarından **Void setup(){.....}** kısmında süslü parantezlerin arasında kullanılmalıdır. Çünkü Arduino'nun kurulumuna ait bir komuttur.
- Pin tanımlama alanına dijital pinlerden herhangi biri yazılabilir. Örneğin Arduino Uno için; 0'dan 13'e kadar 14 adet dijital giriş/çıkış olarak kullanılabilen pinler mevcuttur. Bunlardan herhangi biri yazılabilir.
- Pin numaraları rakam olarak yazılabileceği gibi, bu sayılara bir isim verilip, pin yerine o isim de kullanılabilir. (pin numaralarına isim atama, ilgili komutlar anlatılırken bahsedilecektir.)
- Mode ile belirtilen alana output yazıldığı için bu örnekte 13 numaralı pin, çıkış olarak kullanılacaktır.
- Komutun sonunda (;) noktalı virgül kullanılmalıdır.

4. digitalWrite(pin, value)

Dijital yazma anlamına gelen bu komut, çıkış olarak belirlenen pine değer aktarmak(yazmak) için kullanılır.

Kullanım şekli:

digitalWrite(13, 1);

Açıklama:

- İki farklı argüman vardır. İlki dijital yazma işleminin yapılacağı pin belirlenir. İkincisi ise yazılacak değer belirlenmesi. Yazılacak değer 1 veya 0 olabilir.
- Yazılacak değer 1 veya 0 olarak yazılabileceği gibi 1 = High, 0 = Low olacak biçimde High veya Low da yazılabilir.

- pinMode komutu ile hangi pinler dijital çıkış olarak tanımlandı ise o pinlere yazma işlemi yapılabilir. Giriş olarak tanımlanan bir pine yazma işlemi yapılamaz.
- Örneğin 13. Pine led bağlı olduğunu düşünürsek ve yukarıda verilen şekilde komut çalıştırılırsa; 13. Pine bağlı olan led 1 olacaktır ve ışık verecektir.
- digitalWrite(13,0); şeklinde yazılırsa yanan led sönecektir.
- Bu komut Loop() bloğunda yer almalıdır. Sonunda da (;) noktalı virgül kullanılmalıdır.

5. digitalRead(pin)

Dijital okuma anlamına gelen komut belirtilen pinden dijital olarak okuma işlemi yapar.

Kullanım şekli:

digitalRead(5);

Açıklama:

- Read okuma işlemi olduğundan bu dışarıdan gelen bilgiyi okumak biçiminde olmalıdır.
- Burada dikkat edilecek husus şudur; eğer okunan değer 1 ise “digitalRead()” komutunun geri getirdiği değer 1 olur. Yani okuduğu değer 1 ise digitalRead(5) = 1 olur, okuduğu değer 0 ise digitalRead(5) = 0 olur.
- Dijital okuma işlemi pinMode komutu ile Setup kısmın da giriş olarak tanımlanan pinden okuma işlemi yapabilir.
- Belirlenen pine gelen dijital 1 veya 0 bilgisini algılayabilir.

6. Arduino Gecikme Komutları

***delay(süre)**

delay, gecikme ya da bekleme anlamlarında kullanılır. Bu komut çalıştırıldığı sırada mikrodenetleyici veya Arduino hiçbir şey yapmadan belirttiğimiz süre kadar bekler.

Kullanım şekli:

delay(1000);

Açıklama:

- Parantez içine yazılacak sayı mili saniye cinsinden olmalıdır.
- 1000 mili saniye 1 saniyeye eşittir.

- Loop() bloğunda kullanılır ve sonunda noktalı virgül olmalıdır.

***delayMicroseconds(süre)**

Bekleme komutlarından diğeri ise delaymikroseconds komutudur. Bu komutun delay komutundan farkı süre kısmına yazılan sayı mikro saniye cinsinden olmalıdır.

Kullanımı:

delayMicroseconds(1000);

Açıklama:

- Mikro saniye olarak süre belirtilmeli.
- 1.000.000 mikro saniye 1 saniyeye eşittir.
- Loop() bloğu içinde kullanılmalı sonunda noktalı virgül bulunmalı.

7. **Serial.begin**

Arduino'da seri haberleşmede bant genişliği seçmek için kullanabilirsiniz. Fakat Kod kısmında void setup() bölümüne yazmanız gerekir.

Kullanımı:

Serial.begin(bantgenisligino)

Örnek: *Serial.begin(9600)*

8. **Serial.print:**

Arduino'da seri haberleşmede veri yazdırmak için kullanılır. Println olduğunda yazdırdıktan sonra bir alt satıra geçer.

Kullanımı:

Serial.print("Yazdırılacak metin")

Örnek:

Serial.print("Merhaba")

Serial.println("Merhaba")

9. **Serial.read**

Arduino'da seri haberleşmede veri almak için kullanılır. Veri alırken kullanılan değişken setup() 'dan önce tanımlanmalıdır.

Kullanımı:

girilenDeger = Serial.read();

Örnek:

int girilenDeger=0;

```
girilenDeger = Serial.read();
```

10. Arduino'da if else kullanımı

Diğer programlama dillerinde olduğu gibi Arduino'da da if else'in kullanımı aşağıdaki gibidir. Fakat if-else kullanırken içine yazdığınız kodların hizasına ve parantezlere dikkat etmelisiniz.

```
if(şart ifadesi) {  
  //şart sağlandığında çalışacak kodlar</p>  
} else {  
  //sağlanmadığında çalışacak kodlar</p>  
}
```

11. Arduino'da for kullanımı

Diğer programlama dillerinde olduğu gibi Arduino'da da for döngüsünün kullanımı aşağıdaki gibidir. Fakat for döngüsü kullanırken içine yazdığınız kodların hizasına dikkat etmelisiniz.

```
for(int i=baslangicdegeri;i<sartifadesi;i++) {  
  //kodlar  
}
```

12. Random

Arduino'da rastgele sayı üretmek için kullanabilirsiniz. Yani 1 ile 100 arasında bir sayı üretmek istiyorsanız bu komutu kullanmanız gerekmektedir.

Kullanımı:

```
random(baslangicdegeri,bitisdegeri)
```

Örnek:

```
int sayi=random(1,100) //rasgele 1 ila 100 arasında bir sayı üretir.
```

13. While döngüsü

For gibi while kodları da döngü amacıyla kullanılır.

```
b = 20;
```

```
while( b > 10){
```

```
  // b değişkeninin 10'dan büyük olduğu durumlarda döngü devam eder
```

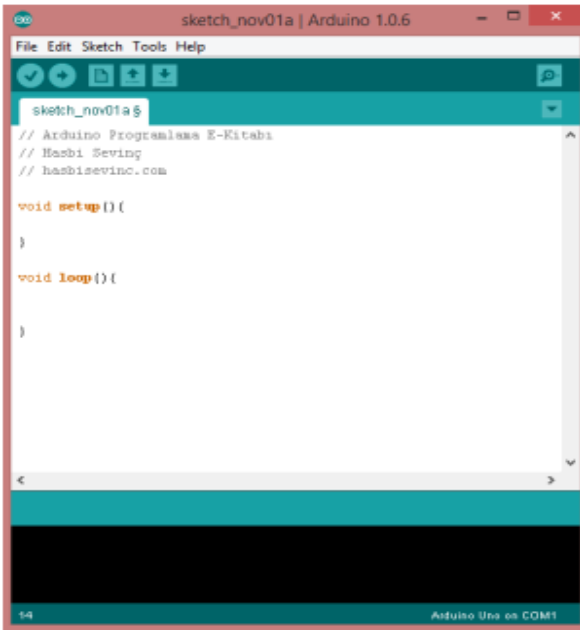
```
  b = b - 1 ;
```

```
  // her döngüde b'nin değerini bir azalttık
```

```
// dikkat edilmelidir ki eğer b'nin değerini değiştirmeseydik
// döngü koşulu hep doğru olacağından program burada takılı kalacaktı
}
```

E. Arduino'nun Kurulumu ve Arduino Programı

Arduino'yu kullanmaya başlıyoruz. Öncelikle arduinomuzun USB kablosunu arduino'ya bağlayalım. Daha sonra bilgisayarımıza takalım. Win 7 ve üst sürümlerinden arduino otomatik olarak yüklenecektir. Biraz bekleddikten sonra artık arduino bilgisayar tarafından tanınacaktır.



Arduino'nun bilgisayar tarafından tanınıp tanınmadığını görmek için aygıt yöneticisinden kontrol edelim ve arduino'nun bağlı olduğu COM Portunu aklımızda tutalım (aygıt yöneticisi: bilgisayarına sağ tıklayın -> yönet -> aygıt yöneticisi) Arduino'yu bilgisayarımıza tanıtırken bizde bir yandan kullanacağımız programı indirelim. Bunun için <http://arduino.cc/en/Main/Software> adresine girelim ve Arduino programını indirip programı kuralım. Programı hatasız bir şekilde yükledikten sonra programı açalım.

Program çalıştığında yandaki gibi bir ekran sizi karşılayacak. Bu ekranın ortasındaki kısma

kodlarımızı yazıyoruz. Programın sağ en altındaki yer bize Arduino Uno üzerinden çalıştığımızı ve arduino'nun COM1 portuna bağlı olduğunu gösteriyor. Eğer bu doğru değilse bu ayarları değiştirmemiz lazım öncelikle.

Tools menüsü: Buradan kullandığımız arduino türünü ve arduinomuzun bağlı olduğu COM portunu seçelim. Eğer bunu bilmiyorsak aygıt yöneticisinden bakabiliriz.

Program compile: Bu buton ile yazdığımız programı kontrol edebiliriz. Eğer kodda hata varsa alttaki siyah bölümde turuncu yazı şeklinde yaptığımız hata ve satırı yazacaktır. Hatırlatma: Bulduğumuz satırın sayısı sol attadır yazmaktadır.

Program Compile & Upload: Bu buton ile önce yazdığımız kod derlenir. Eğer kodda hata yoksa yazılan kod Arduino'nun anlayacağı dile çevrilir ve otomatik olarak Arduino'ya atılır. İşlem sırasında ilerleyen çubuktan işlem durumunu görebilirsiniz. Ayrıca Tam program

atılma sürecinde Arduino üzerinde bulunan Tx ve Rx LED'leri hızlı bir şekilde yanıp sönecektir. Programı bir kere Arduino'ya atmanız yeterli olacaktır.

Serial Monitor: Yeni pencere açan bu buton ile Arduino'dan yolladığımız verileri görebiliriz. İlerleyen konularımızda bu butonu sıkça kullanacağız.

Tinkercad Circuits (Devreler)

Tinkercad, Autodesk firmasının geliştirdiği, internet tarayıcısı üzerinden kullanılabilen, 3 boyutlu tasarım yapılabilen, dijital elektronik ve kod yazma eklentileri ile çalışabilmeyi sağlayan ücretsiz bir uygulamadır.

Tinkercad uygulamasını kullanabilmek için internet bağlantısı olan her yerden; bilgisayar, tablet ve telefonla tasarımlar yapabilir. Oluşturulan modeller, profil üzerinden sisteme kaydedildiği için farklı zamanlarda ve farklı cihazlardan profile girerek modeller üzerinde değişiklik yapılabilir.



Şekil: Autodesk Tinkercad uygulaması

Tinkercad Circuits bölümü, basit elektronik devre kurulumunu anlatmak için kullanılacak sade, görsellik anlamında etkileyici bir online simülasyon programıdır. Circuits ile basit elektrik-elektronik devrelerde kullanılacak araçlar ile sınırsız denemeler yapılabilmektedir. Özellikle kontrol kartlarını kullanmadan önce basit elektrik devrelerinin anlatılması, uygulamaların anlaşılması açısından tavsiye edilmektedir.

Bu bölümde Tinkercad Circuits üzerinden temel elektronik ve Arduino uygulamaları, simülasyon kullanarak yapılacaktır. Bu sayede herhangi bir bileşen ve kart kullanmadan robotik uygulamaların nasıl çalıştığı öğrenilmiş olacaktır.



Şekil: Tinkercad Circuits (Devreler)

Sisteme Üye Olma ve Giriş

Tinkercad sitesi üzerinden tasarım oluşturmak veya Circuits ile simülasyon oluşturabilmek için

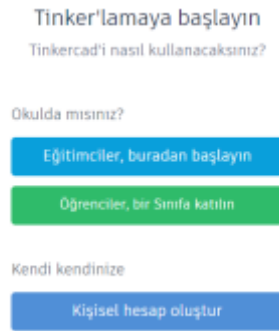
sisteme kayıt olunması gerekmektedir.

<https://www.tinkercad.com> adresine girdikten sonra Hemen Katıl seçeneğini seçerek sisteme üye olunabilir.



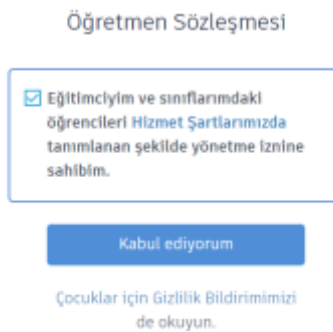
Şekil :Tinkercad sitesi üzerinden kayıt işlemi

Hemen Katıl seçeneğini seçtikten sonra Şekildeki gibi 3 seçenek ekrana gelir.



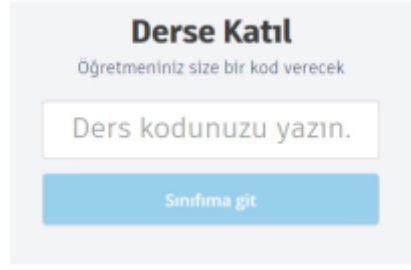
Şekil :Sisteme kayıt olma seçenekleri

Eğitimciler, buradan başlayın: Bu seçenek ile sisteme kayıt olmak istendiğinde Öğretmen Sözleşmesi kabul edilerek öğretmen hesabı oluşturulabilir.



Şekil :Öğretmen sözleşmesi

Öğretmenin oluşturduğu hesap üzerinden sınıflar tanımlanarak öğrencilerin sisteme kayıt olmadan, kod kullanarak giriş yapması sağlanabilir. Öğrenciler, bir Sınıfa katılın: Bu seçenek sayesinde; öğretmen hesabı ile öğrenciler için oluşturulan ders kodları kullanılarak, sisteme kayıt olmadan giriş yapmaları sağlanabilir.



Şekil: Derse katıl seçeneği ile ders kodu girişi

Kişisel Hesap Oluştur:

Bu seçenek ile sistem üzerinde standart bir hesap oluşturularak sistem kullanılabilir. Eğer bir sınıf hesabı oluşturulmayacaksa bu seçenek seçilebilir.

Kayıt Aşaması

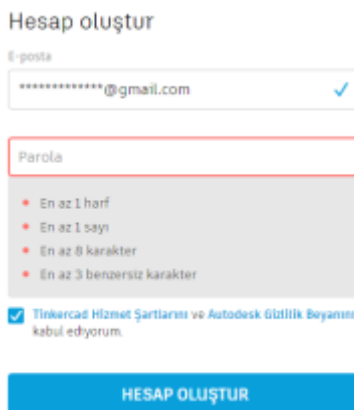
Öğretmen hesabı veya kişisel bir hesap oluştururken sisteme bir e-posta hesabı ile kayıt olunabileceği gibi var olan bir servis (gmail, apple, Microsoft veya facebook) ile otomatik olarak da kayıt olunabilir.

E-postayla Kaydol seçeneği seçilirse hesap oluşturma aşamasında ülke ve doğum tarihi bilgilerinin girildiği bir sayfa açılacaktır. Oluşturulan hesap 18 yaşından küçük bireylere ait olursa kısıtlı bir hesap oluşturulacaktır.



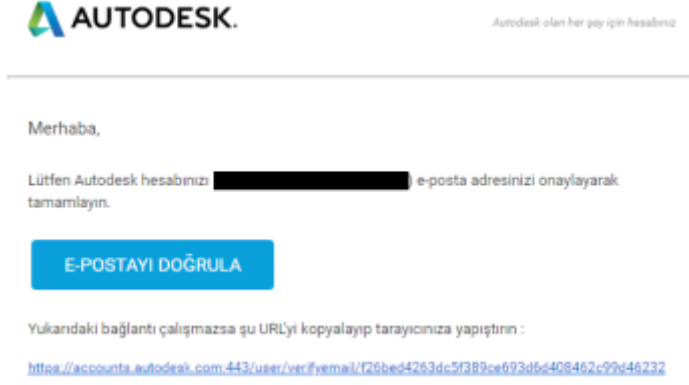
Şekil: Kayıt aşamasında bölge ve tarih bilgisi

Bir sonraki aşamada kullanıcının bir e-posta hesabı ve tinkercad oturumu açmak için kullanacağı parola girmesi gereklidir. Bu adımda oluşturulan parolanın verilen kriterlere uygun olarak girilmesi gereklidir.



Şekil: e-posta adresi ve parola belirleme adımı

Bu işlemten sonra hesap oluşur ve sisteme otomatik olarak giriş sağlanır. Ayrıca e-posta adresine, hesabın herhangi bir kısıtlama olmadan kullanılabilmesi için bir doğrulama mesajı gelir. Buradaki e-postayı doğru linkine tıklanması gerekmektedir.



Şekil:Doğrulama mesajı

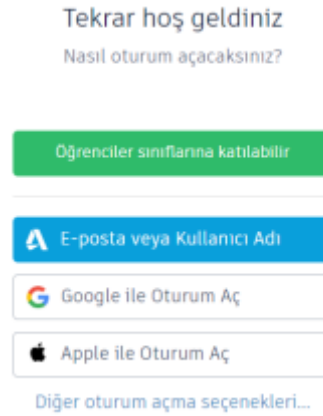
Farklı bir cihazdan veya farklı bir zamanda sisteme giriş yapmak istenildiğinde var olan hesap ile sisteme girmek için ana sayfada bulunan Giriş yap seçeneği seçilir.



Şekil 3.12. Sisteme giriş yapma

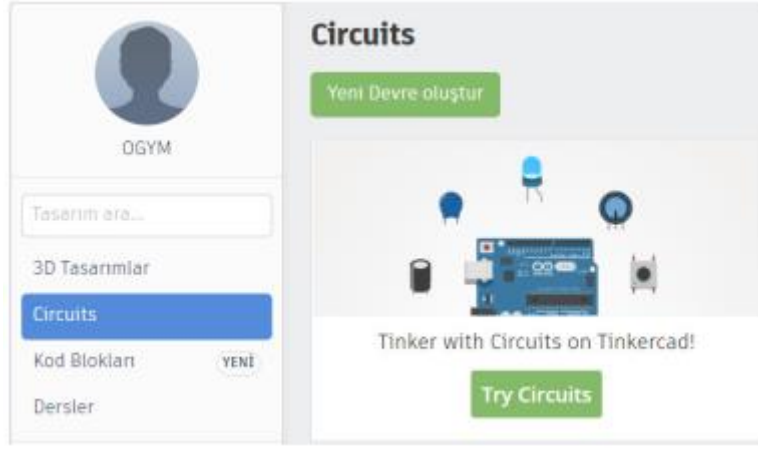
Şekil: Sisteme giriş yapma

Açılan sayfada hesaba uygun olan seçenek seçilerek giriş yapılabilir.



Şekil: Tinkercad oturum açma işlemi

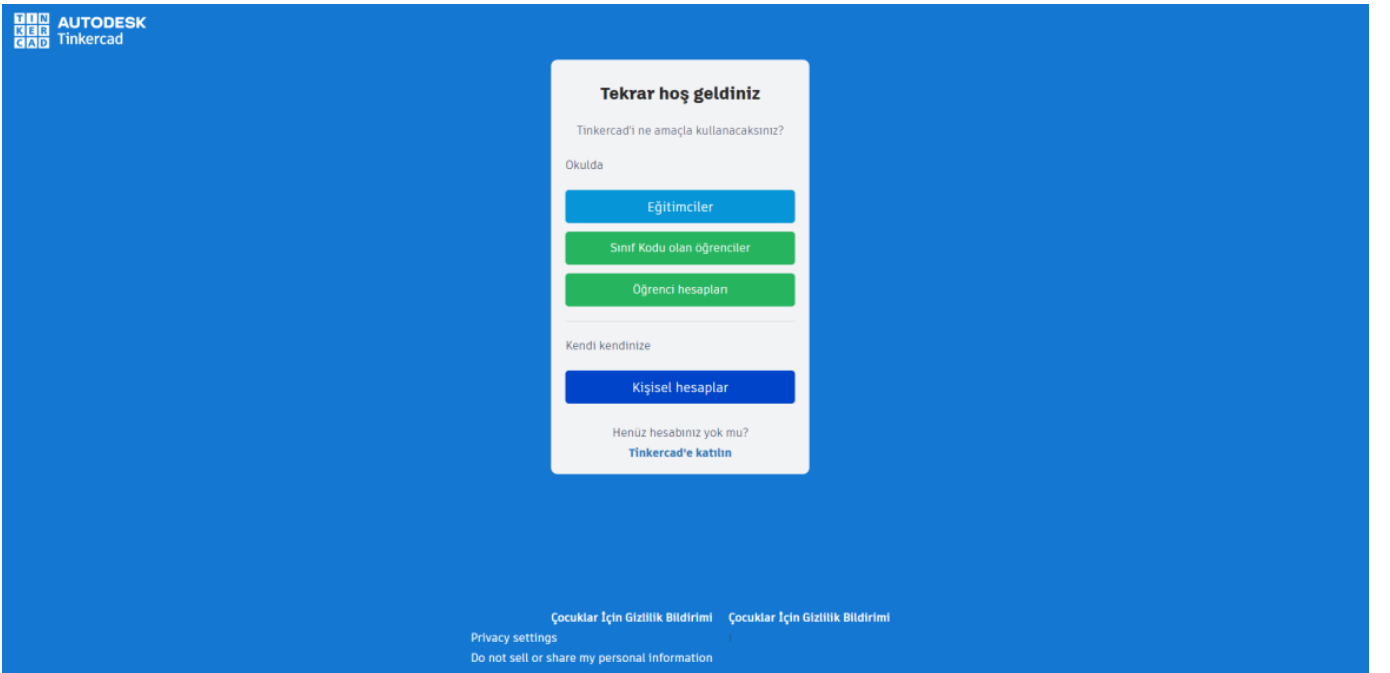
Sisteme giriş yapıldıktan sonra Tinkercad uygulaması kullanılır. Circuits (devreler) özelliğini kullanarak elektronik simülasyonlarını yapabilmek için açılan sayfada sol tarafta bulunan Circuits seçeneği seçilir.



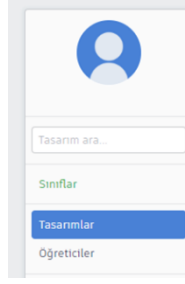
Şekil:Tinkercad Circuits (Devreler) Simülasyonuna geçiş

TINKERCAD İLE ADIM ADIM DEVRE OLUŞTURMA

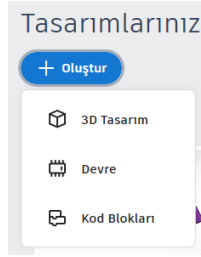
- Devre tasarlamaya başlarken yapacağımız il işlem herhangi bir internet tarayıcısından TinkerCad'i açmak olacaktır. Siteye giriş yapmak için <https://www.tinkercad.com/> adresini kullanıyoruz.
- Siteyi açtıktan sonra hesabımıza giriş yapıyoruz. Giriş kısmında öğretmen ve öğrenciler için farklı giriş seçenekleri mevcut. Eğitimciler Google hesaplarıyla sisteme kolay bir şekilde giriş yapabilirken, öğrenciler öğretmenlerinin oluşturduğu kullanıcı adı ve sınıf kodlarını kullanarak sisteme giriş yapabilirler.



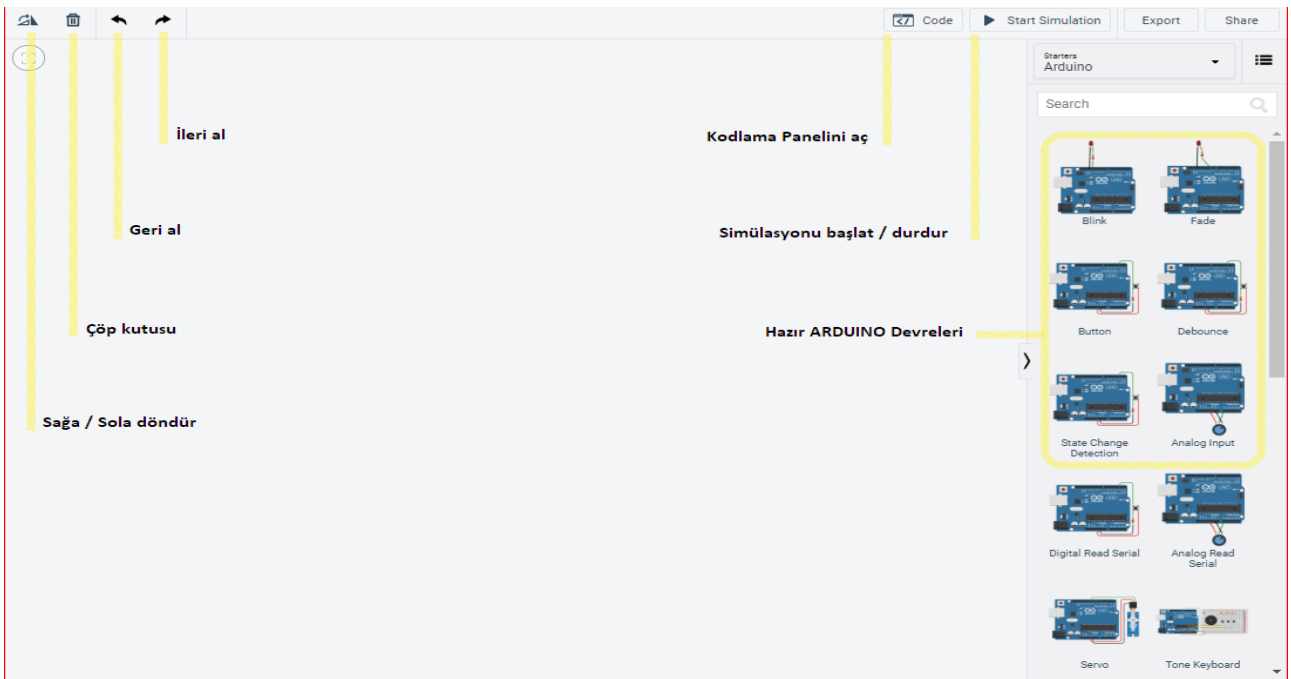
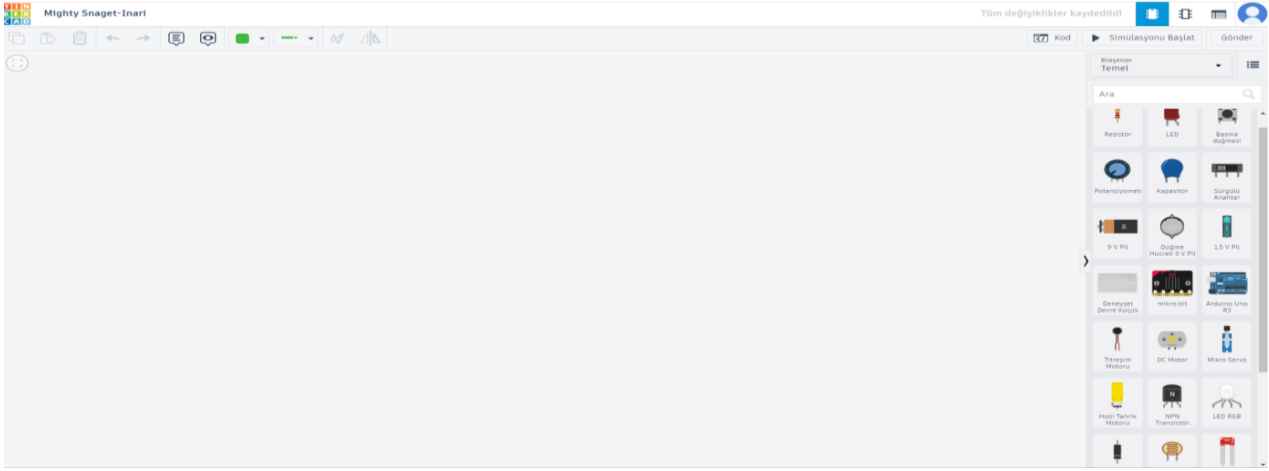
- Tinkercad ana hesabımızı açtıktan sonra tasarımlarımızın bulunduğu sekmeye ("tasarımlar" sekmesi) gelerek yeni bir tasarım oluşturabiliyoruz. Tasarımlarımızda 3d tasarım, devre ve kod blokları oluşturabiliyoruz.



- Bir devre yapmaya karar verdikçe devremizi yapmak için “Oluştur” butonuna tıklayarak gelen seçenekler arasından “devre”yi seçiyoruz.



- Karşımıza devre oluşturup kodlama yapabileceğimiz ekranımız geliyor.



- Aşağıda resimde butonların özelliklerini görebiliyoruz. Bu butonların özellikleri ise şu şekildedir:

Sağa / Sola döndür – çalışma alanına eklediğiniz elektronik komponent veya modüllerin döndürme ayarlarını yapar.

Çöp Kutusu – istenmeyen elemanları ekrandan silebilirsiniz.

Geri Al / İleri Al – Ekranda yaptığınız değişiklikleri bir adım geri giderek evvelki haline dönüştürebilirsiniz.

Kodlama Panelini Aç – Arduino devresinin kodlandığı çalışma paneli. Burada hem GÖRSEL hem de METİN olarak hazırlanmış Arduino kodlarını göreceksiniz. Görsel kodlama çok popüler olan ve yeni başlayanlar için gayet kolay bir yazılım tekniğidir. SCRATCH veya SNAP gibi ortamlarla evvelden çalışmışsanız buna hemen alışırsınız. Değilse bunu kısa zamanda öğrenip uygulamaya başlayabilirsiniz. Metin kodlama ARDUINO IDE ortamında kullanılan aynı dil ve tekniği uygular.

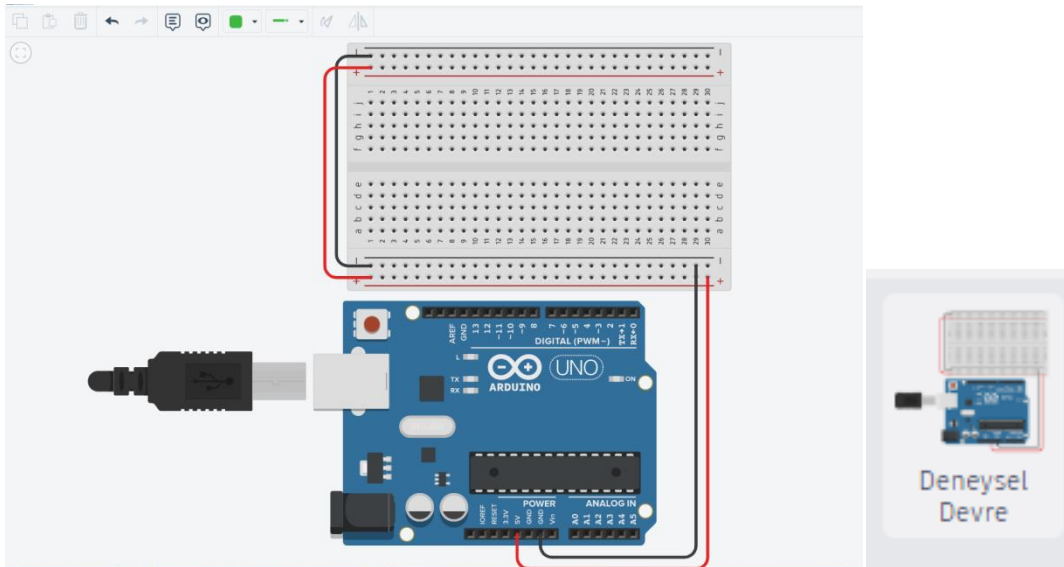
Simülasyonu Başlat / Durdur – ARDUINO devrenizi tamamladıktan sonra bu butona basarsanız çalışmasını izleyebilirsiniz. LED'ler yanıp söner, motorlar döner, elektronik komponentler gözle görülme bile vazifelerini yaparlar.

Hazır ARDUINO Devreleri – Bu panelde önceden hazırlanmış ve bir çok Arduino kitaplarında da bulunan popüler devreler hazır olarak size sunulmuştur.

- Açılan yeni sayfada sağ tarafta olunan bileşenler kısmında programlamak istediğimiz Arduino'yu seçiyoruz.



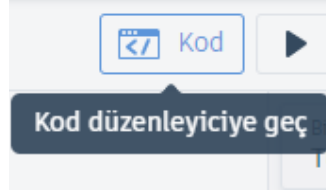
- Deneysel Devre'yi seçersek bizim için bir board ve Arduino kitini ekranın içerisine yerleştiriyoruz.



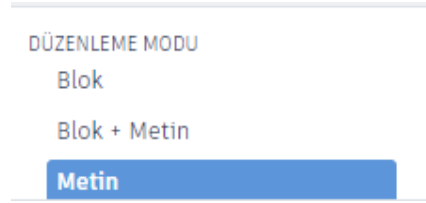
- Bileşenler'den "Tümü"ne tıkladığımız zaman ise devremiz için gerekli olan tüm devre elemanlarına erişim sağlayabiliyoruz.



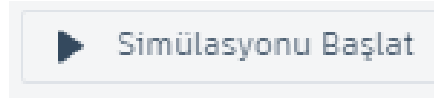
- Devremizi tasarladıktan sonra Arduinomuzu programlamak için sağ tarafta bulunan kod kısmına tıklayıp programlamaya başlıyoruz.



- Programlamayı blok tabanlı programlama veya metin tabanlı programlama seçerek kodlarımızı yazıyoruz. İstersek te hem metin hem de blok tabanlı programlamayı aynı ekranda görebiliyoruz.



- Simülasyonu başlat butonuna basarak yazmış olduğumuz kodları ve devremizi test edebiliriz.

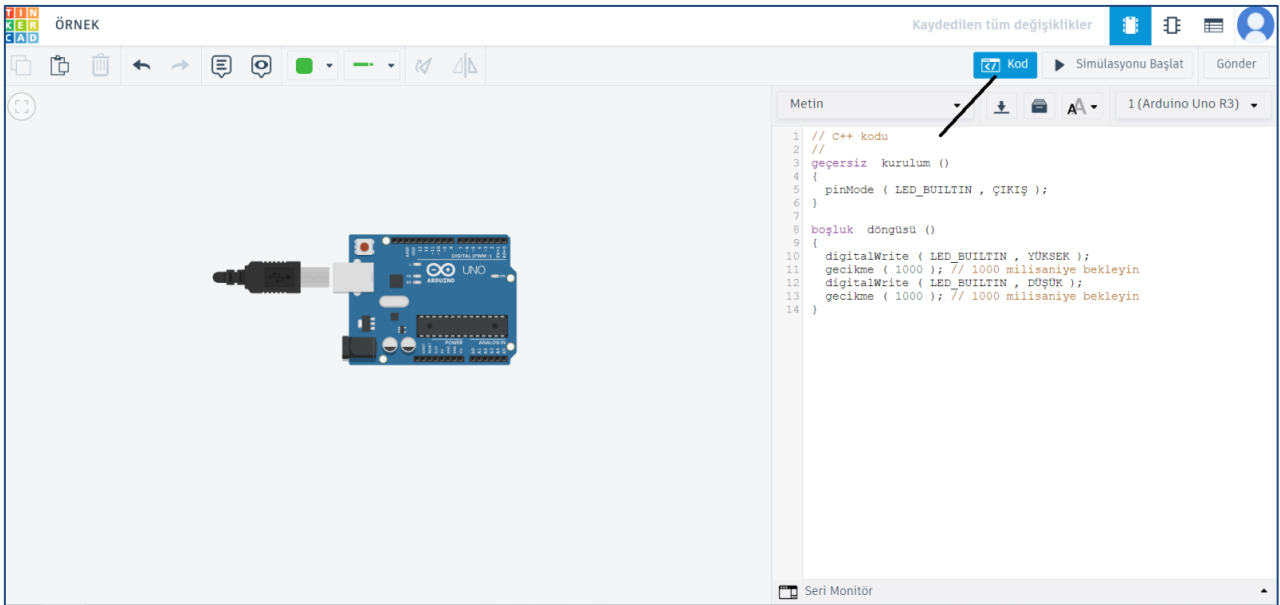
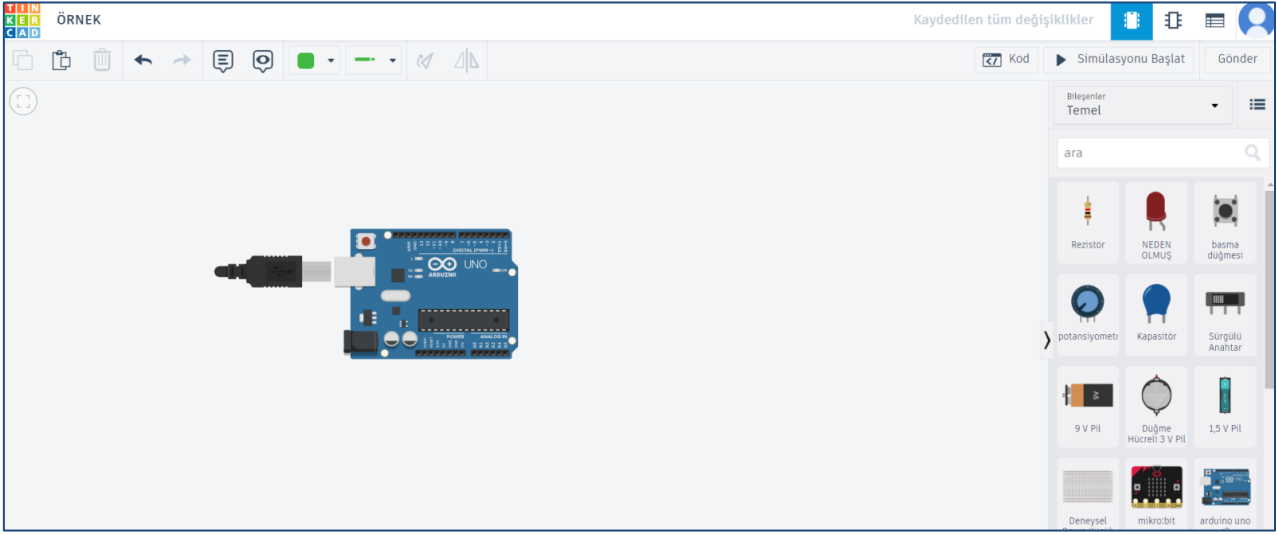
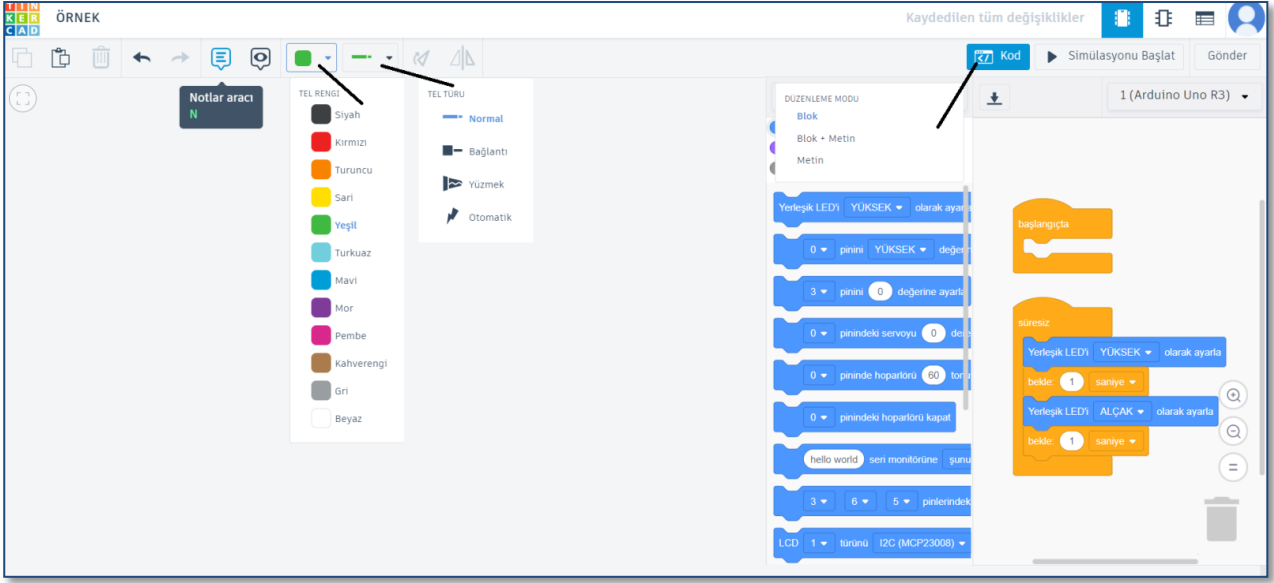


- Devre tasarımı bittikten sonra sayfamızın sol üst köşesinde bulunan isimlendirme kısmında projemizi isimlendiririz böylece bir sonraki sefer bulmak istediğimiz projemizi daha kolay bulabiliriz.



devre isimlendirme

ARDUINO/TINKERCAD UYGULAMA OLUŞTURMA



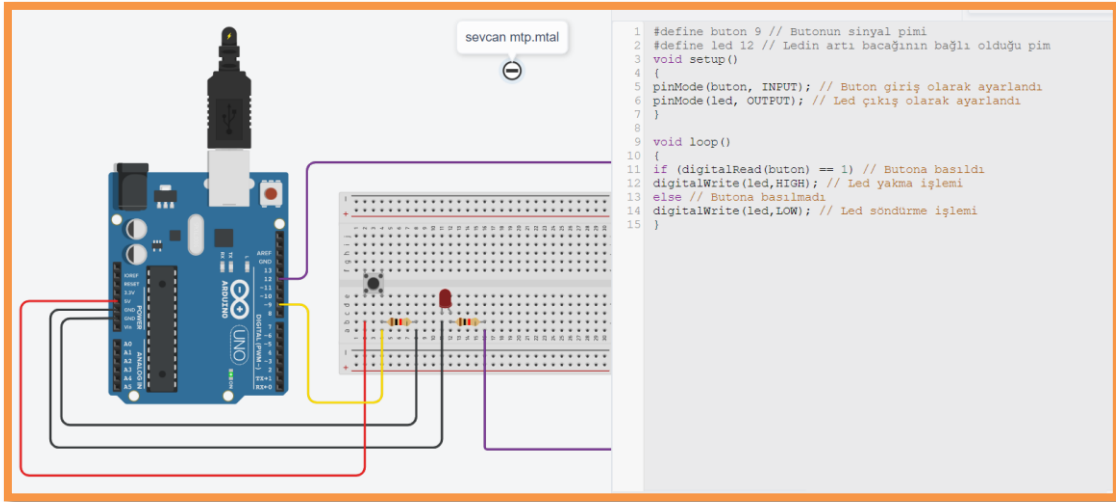
ARDUINO ile TINKERCAD PROJEMİZDE YAPTIĞIMIZ UYGULAMALARIMIZ

▪ BUTON LED UYGULAMASI:

UYGULAMA KOD YAZILIMI:

```
#define buton 9 // Butonun sinyal pimi
#define led 12 // Ledin artı bacağının bağlı olduğu pim
void setup()
{
  pinMode(buton, INPUT); // Buton giriş olarak ayarlandı
  pinMode(led, OUTPUT); // Led çıkış olarak ayarlandı
}

void loop()
{
  if(digitalRead(buton) == 1) // Butona basıldı
    digitalWrite(led,HIGH); // Led yakma işlemi
  else // Butona basılmadı
    digitalWrite(led,LOW); // Led söndürme işlemi
}
```



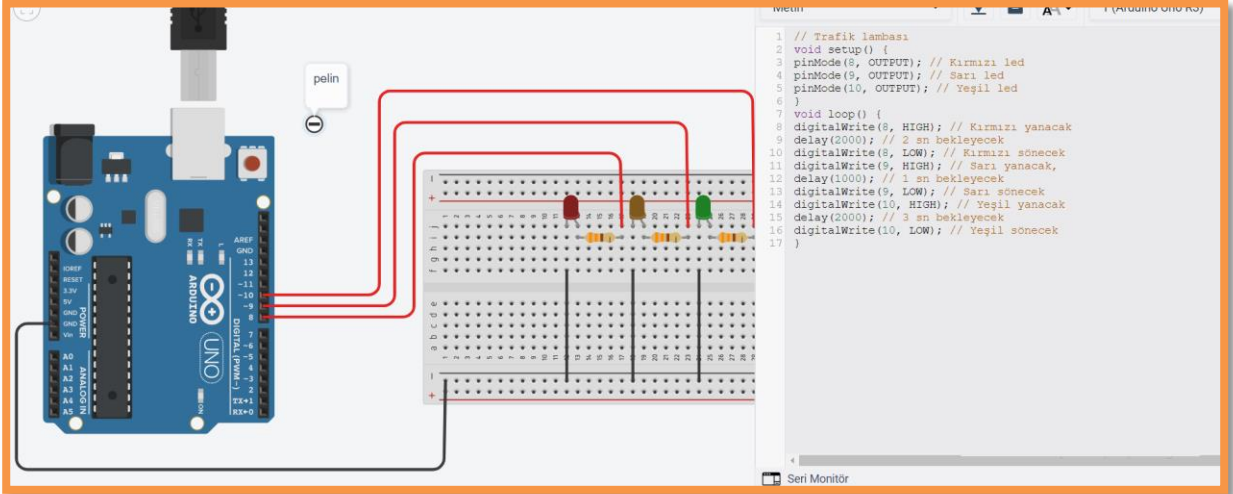
▪ TRAFİK IŞIKLARI UYGULAMASI:

UYGULAMA KOD YAZILIMI:

```
Void setup() {
  pinMode(5, OUTPUT); // Kırmızı led
  pinMode(6, OUTPUT); // Sarı led
  pinMode(7, OUTPUT); // Yeşil led
}

void loop() {
  digitalWrite(5, HIGH); // Kırmızı yanacak
  delay(2000); // 4 sn bekleyecek
  digitalWrite(5, LOW); // Kırmızı sönecek
  digitalWrite(6, HIGH); // Sarı yanacak
  delay(1000); // 1 sn bekleyecek
  digitalWrite(6, LOW); // Sarı sönecek
  digitalWrite(7, HIGH); // Yeşil yanacak
  delay(2000); // 3 sn bekleyecek
}
```

```
digitalWrite(7, LOW); // Yeşil sönecek  
}
```



▪ KARAŞİMŞEK UYGULAMASI:

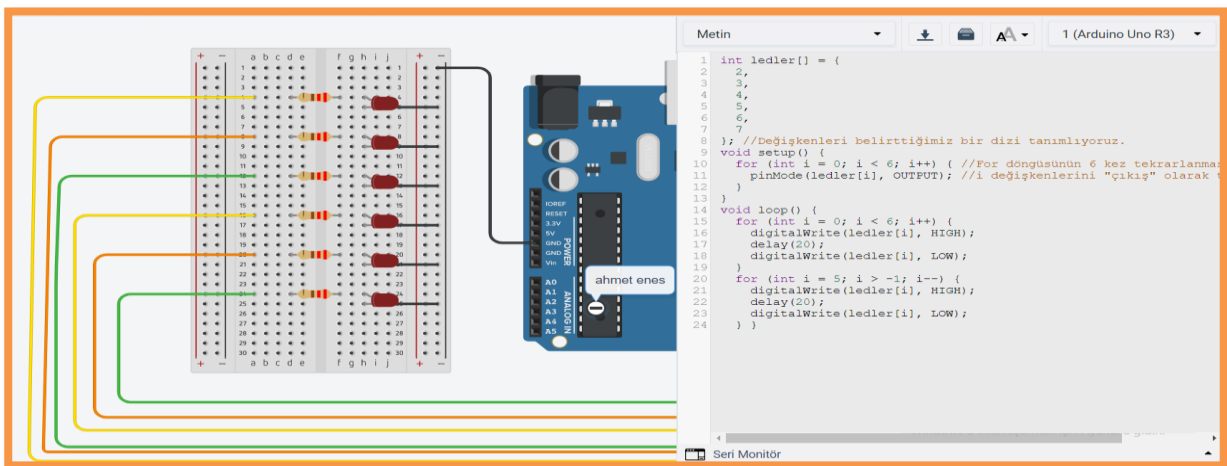
UYGULAMA KOD YAZILIMI:

```
void setup() {  
  pinMode (2, OUTPUT);  
  pinMode (3, OUTPUT);  
  pinMode (4, OUTPUT);  
  pinMode (5, OUTPUT);  
  pinMode (6, OUTPUT);  
  pinMode (7, OUTPUT);  
  pinMode (8, OUTPUT);  
  pinMode (9, OUTPUT);  
  pinMode (10, OUTPUT);  
  pinMode (11, OUTPUT);  
}  
void loop() {  
  digitalWrite (2, HIGH);  
  delay (120);  
  digitalWrite (2, LOW);  
  digitalWrite (3, HIGH);  
  delay (120);  
  digitalWrite (3, LOW);  
  digitalWrite (4, HIGH);  
  delay (120);  
  digitalWrite (4, LOW);  
  digitalWrite (5, HIGH);  
  delay (120);  
  digitalWrite (5, LOW);  
  digitalWrite (6, HIGH);  
  delay (120);  
  digitalWrite (6, LOW);  
  digitalWrite (7, HIGH);  
  delay (120);  
  digitalWrite (7, LOW);
```

```

digitalWrite (8, HIGH);
delay (120);
digitalWrite (8, LOW);
digitalWrite (9, HIGH);
delay (120);
digitalWrite (9, LOW);
digitalWrite (10,HIGH);
delay (120);
digitalWrite (10, LOW);
digitalWrite (11, HIGH);
delay (120);
digitalWrite (11, LOW);
digitalWrite (10, HIGH);
delay (120);
digitalWrite (10, LOW);
digitalWrite (9, HIGH);
delay (120);
digitalWrite (9, LOW);
digitalWrite (8, HIGH);
delay (120);
digitalWrite (8, LOW);
digitalWrite (7, HIGH);
delay (120);
digitalWrite (7, LOW);
digitalWrite (6, HIGH);
delay (120);
digitalWrite (6, LOW);
digitalWrite (5, HIGH);
delay (120);
digitalWrite (5, LOW);
digitalWrite (4, HIGH);
delay (120);
digitalWrite (4, LOW);
digitalWrite (3, HIGH);
delay (120);
digitalWrite (3, LOW);
digitalWrite (2, HIGH);
delay (120);
digitalWrite (2, HIGH);
}

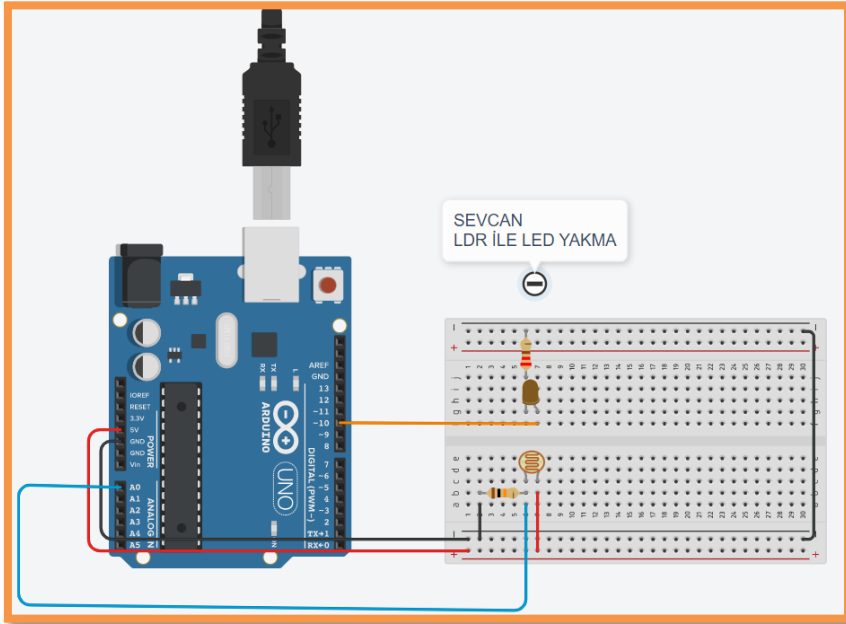
```



▪ **LDR ile LED YAKMA:**

UYGULAMA KOD YAZILIMI:

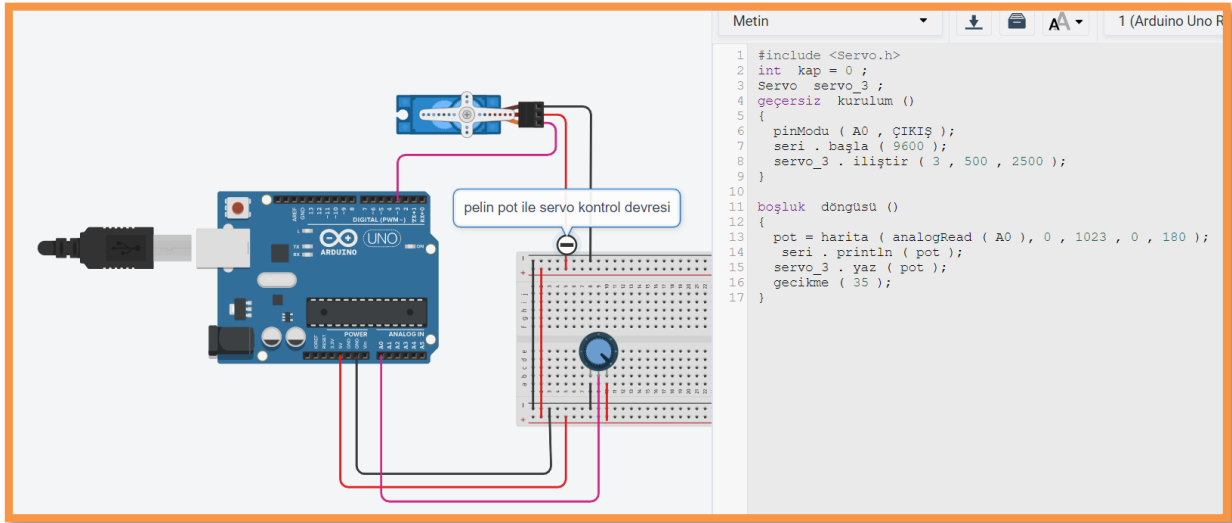
```
#define ldr A0
#define led 10
int vldr=0;
void setup()
{
  pinMode(ldr,INPUT);
  pinMode(led,OUTPUT);
  Serial.begin(9600);
}
void loop()
{
  vldr=analogRead(ldr);
  if(vldr<700)
  {
    digitalWrite(led,1);
  }
  else
  {
    digitalWrite(led,0);
  }
  Serial.println(vldr);
  delay(100);
}
```



▪ POTANSİYOMETRE İLE SERVO MOTOR KONTROLÜ:

UYGULAMA KOD YAZILIMI:

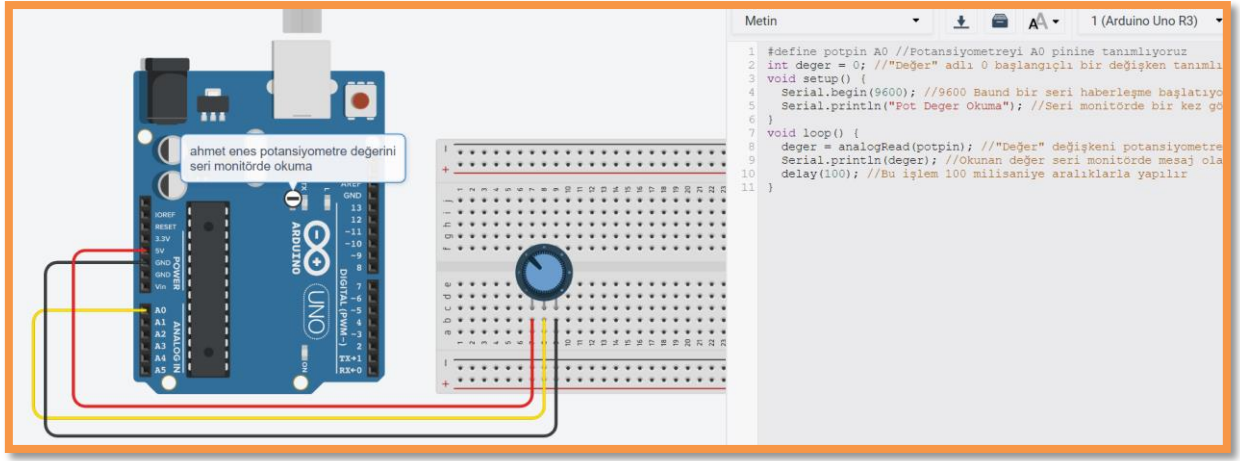
```
int pot=0;
Servo servo_3;
void setup()
{
  pinMode(A0, OUTPUT);
  Serial.begin(9600);
  servo_3.attach(3,500,2500);
}
void loop()
{
  pot=map(analogRead(A0),0,1023,0,180);
  Serial.println(pot);
  servo_3.write(pot);
  delay(35);
}
```



▪ POTANSİYOMETRE/SERİ MONİTÖRDE DEĞER OKUMA:

UYGULAMA KOD YAZILIMI:

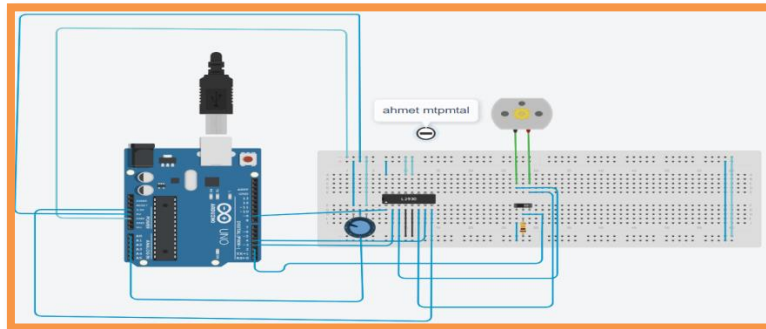
```
#define potpin A0 //Potansiyometreyi A0 pinine tanımlıyoruz
int deger = 0; //"Değer" adlı 0 başlangıçlı bir değişken tanımlıyoruz
void setup() {
  Serial.begin(9600); //9600 Baund bir seri haberleşme başlatıyoruz
  Serial.println("Pot Deger Okuma"); //Seri monitörde bir kez gönderilen bir mesajı tanımlıyoruz
}
void loop() {
  deger = analogRead(potpin); //"Değer" değişkeni potansiyometrenin değerini okur ve değişiklikler olur
  Serial.println(deger); //Okunan değer seri monitörde mesaj olarak gönderilir
  delay(100); //Bu işlem 100 milisaniye aralıklarla yapılır
}
```



▪ L293D MOTOR SÜRÜCÜ ENTEGRESİ KULLANMA:

UYGULAMA KOD YAZILIMI:

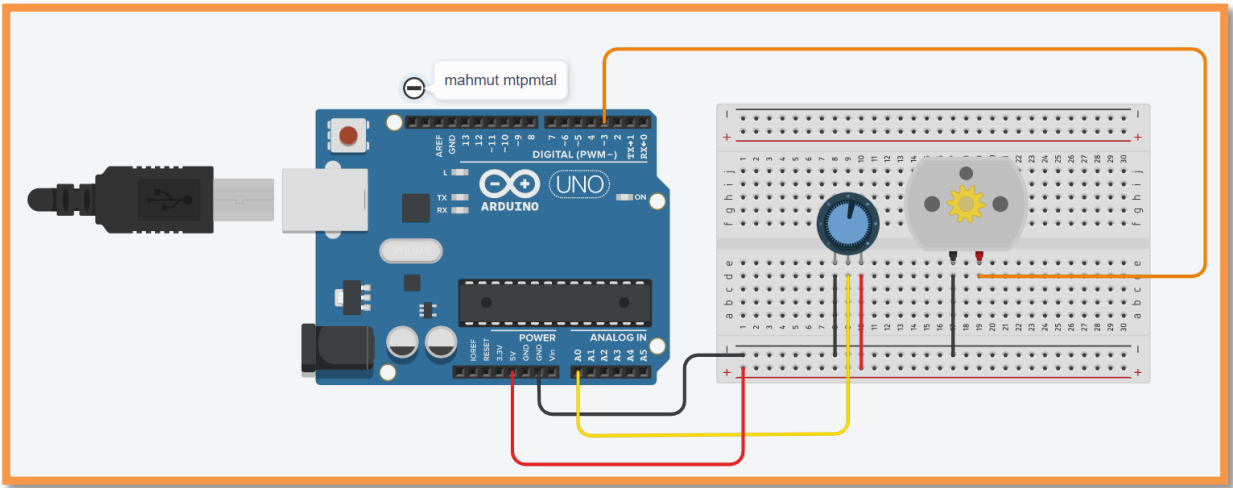
```
#define switchPin 2
#define motorPin1 3
#define motorPin2 4
#define speedPin 9
#define potPin 0
int Mspeed = 0;
void setup() {
  pinMode(switchPin, INPUT);
  pinMode(motorPin1, OUTPUT);
  pinMode(motorPin2, OUTPUT);
  pinMode(speedPin, OUTPUT);
}
void loop() {
  Mspeed = analogRead(potPin)/4;
  analogWrite(speedPin, Mspeed);
  if (digitalRead(switchPin))
  {
    digitalWrite(motorPin1, LOW);
    digitalWrite(motorPin2, HIGH);
  }
  else
  {
    digitalWrite(motorPin1, HIGH);
    digitalWrite(motorPin2, LOW);
  }
}
```



▪ POTANSİYOMETRE İLE DC MOTOR HIZ KONTROLÜ-1:

UYGULAMA KOD YAZILIMI:

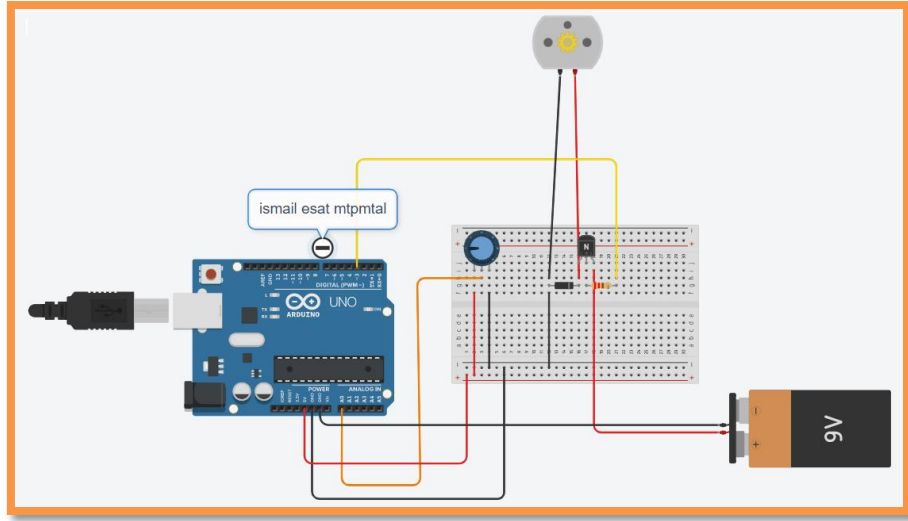
```
void setup()
{
  pinMode(A0, INPUT);
  pinMode(3, OUTPUT);
  Serial.begin(9600);
}
void loop()
{
  analogWrite(3, map(analogRead(A0), 0, 1023, 0, 255));
  Serial.println(map(analogRead(A0), 0, 1023, 0, 255));
  delay(10);
}
```



▪ POTANSİYOMETRE İLE DC MOTOR HIZ KONTROLÜ-2:

UYGULAMA KOD YAZILIMI:

```
int pot = A0;
int motor = 3;
int potDeger;
int devir;
void setup() {
  pinMode(pot, INPUT);
  pinMode(motor, OUTPUT);
}
void loop() {
  potDeger = analogRead(pot);
  devir= map(potDeger, 0, 1023, 0, 255);
  analogWrite(motor,devir);
}
```

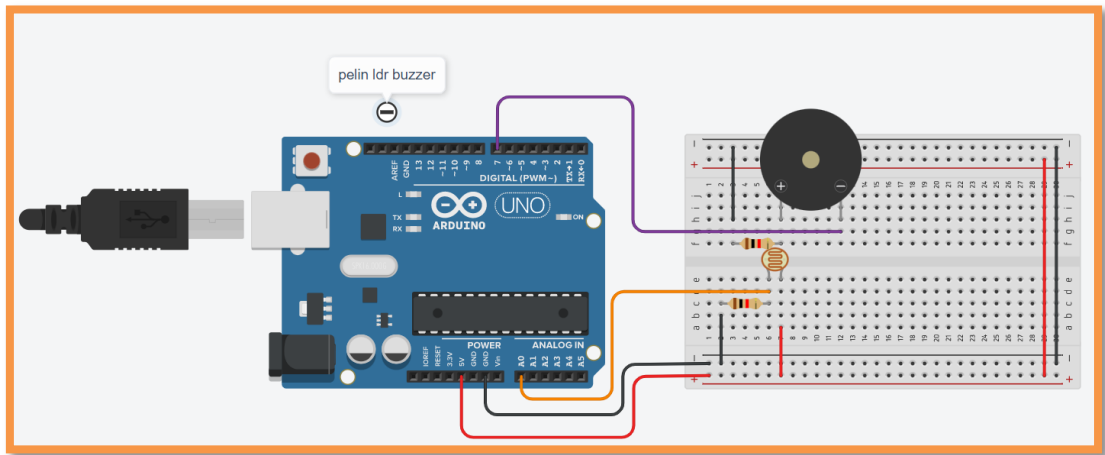
▪ **LDR İLE BUZZER UYGULAMASI:**

UYGULAMA KOD YAZILIMI:

```

int sensor = A0;
int ledverde=7;
void setup()
{
  Serial.begin (9600);
  pinMode(ledverde,OUTPUT);
}
void loop()
{
  int valor;
  valor = analogRead (sensor);
  Serial.println (valor);
  delay (500);
  if (valor>500)
  { digitalWrite (ledverde,HIGH);}
  else
  {
  digitalWrite (ledverde,LOW);}
}

```



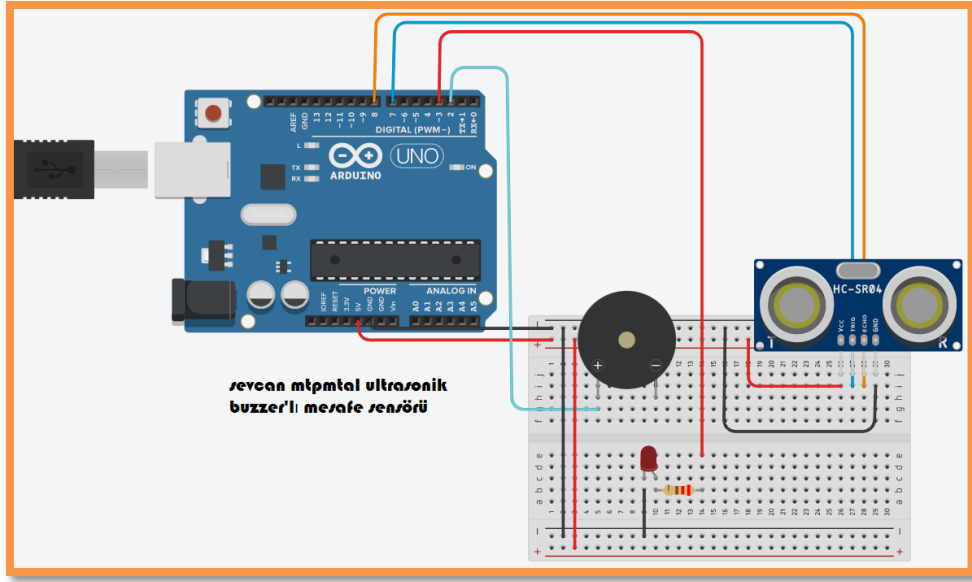
▪ ULTRASONİK MESAFE SENSÖRÜ VE SESLİ UYARILI PARK SİSTEMİ:

UYGULAMA KOD YAZILIMI:

```
int tetikpin = 7;
int alicipin = 8;
int buzzer =2;
int led=3;
float sure;
float mesafe;
void setup() {
  Serial.begin(9600);
  pinMode(tetikpin,OUTPUT);
  pinMode(alicipin,INPUT);
  pinMode(led,OUTPUT);
  pinMode(buzzer,OUTPUT);
}
void loop() {

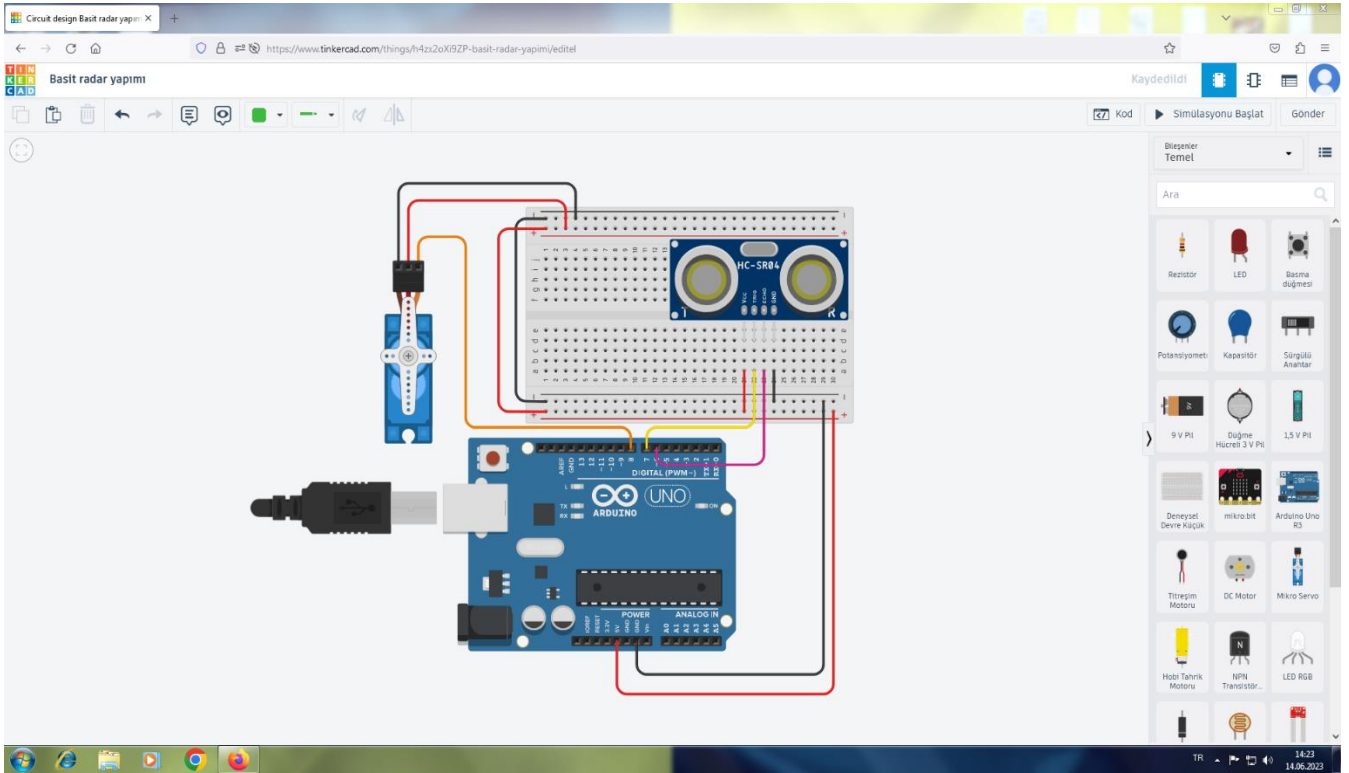
  digitalWrite(tetikpin,HIGH);
  delayMicroseconds(1000);
  digitalWrite(tetikpin,LOW);
  sure = pulseIn(alicipin,HIGH);
  mesafe = (sure/2) / 29.1; //20 derece için

  if(mesafe<=50)
  {
    digitalWrite(buzzer,HIGH);
    digitalWrite(led,HIGH);
    Serial.print("Uzaklik =");
    Serial.print(mesafe);
    Serial.print(" buzzer ve led calisiyor ");
    Serial.println("");
  }
  else
  {
    digitalWrite(buzzer,LOW);
    digitalWrite(led,LOW);
    Serial.print("Uzaklik =");
    Serial.print(mesafe);
    Serial.print(" buzzer ve led calismiyor ");
    Serial.println("");
  }
}
```



ÖRNEK UYGULAMALAR

1. Basit Radar Yapımı



Ultrasonik mesafe sensörü olan HC-SR04, ortama ses dalgaları gönderir. Nesnelere çarpan ses dalgaları tekrar ultrasonik sensöre ulaşır. Sinyalin havadaki hızına bağlı olarak nesnenin mesafesi tespit edilmiş olur. Ultrasonik sensör, servo motor üzerine monte edildikten sonra yazılan kodlarla sistemin yaklaşık 180 derecelik bir açıda tarama yapması ve bu alandaki nesnelerin yerlerini tespit etmesi sağlanır.

Gerekli Malzemeler:

- 1 adet Arduino R3 kart

- 1 adet HC-SR04 ultrasonik mesafe sensörü
- 1 adet sensör tutucu
- 1 adet SG90 Servo motor
- Bağlantı kabloları (jumper)

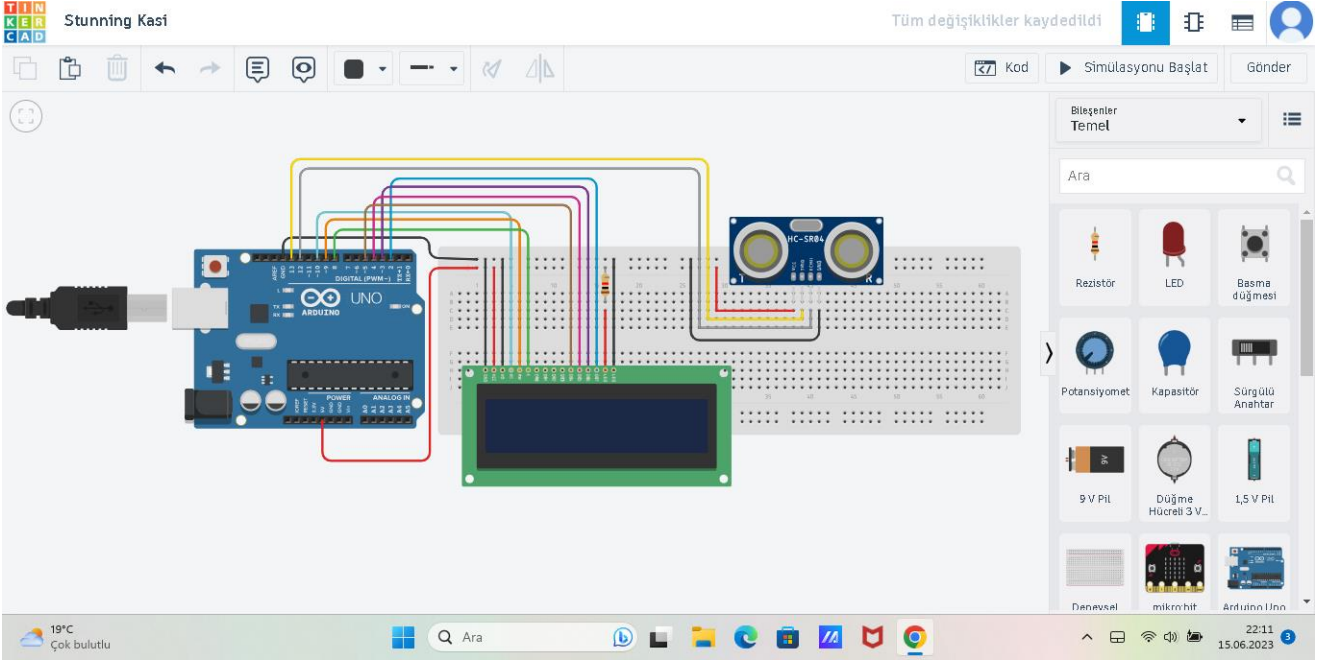
Devre Bağlantıları:

- 1) Servo motorun kahverengi kablosunu Arduino kartın GND pinine, kırmızı kablosunu 5V pinine ve sarı kabloyu da 8 numaralı pine bağlayalım.
- 2) Ultrasonik mesafe sensörünün GND ucunu, Arduino kartın GND pinine, trig ucunu 7 numaralı pine, echo ucunu 8 numaralı pine ve VCC ucunu da 5V pinine bağlayalım.
- 3) Bağlantıları tamamladıktan sonra Arduino IDE programı üzerinden kod yazımına geçelim.

Devre Kodları:

```
#include<Servo.h>
Servo sMotor;
#define trig 7
#define echo 6
void setup(){
  pinMode(trig,OUTPUT);
  pinMode(echo,INPUT);
  sMotor.attach(8);
}
void loop(){
int olcum=mesafe();
  if(olcum!=0){
    if(olcum<=75){
      sMotor.write(70);
      delay(1000);
      sMotor.write(20);
      delay(1000);
    }
  }
}
int mesafe(){
  long sure,uzaklik;
  digitalWrite(trig,LOW);
  delayMicroseconds(2);
  digitalWrite(trig,HIGH);
  delayMicroseconds(10);
  digitalWrite(trig,LOW);
  sure=pulseIn(echo,HIGH);
  uzaklik=sure/58.2;
  delay(50);
  return uzaklik;
}
```

2. Uzunlukölçer Yapımı



Bu projede mesafe ölçümü yapmamızı sağlayan sensör olan HC-SR04, ultrasonik mesafe sensörüdür. Çalışma mantığı olarak ultrasonik ses dalgaları yayar ve bu dalgaları geri toplar. Gönderilen ultrasonik ses dalgaları bir yüzeye çarptıktan sonra geri dönerler. Geri dönen dalgaları yakalayan bu sensör elde ettiği verileri Arduino karta aktarır. Arduino'da işlenen bu veriler ölçülen değerlerin santimetre cinsinden verilere dönüştürülmesi ile anlamlı hale gelir. Bu şekilde dijital bir metre elde etmiş oluruz.

Gerekli Malzemeler:

- 1 adet Arduino Uno R3
- 1 adet deneysel devre (Breadboard)
- 1 adet 2x16 LCD Ekran
- 1 adet ultrasonik Mesafe Sensörü (HC-SR04)
- 1 adet direnç (220Ω)
- Jumper kablolar

Devre Bağlantıları:

- 1) HC-SR04 ultrasonik mesafe sensörünün trig pinini Arduino 13 numaralı pine, echo pinini Arduino 12 numaralı pine bağlayalım.
- 2) I2C LCD ekranın RS pinini Arduino 10 numaralı pine, RW pinini Arduino 9 numaralı pine, E pinini Arduino 8 numaralı pine, DB4 pinini Arduino 5 numaralı pine, DB5 pinini Arduino 4 numaralı pine, DB6 pinini Arduino 3 numaralı pine, DB7 pinini Arduino 2 numaralı pine bağlayalım.
- 3) Hem ultrasonik mesafe sensöründeki hem de I2C LCD ekrandaki VCC pinini Arduino 5V pinine, GND pinini de Arduino GND pinine bağlıyoruz.

Devre Kodları:

```
int maxMesafe = 200;
int minMesafe = 5;
#include <LiquidCrystal.h>
LiquidCrystal LCD(10,9,8,5,4,3,2);

#define trigPin 13
```

```

#define echoPin 12

void setup()
{
  Serial.begin (9600);
  pinMode(trigPin, OUTPUT);
  pinMode(echoPin, INPUT);

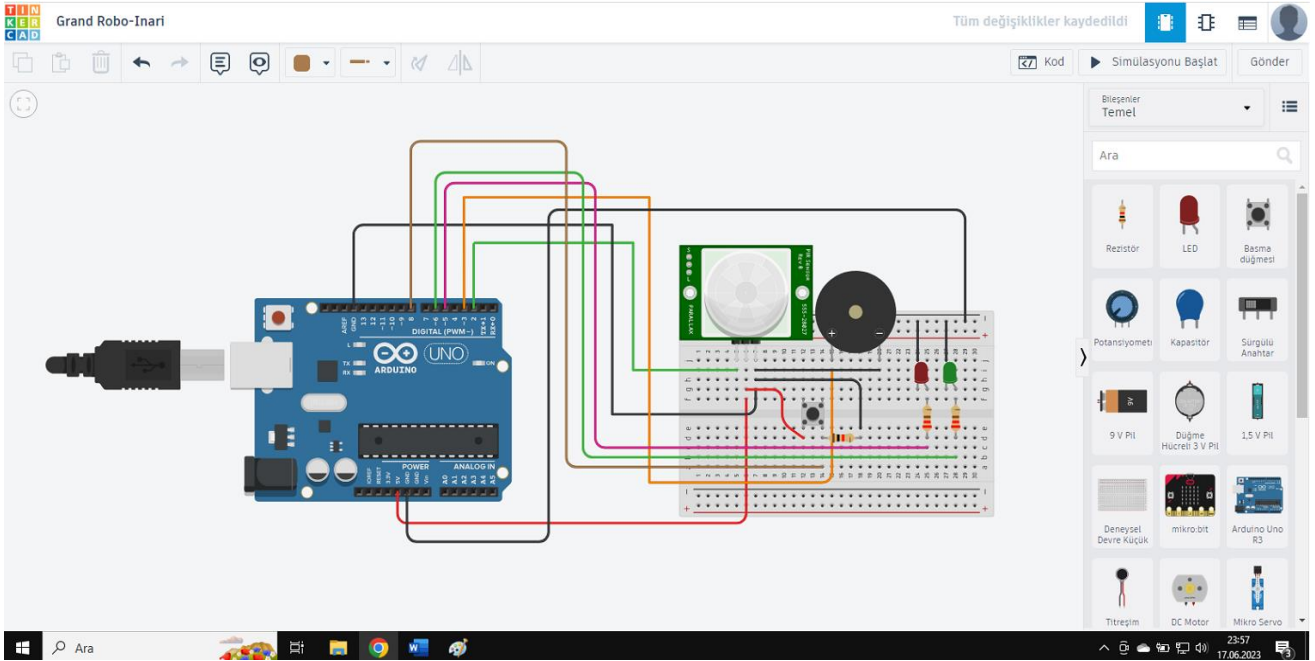
  LCD.begin(16,2);
  LCD.setCursor(0,0);
  LCD.print("DISTANCE");
}
void loop() {
  long duration, distance;
  digitalWrite(trigPin, LOW);
  delayMicroseconds(2);
  digitalWrite(trigPin, HIGH);
  delayMicroseconds(10);
  digitalWrite(trigPin, LOW);
  duration = pulseIn(echoPin, HIGH);
  distance = (duration/2) / 29.1;

  if (distance >= maxMesafe)
  {
    Serial.println("- Mesafe Uzak! -");
    LCD.setCursor(0,1);

    LCD.print("- Mesafe Uzak! -");
    LCD.print("          ");
    delay(500);
  }
  else if (distance <= minMesafe)
  {
    Serial.println("- Mesafe Yakin -");
    LCD.setCursor(0,1);
    LCD.print("- Mesafe Yakin -");
    LCD.print("          ");
    delay(500);
  }
  else {
    LCD.setCursor(0,1);
    LCD.print("          ");
    LCD.setCursor(0,1);
    LCD.print("Mesafe: ");
    LCD.setCursor(7,1);
    LCD.print(distance);
    LCD.print("cm");
    delay(500);
  }
}

```

3. Hırsız Alarmı



Hareket sensörü (PIR sensör), ortamdaki sıcaklık ve kızılötesi dalga değişimlerini algılamaya yarayan bir sensördür. Ortamda bir dalga değişimi olduğunda bunu algılar. Bu projede PIR sensör, ortamda bir hareket olduğunda bunu algılayarak buzzerın ses çıkarmasını ve ışıkların yanıp sönmelerini sağlayarak uyarı verecektir. Alarmı kapatmak için butona basılmalıdır. Hareket yok iken her şeyin yolunda olduğunu gösteren yeşil renkli led sürekli ışık verecektir.

Gerekli malzemeler:

- 1 adet Arduino Uno R3
- 1 adet deneysel devre (Breadboard)
- 1 adet hareket sensörü (PIR sensör)
- 1 adet buzzer
- 1 adet buton
- 2 adet led
- 3 adet direnç (220Ω)
- Jumper kablolar

Devre Bağlantıları:

- 1) Hareket sensörünün sinyal ucunu Arduino 2 numaralı pine, güç ucunu Arduino 5V pinine, toprak (GND) ucunu Arduino GND pinine bağlayalım.
- 2) Buzzerın (+) ucunu Arduino 3 numaralı pine, (-) ucunu GND pinine, Butonun bir ucuna direnç bağlayıp bu ucu GND hattına ve yine aynı hizadan bir kablo ile Arduino 8 numaralı pine bağlayalım.
- 3) Kırmızı ve yeşil ledlerin (-) uçları Arduino GND pinine, kırmızı ledin (+) ucunu Arduino 5 numaralı pine, yeşil ledin (+) ucunu Arduino 6 numaralı pine balayalım.

Devre Kodları:

```
int sensor = 2;  
int buzzer = 3;  
int kirmizi = 5;  
int yesil = 6;  
int buton = 8;  
bool alarmDurum = false;
```

```

void setup() {
  pinMode(sensor,INPUT);
  pinMode(kirmizi,OUTPUT);
  pinMode(yesil,OUTPUT);
  pinMode(buton,INPUT);

}

void loop() {
  if(digitalRead(sensor) == 1){
    alarmDurum = true;
  }

  if(digitalRead(buton) == 1){
    alarmDurum = false;
  }

  if(alarmDurum == false){
    digitalWrite(yesil,HIGH);
    digitalWrite(kirmizi,LOW);
    noTone(buzzer);
  }else{
    digitalWrite(yesil,LOW);
    digitalWrite(kirmizi,HIGH);
    tone(buzzer,250);
    delay(100);
    digitalWrite(yesil,HIGH);
    digitalWrite(kirmizi,LOW);
    noTone(buzzer);
    delay(100);
  }
}

```


KAYNAKÇA

<https://www.hosting.com.tr/blog/arduino/>

<https://www.robocombo.com/blog/icerik/arduino-nedir-arduino-ile-neler-yapilir>

<https://maker.robotistan.com/arduino-nedir-ne-ise-yarar/#:~:text=Arduino%2C%20a%C3%A7%C4%B1k%20kaynakl%C4%B1%20bir%20elektronik,eri%C5%9Filebilir%20hale%20getirmek%20i%C3%A7in%20tasarlanm%C4%B1%C5%9Ft%C4%B1r.>

<https://blog.direnc.net/led-nedir-nerelerde-kullanilir-cesitleri-ve-sembolleri/>

<https://www.pino.io/blogs/blog/blok-ve-metin-tabanlı-kodlama>

<https://www.dudigan.com/2015/12/arduino-komutlari/>

<https://www.2zsyazilim.com/blog/arduino/arduinoda-kullanilan-kodlar/>

https://cagataymuhammet.github.io/dosyalar/temel_elektronik_ve_arduino.pdf

https://esenyurtkatipcelebiaihl.meb.k12.tr/meb_iys_dosyalar/34/39/766148/dosyalar/2022_03/13144210_DEVRE-SIMULASYONU-VE-DEVRE-CIZIMLERI.pdf

GÖREV DAĞILIMI

KONULAR	OKUL ADI
Arduino'nun Tanımı,Tarihçesi ve Temel Giriş-Çıkış Bilgileri	Baki,Mekteb-Lisey Kompleksi
Arduino'nun Text Tabanlı Kodlanması	Fatih MTAL
Thinkercad nedir? Devre Simülasyonu nedir?Program Arayüzü	Gazi Anadolu Lisesi
Thinkercad ile adım adım devre oluşturma	Fatma Hacı Hüseyin Akgül MTAL
Thinkercad ile Arduino Uygulamaları Oluşturma	Mehmet Tuza Pakpen MTAL
Projemizde yaptığımız uygulamalar hariç 3tane örnek uygulama	Atakent Şehit Selçuk Pakcer Anadolu Lisesi