

**Bu kitaba sığmayan  
daha neler var!**



Karekodu okutun, bu kitapla ilgili EBA içeriklerine ulaşın!

**eba**  
www.eba.gov.tr



**ÖDS**

**ÖĞRENCİ/ÖĞRETMEN  
DESTEK SİSTEMİ**

<https://ods.eba.gov.tr>

- Konu Anlatımlı Ders Videoları
- Soru Çözüm Videoları
- Ders Anlatım Videoları
- Çoktan Seçmeli Sorular



**BU DERS KİTABI MİLLÎ EĞİTİM BAKANLIĞINCA  
ÜCRETSİZ OLARAK VERİLMİŞTİR.  
PARA İLE SATILAMAZ.**

ISBN: 978-975-11-6234-2

Bandrol Uygulamasına İlişkin Usul ve Esaslar Hakkında Yönetmelik'in 5'inci Maddesinin İkinci Fıkrası Çerçevesinde Bandrol Taşınması Zorunlu Değildir.

BİLİŞİM TEKNOLOJİLERİ ALANI

ROBOTİK VE KODLAMA

10 DERS MATERYALI

MESLEKİ VE TEKNİK ANADOLU LİSESİ

BİLİŞİM TEKNOLOJİLERİ ALANI

ROBOTİK VE KODLAMA

10 DERS  
MATERYALI





MESLEKİ VE TEKNİK ANADOLU LİSESİ  
BİLİŞİM TEKNOLOJİLERİ ALANI

# ROBOTİK VE KODLAMA

# 10

Ders Materyali

## YAZARLAR

Aydın ALTIN

Erdal AKPINAR

Habibe KARAYİĞİT

İsa TETİK



MİLLÎ EĞİTİM BAKANLIĞI YAYINLARI .....	8042
YARDIMCI VE KAYNAK KİTAPLAR DİZİSİ .....	1970

Her hakkı saklıdır ve Millî Eğitim Bakanlığına aittir. Ders materyalinin metin, soru ve şekilleri kısmen de olsa hiçbir surette alınıp yayımlanamaz.

## HAZIRLAYANLAR

Dil Uzmanı

**Osman Nuri GÜVEN**

Program Geliştirme Uzmanı

**Büşra KİREZ**

Ölçme ve Değerlendirme Uzmanı

**Arzu DURSUN URGUN**

Rehberlik Uzmanı

**Gülşen YALIN**

Görsel Tasarım Uzmanı

**Adike Candan DOĞRUÖZ**

ISBN: 978-975-11-6234-2

Millî Eğitim Bakanlığının 24.12.2020 gün ve 18433886 sayılı oluru ile Meslekî ve Teknik Eğitim Genel Müdürlüğünce ders materyali olarak hazırlanmıştır.



## İSTİKLÂL MARŞI

Korkma, sönmez bu şafaklarda yüzen al sancak;  
Sönmeden yurdumun üstünde tüten en son ocak.  
O benim milletimin yıldızıdır, parlayacak;  
O benimdir, o benim milletimindir ancak.

Çatma, kurban olayım, çehreni ey nazlı hilâl!  
Kahraman ırkıma bir gül! Ne bu şiddet, bu celâl?  
Sana olmaz dökülen kanlarımız sonra helâl.  
Hakkıdır Hakk'a tapan milletimin istiklâl.

Ben ezelden beridir hür yaşadım, hür yaşarım.  
Hangi çılgın bana zincir vuracakmış? Şaşarım!  
Kükremiş sel gibiyim, bendimi çiğner, aşarım.  
Yırtarım dağları, enginlere sığmam, taşarım.

Garbın âfâkını sarmışsa çelik zırhlı duvar,  
Benim iman dolu göğsüm gibi serhaddim var.  
Ulusun, korkma! Nasıl böyle bir imanı boğar,  
Medeniyet dediğin tek dişi kalmış canavar?

Arkadaş, yurduma alçakları uğratma sakın;  
Siper et gövdeni, dursun bu hayâsızca akın.  
Doğacaktır sana va'dettiği günler Hakk'ın;  
Kim bilir, belki yarın, belki yarından da yakın.

Bastığın yerleri toprak diyerek geçme, tanı:  
Düşün altındaki binlerce kefensiz yatanı.  
Sen şehit oğlusun, incitme, yazıktır, atanı:  
Verme, dünyaları alsan da bu cennet vatanı.

Kim bu cennet vatanın uğruna olmaz ki feda?  
Şüheda fışkıracak toprağı sıksan, şüheda!  
Cânı, cânânı, bütün varımı alsın da Huda,  
Etmesin tek vatanımdan beni dünyada cüda.

Ruhumun senden İlahî, şudur ancak emeli:  
Değmesin mabedimin göğsüne nâmahrem eli.  
Bu ezanlar -ki şehadetleri dinin temeli-  
Ebedî yurdumun üstünde benim inlemeli.

O zaman vecd ile bin secde eder -varsa- taşım,  
Her cerîhamdan İlahî, boşanıp kanlı yaşım,  
Fışkırır ruh-ı mücerret gibi yerden na'sım;  
O zaman yükselerek arşa değer belki başım.

Dalgalan sen de şafaklar gibi ey şanlı hilâl!  
Olsun artık dökülen kanlarımın hepsi helâl.  
Ebediyyen sana yok, ırkıma yok izmihlâl;  
Hakkıdır hür yaşamış bayrağımın hürriyyet;  
Hakkıdır Hakk'a tapan milletimin istiklâl!

**Mehmet Âkif Ersoy**

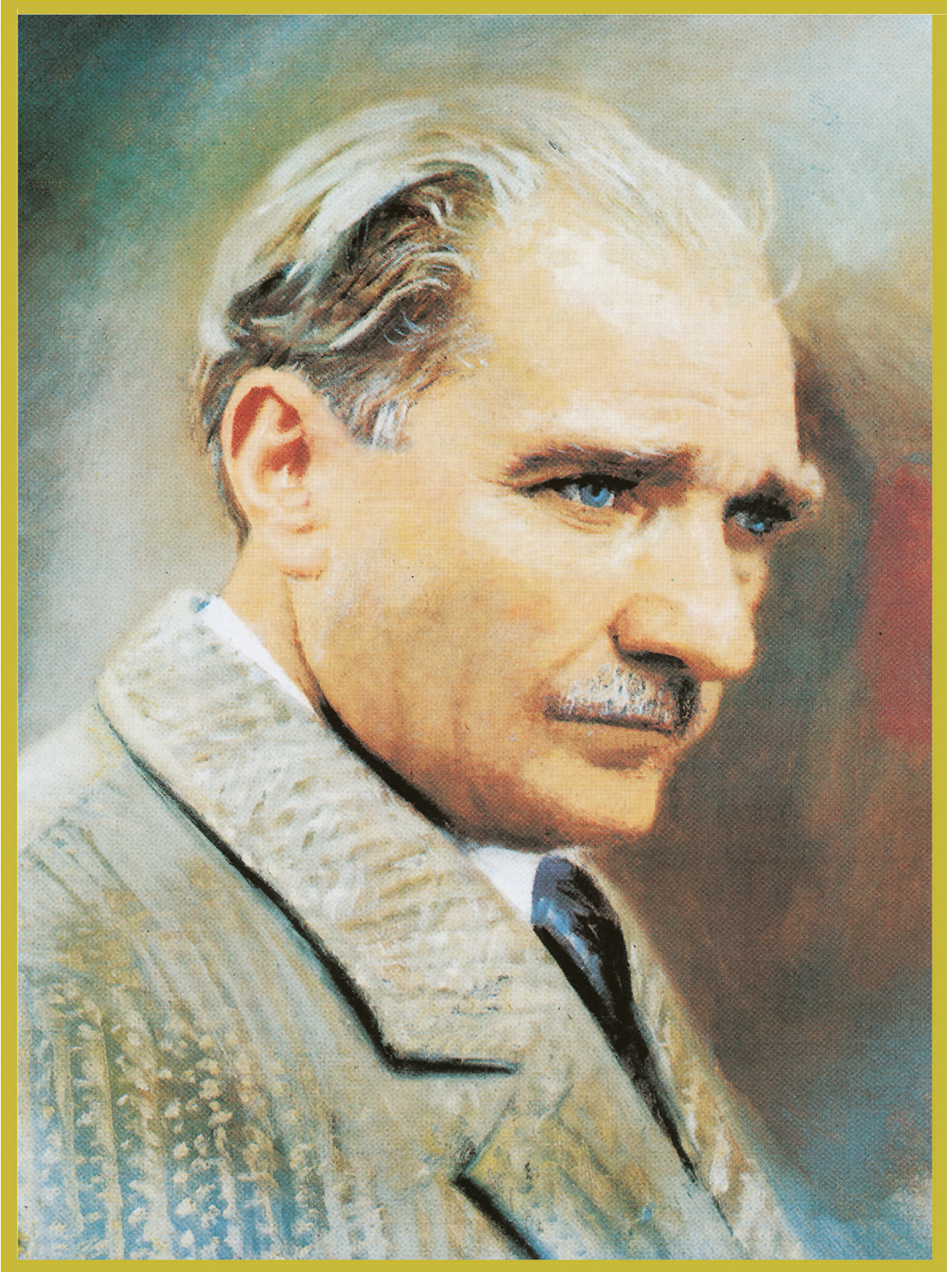
## GENÇLİĞE HİTABE

Ey Türk gençliği! Birinci vazifen, Türk istiklâlini, Türk Cumhuriyetini, ilelebet muhafaza ve müdafaa etmektir.

Mevcudiyetinin ve istikbalinin yegâne temeli budur. Bu temel, senin en kıymetli hazinendir. İstikbalde dahi, seni bu hazineden mahrum etmek isteyecek dâhilî ve hâricî bedhahların olacaktır. Bir gün, istiklâl ve cumhuriyeti müdafaa mecburiyetine düşersen, vazifeye atılmak için, içinde bulunacağın vaziyetin imkân ve şeraitini düşünmeyeceksin! Bu imkân ve şerait, çok namüsaît bir mahiyette tezahür edebilir. İstiklâl ve cumhuriyetine kastedecek düşmanlar, bütün dünyada emsali görülmemiş bir galibiyetin mümessili olabilirler. Cebren ve hile ile aziz vatanın bütün kaleleri zapt edilmiş, bütün tersanelerine girilmiş, bütün orduları dağıtılmış ve memleketin her köşesi bilfiil işgal edilmiş olabilir. Bütün bu şeraitten daha elîm ve daha vahim olmak üzere, memleketin dâhilinde iktidara sahip olanlar gaflet ve dalâlet ve hattâ hıyanet içinde bulunabilirler. Hattâ bu iktidar sahipleri şahsî menfaatlerini, müstevlîlerin siyasî emelleriyle tevhit edebilirler. Millet, fakr u zaruret içinde harap ve bîtap düşmüş olabilir.

Ey Türk istikbalinin evlâdı! İşte, bu ahval ve şerait içinde dahi vazifen, Türk istiklâl ve cumhuriyetini kurtarmaktır. Muhtaç olduğun kudret, damarlarındaki asil kanda mevcuttur.

Mustafa Kemal Atatürk



MUSTAFA KEMAL ATATÜRK





## İÇİNDEKİLER

DERS MATERYALİNİN TANITIMI.....	11
---------------------------------	----

# 1. ÖĞRENME BİRİMİ

<b>ROBOTİK İÇİN MİKRODENETLEYİCİ KART .....</b>	<b>16</b>
---	-----------

<b>1.1. MİKRODENETLEYİCİ KART YAPISI VE ÇEŞİTLERİ .....</b>	<b>18</b>
---	-----------

1.1.1. Mikrodenetleyici Kartlar ve Yapısı .....	18
---	----

1.1.2. Uygulama Kartı ve Çeşitleri .....	20
--	----

1.1.3. Devre Tasarım Programı Kullanarak Devre Simülasyonu.....	21
---	----

<b>1.2. ROBOT TÜRLERİ VE EĞİTSEL AMAÇLI ROBOTLAR .....</b>	<b>28</b>
--	-----------

1.2.1. Robot Kavramı .....	28
----------------------------	----

1.2.2. Robot Türleri .....	29
----------------------------	----

1.2.3. Eğitsel Amaçlı Robotlar .....	32
--------------------------------------	----

<b>1.3. ROBOTTA MEKANİK / ELEKTROMEKANİK BİLEŞENLER.....</b>	<b>37</b>
--	-----------

1.3.1. Robot Gövdeleri .....	37
------------------------------	----

1.3.2. Motorlar .....	38
-----------------------	----

1.3.3. Tekerlek, Ayak ve Paletler .....	38
---	----

1.3.4. Eklenti ve Bağlantı Bileşenleri.....	39
---	----

1.3.5. Vida, Somun ve Rondela Bileşenleri.....	39
--	----

1.3.6. Amortisör, Yay ve Esnek Bileşenler .....	40
---	----

1.3.7. Mekanik veya Vakumlu Nesne Tutucu Bileşenler .....	40
---	----

<b>1.4. ROBOTTA ELEKTRONİK BİLEŞENLER.....</b>	<b>43</b>
--	-----------

1.4.1. Motor Sürücü Kartları .....	43
------------------------------------	----

1.4.1.1. DC Motor Sürücü Kartları .....	43
---	----

1.4.1.2. Servo Motor Sürücü Kartları .....	43
--	----

1.4.1.3. Özel Robot Kartları .....	44
1.4.2. Sensörler.....	45
1.4.3. Kablosuz Erişim Kartları.....	45
<b>ÖLÇME VE DEĞERLENDİRME .....</b>	<b>47</b>
SIRA SİZDE KONTROL LİSTELERİ VE DEĞERLENDİRME ÖLÇEKLERİ ...	50

## 2. ÖĞRENME BİRİMİ

<b>MİKRODENETLEYİCİ KART PROGRAMLAMA.....</b>	<b>52</b>
<b>2.1. SİSTEM GEREKSİNİMLERİNE UYGUN MİKRODENETLEYİCİ KART</b>	<b>54</b>
<b>2.2. MİKRODENETLEYİCİ KARTIN BİLGİSAYAR BAĞLANTISI</b>	
<b>VE ÖRNEK PROGRAM YÜKLENMESİ .....</b>	<b>58</b>
<b>2.3. MİKRODENETLEYİCİ KART İLE LED UYGULAMALARI.....</b>	<b>60</b>
2.3.1. Mikrodenetleyici Kart ve Bir Adet LED'in Kullanımı.....	62
2.3.2. Mikrodenetleyici Kart ile 2 LED'in Kullanımı .....	64
2.3.3. Mikrodenetleyici Kart ile 5 LED'in Kullanımı .....	66
2.3.4. Mikrodenetleyici Kart ile 7 LED'in Kullanımı (for döngüsü ile) .....	68
2.3.5. Mikrodenetleyici Kart ile Trafik Lambası Kontrolü .....	70
<b>2.4. MİKRODENETLEYİCİ KART İLE DİJİTAL GİRİŞ UYGULAMALARI....</b>	<b>75</b>
2.4.1. Anahtar Kullanımı.....	75
2.4.2. Buton Kullanımı.....	80
<b>2.5. MİKRODENETLEYİCİ KART İLE SERİ PORT UYGULAMALARI .....</b>	<b>85</b>
<b>2.6. MİKRODENETLEYİCİ KART İLE DİZİ UYGULAMALARI .....</b>	<b>89</b>
<b>2.7. MİKRODENETLEYİCİ KART İLE ANALOG GİRİŞ UYGULAMALARI..</b>	<b>97</b>
2.7.1. Potansiyometre ile LED'lerin Yanıp Sönme Hızının Ayarlanması .....	101
2.7.2. Potansiyometre Seviyesinin LED ile Gösterilmesi .....	103
<b>2.8. IŞIK ETKİLİ DİRENÇ (LDR) UYGULAMALARI .....</b>	<b>104</b>

<b>2.9. MİKRODENETLEYİCİ KART İLE ANALOG ÇIKIŞ (PWM)</b>	
<b>UYGULAMALARI</b> .....	<b>110</b>
<b>2.10. MİKRODENETLEYİCİ KART İLE RGB LED UYGULAMALARI</b> .....	<b>115</b>
<b>2.11. MİKRODENETLEYİCİ KART İLE BUZZER UYGULAMALARI</b> .....	<b>123</b>
<b>2.12. MİKRODENETLEYİCİ KART İLE 7 SEGMENT DİSPLAY</b>	
<b>UYGULAMALARI</b> .....	<b>132</b>
<b>2.13. MİKRODENETLEYİCİ KART İLE IR ALICI UYGULAMALARI</b> .....	<b>147</b>
2.13.1. Uzaktan Kumandanın Kodunun Çözülmesi .....	148
2.13.2. RGB LED ile Uzaktan Kumandanın Kullanımı .....	153
2.13.3. 7 Segment Display'in Uzaktan Kumanda ile Kullanımı .....	158
<b>2.14. MİKRODENETLEYİCİ KART İLE MESAFE SENSÖRÜ</b>	
<b>UYGULAMALARI</b> .....	<b>164</b>
<b>2.15. MİKRODENETLEYİCİ KART İLE LCD UYGULAMALARI</b> .....	<b>171</b>
2.15.1. LCD Bağlantıları.....	172
2.15.2. LCD Ekranda Yazılan Yazının Sağa veya Sola Kaydırılması.....	175
2.15.3. LDR ile Ölçülen Işık Şiddetinin LCD Ekranda Gösterilmesi .....	179
2.15.4. Sıcaklık Sensörü ile Okunan Değerlerin LCD Ekranda Gösterilmesi.....	181
2.15.5. Mesafe Sensöründe Okunan Değerlerin LCD Ekranda Gösterilmesi .....	184
<b>ÖLÇME VE DEĞERLENDİRME</b> .....	<b>187</b>
<b>SIRA SİZDE KONTROL LİSTESİ</b> .....	<b>189</b>

## **3. ÖĞRENME BİRİMİ**

<b>ROBOT TABANLI PROJE GELİŞTİRME</b> .....	<b>190</b>
<b>3.1. EĞİTSEL ROBOT BİLEŞENLERİ</b> .....	<b>192</b>
3.1.1. Robot Gövdesi .....	193
3.1.2. Mikrodenetleyici Kartı.....	193

3.1.3. Motorlar .....	194
3.1.4. Tekerlekler.....	196
3.1.5. Motor Sürücü .....	197
3.1.6. Enerji Kaynağı.....	198
<b>3.2. EĞİTSEL ROBOTUN DEVRE ŞEMASI.....</b>	<b>198</b>
<b>3.3. EĞİTSEL ROBOTUN MONTAJI .....</b>	<b>199</b>
<b>3.4. MOTORLARI SADECE GERİ YÖNDE ÇALIŞTIRMA.....</b>	<b>202</b>
<b>3.5. EĞİTSEL ROBOTUN PROGRAMLANMASI .....</b>	<b>204</b>
<b>3.6. EĞİTSEL ROBOTUN UZAKTAN KUMANDA İLE KONTROLÜ .....</b>	<b>212</b>
<b>3.7. EĞİTSEL ROBOT İLE ENGELDEN KAÇMA.....</b>	<b>218</b>
<b>3.8. EĞİTSEL ROBOT İLE ÇİZGİ İZLEME.....</b>	<b>222</b>
<b>3.9. EĞİTSEL ROBOTUN BLUETOOTH KONTROLÜ .....</b>	<b>230</b>
<b>3.10. SERVO MOTOR İLE ROBOT UYGULAMALARI .....</b>	<b>238</b>
3.10.1. Açık Vererek Servo Motor Kontrolü.....	239
3.10.2. Potansiyometre ile Servo Motor Kontrolü .....	241
<b>ÖLÇME VE DEĞERLENDİRME.....</b>	<b>246</b>
SIRA SİZDE KONTROL LİSTESİ .....	247
EK – 1: ROBOT GÖVDESİNİN ÖLÇÜLERİ .....	248
EK – 2: MESAFE SENSÖRÜ APARATININ ÖLÇÜLERİ .....	250
EK – 3: ÇİZGİ İZLEME SENSÖRÜ APARATININ ÖLÇÜLERİ.....	251
CEVAP ANAHTARLARI .....	252
GÖRSEL KAYNAKÇASI .....	253

## DERS MATERYALİNİN TANITIMI

Öğrenme biriminin konularını gösterir.

Öğrenme biriminin adını gösterir.

Etkileşimli kitap, video, ses, animasyon, uygulama, oyun, soru vb. ilave kaynaklara ulaşabileceğiniz karekodu gösterir.

**KONULAR**

- 1.1. MİKRODENETLEYİCİ KART YAPISI VE ÇEŞİTLERİ
- 1.2. ROBOT TÜRLERİ VE EĞİTSEL AMAÇLI ROBOTLAR
- 1.3. ROBOTTA MEKANİK / ELEKTROMEKANİK BİLEŞENLER
- 1.4. ROBOTTA ELEKTRONİK BİLEŞENLER

**NELER ÖĞRENECEKSİNİZ?**

- Mikrodenetleyici ve mikrodenetleyicinin önemi
- Mikrodenetleyicinin robotik sistemlerdeki görevi
- Mikrodenetleyici seçiminin nelere göre yapılacağı
- Uygulama kartı kavramının robotik projelerdeki önemi
- Uygulama kartı seçimi
- Simülasyon programıyla uygulama kartını ve kodları çalıştırma
- Robot kavramı, robot türleri ve eğitilmiş robotların farkı
- Robot tasarımı ve çalışmalarını için gerekli ortamı sağlama

**Hazırlık Çalışmaları**

1. Çevrenizde elektrikle çalışan makinelerden birinin görevini açıklayınız.
2. Gelecekte robotlar dünyamızı ele geçirebilir mi? Bu konuda neler düşünüyorsunuz?
3. BTT dersinde öğrendiğiniz bilgisayar donanım parçası mikrodenetleyicinin görevlerinden hatırladıklarınızı söyleyiniz.
4. Web 2.0 araştırmaları araştırarak edindiğiniz bilgileri sınıfınızda arkadaşlarınızla paylaşınız.

**1. ÖĞRENME BİRİMİ**

Robotik İçin Mikrodenetleyici Kart

**Anahtar Kelimeler**

- Mikrodenetleyici kartı
- Uygulama kartları
- Robotlar
- Robot türleri
- Eğitilmiş robotlar

Derse başlamadan yapılacak olan hazırlıkları gösterir.

Öğrenme biriminde neler öğrenileceğini gösterir.

Öğrenme birimindeki anahtar kelimeleri gösterir.

Konu başlığını gösterir.

Alt konu başlıklarını gösterir.

Görselleri gösterir.

Etkileşimli kitap, video, ses, animasyon, uygulama, oyun, soru vb. ilave kaynaklara ulaşabileceğiniz karekodu gösterir.

Program kodlarını gösterir.

Öğrencilerin yapacağı uygulamaları gösterir.

Yaptığınız uygulamaların değerlendirmelerini gösterir.

**1. Öğrenme Birimi: Robotik için Mikrodenetleyici Kart**

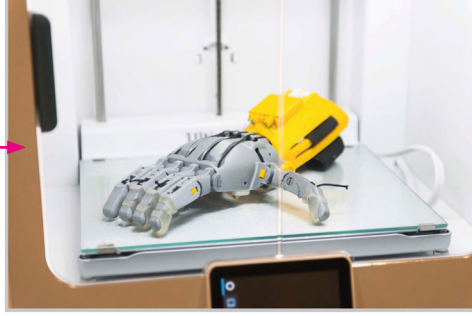
### 1.3. ROBOTTA MEKANİK / ELEKTROMEKANİK BİLEŞENLER

Robotlar, programlanabilen elektronik kartlar haricinde gövdesini ve hareketli parçalarını oluşturan plastik veya metal bileşenlerden meydana gelmektedir. Robotun **mekanik / elektromekanik bileşenleri** 7 kısımda incelenebilir:

- Robot gövdeleri
- Motorlar
- Tekerlek, ayak ve paletler
- Ekilenti ve bağlantı bileşenleri
- Vida, somun ve rondela bileşenleri
- Amortisör, yay ve esnek bileşenler
- Mekanik veya vakumlu nesne tutucu bileşenler

#### 1.3.1. Robot Gövdeleri

Robotun tüm bileşenlerini üzerinde bulunduran yapı; plastik, pleksi glass, ağaç, metal gibi malzemelerden üretilmiş en temel robot bileşendir. Görsel 1.34'te üzerine motorlar monte edilmiş, 3 boyutlu yazıcıyla üretilmiş bir robot el modeli görülmektedir.



Görsel 1.34: 3 boyutlu yazıcıyla üretilmiş bir robot el modeli

Bilişim Teknolojileri Alanı 35


**2. Öğrenme Birimi: Mikrodenetleyici Kart Programlama**

### 3. Adım: Mikrodenetleyici kart ile 7 LED'li uygulamanın kodları aşağıdaki gibidir.

```
int i;
void setup() {
  for (i=2; i<=8; i++) pinMode(i, OUTPUT);
}
void loop() {
  for (i=2; i<=8; i++) { digitalWrite(i, 1); delay(1000); }
  for (i=2; i<=8; i++) { digitalWrite(i, 0); delay(1000); }
}
```

**Sıra Sizde**

Görsel 2.21 veya Görsel 2.22'de devresi verilen 7 LED'li uygulama devre elemanlarını breadboard üzerine yerleştiriniz. Mikrodenetleyici kart ile 7 LED'li uygulamanın program kodlarını mikrodenetleyici IDE programında yazınız. Tüm işlemler bittikten sonra programı mikrodenetleyici karta yükleyip, devrenin çalışmasını gözlemleyiniz.

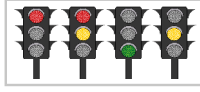


**Değerlendirme**

Çalışmalarınız öğrenme biriminin sonunda yer alan Kontrol Listesi kullanılarak değerlendirilecektir. Çalışmalarınızı yaparken değerlendirme ölçütlerini dikkate alınız.

**2.3.5. Mikrodenetleyici Kart ile Trafik Lambası Kontrolü**

Bu bölümde trafik lambası uygulaması yapılacaktır. Kavşaklarda ve yaya geçitlerinde trafikteki geçişlerin düzenli bir şekilde olmasını sağlamak için trafik lambaları kullanılır. Görsel 2.23'te bir trafik ışığı sinyalizasyon devresinde trafik ışıklarının yanıp sönmeye sıralaması görülmektedir.



Görsel 2.23: Trafik ışıklarının yanıp sönmeye sıralaması

**7. Uygulama**

Trafik ışıklarının yanıp sönmeye sıralamasını aşağıdaki adımları takip ederek yapınız.

**Gerekli Devre Elemanları**

- 1 adet mikrodenetleyici kart
- 1 adet breadboard
- 3 adet LED
- 3 adet 220Ω (kırmızı, kırmızı, kahverengi, altın rengi) direnç
- Bağlantı kabloları

Bilişim Teknolojileri Alanı 70

Yapılacak örnek uygulamaları gösterir.

Önemli notları gösterir.

## 2. Öğrenme Birimi: Mikrodenetleyici Kart Programlama

**Sıra Sizde**

Görsel 2.66 veya Görsel 2.67'de devresi verilen ortak katotlu RGB LED uygulamasının uygulamasını breadboard üzerine kurunuz. Bu uygulamaya ait program kodlarını mikrodenetleyici IDE programında yazınız. Tüm işlemler bittikten sonra programı mikrodenetleyici karta yükleyip, devrenin çalışmasını gözlemleyiniz.

**Değerlendirme**

Çalışmalarınız öğrenme biriminin sonunda yer alan Kontrol Listesi kullanılarak değerlendirilecektir. Çalışmalarınızı yaparken değerlendirme ölçütlerini dikkate alınız.

**27. Uygulama**

Bu uygulamadaki amaç, 3 adet potansiyometre ile ortak katotlu RGB LED ile renklerin parlaklığını kontrol eden uygulamayı gerçekleştirmektir.

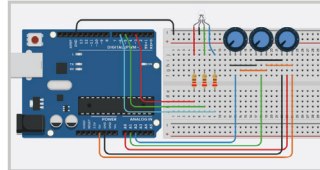
**Gerekli Devre Elemanları**

- 1 adet mikrodenetleyici kart
- 1 adet breadboard
- 3 adet potansiyometre
- 1 adet ortak katotlu RGB LED
- 3 adet 220Ω (kırmızı, kırmızı, kahverengi, altın rengi) direnç
- Bağlantı kabloları

**1. Adım:** Görsel 2.68 ve Görsel 2.69'da 3 adet potansiyometre ile ortak katotlu RGB LED ile renklerin parlaklığını kontrol eden uygulama görülmektedir. Uygulamada, ortak anotlu RGB LED kullanılırsa LED'in ortak ucu, mikrodenetleyici kartın GND pini yerine 5V pinine bağlanır.

**Önemli**

Uygulamada kullanılan RGB LED çeşidine göre program kodları değişmez.



Görsel 2.68: Ortak katotlu RGB LED'in potansiyometre ile kontrolü uygulama devresi

116 Bilişim Teknolojileri Alanı

Ölçme ve değerlendirme sorularını gösterir.

## 1. Öğrenme Birimi: Robotik İçin Mikrodenetleyici Kart

### ÖLÇME VE DEĞERLENDİRME

- Elektronik ve robotik sistemlerin beyni aşağıdakilerden hangisidir?  
A) ROM  
B) LED'ler  
C) G / Ç Birimi  
D) A / D Çeviriciler  
E) Mikrodenetleyici
- Mikrodenetleyicilerde sistemin çalışması için kaydedilmiş programları tutan birim aşağıdakilerden hangisidir?  
A) ROM
- Uygulama kartı ile mikrodenetleyici arasındaki ilişki hangisi **olmaz**?  
A) Uygulama kartı mikrodenetleyiciyi kapsar.  
B) Mikrodenetleyici uygulama kartının beynidir.  
C) Uygulama kartları mikrodenetleyici olmadan da çalışır.  
D) Uygulama kartları, üzerlerinde farklı birimleri barındırır.  
E) Mikrodenetleyici çeşidi, uygulama kartında hedeflenen amaca göre farklılık gösterebilir.
- Bilişim robotların atası olan ve algı-

Yaptığınız uygulamaların değerlendirme ölçüklerini gösterir.

## 1. Öğrenme Birimi: Robotik İçin Mikrodenetleyici Kart

### SIRA SIZDE KONTROL LİSTELERİ VE DEĞERLENDİRME ÖLÇEKLERİ

1. Değerlendirme Ölçeği

Ölçütler	4 (Çok iyi)	3 (iyi)	2 (Orta)	1 (Zayıf)
Özgün bir proje konusu belirler.				
Proje konusunun ayrıntılarını hazırlar.				
Projesini sunar.				
Zamanı verimli kullanır.				

1. Kontrol Listesi

Ölçütler	Evet	Hayır
Kontrol ilkelerine göre çalışan robotları internetten araştırır.		

# KONULAR

- 1.1. MİKRODENETLEYİCİ KART YAPISI VE ÇEŞİTLERİ
- 1.2. ROBOT TÜRLERİ VE EĞİTSEL AMAÇLI ROBOTLAR
- 1.3. ROBOTA MEKANİK / ELEKTROMEKANİK BİLEŞENLER
- 1.4. ROBOTA ELEKTRONİK BİLEŞENLER

## NELER ÖĞRENECEKSİNİZ?

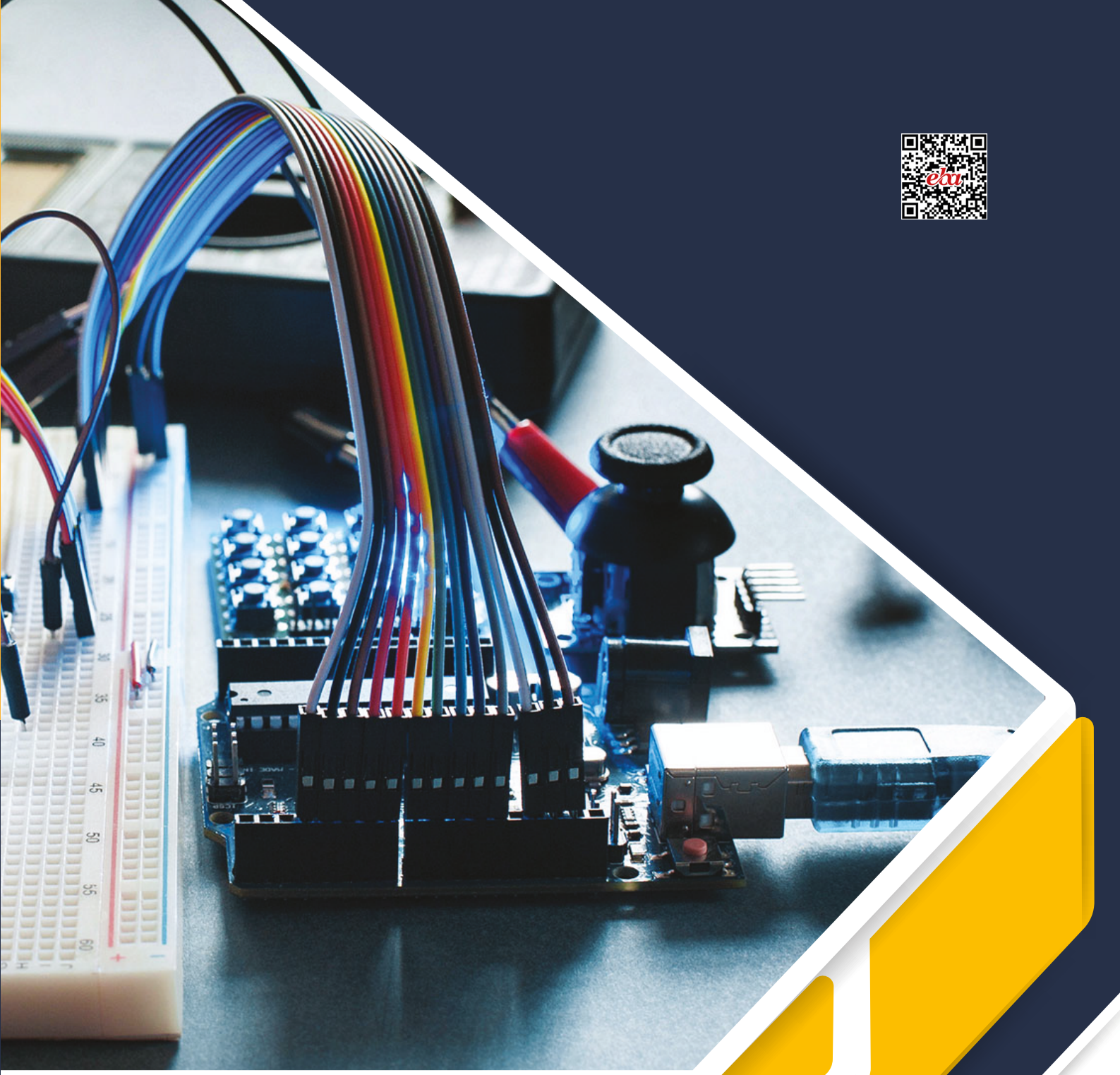
- Mikrodenetleyici ve mikrodenetleyicinin önemi
- Mikrodenetleyicinin robotik sistemlerdeki görevi
- Mikrodenetleyici seçiminin nelere göre yapılacağı
- Uygulama kartı kavramının robotik projelerdeki önemi
- Uygulama kartı seçimi
- Simülasyon programıyla uygulama kartını ve kodları çalıştırma
- Robot kavramı, robot türleri ve eğitsel robotların farkı
- Robot tasarımı ve çalışmaları için gerekli ortamı sağlama



### Hazırlık Çalışmaları

1. Çevrenizde elektrikle çalışan makinelerden birinin görevini açıklayınız.
2. Gelecekte robotlar dünyamızı ele geçirebilir mi? Bu konuda neler düşünüyorsunuz?
3. BTT dersinde öğrendiğiniz bilgisayar donanım parçası mikroişlemcinin görevlerinden hatırladıklarınızı söyleyiniz.
4. Web 2.0 araçlarını araştırarak edindiğiniz bilgileri sınıfınızda arkadaşlarınızla paylaşınız.





# 1. ÖĞRENME BİRİMİ

Robotik için  
Mikrodenetleyici Kart

## Temel Kavramlar

- Mikrodenetleyici kartı
- Uygulama kartları
- Robotlar
- Robot türleri
- Eğitsel robotlar

## 1.1. MİKRODENETLEYİCİ KART YAPISI VE ÇEŞİTLERİ

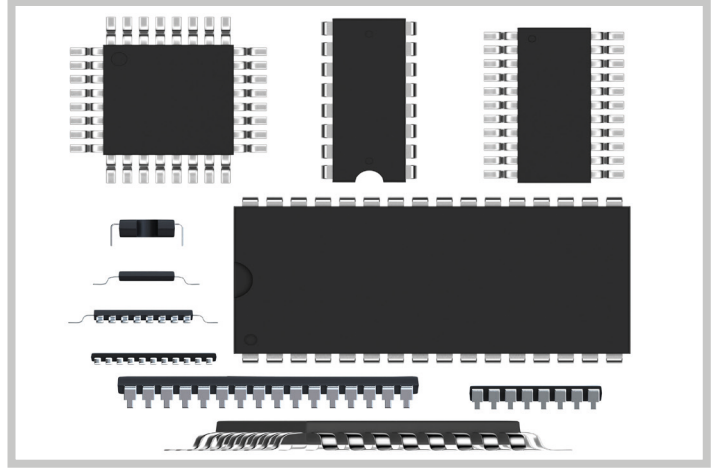
Elektronik makineler günümüzde insan hayatını kolaylaştıran popüler cihazlardır. Elektronik makinelerin beyni olarak nitelendirilen mikrodenetleyiciler, bu cihazların farklı görevleri yerine getirmesini sağlar. Elektronik sistemler, mikrodenetleyiciler ile kontrol edilir. Sadece evlerde değil sağlık, eğitim, finans, gıda, inşaat, otomotiv gibi alanlarda kullanılan robotik teknolojisine sahip cihazların tamamı mikrodenetleyicilerin kontrolünde tüm insanların yaşamını kolaylaştırmaktadır.

### 1.1.1. Mikrodenetleyici Kartlar ve Yapısı

Mikrodenetleyiciler, bir programı hafızasına alarak derleyen ve elektronik cihazın amacına göre sonuçlar elde eden küçük bilgisayarlardır. Mikrodenetleyiciler, çalıştırılması istenen programı hazırlayıp kontrolünü yapabilme yetisine sahiptir ve gerçek zamanlı uygulamaları çalıştırmak için tasarlanmıştır (Görsel 1.1).

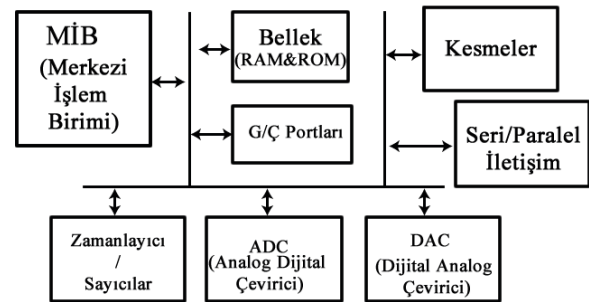
Bir mikrodenetleyicinin yapısında Görsel 1.2'de gösterilen birimler bulunur. Bunlar:

- MİB (Merkezi İşlem Birimi), bellekte sistemin çalışması için kaydedilmiş programların çalıştırılmasını ve diğer birimlerle iletişimini sağlar.
- RAM (Random Access Memory / Rastgele Erişimli Bellek) birimi, mikrodenetleyicilerde bilgileri geçici olarak tutar.
- ROM (Read Only Memory / Sadece Okunabilir Bellek) birimi, mikrodenetleyicilerde sistemin çalışması için kaydedilmiş programları tutar.
- G / Ç Portları, dış ortama gerekli sinyallerin gönderilmesinde veya dış ortamdan istenen sinyallerin alınmasında kullanılır.
- Seri / Paralel iletişim birimlerine, **haberleşme portu** adı da verilir. Seri portlar 9 ya da 25 pinli olabilir. Paralel portlar ise 25 pinli, bilgisayar tarafı dışı olan konnektörlerden oluşmaktadır. Pin sayısı yeterli ve veri aktarılacak mesafe düşük ise paralel bağlantı tipi tercih edilebilir.



Görsel 1.1: Mikrodenetleyiciler

### MİKRODENETLEYİ İÇ YAPISI



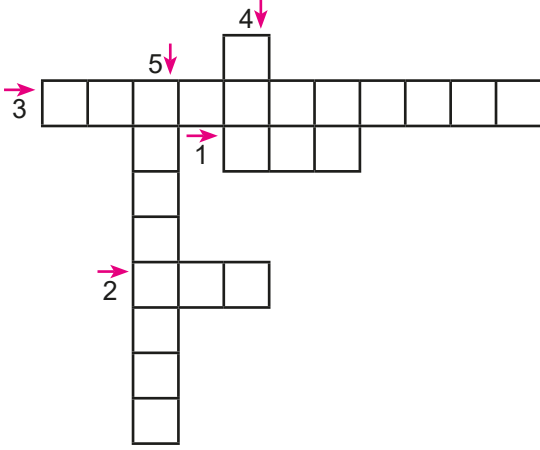
Görsel 1.2: Basit bir mikrodenetleyici iç yapısı

- A / D çeviriciler, çevresel ortamdan alınan analog sinyalleri dijital yani sayısal değerlere çevirirken, D / A çeviriciler sayısal (dijital) değerleri analog sinyallere dönüştürür.
- Zamanlayıcı / Sayıcı birimi mikrodenetleyici içinde zamanlama ve sayma görevlerini gerçekleştirerek program akışını bozmadan kesme işlemlerini yerine getirir. Kesme işlemi tamamlandıktan sonra ana program kaldığı yerden devam eder.



## Sıra Sizde

Tanımları verilen mikrodenetleyici birimlerinin adlarını bulmacadaki ilgili boşluklara yazınız.



1. Mikrodenetleyicide programların çalışmasını sağlayan birimdir.
2. Mikrodenetleyicide bilgileri geçici olarak tutan bellek birimidir.
3. Mikrodenetleyicide dış ortama sinyal gönderir ve dış ortamdan da sinyalleri alır.
4. Mikrodenetleyicide kaydedilmiş programları tutar.
5. Mikrodenetleyicide sinyalleri Analogdan Dijitale veya Dijitalden Analoga çeviren birimlerin genel adıdır.

Mikrodenetleyiciler, yapılarındaki donanım parçalarını (MİB, RAM, ROM, G / Ç birimleri vb.) tuttukları için **gömülü sistemler** olarak da adlandırılır. Genellikle sensörlerden aldıkları çevresel verileri oluşturacakları sistemin amaçlarına uygun şekilde işleyerek çalıştırır.

Bir robotik projesi için mikrodenetleyici seçerken;

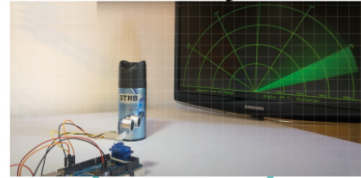
- Kaynak ve kütüphanelerinin çok olmasına,
- Ucuz ve kolay elde edilmesine,
- Programlama kolaylığına,
- Birçok modüle sahip olmasına,
- Mikrodenetleyici ek donanımlarının (shield) olmasına dikkat edilmelidir.



## Sıra Sizde

Görsel 1.3'te gösterilen basit bir radar sistemi yapım süreci şu şekilde olmaktadır: Radar sistemi, tanımlanan mesafelerde önüne gelen nesnelere tanıyan ve bunları yazılı veya görsel şekilde ileten yapılardır. Görsel 1.3'teki gibi basit bir radar

## Radar Projesi!



- MİKRODENETLEYİCİ
- PRO SERVO MOTOR
- 9GHCSR-04 ULTRASONİK SENSÖR
- JUMPER KABLO



Görsel 1.3: Radar projesi için gerekli araçlar

## 1. Öğrenme Birimi: Robotik İçin Mikrodenetleyici Kart

projesi için hareket algılayıcı sensöre, servo motora (dönen kol), kablolar ve radar sisteminin kontrolü için mikrodenetleyiciye ihtiyaç vardır. Kompleks bir uygulama projesi olmadığından pahalı bir mikrodenetleyiciye de ihtiyaç yoktur.

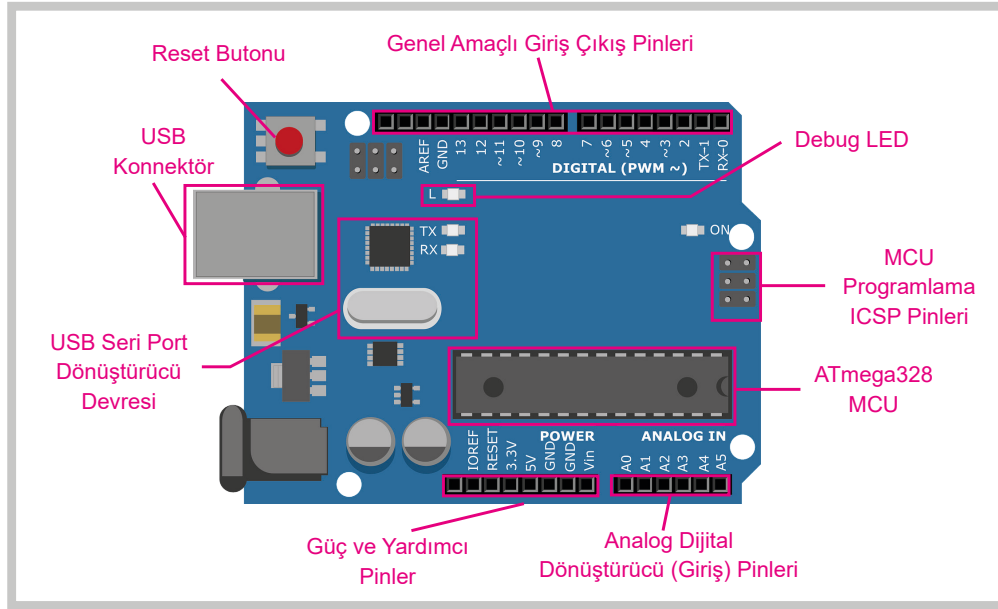
Görsel 1.3'teki gibi projelere ilişkin internette bir araştırma yaparak basit malzemelerden oluşan bir proje belirleyiniz. Görsel 1.3'tekine benzer bir posteri Web 2.0 araçlarından birini kullanarak hazırlayınız.



Hazırlayacağınız çalışma kitap sonunda yer alan 1. Değerlendirme Ölçeği kullanılarak değerlendirilecektir. Çalışmanızı yaparken değerlendirme ölçütlerini dikkate alınız.

### 1.1.2. Uygulama Kartı ve Çeşitleri

Uygulama kartları, mikrodenetleyici yapısına ek olarak projelerin yapımını kolaylaştıran birimleri üzerinde taşır (Görsel 1.4). Uygulama kartı seçilirken uygulamaların fiziki yapımını ve çalışmasını hızlandıracak, pratikliğini arttıracak kartlar seçilmelidir.

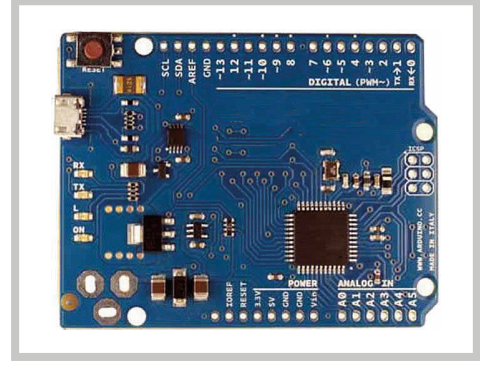


Görsel 1.4: Açık kaynak kod uygulamalarına izin veren pratik bir uygulama kartı

Görsel 1.4'teki gibi kullanım kolaylığına sahip açık kaynak bir uygulama kartının en temel özellikleri şunlardır:

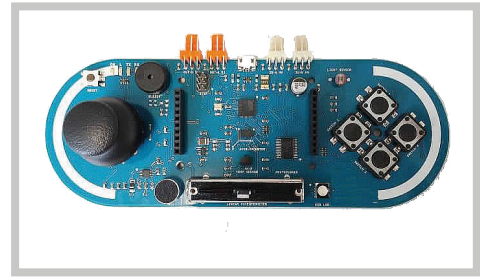
- USB üzerinden programlama,
- Voltaj regülatörü ve güç bağlantısı,
- Tanımlanmış Giriş / Çıkış bağlantıları,
- Debug (Hata ayıklama),
- Güç ve TX / RX(Gönderilen / Alınan veri) LED' ler,
- Reset (Yeniden başlatma) butonu,
- ICSP (In-Circuit Serial Programmer-dâhilî seri programlayıcı) bağlantıları.

İçinde USB özelliği barındıran Görsel 1.5'tekine benzer mikrodenetleyiciye sahip olan bir uygulama kartı, başka bir çipe ihtiyaç duymadan USB üzerinden bağlantısını gerçekleştirerek bilgisayara fare veya klavye olarak bağlanabilir.



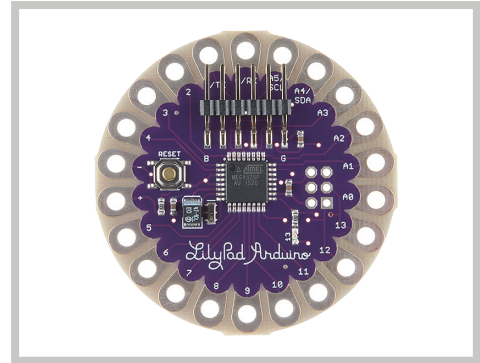
**Görsel 1.5: USB özelliği barındıran mikrodenetleyiciye sahip bir uygulama kartı**

Görsel 1.6'daki gibi bir başka uygulama kartı modelinde ise uygulama kartı üzerinde sensörler bulunmaktadır. Bu uygulama kartıyla birlikte araştırmacılar, eklentilere ihtiyaç duymadan ve fazla bir elektronik bilgi gerektirmeden kolaylıkla çalışmalarını gerçekleştirebilirler. Bu uygulama kartı tiplerinin üzerinde potansiyometre (değeri değişebilen direnç), ışık ve ses sensörü, sıcaklık sensörü, ses üretici, mini analog joystick, renkli LED'ler ve ivmeölçer gibi parçalar dâhil olarak bulunmaktadır. Ayrıca mikrodenetleyicilerin de USB özelliği bulunur.



**Görsel 1.6: Sensörleri üzerinde bulunan pratik bir uygulama kartı**

Giyilebilir cihazlar ve e-tekstil için tasarlanmış bir mikrodenetleyiciden oluşan uygulama kartı Görsel 1.7'de gösterilmektedir. Kumaşa benzer şekilde monte edilmiş güç kaynaklarına, sensörlere ve iletken ipliğe sahip mekanizmaları hareket ettiren sistemlere dikilebilecek şekilde kullanılabilir.



**Görsel 1.7: Giyilebilir cihaz tasarımı için kullanılan bir uygulama kartı**

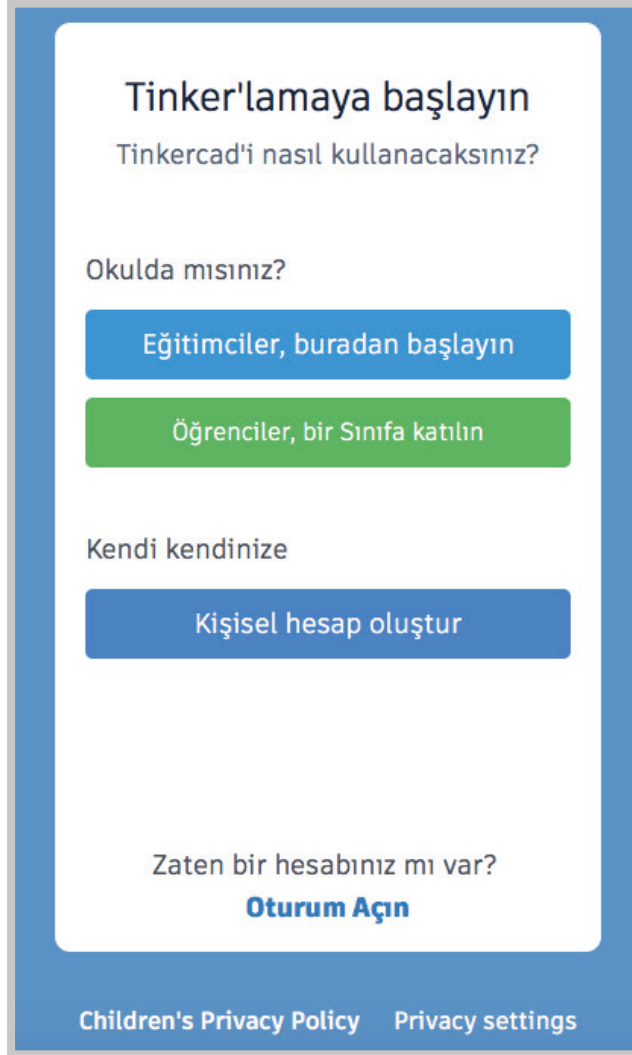
### 1.1.3. Devre Tasarım Programı Kullanarak Devre Simülasyonu

Uygulama kartı ile başlangıç ve orta düzey uygulamalar yapabilmek için elinizde bir uygulama kartı ve fiziki elemanlar olmasına gerek yoktur. Devre simülasyonu programları kullanılarak fiziksel malzemeler olmadan da basit ve orta düzey devreler geliştirilebilir veya fiziki devreler kurulmadan önce kontrol amaçlı bu ortamlar kullanılabilir. Devre simülasyonu için çevrimiçi "Devre Tasarım Programı" kullanılmaktadır. Programın kurulumu ve uygulama devresi oluşturma aşamaları aşağıda anlatılmıştır.

## 1. Uygulama

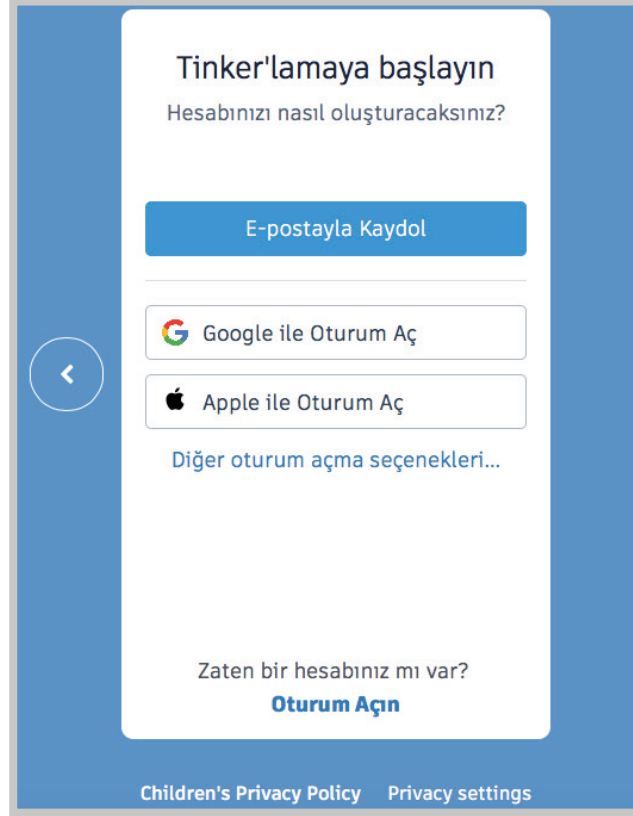
Aşağıdaki adımlara göre Devre Tasarım programı kurulumunu yapınız.

**1. Adım:** Devre Tasarım programı ortamına giriş yapmak için öncelikle üye olmalısınız. Bu aşamadan sonra internet tarayıcısını açınız, adres satırına <https://www.tinkercad.com/> yazınız ve siteye giriş yapıp butonuna tıklayarak, Görsel 1.8'de gösterilen seçeneklerden uygun olan biri ile hesap oluşturunuz ve tasarım ortamına geçiş yapınız.



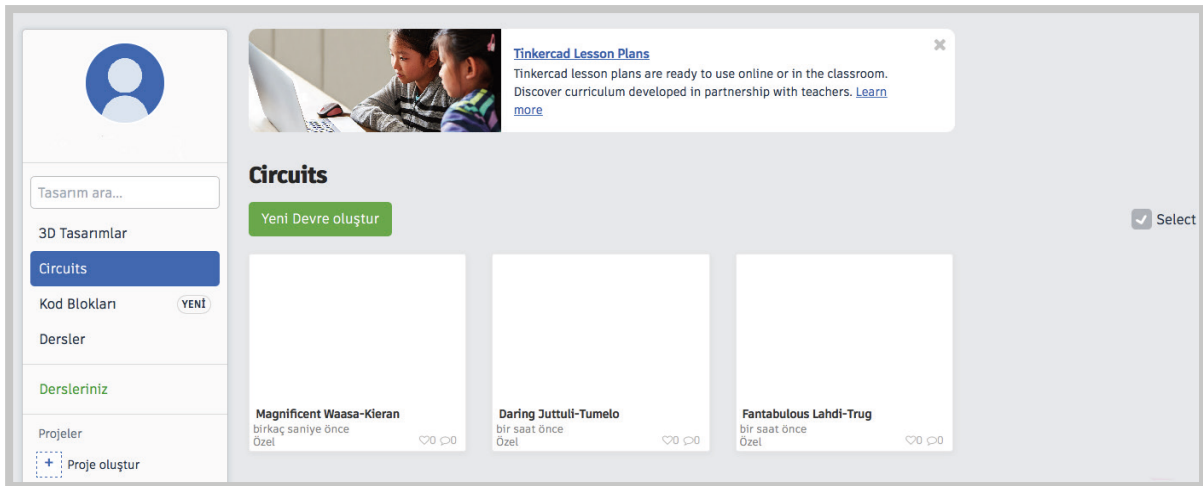
Görsel 1.8: Devre tasarım programı üyelik ekranı

**2. Adım:** Görsel 1.9'da gösterildiği gibi Gmail ve Apple hesaplarıyla oturum açılmışsa kaydolmadan bu hesaplar üzerinden sisteme giriş yapılabilir.



Görsel 1.9: Bilgisayarlı çizim programı hesap ekranı

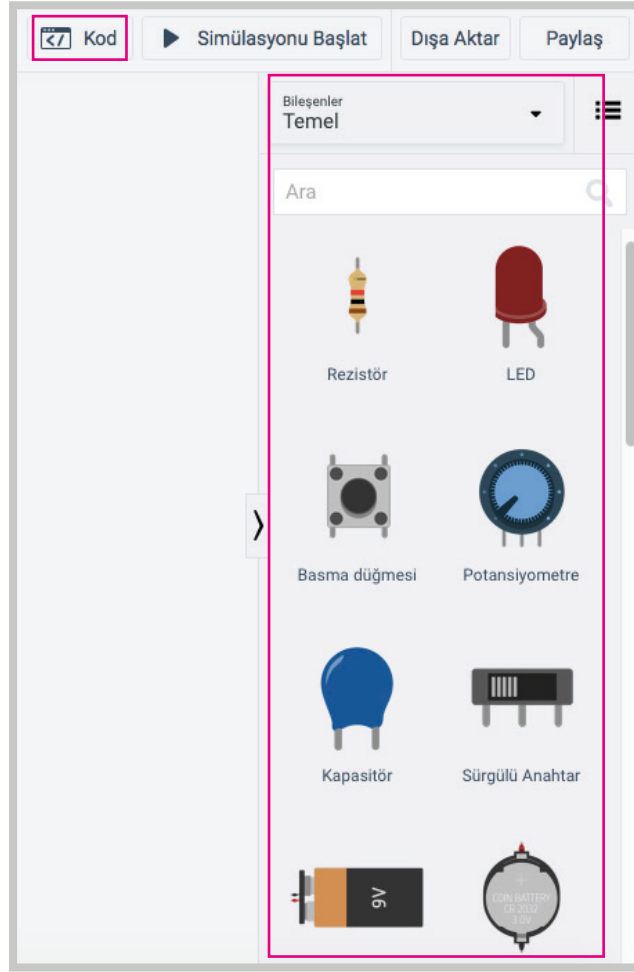
**3. Adım:** Görsel 1.10'da gösterildiği gibi tasarım ortamında öncelikli olarak 3 boyutlu seçenek ekrana gelir. **Devreler (Circuits)** seçeneği ile giriş yapınız. Karşınıza **Devre tasarım ekranı** gelecektir.



Görsel 1.10: Devre tasarım ekranı

## 1. Öğrenme Birimi: Robotik İçin Mikrodenetleyici Kart

**4. Adım:** Görsel 1.11’de gösterildiği gibi devre elemanlarını sürükleyip bırakarak sağdaki devre elemanlarının olduğu panelden tasarım alanına alınız. Devre tasarlandıktan sonra kod alanına giriş yapınız. Blok veya metin tabanlı kodlar oluşturularak **Simülasyonu Başlat** seçeneği ile sonuçları gözlemleyiniz.



Görsel 1.11: Tinkercad ekranı bölümlenmesi

### 2. Uygulama

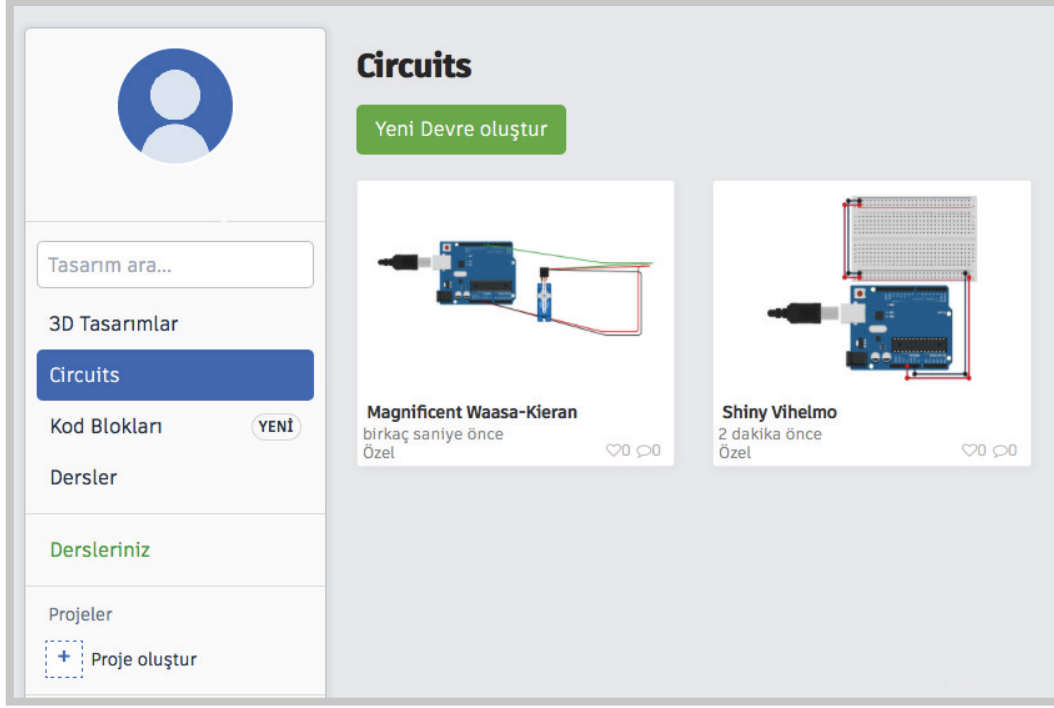
“Devre Tasarım Programı”nda uygulama kartının 13 No.lu pini üzerinde bulunan LED’i yakıp söndüren göz kırpması (blink) uygulamasını yapınız.

### Önemli

LED’ler ışık yayan elemanlardır. Bir kısa bir de uzun bacağına sahiptir. Kısa bacak GND (topraklama), uzun bacak ise dijital girişine bağlanmaktadır. LED’ler 3V ile çalıştılarından uygulama kartının 5V’luk gerilimini dengelemek için direnç (rezistans) elemanları da kullanılmalıdır. LED’in yanıp sönmelerini sağlamak için LED’i bir güç kaynağına bağlamak gereklidir.

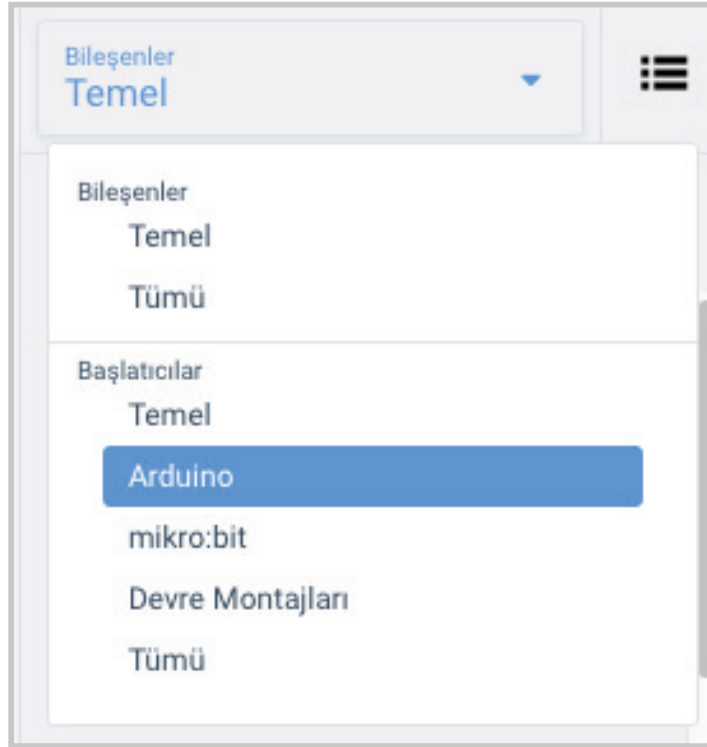


**1. Adım:** Görsel 1.12'de gösterildiği gibi “Devre Tasarım Programı” içeriğinde **Circuits** ekranından **Yeni Devre oluştur** seçeneğini tıklayınız.



Görsel 1.12: Circuits (Devreler) seçeneğinden yeni devre oluşturma işlemi

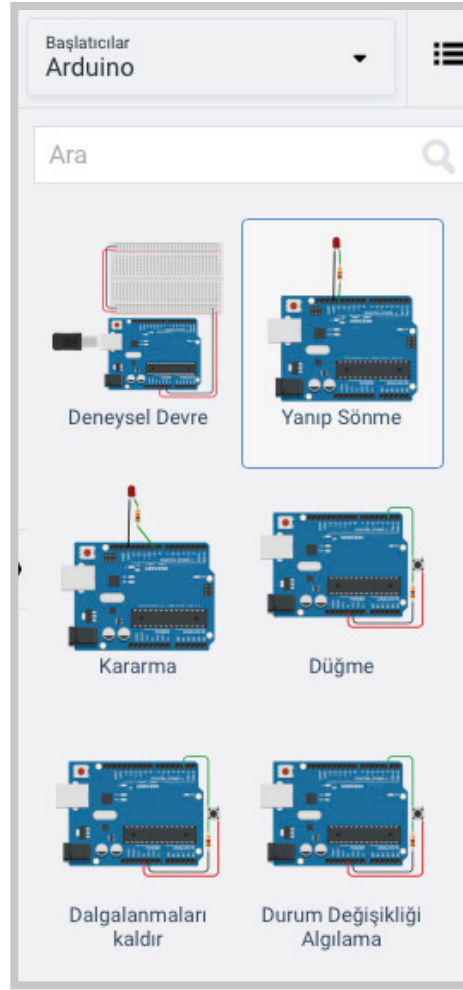
**2. Adım:** Görsel 1.13'te gösterildiği gibi **Bileşenler** seçeneğinden **Arduino**'yu tıklayınız.



Görsel 1.13: Bileşenler seçeneğinden Arduino devrelerin seçimi

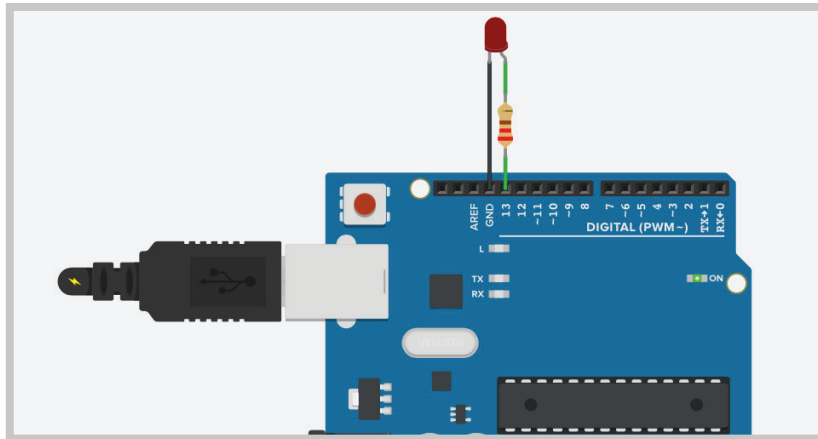
## 1. Öğrenme Birimi: Robotik İçin Mikrodenetleyici Kart

**3. Adım:** Görsel 1.14'te gösterildiği gibi **Arduino** seçeneğinden **Yanıp Sönme** devresini seçiniz.



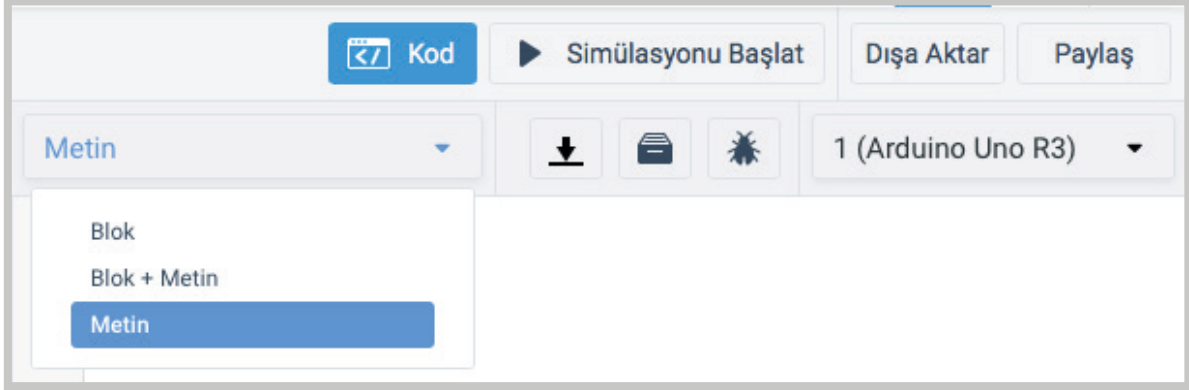
Görsel 1.14: Arduino yanıp sönme devresi

**4. Adım:** Görsel 1.15'te gösterildiği gibi yanıp sönme devresi LED'in uzun bacağıının, 13 No.lu pine dirençle beraber takıldığı ve kısa ucunun GND (topraklama) pinine takıldığı bir uygulamadır. 13 No.lu pinin alt tarafında bulunan L harfi de mikrodenetleyici üzerindeki LED'i ifade eder. Devre çalıştırıldığında kırmızı LED ile beraber L harfinin olduğu LED'in de yanıp söndüğü görülecektir.



Görsel 1.15: Çalışma alanına yanıp sönme devresinin entegre edilmesi

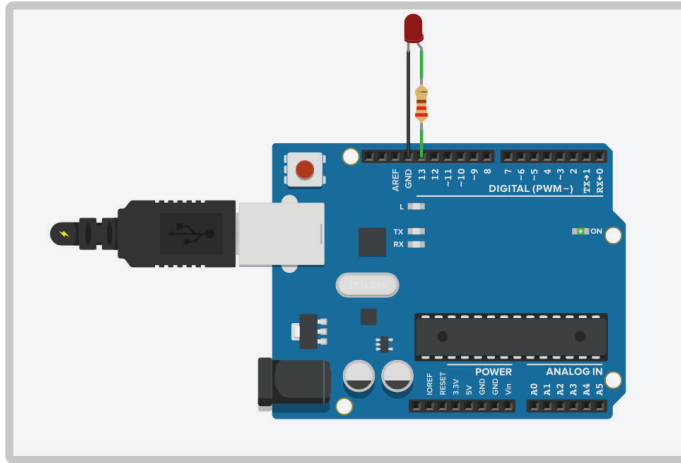
**5. Adım:** Programdaki hazır kodları açınız ve inceleyiniz. Görsel 1.16'da gösterildiği gibi sağ üst köşede **Kod** sekmesi tıklandığında birden çok seçenek çıkmaktadır. Buradan **Metin** seçeneği tıklanırsa mikrodenetleyici IDE programı kurulmadan simülasyon olarak kullanılabilir.



Görsel 1.16: Kod kısmındaki Metin seçeneğinden programın kodlarının incelenmesi

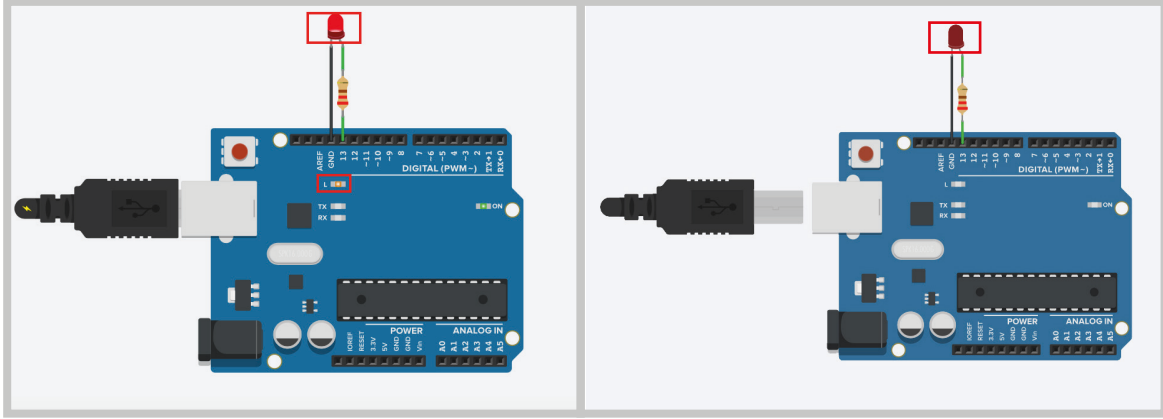
**6. Adım:** Görsel 1.17'de gösterildiği gibi hazır kodlar incelendiğinde **sayısal giriş pininin 13** olarak ayarlandığı görülmektedir. HIGH durumunda LED'in 1 saniye yanacağı ve LOW durumunda LED'in 1 saniyelik süreyle söneceği bir kodlama bloku oluşturulmuştur.

```
/*
  Bu program mikrodenetleyicinin 13.pininin
  (dâhilî LED) yanıp sönmelerini sağlar.
*/
void setup()
{
  pinMode(13, OUTPUT);
}
void loop()
{
  // LED'i açın (HIGH-YÜKSEK voltaj seviyesi)
  digitalWrite(13, HIGH);
  delay(1000); // 1000 milisaniye (1 saniye) bekleyin.
  // Voltajı LOW yazarak LED'i kapatın.
  digitalWrite(13, LOW);
  delay(1000); //1000 milisaniye (1 saniye) bekleyin.
}
```



Görsel 1.17: Yanıp sönmeye devresinin görünümü

**7. Adım:** Görsel 1.18'de **Simülasyonu Başlat** tıklandıktan sonraki birer saniyelik LED yakma söndürme değişimi görülmektedir.



Görsel 1.18: Devrenin yanması ve sönməsi

## 1.2. ROBOT TÜRLERİ VE EĞİTSEL AMAÇLI ROBOTLAR

Karel Capek (Karel Kapek), 1921 yılında prömiyeri yapılan **Rossum'un Evrensel Robotları** adlı bilim kurgu oyununda, tekrar edilen basit işleri yapabilen, insana benzer yaratıkları anlatmak için ilk defa robot kelimesini kullanmıştır. Karel Capek yaşasaydı muhtemelen günümüzde robotların insan yaşamına bu kadar dâhil olmasına çok şaşırırdı.

### 1.2.1. Robot Kavramı

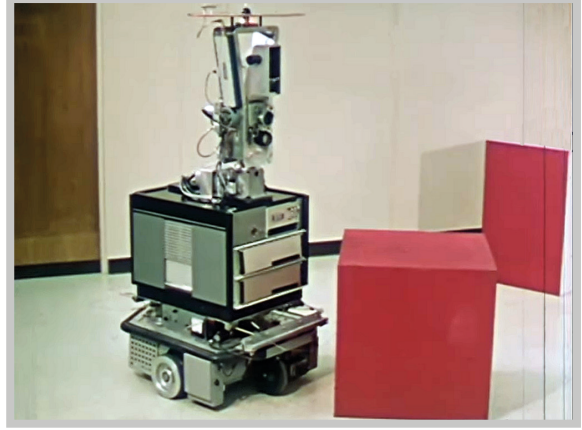
Robotlar, bir programlama algoritması ile çalıştırılan elektromekanik parçalardan oluşan akıllı sistemlerdir. Robotik bir sistem çevreyi algılar ve çevreden gerekli olan verileri toplar. Bu verileri amaçlarına uygun bir şekilde değerlendirerek bir sonuç üretir. Robotlar bu işlemleri kendi kendine (otonom) veya bir kullanıcının çalıştırmasıyla yerine getirebilmektedir. Robotlar, farklı kontrol yöntemleriyle kullanılır. Kontrol yöntemlerine göre robotlar şu şekilde ayrılır:

1. Etki-Tepki (Algılama-Cevap) prensibiyle çalışıyorsa bu tür robotlar **Tepkisel (Reactive)** kontrol robotlarıdır. 1990'ların sonlarında uluslararası büyük satranç ustası Garry Kasparov'u satrançta yenen süper bilgisayar Deep Blue, (Görsel 1.19) bu tür makinelerin mükemmel bir örneğidir. Deep Blue, bir satranç tahtasındaki taşları tanımlayabilmekte ve her birinin nasıl hareket ettiğini bilmekteydi. Kendisi ve rakibi için sonraki hamlelerin neler olabileceği konusunda tahminlerde bulunup olasılıklar arasından en uygun hamleleri seçebilmekteydi.



Görsel 1.19: Deep Blue

2. Algılama-Planlama-Hareket prensibiyle çalışan robotlara **Bilinçli (Deliberative)** kontrol robotları denir. Bilinçli kontrol robotlarına bu robot türlerinin atası olan, 1966 ve 1972 yılları arasında Charles Rosen (Çarls Rozın) tarafından geliştirilen Görsel 1.20'deki Shakey robot örnek verilebilir. Bu robot kendi eylemleri hakkında mantık yürütebilen ilk genel amaçlı mobil robottur. Shakey robot çevresini algılayarak gerçek verileri elde edebilmekteydi. Bu veriler üzerinde planlar oluşturup plan uygulamasındaki hataları düzeltebilmekte ve sıradan İngilizce kullanarak iletişim kurabilmekteydi.



Görsel 1.20: Bilinçli robotların atası Shakey robotu

3. Düşünme-Hareket olaylarının paralel yaşandığı kontrol robotlarına **Karma (Hibrit)** kontrol robotları denir.

4. Hibrit kontrol mantığında hibrit kontrole alternatif kontrol tipi robotlarına **Davranışsal (Behavioral)** kontrol robotları denir.



### Sıra Sizde

Bir arkadaşınızla Karma (Hibrit) ve Davranışsal (Behavioral) kontrol ilkelerine göre çalışan robotları araştırarak, görsel bir çalışma (sunum, poster gibi) hazırlayınız. Hazırladığınız çalışmanın sunumunu sınıfta yapınız.

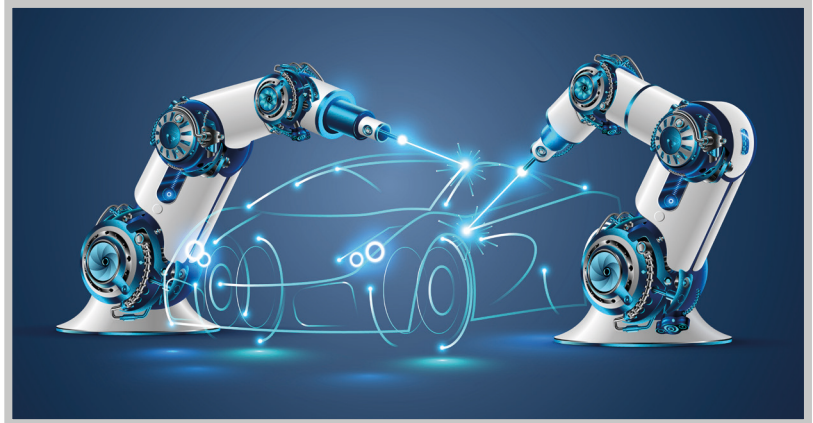


### Değerlendirme

Hazırlayacağınız çalışma öğrenme biriminin sonunda yer alan 1. Kontrol Listesi kullanılarak değerlendirilecektir. Çalışmanızı yaparken değerlendirme ölçütlerini dikkate alınız.

### 1.2.2. Robot Türleri

Günümüzde çok farklı uygulama alanlarına göre robotlar üretilmektedir. Ev robotları, tıbbi robotlar, askerî robotlar, uzay robotları, hobi robotları ve eğitsel robotlar sayılabilecek başlıca robot türlerindedir. Robot türlerinden biri olan ve endüstriyel üretim için kullanılan endüstriyel robotların en önemli özelliği bir kola benzemesidir. Bu robotlar yük taşıma, parçaların kesilmesi, tutulması, bir yerden bir yere taşınması gibi farklı işlevleri gerçekleştirir. Görsel 1.21'de gösterilen robotlar, kaynakçı endüstri robotlarıdır ve araba parçalarının birleştirilmesinde sıklıkla kullanılır.



Görsel 1.21: Kaynakçı endüstri robotları

## 1. Öğrenme Birimi: Robotik İçin Mikrodenetleyici Kart

Ev robotları, ev işlerine yardımcı olmak için geliştirilmiş robotlardır. Görsel 1.22’de bir robot süpürge görülmektedir. Yapay zekâ algoritmaları ile çalışan robot süpürgeler, temizleyeceği yerlerin bir haritasını oluşturarak bu haritaya göre hareket eder. Robot süpürgeler; yapılarındaki sensörlerle boşlukları hesaplayıp, duvarların veya kanepelerin çok yakınından geçerek temizlik yapabilmektedir.



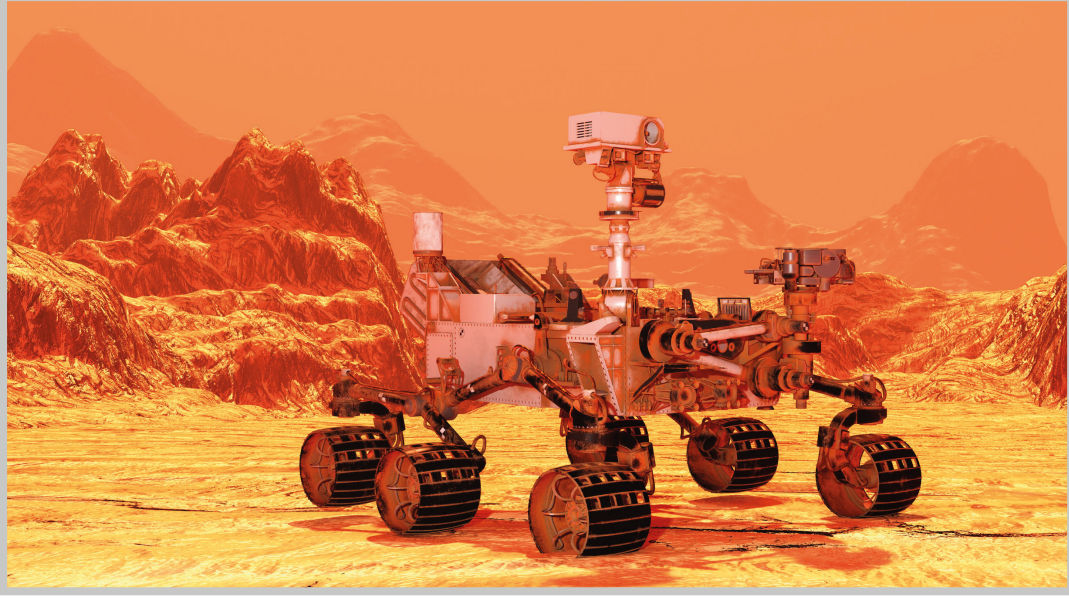
Görsel 1.22: Ev robotu

Tıbbi robotlar, fabrikalarda ilaç üretimi ve dağıtımı için kullanılan robotlardır. Esas önem arz ettikleri yerler ise cerrahi operasyonlardaki katkılarıdır. Görsel 1.23’te doktora yardımcı olan bir cerrahi robot görülmektedir.



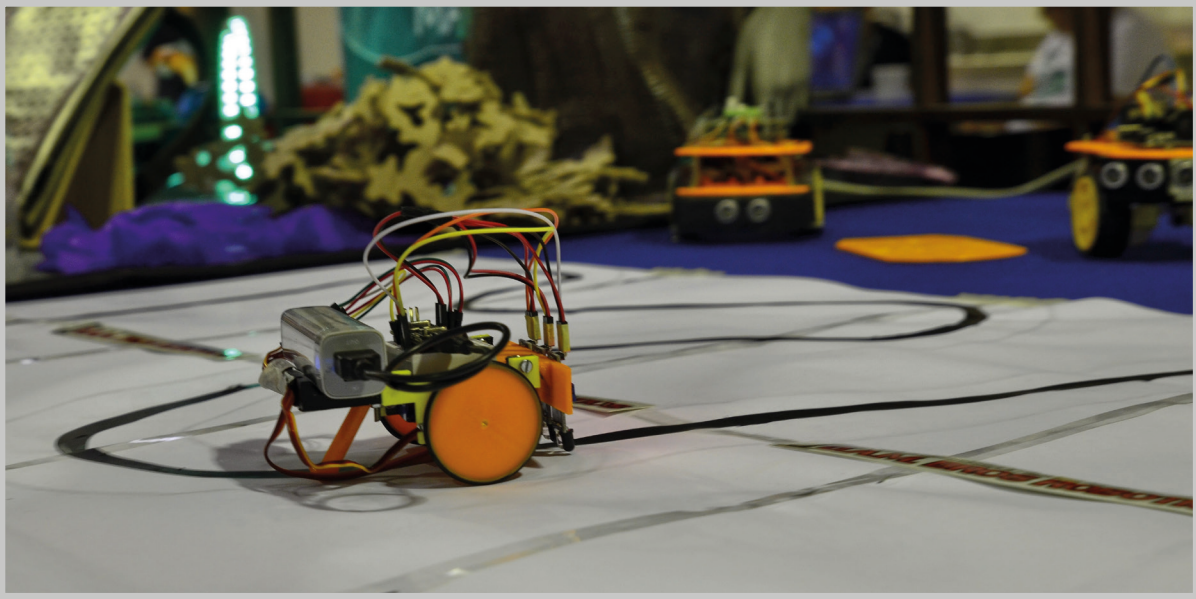
Görsel 1.23: Cerrahi alanda kullanılan bir tıp robotu

Uzay robotları; gezegenlerin keşfinde, uzay istasyonlarında ve uzayla ilgili farklı görevlerde kullanılmaktadır. Görsel 1.24'te Mars'a gönderilen uzay aracı Rover görülmektedir. Bu uzay robotuna isim vermek amacıyla NASA bir yarışma düzenlemiş ve yarışmayı kazanan 13 yaşında bir öğrenci, robota "Azim" anlamına gelen "Perseverance" adını vermiştir.



**Görsel 1.24: Perseverance, Mars Rover uzay aracı**

Hobi ve yarışma robotları, kişisel olarak yapılan bir görevi yerine getirmede kazandırılan özellik ve bu özelliklerin kullanımına dayalı yarışmalar için yapılan robotlardır. Çizgi takip eden, mini sumo, yumurta toplayan, engelden kaçan olmak üzere birçok yarışma robotu türü bulunur. Çizgi izleyen robot, gideceği yolu sensörlerle renk farklılıklarını seçerek takip eden robot tipidir (Görsel 1.25). Robotun takip edeceği yol siyah zeminin üzerine beyaz yol ya da beyaz zeminin üzerine siyah yol şeklinde olabilir.



**Görsel 1.25: Çizgi takip eden robot**

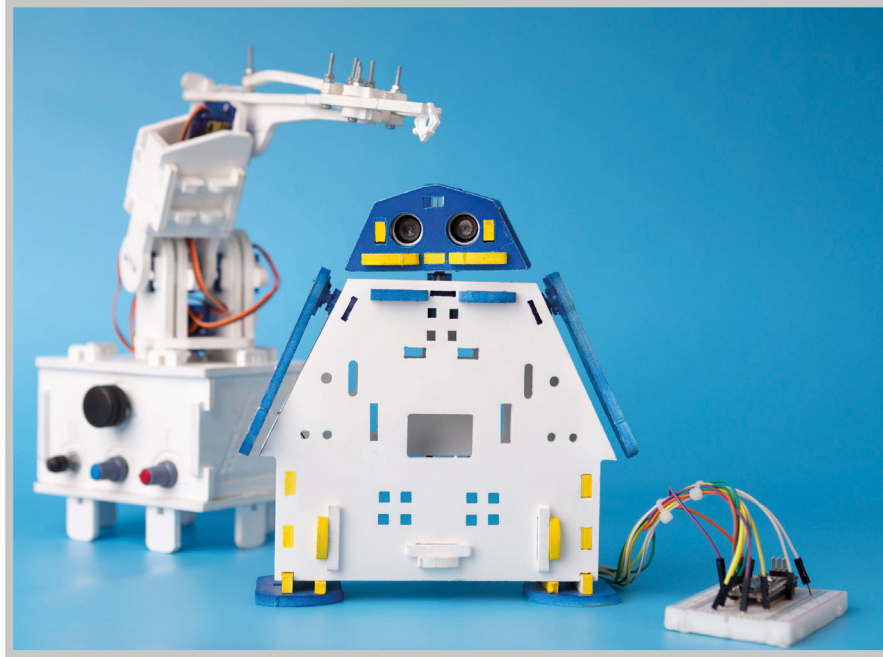
Sanal robotlar, simülasyonları ya da tekrarlanan görevleri gerçekleştirmek için yazılan programlardan oluşan, fiziksel olarak var olmayan robotlardır. Çağrı merkezlerinde kullanılan robotlar bu türlere en iyi örneklerden biridir. On-line sanal robot uygulamaları kullanılarak hem zamandan hem de maliyetten tasarruf sağlanabilmektedir. İnternette birçok sanal robot uygulaması bulunmaktadır. On-line sanal robot uygulamalarının en büyük avantajı, program kurma işlemine gerek kalmamasıdır.

### 1.2.3. Eğitsel Amaçlı Robotlar

Eğitsel amaçlı robotlar, öğrencileri çok erken yaşlardan itibaren etkileşimli olarak **Robotik ve Programlama** ile tanıştırmak için tasarlanmış bir disiplindir. Öğrencilerin problem çözme, eleştirel bakış açısı kazanma ve karar verme stratejilerini geliştirmeleri açısından beceri kazandırmayı amaçlamaktadır.

Eğitsel amaçlı robot türleri şunlardır:

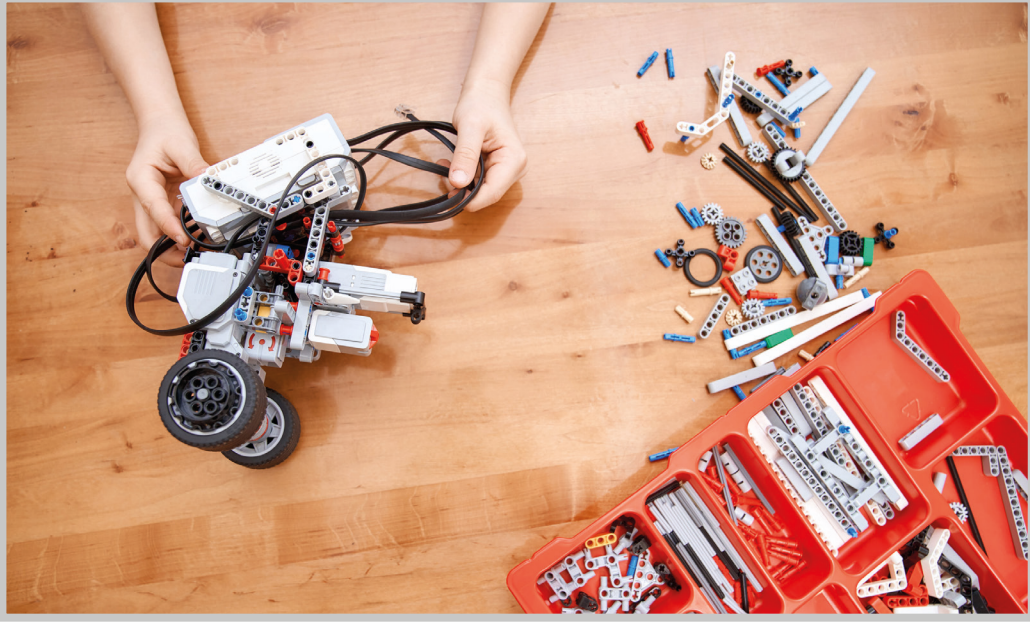
**1. Blok Tabanlı Robot Montaj Setleri:** Blok tabanlı robot montaj setleri özellikle çocukların kendi robot tasarımlarını yapabilmeleri için birbirine kolayca takılabilen parçalardan oluşur. Blok tabanlı robot montaj setleri mantıksal düşünme ve tasarım becerilerini geliştirmekle kalmayıp blok tabanlı programlamanın temellerini de öğretir. Örneğin Görsel 1.26'da bloklardan yapılmış akıllı bir ev gösterilmektedir. Akıllı evlerde servo motor ile kapının açılıp kapanması, kapı ziline uzaktan çalınması, evin sıcaklığının ölçülmesi ve ışıkların yönetilmesi gibi işlemler uzaktan erişimle yapılabilir.



Görsel 1.26: Bloklarla yapılmış akıllı bir ev

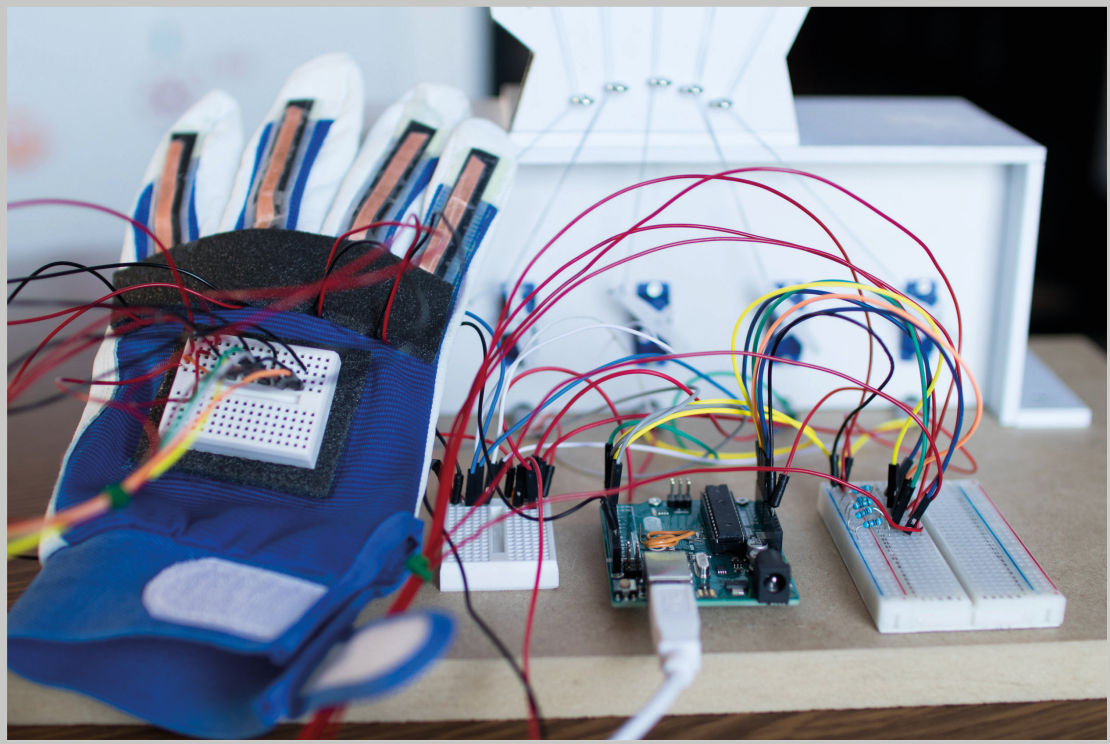
**2. Düşük Maliyetli Mobil Robot Tasarım Kitleri:** Düşük maliyetli mobil robot tasarım kitleri, temel bilgi düzeyine sahip herkesin kullanabileceği, montajlanmamış ve basit düzeyde algılayıcılar içeren ucuz kitlerdir. Görsel 1.27'de düşük maliyetli eğitim için kullanılan robot tasarım kiti görülmektedir.





Görsel 1.27: Laboratuvarlarda kullanılan bir mobil robot tasarım kiti

**3. Açık Kaynak Mobil Robot Platformları:** Açık kaynak mobil robot platformları, ücretsiz bir biçimde donanım ve yazılımlardan faydalanma olanağı sağlayan, daha çok eğitim amaçlı üretilen robot setleridir. Görsel 1.28'deki gibi bir mobil robot platformu; kullanıcıların projelerini, uygulamalarını hayata geçirmek için ucuz ve kullanışlı platformlar sunar.



Görsel 1.28: Mobil robot platformu

**4. Tam Monte Edilmiş Mobil Robotlar:** Tam monte edilmiş mobil robotlar montajı tamamlanmış, bir amaç için tasarlanmış ve kullanıma hazır bir durumda satılan robotlardır. Görsel 1.29'da cam silme robotu tam monte edilmiş mobil robotlara bir örnektir. Cam silme robotu üzerindeki sensörlerle pencere üzerinde çok yönlü hareket sağlayarak camları siler.



Görsel 1.29: Cam silme robotu (cam-bot)

**5. Minyatür Sürü Robotları:** Aynı benzerlik ve kullanışlıkta olan robotların ortak bir iş için birlikte çalışmasıyla oluşan robot türleridir. Görsel 1.30'da gökyüzünde uçan güvenlik kameralı güvenlik uçağı sürüsü görülmektedir. Bu sürü çevrede olabilecek herhangi bir tehlikeyi kaydeder ve önlem alınması için farklı yöntemlerle tepki verir.



Görsel 1.30: Güvenlik uçağı sürüsü



### Sıra Sizde

Küçük gruplar oluşturarak eğitsel robot türleriyle (1. ,2. ,3. ,4. ve 5. madde) ilgili bir araştırma yapınız ve yukarıda bahsedilen her bir eğitsel robot türüne birer örnek olacak şekilde poster hazırlayınız.

## Değerlendirme

Hazırlayacağınız çalışma öğrenme biriminin sonunda yer alan 2. Değerlendirme Ölçeği kullanılarak değerlendirilecektir. Çalışmanızı yaparken değerlendirme ölçütlerini dikkate alınız.

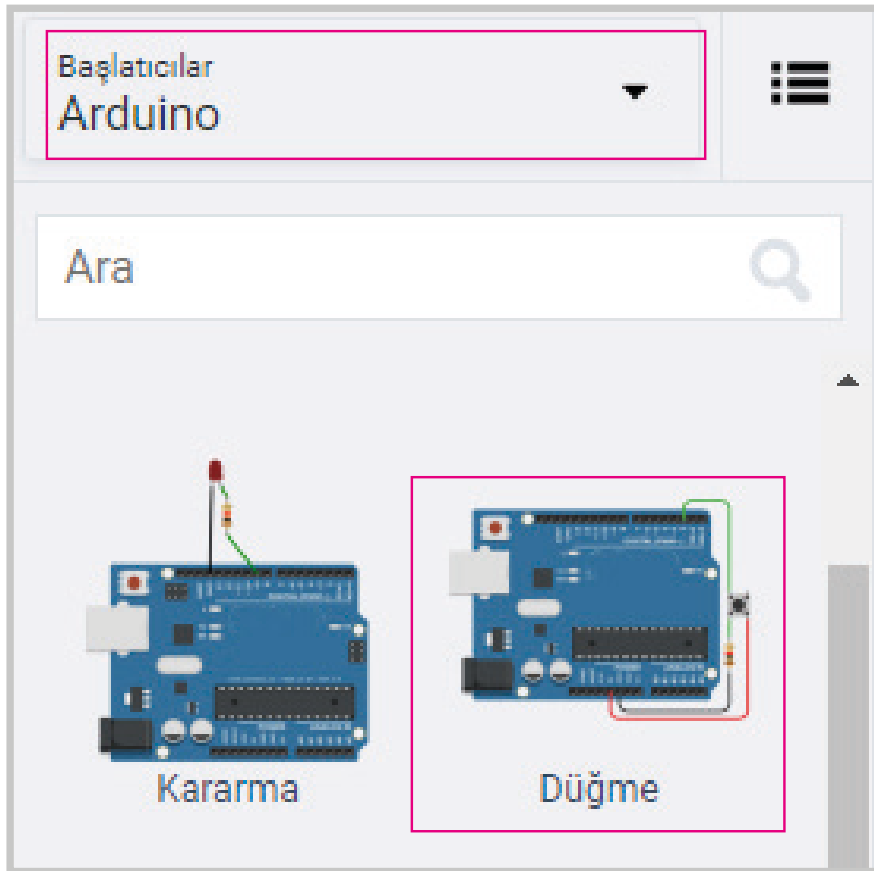
## 3. Uygulama

Eğitimde en çok uygulanan robot türlerinin temel mantığı robot programlamayı bilmekten ve tanımaktan geçer. İnternette on-line kullanılabilen “Devre Tasarım Programı” programında başlangıç uygulama örnekleri bulunmaktadır. Bu uygulamaların hem devre şeması hem de kodları çalıştırılmaya hazırdır. Bu uygulamalardan biri olan düğme (buton) uygulamasını ve kodlarını inceleyiniz.

## Önemli

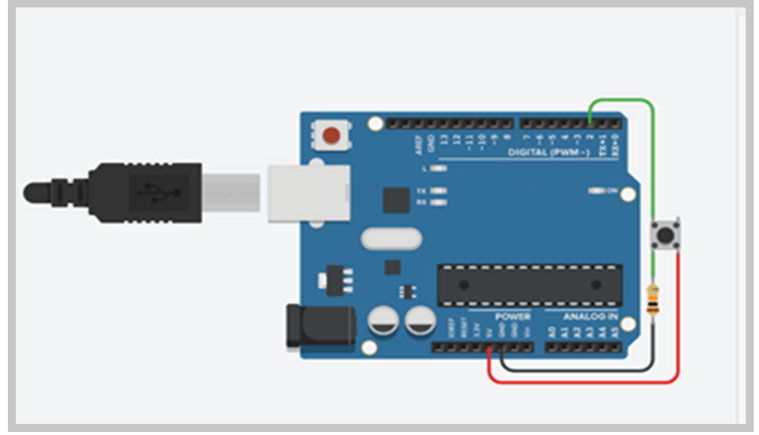
Düğmeler (Butonlar), üzerine basıldığında elektrik akımının iletimine izin veren, bırakıldığında ise devredeki akımı kesen elektronik bir malzemedir.

**1. Adım:** Görsel 1.31’de gösterildiği gibi **Başlatıcılar** seçeneğinden Arduino’yu seçiniz, çıkan seçeneklerden **Düğme** örneğine tıklayınız ve çalışma alanına ekleyiniz. Örnek uygulama incelendiğinde, butonun bir bacağı DIGITAL 2 No.lu pine, butonun 2 numaralı pinle kısa devre hâlinde olan diğer bacağı GND (topraklama) girişine bağlanmıştır. Butonun diğer bacağına ise 5V çıkış uygulanmıştır.



Görsel 1.31: Düğme uygulama devresinin çalışma alanına eklenmesi

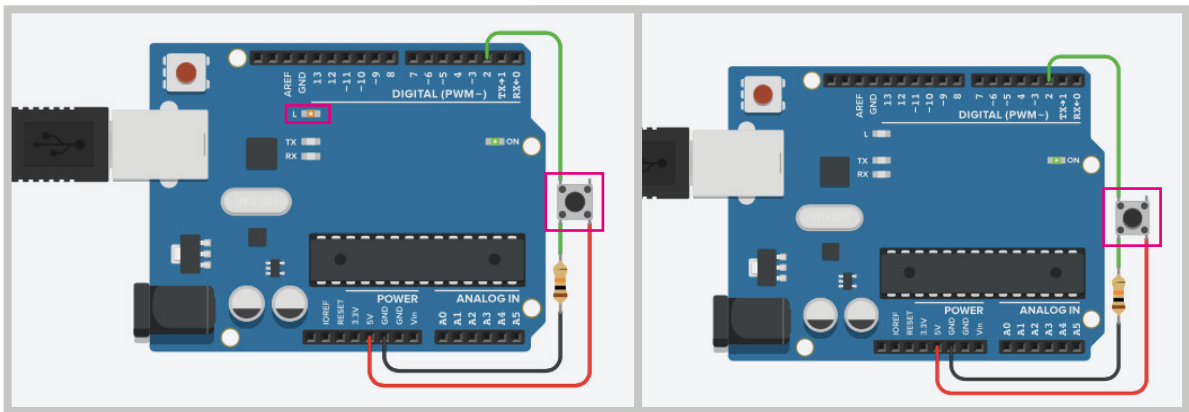
**2. Adım:** Görsel 1.32'de yer alan düğme uygulama devresi incelendiğinde, fiziksel devreye ait kodların aşağıdaki şekilde olduğu görülür. Programda **void setup** fonksiyonu içinde; 2 No.lu pin düğme (buton) giriş ve 13 No.lu pin ise Arduino tümleşik LED olarak tanımlanmıştır. **void loop** fonksiyonu içinde butona basıldığında LED yanacak ve butondan elinizi kaldırdığınızda LED sönecektir.



Görsel 1.32: Düğme uygulamasının devresi

```
int buttonState = 0;
void setup()
{
  pinMode(2, INPUT);
  pinMode(13, OUTPUT);
}
void loop()
{
  buttonState = digitalRead(2); // buton değerinin durumunu oku.
                                // düğmeye basılıp basılmadığını kontrol et.
  if (buttonState == HIGH) {
    digitalWrite(13, HIGH); // LED'i aç.
  } else {
    digitalWrite(13, LOW); //LED'i kapat.
  }
  delay(10); //Geciktirme işlemi
}
```

**3. Adım:** Görsel 1.33'te gösterildiği gibi **Simülasyonu Başlat** tıklanıp uygulamadaki buton basılı tutulduğunda 13 No.lu LED yanacak ve buton bırakıldığında Görsel 1.33'te gösterildiği gibi LED sönecektir.



Görsel 1.33: Buton düğmesine basıldığında LED'in yanması ve bırakıldığında LED'in sönməsi

## 1.3. ROBOTTA MEKANİK / ELEKTROMEKANİK BİLEŞENLER

Robotlar, programlanabilen elektronik kartlar haricinde gövdesini ve hareketli parçalarını oluşturan plastik veya metal bileşenlerden meydana gelir. Robotun **mekanik / elektromekanik bileşenleri** 7 kısımda incelenebilir:

- Robot gövdeleri
- Motorlar
- Tekerlek, ayak ve paletler
- Eklenti ve bağlantı bileşenleri
- Vida, somun ve rondela bileşenleri
- Amortisör, yay ve esnek bileşenler
- Mekanik veya vakumlu nesne tutucu bileşenler

### 1.3.1. Robot Gövdeleri

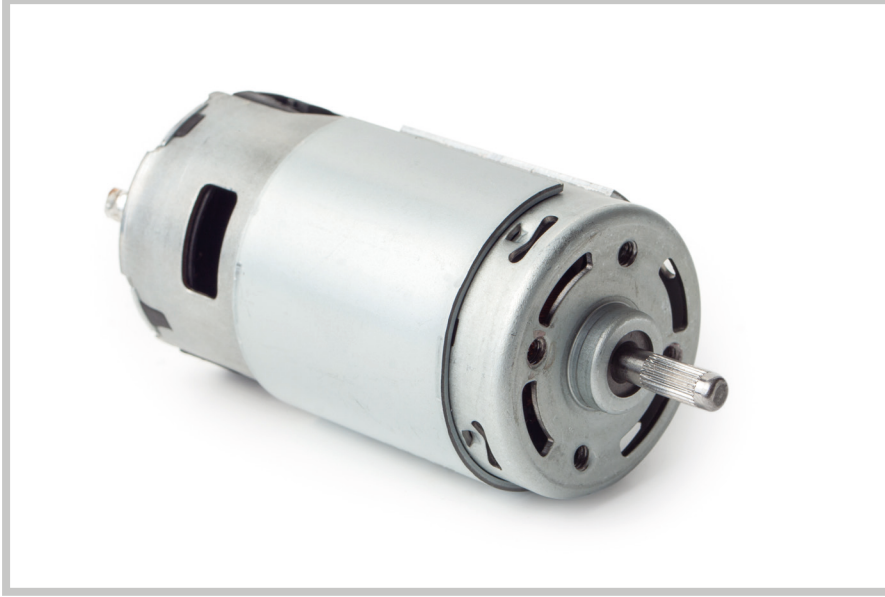
Robotun tüm bileşenlerini üzerinde bulunduran yapı; plastik, pleksi glass, ağaç, metal gibi malzemelerden üretilmiş en temel robot bileşenidir. Görsel 1.34'te üzerine motorlar monte edilmiş, 3 boyutlu yazıcıyla üretilmiş bir robot el modeli görülmektedir.



Görsel 1.34: 3 boyutlu yazıcıyla üretilmiş bir robot el modeli

### 1.3.2. Motorlar

Motorlar, robotlarda hareket imkânı sağlayan en önemli bileşenlerden biridir. Temel prensip olarak, elektrik enerjisini hareket enerjisine dönüştüren elektrik makineleridir. Robotlarda genelde DC motor (Görsel 1.35) kullanılır. Robotların işlevine ve boyutuna göre çeşitli büyüklük ve hızlarda DC motorlar kullanılır.



Görsel 1.35: DC motor örneği

### 1.3.3. Tekerlek, Ayak ve Paletler

Robotun motordan aldığı dönme hareketini yere iletmesi için tekerlekler, ayaklar ve paletler kullanılır. Görsel 1.36'da farklı özelliklere sahip robotlar görülmektedir. Robotların türüne ve kullanım alanına göre farklı tekerlek yapısına sahip 2, 3, 4 veya daha fazla tekerlekli robotlar üretilebilir. Engelibeli arazilerde tekerlek kullanımı zor olacağından daha çok yürüme yeteneğine sahip robotlar veya paletli robotlar kullanılır.



a) Tekerlekli robot

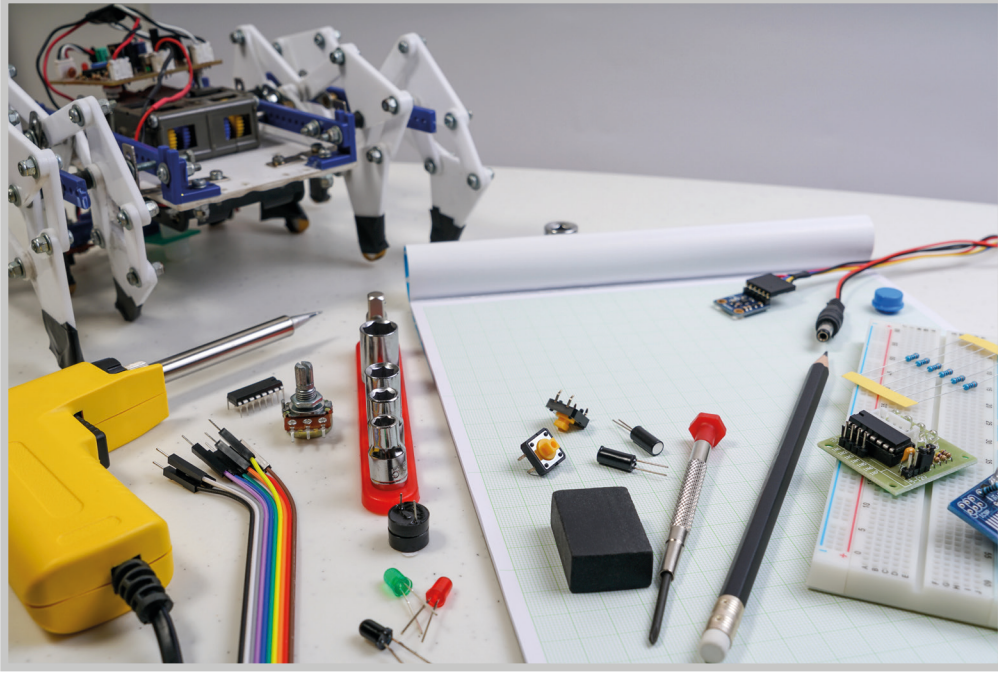
b) Ayaklı robot

c) Paletli robot

Görsel 1.36: Çeşitli özelliklere sahip robotlar

## 1.3.4. Eklenti ve Bağlantı Bileşenleri

Robotlarda motorları gövdeye sabitlemek, yürüme yetisi için gerekli ayakları ve mekaniksel bağlantıları takabilmek için küçük parçalardan oluşan, bağlantı elemanlarına ihtiyaç vardır (Görsel 1.37).



Görsel 1.37: Yürüyen robotlardaki eklenti ve bağlantı bileşenleri

## 1.3.5. Vida, Somun ve Rondela Bileşenleri

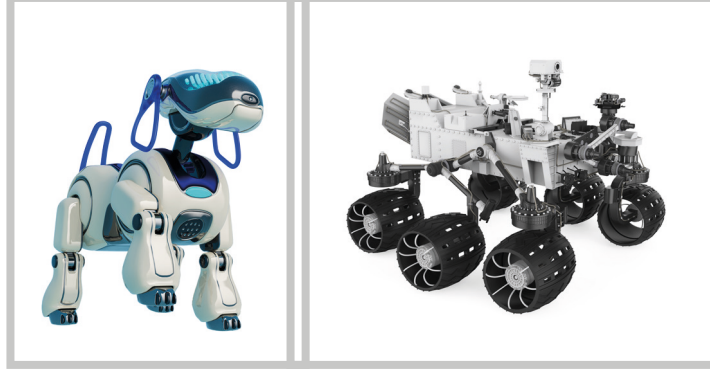
Robotlardaki motor, tekerlek, daha küçük eklentiler ve bağlantı bileşenlerini birbirine tutturmak için Görsel 1.38’de gösterildiği gibi vida, somun gibi yardımcı aparatlara ihtiyaç vardır. Bu yardımcı aparatlar robotun büyüklüğüne ve kullanım yerine göre değişiklik gösterebilir.



Görsel 1.38: Robotlardaki eklenti ve bağlantı bileşenleri için vida ve somunlar

### 1.3.6. Amortisör, Yay ve Esnek Bileşenler

Özellikle arazi robotlarında yoldan gelen sarsıntıları üzerine alarak, robotun devrilmemesini ve yolda rahat gitmesini sağlamak için amortisörler veya yaylar kullanılır. Görsel 1.39'da gösterildiği gibi yürüyen robotlarda, özellikle yürürken daha fazla esneklik sağlamak için kol ve bacaklardaki motorların uçlarında yaylar kullanılmıştır. Yine aynı görselde arazi robotunun tekerleklerine takılan amortisörler görülmektedir.



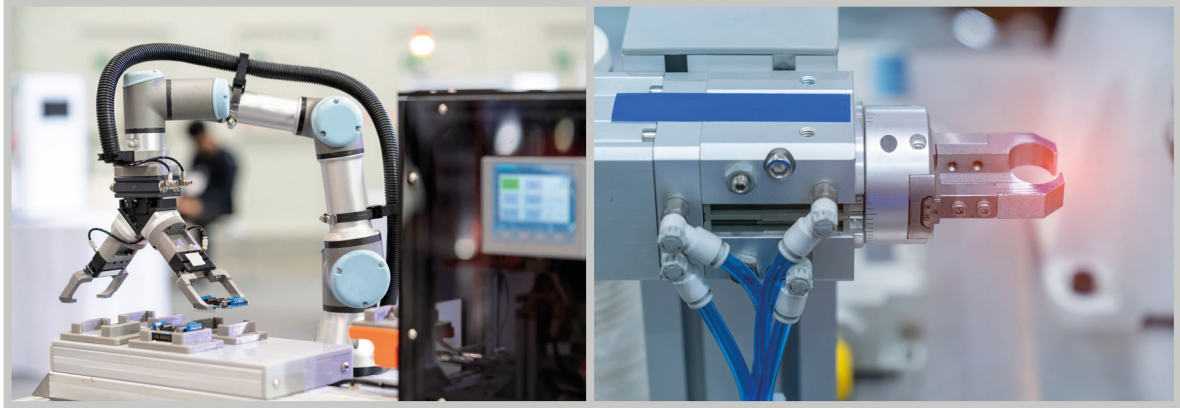
a) Yürüyen robotlarda

b) Arazi robotlarında

**Görsel 1.39: Robotlarda esnekliği sağlamak için kullanılan yaylar**

### 1.3.7. Mekanik veya Vakumlu Nesne Tutucu Bileşenler

Robot kolu bulduran araçlarda bir nesnenin bir yerden alınıp başka bir yere taşınmasında özellikle mekanik tutucular veya pnömatik vakumlu tutucular kullanılır (Görsel 1.40). Mekanik tutucularda bir motor aracılığıyla tutucu ağızının açılıp kapanması sağlanır. Vakumlu tutucularda kompresör tarafından üretilen basınç, hava yardımıyla bir çekme kuvveti oluşturur. Bu sayede tutulan nesneler bir yerden başka bir yere taşınabilir.



a) Gripper mekanik tutucu

b) Pnömatik vakumlu tutucu

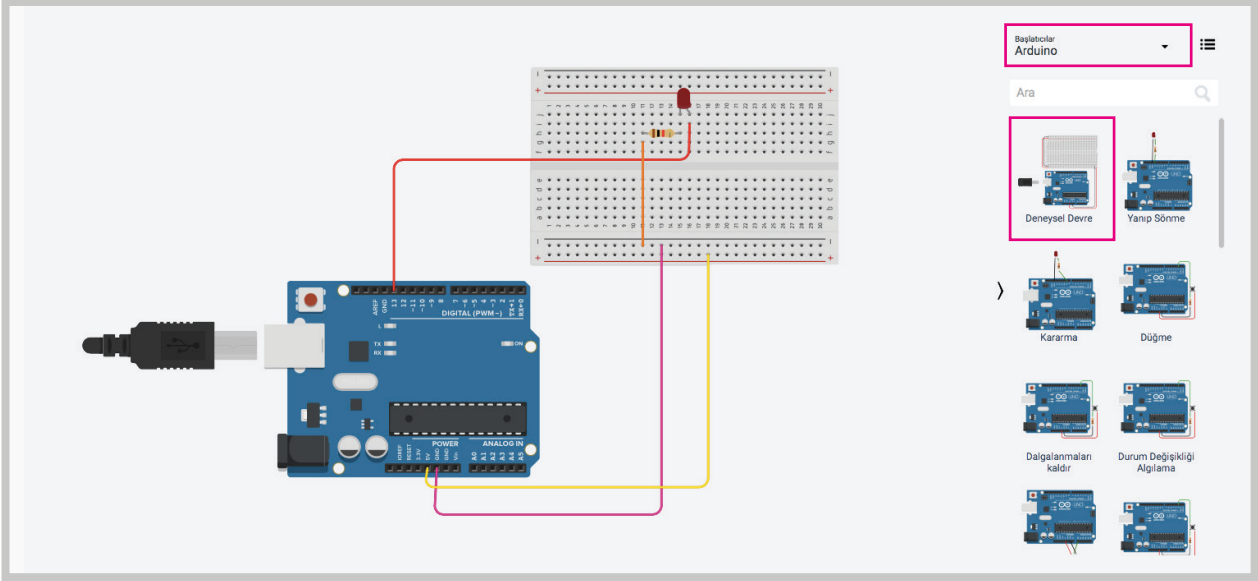
**Görsel 1.40: Robotlardaki nesne tutucu bileşenler**

## 4. Uygulama

Robotta mekanik / elektromekanik bileşenleri inceledikten sonra basit anlamda kullanılan elektronik parçalarla on-line uygulamalara devam edilecektir. "Devre Tasarım Programı" programında breadboard (uygulama tahtası) üzerinde, birer saniye aralıklarla LED yakıp söndürme olayını gerçekleştiren simülasyon devresini kurunuz.

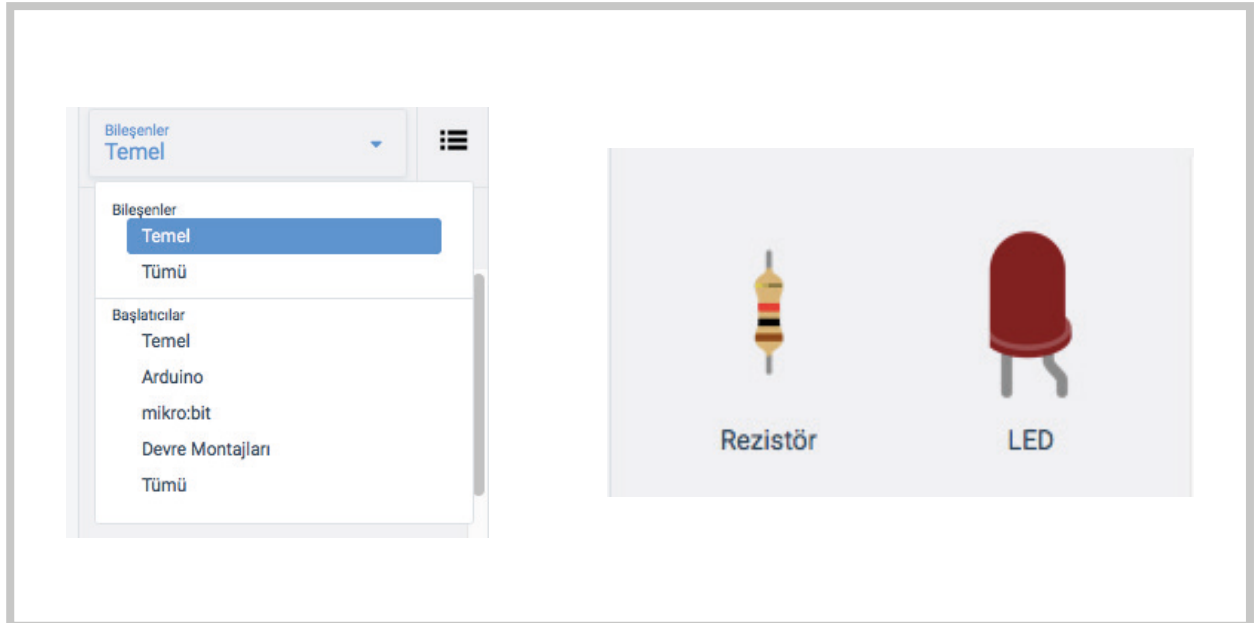


**1. Adım:** Breadboard, devre kurmayı kolaylaştıran ve pratik değişiklikler yapmaya imkân tanıyan, kullanımı kolay bir platformdur. “Devre Tasarım Programı” programında breadboardu kullanmak için **Başlatıcılar** seçeneğinden **Arduino**'yu seçiniz ve **Deneyisel Devre**'yi tıklayarak çalışma alanına bırakınız (Görsel 1.41).



Görsel 1.41: Deneyisel devrenin çalışma alanına bırakılması

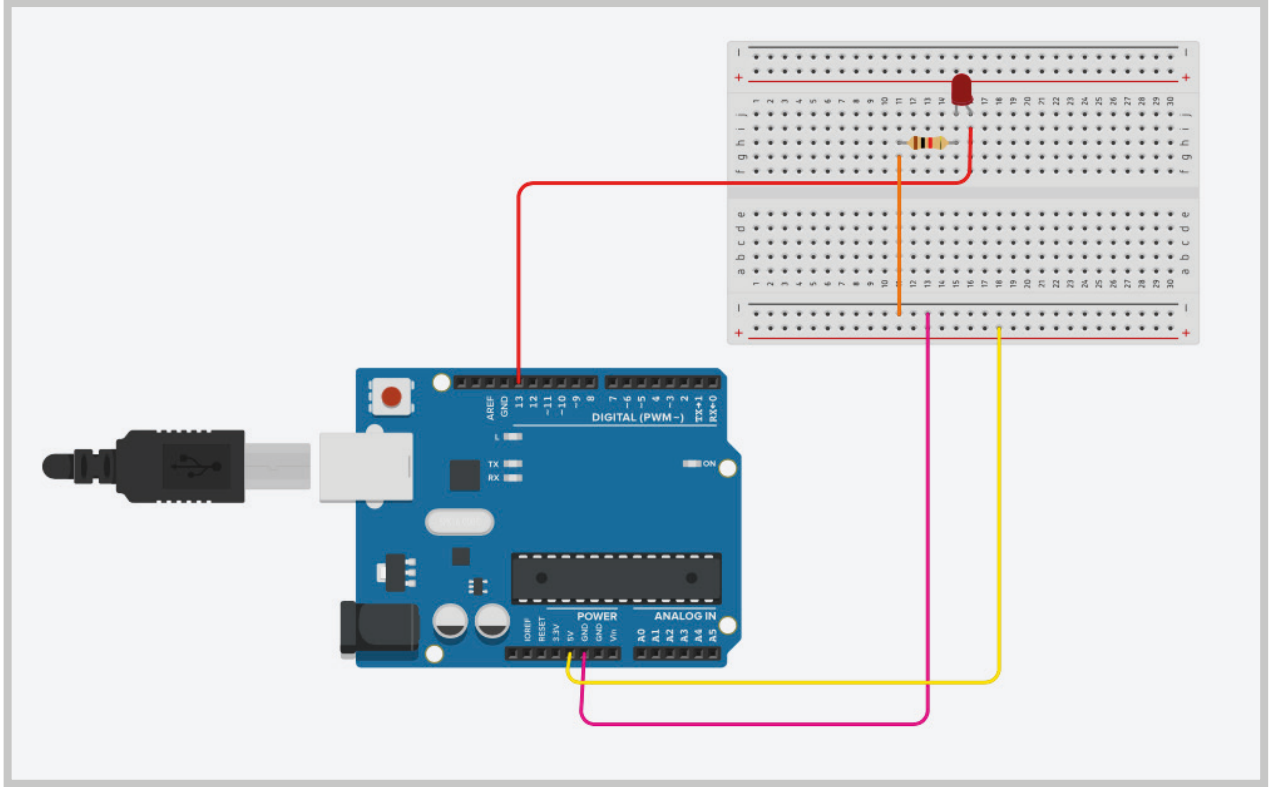
**2. Adım:** Devreyi yapmak için bir LED, bir direnç (rezistör) ve 4 jumper kablo kullanılacaktır. **Başlatıcılar** seçeneğinden **Temel** tıklanarak gerekli olan malzemeleri kullanınız (Görsel 1.42).



Görsel 1.42: Devre elemanlarının seçilmesi

## 1. Öğrenme Birimi: Robotik İçin Mikrodenetleyici Kart

**3. Adım:** Devreyi Görsel 1.43'te gösterildiği gibi düzenleyiniz.



Görsel 1.43: Devre elemanlarının devreye yerleştirilmesi

**4. Adım:** Devre çalıştırıldığında birer saniye aralıklarla kırmızı LED'in yanıp sönmelerini sağlayacak devre kodlarını oluşturunuz. **Kod** seçeneğine tıklayarak, **Metin** seçeneğine aşağıdaki kodları yazınız.

```
void setup()  
{  
  pinMode(13, OUTPUT);  
}  
void loop()  
{  
  digitalWrite(13, HIGH); //LED yandı.  
  delay(1000); //1 saniye açık kalsın.  
  digitalWrite(13, LOW); //LED söndü.  
  delay(1000); //1 saniye kapalı kalsın.  
}
```

## 1.4. ROBOTTA ELEKTRONİK BİLEŞENLER

Bir robottaki mekaniksel parçaları kolay bir şekilde kontrol etmek ve dışarıdan alınan fiziksel değişikliklere göre farklı yönlendirmelerde bulunmak için elektronik bileşenlere ihtiyaç vardır. Robotlarda kullanılan **elektronik bileşenler**;

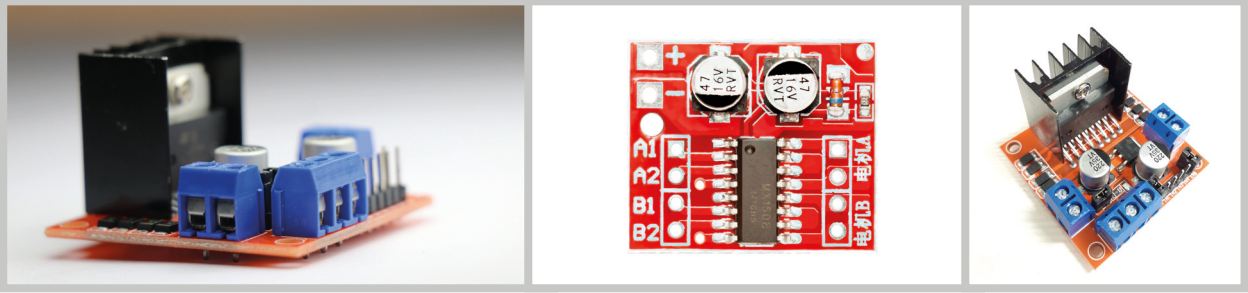
- Mikrodenetleyici kartlar,
- Motor sürücü kartları,
- Sensörler,
- Kablosuz erişim kartları şeklinde sıralanabilir.

### 1.4.1. Motor Sürücü Kartları

Mikrodenetleyicinin çıkış akımının motoru döndürebilecek seviyede olmaması ve mikrodenetleyicilerin zarar görmemesi için sürücü devrelerine ihtiyaç duyulmuştur.

#### 1.4.1.1. DC Motor Sürücü Kartları

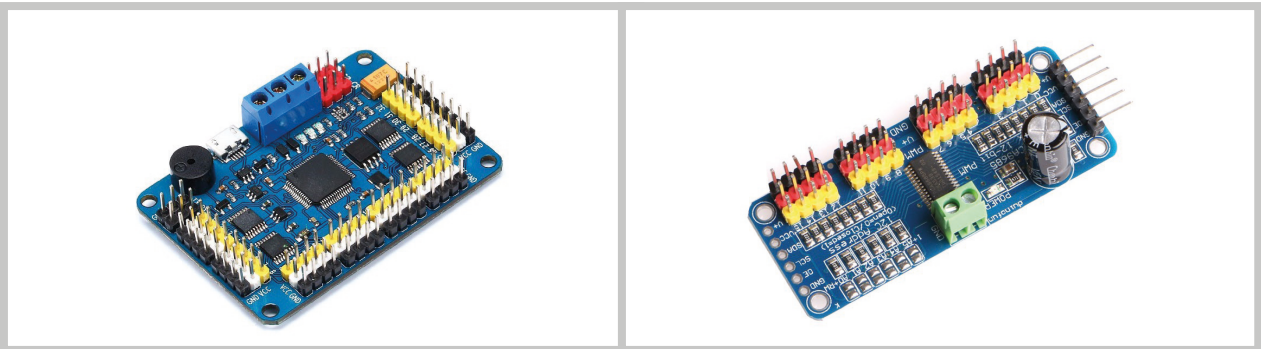
DC Motor sürücüler, mikrodenetleyiciden gelen motor kontrol sinyallerini DC motora ileten elektronik devrelerdir. Görsel 1.44'te görüldüğü gibi çok çeşitli özelliklerde ve boyutlarda (L293, L298N vb.) motor sürücüler mevcuttur.



Görsel 1.44: Robotlarda kullanılan motor sürücü kart örnekleri

#### 1.4.1.2. Servo Motor Sürücü Kartları

Özellikle insansı robotlar (16 ile 19 arası servo motorlu), örümcek robotlar (12 ile 32 arası servo motorlu) gibi mikrodenetleyici portlarının yeterli gelmediği, çok sayıda servo motorun kontrolünün gerekli olduğu robotlarda ilave servo motor sürücü kartları kullanılır. Görsel 1.45.a'da 32 servo motora kadar kontrol edebilen, Görsel 1.45.b'de ise 16 servo motora kadar kontrol edebilen kontrol kartları görülmektedir. Servo motorlar kontrol kartları üzerindeki portlara kolaylıkla bağlanacak şekilde tasarlanmıştır.



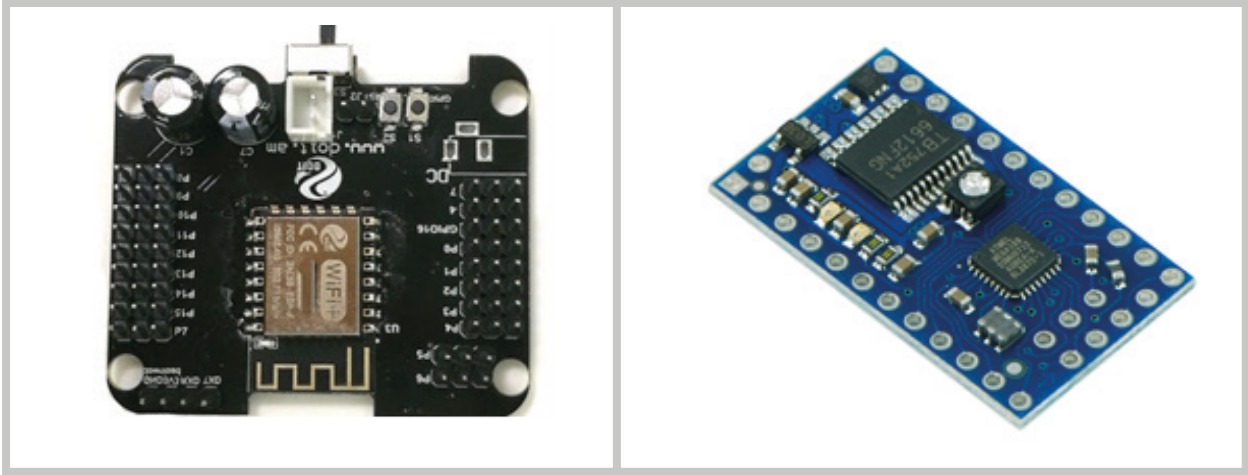
a) 32 servo motor için

b) 16 servo motor için

Görsel 1.45: Robotlarda kullanılan servo motor sürücü kartları

## 1.4.1.3. Özel Robot Kartları

Özel robot kartları robot projeleri için tasarlanan özel üretim kontrol kartlarıdır. Wi-Fi bağlantısı imkânı sağlayan, 17 servo motoru kontrol edebilecek porta sahip insansı robotlar veya örümcek robotlar için geliştirilmiş özel robot kartları vardır. Görsel 1.46.a'da 17 servo motorlu özel robot kartı görülmektedir. Bazı robot kartları içinde motor sürücüsünü de barındırır. Küçük bir kart üzerine hem mikrodenetleyici hem de motor sürücüsü entegresi yerleştirilmiştir. Görsel 1.46.b'de mikrodenetleyici ve motor sürücüsü birlikte kullanılan özel üretim bir robot kartı görülmektedir.

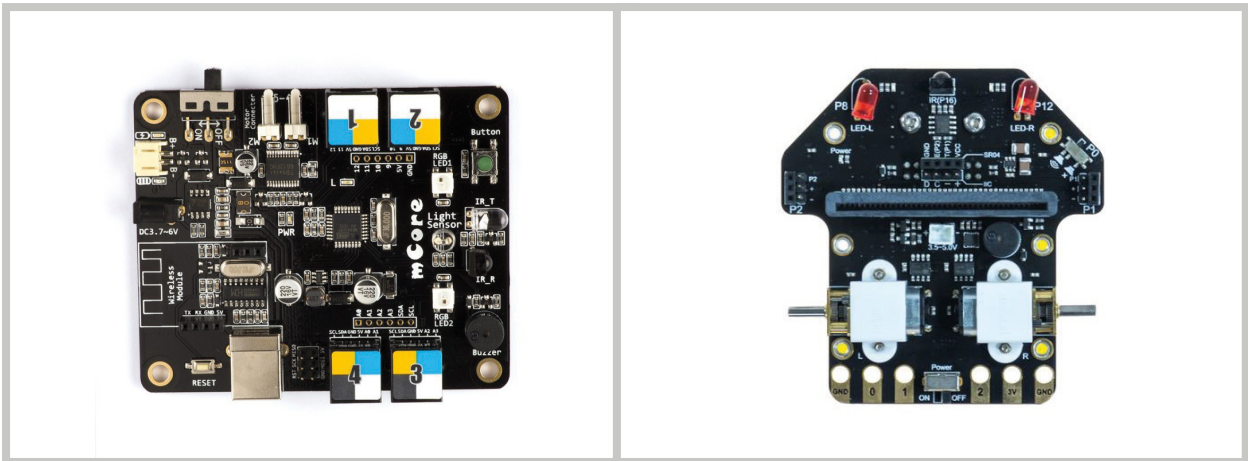


a) 17 servo motorlu insansı robot için

b) Mikrodenetleyici ve motor sürücüsü birlikte

**Görsel 1.46: Robotlarda kullanılan özel üretim kontrol kartları**

Görsel 1.47.a,b'de gösterildiği gibi bazı özel robot kartlarında ise mikrodenetleyici ve motor sürücüsüne ilave olarak sensörler, RGB LED'ler, IR alıcı, buzzer gibi ilave bileşenler de eklenerek üretilir. Bu sayede robot, işlevleri yanında farklı sensör uygulamaları da yapabilecek yetenekler kazanır.



a) Mikrodenetleyici ve motor sürücüsü birlikte

b) Motor, motor sürücüsü ve sensörlerle birlikte

**Görsel 1.47: Robotlarda kullanılan özel üretim kontrol kartları**

## 1.4.2. Sensörler

Sensörler robotlarda mikrodenetleyicinin ısı, ışık ve ses gibi dışarıdan gelen fiziksel olayları algılayıp yorumlamasına yardımcı olan robot bileşenleridir. Robotlarda kullanım amacına göre çeşitli özelliklerde sensörler kullanılır. Görsel 1.48'de çizgi izleyen robotlarda kullanılan çizgi izleme sensörü, renk algılayan robotlarda kullanılan renk sensörü ve engeli algılayan robotlarda kullanılan ultrasonik mesafe sensörü görülmektedir. Bu sensörlere ek olarak gyro sensörü, ivme sensörü, alev sensörü, gaz sensörü, titreşim sensörü gibi birçok sensör bulunmaktadır.



a) Çizgi izleme sensörü

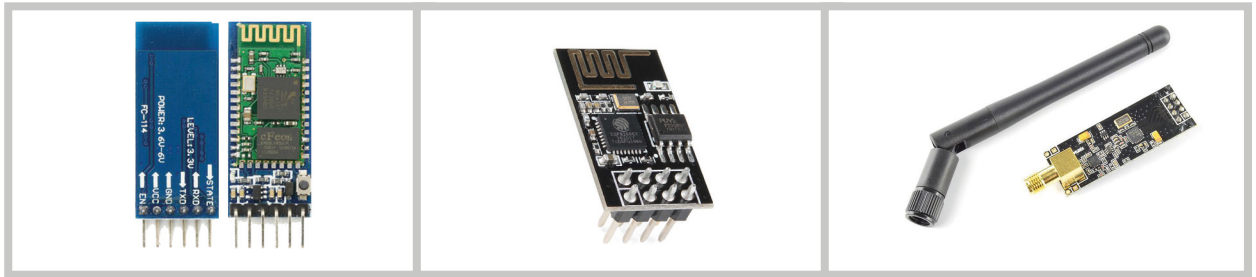
b) Renk sensörü

c) Ultrasonik mesafe sensörü

Görsel 1.48: Robotlarda kullanılan örnek sensör kartları

## 1.4.3. Kablosuz Erişim Kartları

Robotları bilgisayar, tablet, cep telefonu ve oyun kumandası gibi uzaktan erişimle kontrol edebilmek için bazı ilave kartlara ihtiyaç duyulur. Kablosuz erişim kartları olarak ifade edilen bu kartların bir örneğine Görsel 1.49'da yer verilmiştir. Bluetooth özelliğe sahip cihazlarla haberleşebilmek için bluetooth kartı, Wi-Fi üzerinden haberleşme olanağı sağlayan ESP8266 Wi-Fi kartı, iki farklı mikrodenetleyiciyi Wi-Fi üzerinden haberleştirmek için NRF24L01 kartı kullanılabilir. Bu kartlara ek olarak farklı boyut ve özelliklerde kablosuz erişim kartları da bulunur.



a) Bluetooth

b) ESP8266 Wi-Fi

c) NRF24L01

Görsel 1.49: Kablosuz erişim kartları



### Sıra Sizde

Sınıfınızda, uygun olabilecek yeterli sayıda grup oluşturarak, her bir grubun aşağıda yer verilen robot türlerinden birini seçmesini sağlayınız. Belirlediğiniz robot türünün elektronik parçaları ile ilgili bir araştırma yapınız. Araştırmalarınızın sonucunda bir sunum hazırlayarak sınıfınızla paylaşınız.

- Bluetooth Kontrollü Robot
- Engelden Kaçan Robot
- Çizgi İzleyen Robot
- Mini Sumo Robot
- Yumurta Toplama Robotu
- Endüstriyel Robot Kol

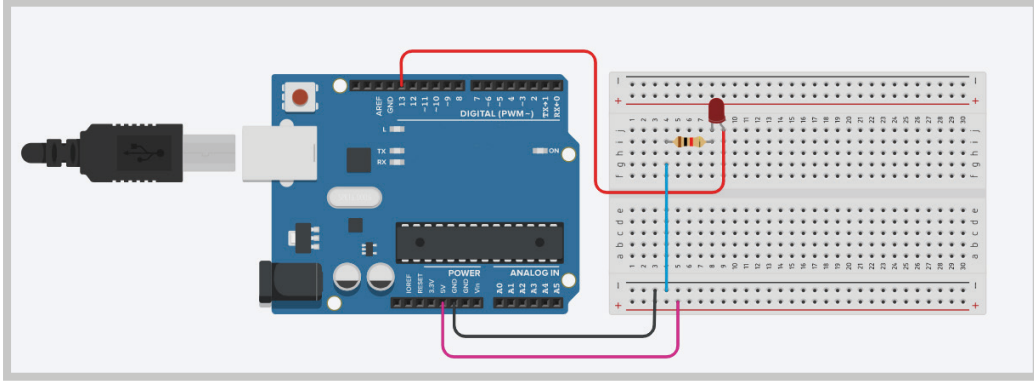
## 1. Öğrenme Birimi: Robotik İçin Mikrodenetleyici Kart

### Değerlendirme

Hazırlayacağınız çalışma öğrenme biriminin sonunda yer alan 3. Değerlendirme Ölçeği kullanılarak değerlendirilecektir. Çalışmanızı yaparken değerlendirme ölçütlerini dikkate alınız.

### Sıra Sizde

3. Uygulamada yapılan LED yakıp söndürme çalışmasını 2 saniye aralıklı döngü ile yapınız (Görsel 1.50).



Görsel 1.50: LED yakıp söndürme devresi

### Değerlendirme

Hazırlayacağınız çalışma öğrenme biriminin sonunda yer alan 2. Kontrol Listesi kullanılarak değerlendirilecektir. Çalışmanızı yaparken değerlendirme ölçütlerini dikkate alınız.

## ÖLÇME VE DEĞERLENDİRME

1. Elektronik ve robotik sistemlerin beyni aşağıdakilerden hangisidir?

- A) ROM
- B) LED'ler
- C) G / Ç Birimi
- D) A / D Çeviriciler
- E) Mikrodenetleyici

2. Mikrodenetleyicilerde sistemin çalışması için kaydedilmiş programları tutan birim aşağıdakilerden hangisidir?

- A) ROM
- B) RAM
- C) G / Ç Birimi
- D) A / D Çeviriciler
- E) Mikrodenetleyici

3. I. Ucuz olması  
II. Programlama kolaylığı  
III. Kolay bulunabilir olması  
IV. Kaynak ve kütüphanelerin fazlalığı

**Yukarıda verilenlerden hangileri mikrodenetleyici seçiminde önemlidir?**

- A) I ve II
- B) II ve IV
- C) I ve III
- D) I,II ve III
- E) I, II, III ve IV

4. Uygulama kartı ile mikrodenetleyici arasındaki ilişki hangisi olamaz?

- A) Uygulama kartı mikrodenetleyiciyi kapsar.
- B) Mikrodenetleyici uygulama kartının beynidir.
- C) Uygulama kartları mikrodenetleyici olmadan da çalışır.
- D) Uygulama kartları, üzerlerinde farklı birimleri barındırır.
- E) Mikrodenetleyici çeşidi, uygulama kartında hedeflenen amaca göre farklılık gösterebilir.

5. Bilinçli robotların atası olan ve algılama-plan-hareket prensibiyle çalışan robot aşağıdakilerden hangisidir?

- A) IHA
- B) Rover
- C) Shakey
- D) Deep Blue
- E) Perseverance

6. I. Robot süpürgeler  
II. Cerrahi robotlar  
III. Deep Blue  
IV. Kaynakçı robotlar

**Yukarıda verilen ifadelerden hangisi veya hangileri endüstri robotuna bir örnektir?**

- A) Yalnız I
- B) Yalnız IV
- C) I ve II
- D) I ve III
- E) III ve IV

7. Aşağıdakilerden hangisi hobi ve yarışma robotlarına örnek değildir?

- A) Çizgi Takip
- B) Mini sumo
- C) Yumurta Toplayan
- D) Engelden Kaçan
- E) Kaynakçı

8. Aşağıdakilerden hangisi robot simülâtörünün faydalarından değildir?

- A) Robot programlamayı öğretir.
- B) Gerçekçi sonuçlar elde edilmez.
- C) Malzemelerin alınmasına gerek yoktur.
- D) Sanal ortamda robota erişmenizi sağlar.
- E) Elde edilen skorları kaydetmenize olanak tanır.

9. Aşağıdakilerden hangisi robotta mekanik / elektronik bileşenlerden değildir?

- A) Motorlar
- B) Amortisör
- C) Tekerlek ayak
- D) Sensörler
- E) Vida, somun

10. Robotun tüm parçalarını üzerinde bulunduran plastik, pleksi glass, ağaç gibi metallerden üretilen en temel mekanik / elektromekanik bileşen aşağıdakilerden hangisidir?

- A) Motorlar
- B) Sensörler
- C) Tekerlek ayak
- D) Robot gövdesi
- E) Vakumlu nesne tutucu

11. Robota hareketlilik imkânı tanıyan mekanik / elektromekanik bileşen aşağıdakilerden hangisidir?

- A) Motorlar
- B) Sensörler
- C) Robot gövdesi
- D) Tekerlek ayak
- E) Vakumlu nesne tutucu

12. Robotlardaki küçük eklentiler ve bağlantı bileşenlerini tutmak için kullanılan mekanik / elektromekanik bileşen aşağıdakilerden hangisidir?

- A) Motorlar
- B) Sensörler
- C) Tekerlek ayak
- D) Eklenti ve bağlantı bileşenleri
- E) Vida, somun ve rondela bileşenleri

13. Robotta elektronik bileşenlerden olan sensörlerin görevi aşağıdakilerden hangisidir?

- A) Robotun sarsıntıya uymasını engeller.
- B) Robotlara uzaktan erişmek için kullanılır.
- C) Isı, ışık, ses gibi dışarıdan gelen olayları algılar.
- D) Motor kontrol sinyallerini motora ileten elektronik devrelerdir.
- E) Bir nesnenin bir yerden alınıp başka bir yere taşınmasını sağlar.

14. Aşağıdakilerden hangisi robotta bulunan elektronik bileşenlerden değildir?

- A) Motorlar
- B) Sensörler
- C) Mikrodenetleyici kart
- D) DC Motor sürücüler
- E) Kablosuz erişim kartları



**15. Mikrodenetleyiciden gelen motor sinyallerini motora ileterek, robotların hareketlenmesini sağlayan elektronik devre aşağıdakilerden hangisidir?**

- A) Sensörler
- B) Özel robot kartı
- C) DC motor sürücü kartı
- D) Servo motor sürücü kartı
- E) Kablosuz erişim kartları

**16. Robotları farklı elektronik araçlarla uzaktan erişimle kontrol etmeyi sağlayan elektronik parça aşağıdakilerden hangisidir?**

- A) Sensörler
- B) Özel robot kartı
- C) DC motor sürücü kartı
- D) Kablosuz erişim kartları
- E) Servo motor sürücü kartı

Meraklısına

Doğadaki canlıların farklı yeteneklerinden ilham alan robotlarla ilgili detaylı ve ilginç gelişmeleri içeren makaleye aşağıda belirtilen linkten ulaşabilirsiniz.

<https://bilimgenc.tubitak.gov.tr/makale/dogadan-ilham-alan-robotlar>

## SIRA SİZDE KONTROL LİSTELERİ VE DEĞERLENDİRME ÖLÇEKLERİ

### 1. Değerlendirme Ölçeği

Ölçütler	4 (Çok iyi)	3 (İyi)	2 (Orta)	1 (Zayıf)
Özgün bir proje konusu belirler.				
Proje konusunun ayrıntılarını hazırlar.				
Projesini sunar.				
Zamanı verimli kullanır.				

### 1. Kontrol Listesi

Ölçütler	Evet	Hayır
Kontrol ilkelerine göre çalışan robotları internetten araştırır.		
Sunum için gerekli dokümanları toplar.		
Çalışmada uyuma ve düzene önem verir.		
Zamanı verimli ve iyi kullanır.		

### 2. Değerlendirme Ölçeği

Ölçütler	4 (Çok iyi)	3 (İyi)	2 (Orta)	1 (Zayıf)
Robot türleriyle ilgili araştırma yapar.				
Robot türleriyle ilgili gerekli dokümanları toplar.				
Çalışmada uyuma önem verir.				
Çalışmada düzene önem verir.				
Zamanı verimli kullanır.				

## 3. Değerlendirme Ölçeği

Ölçütler	4 (Çok iyi)	3 (İyi)	2 (Orta)	1 (Zayıf)
Robot türü ile ilgili elektronik parçalar için araştırma yapar.				
Robot çalışması için gerekli verileri (resim, yazı, ses vb.) toplar.				
Robot türünün elektronik parçaları ile ilgili sunumunu yapar.				

## 2. Kontrol Listesi

Ölçütler	Evet	Hayır
Simülasyon programını başlatır.		
LED yakıp söndürme uygulamasının algoritmasını yazar.		
DeneySEL devre elemanlarını çalışma ortamına taşır.		
Deneydeki direnç (rezistör) ve LED elemanlarını çalışma alanına taşıyarak bağlantılarını yapar.		
Simülasyon programında hazır LED yakıp söndürme kodlarını düzenler.		
Simülasyonu çalıştırarak değişimi görür.		

# KONULAR

- 2.1. SİSTEM GEREKSİNİMLERİNE UYGUN MİKRODENETLEYİCİ KART
- 2.2. MİKRODENETLEYİCİ KARTIN BİLGİSAYAR BAĞLANTISI VE ÖRNEK PROGRAM YÜKLENMESİ
- 2.3. MİKRODENETLEYİCİ KART İLE LED UYGULAMALARI
- 2.4. MİKRODENETLEYİCİ KART İLE DİJİTAL GİRİŞ UYGULAMALARI
- 2.5. MİKRODENETLEYİCİ KART İLE SERİ PORT UYGULAMALARI
- 2.6. MİKRODENETLEYİCİ KART İLE DİZİ UYGULAMALARI
- 2.7. MİKRODENETLEYİCİ KART İLE ANALOG GİRİŞ UYGULAMALARI
- 2.8. IŞIK ETKİLİ DİRENÇ (LDR) UYGULAMALARI
- 2.9. MİKRODENETLEYİCİ KART İLE ANALOG ÇIKIŞ (PWM) UYGULAMALARI
- 2.10. MİKRODENETLEYİCİ KART İLE RGB LED UYGULAMALARI
- 2.11. MİKRODENETLEYİCİ KART İLE BUZZER UYGULAMALARI
- 2.12. MİKRODENETLEYİCİ KART İLE 7 SEGMENT DISPLAY UYGULAMALARI
- 2.13. MİKRODENETLEYİCİ KART İLE IR ALICI UYGULAMALARI
- 2.14. MİKRODENETLEYİCİ KART İLE MESAFE SENSÖRÜ UYGULAMALARI
- 2.15. MİKRODENETLEYİCİ KART İLE LCD UYGULAMALARI

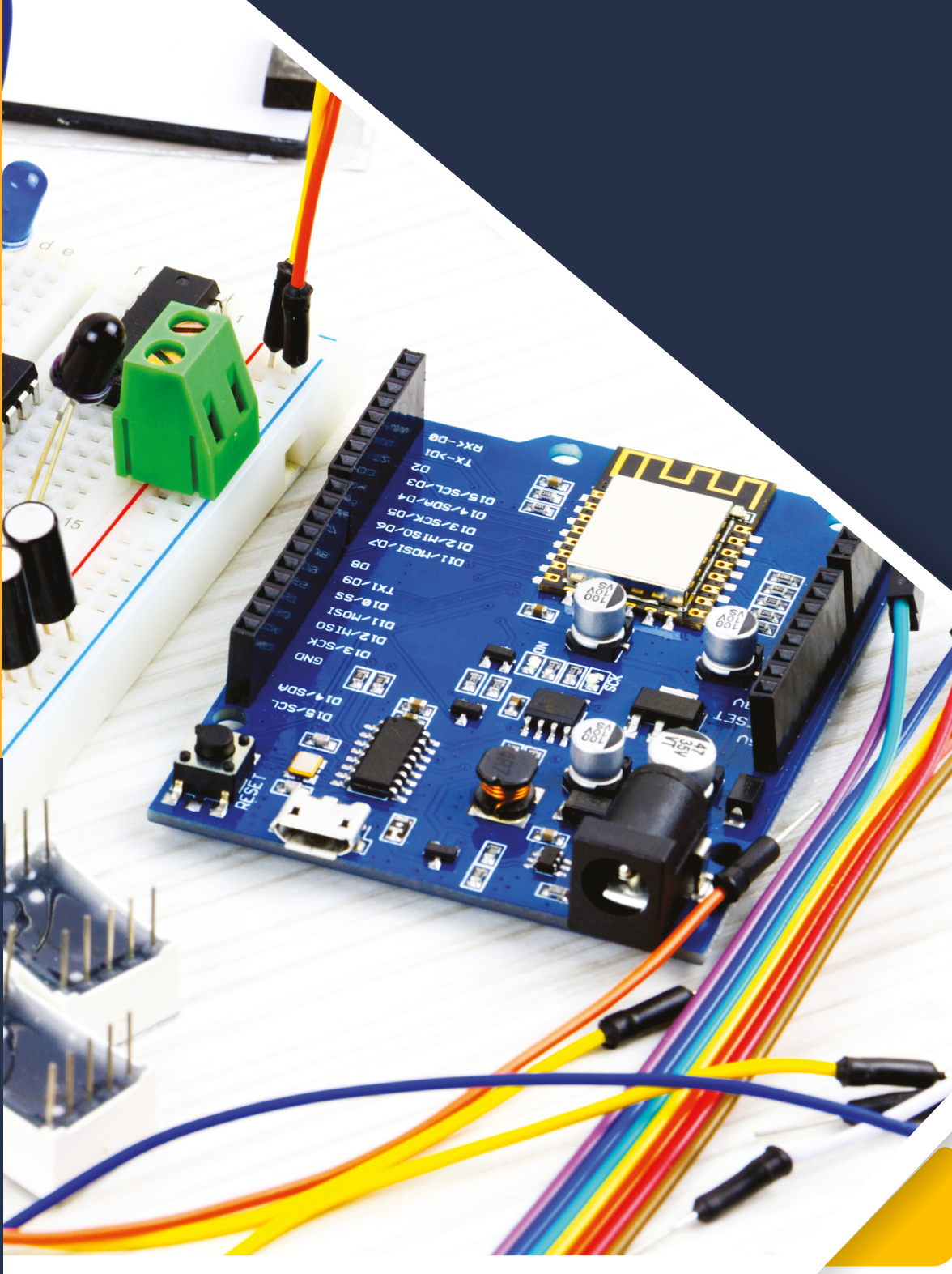
## NELER ÖĞRENECEKSİNİZ?

- Mikrodenetleyici kart yazılımının kurulumu
- Bilgisayar ile uygulama kartı arasında bağlantı kurma
- LED, direnç ve diğer devre elemanlarını breadboard üzerine yerleştirme
- Mikrodenetleyici kartının pin bağlantıları
- Mikrodenetleyici IDE uygulamalarının kod yapısı
- Dijital pinler içinde yer alan PWM pinleri
- Potansiyometreden analog değer okuma
- Seri port ekranından bilgi takibi yapma
- Buton ve anahtar ile dijital giriş uygulamaları
- Işığa duyarlı dirençleri ve sıcaklık sensörünü kullanma
- 7 Segment display yapısı ve uygulamaları
- LCD ekran yapısı ve uygulamaları
- Buton yapısı ve uygulamaları
- Uygulamalar için gerekli ortamı kurma



### Hazırlık Çalışmaları

1. Analog giriş, PWM, dijital giriş ve dijital çıkış kavramları size ne ifade ediyor?
2. Robotik kodlamada kullanılan elektronik bileşenleri araştırınız. Araştırmanızın sonuçlarını arkadaşlarınızla paylaşınız.
3. Elektronik parçalardan oluşan makineler, istenen işlevi nasıl yerine getirmektedir? Bu konuda neler düşünüyorsunuz.



## 2. ÖĞRENME BİRİMİ

### Mikrodenetleyici Kart Programlama

#### Temel Kavramlar

- Mikrodenetleyici kartları
- Mikrodenetleyici IDE programı
- Breadboard
- Tinkercad
- Direnç
- LED, LDR
- 7 segment display,
- Ultrasonik mesafe sensörü
- Seri iletişim
- Buzzer
- Sıcaklık sensörü,
- 1602 LCD ekran

### 2.1. SİSTEM GEREKSİNİMLERİNE UYGUN MİKRODENETLEYİCİ KART

Projenin temelini oluşturacak hazırlıklardan biri de sisteme uygun mikrodenetleyici kartı seçmek ve geliştirme ortamını çalışmalarınız için hazır hâle getirmektir.

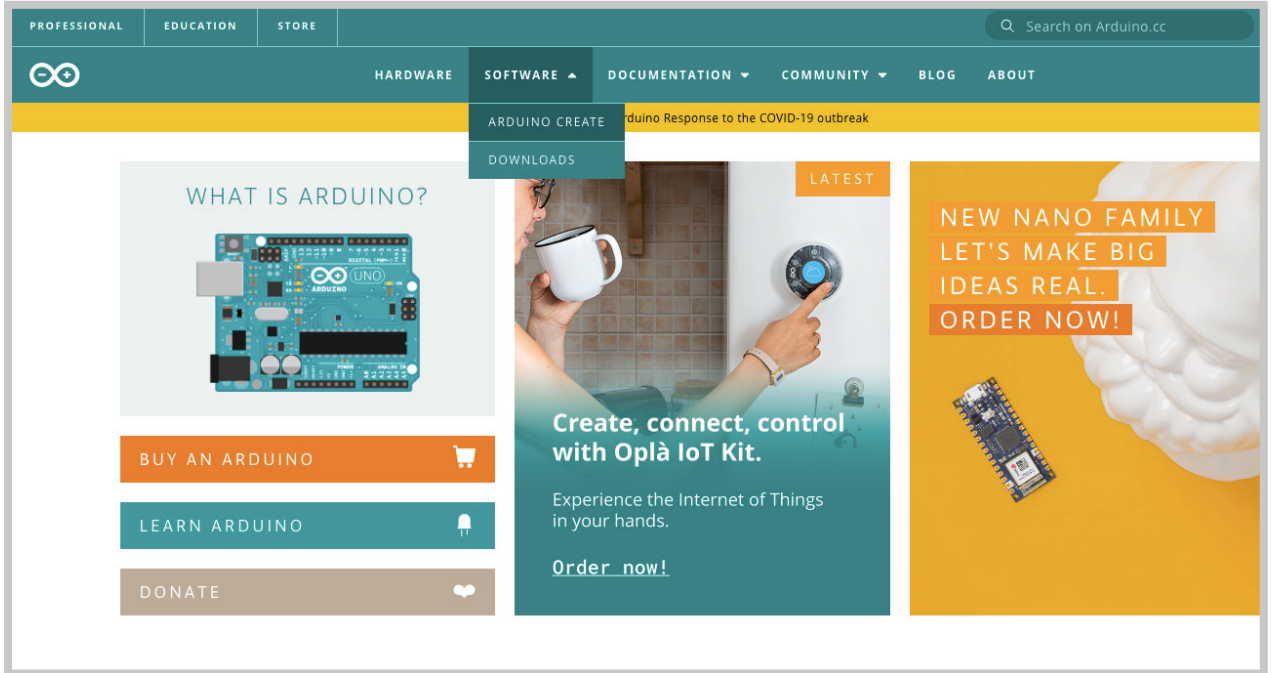
#### 2.1.1. Mikrodenetleyici IDE Kurulumu

IDE, İngilizce Integrated Development Environment (Tümleşik Geliştirme Ortamı) kelimelerinin kısaltmasından oluşmuştur. Her mikrodenetleyiciye özel olarak geliştirilmiş mikrodenetleyici IDE programları mevcuttur.

#### 1. Uygulama

Bu öğrenme birimi altındaki uygulamalarda, kullanılacak program kodlarının derlenebilmesi ve mikrodenetleyici karta yüklenebilmesi için ihtiyacınız olacak masaüstü tabanlı mikrodenetleyici IDE kurulumu gerçekleştirilecektir. Mikrodenetleyici kartın sayfasındaki (<https://www.arduino.cc/en/main/software>) yazılımlarını yükleyerek hem sürücülerini bilgisayara yüklemiş olursunuz hem de mikrodenetleyici kartını bilgisayara tanıtırıp kullanmaya başlayabilirsiniz.

**1. Adım:** Görsel 2.1'de uygulama kartının resmi internet sayfasını açarak **SOFTWARE** menüsünden **DOWNLOADS** seçeneğine tıklayınız.



Görsel 2.1: Mikrodenetleyici kartının resmi internet sayfası

**2. Adım:** Görsel 2.2'de verilen indirme seçeneklerinden kullandığınız işletim sistemine göre seçim yapınız.



**Downloads**

**Arduino IDE 1.8.13**

The open-source Arduino Software (IDE) makes it easy to write code and upload it to the board. This software can be used with any Arduino board.

Refer to the [Getting Started](#) page for Installation instructions.

**SOURCE CODE**

Active development of the Arduino software is [hosted by GitHub](#). See the instructions for [building the code](#). Latest release source code archives are available [here](#). The archives are PGP-signed so they can be verified using [this](#) gpg key.

**DOWNLOAD OPTIONS**

**Windows** Win 7 and newer  
**Windows** ZIP file

**Windows app** Win 8.1 or 10 [Get](#)

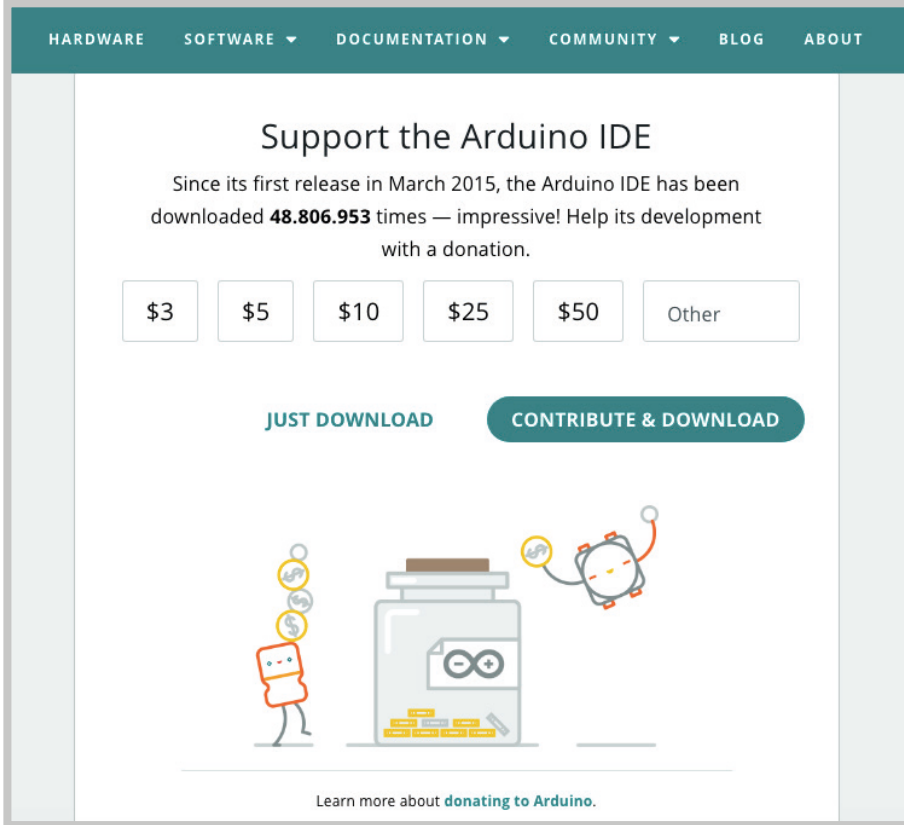
**Linux** 32 bits  
**Linux** 64 bits  
**Linux** ARM 32 bits  
**Linux** ARM 64 bits

**Mac OS X** 10.10 or newer

[Release Notes](#) [Checksums \(sha512\)](#)

**Görsel 2.2: İşletim sistemine göre seçim yapılması**

**3. Adım:** Kurulumun şu ana kadar 48.806.953 kez indirildiği Görsel 2.3'te görülmektedir. Bağlantıya tıkladığında **Contribute to the Arduino Software** sayfası karşınıza gelecektir. Kullanılacak uygulama kartı açık kaynak kod olduğu için geliştirilmesine destek vermeniz yani bağış yapmanız istenmektedir. Bu seçenekleri değerlendirebilir ya da sadece indir anlamına gelen **JUST DOWNLOAD** seçeneğine tıklayarak bir sonraki adıma geçebilirsiniz.



**Support the Arduino IDE**

Since its first release in March 2015, the Arduino IDE has been downloaded **48.806.953** times — impressive! Help its development with a donation.

[\\$3](#) [\\$5](#) [\\$10](#) [\\$25](#) [\\$50](#) [Other](#)

[JUST DOWNLOAD](#) [CONTRIBUTE & DOWNLOAD](#)

[Learn more about donating to Arduino.](#)

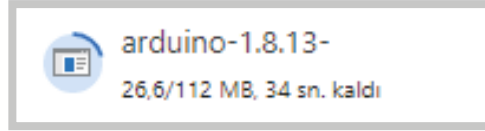
**Görsel 2.3: Bağış yaparak ya da yapmadan indirme ekranı**

## 2. Öğrenme Birimi: Mikrodenetleyici Kart Programlama



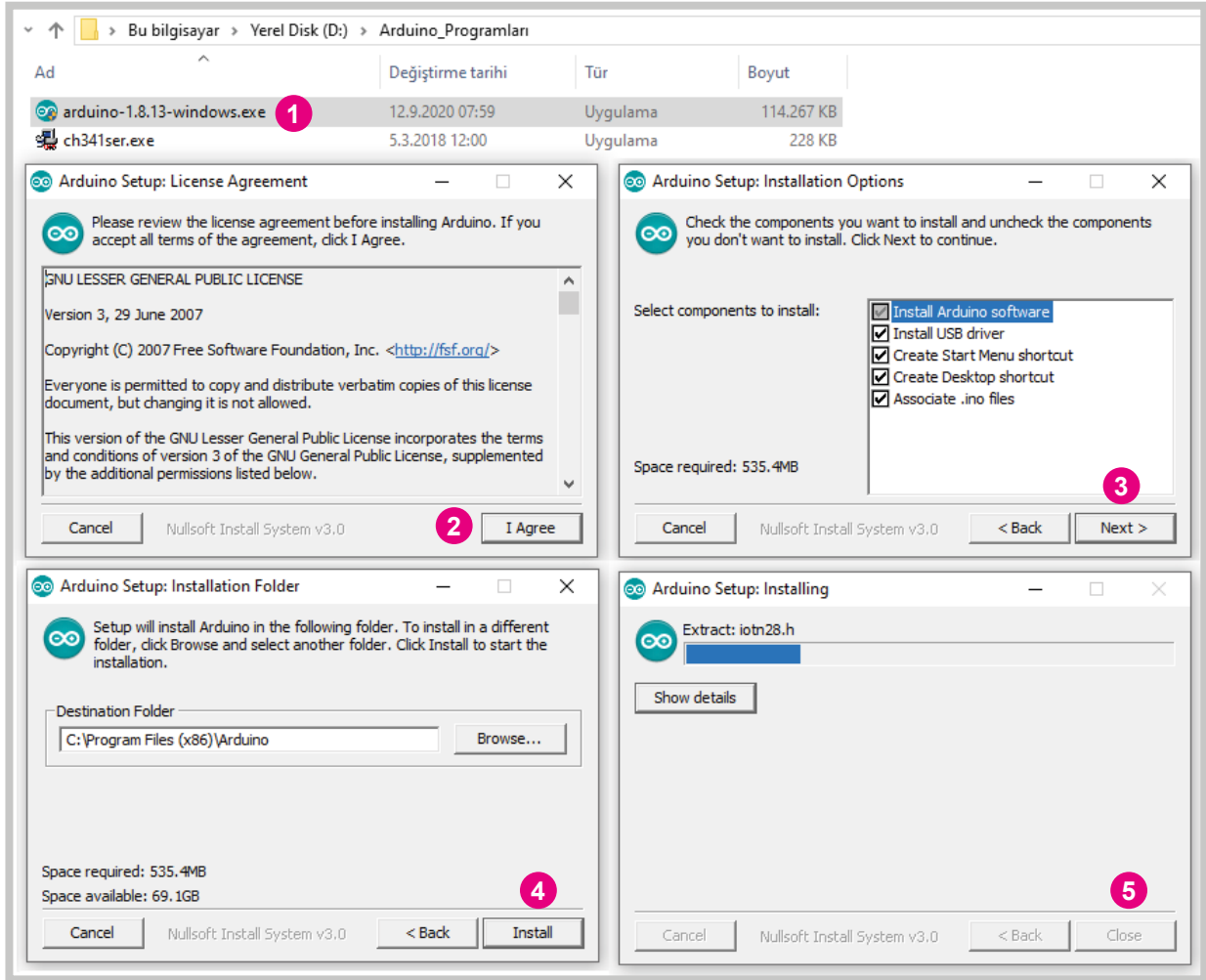
Açık kaynak yazılımlar, bir bilgisayar yazılımının başka kişilerce görülmesine, kullanılmasına ve değiştirilmesine imkân tanıyan yazılımlardır.

**4. Adım:** Görsel 2.4'te bilgisayarın sol alt köşesinde 112 MB kapasitesinde kurulum dosyasının indirilmekte olduğu görülmektedir.



Görsel 2.4: Kurulum dosyasının indirilmesi

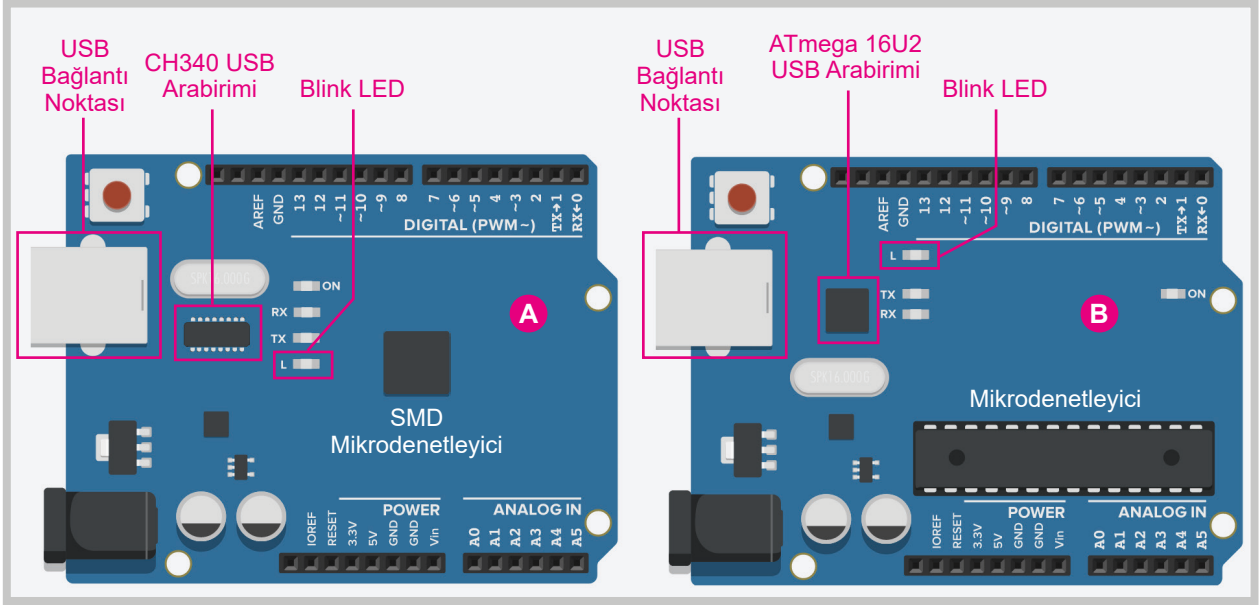
**5. Adım:** Kurulum dosyası bilgisayara indirildikten sonra Görsel 2.5'teki kalan adımları takip ederek kurulum işlemini gerçekleştiriniz.



Görsel 2.5: Mikrodenetleyici IDE programının kurulum aşamaları



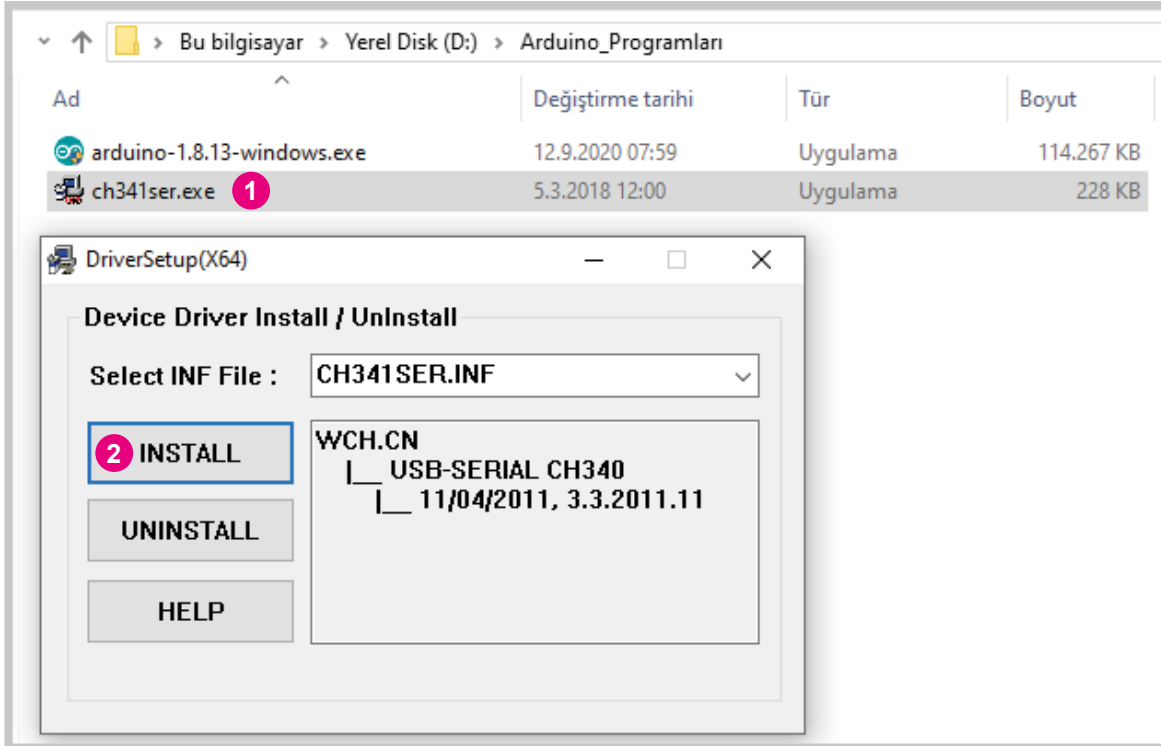
Mikrodenetleyici kart çeşitlerinden yaygın olarak kullanılan iki farklı kart yapısı Görsel 2.6'da gösterilmiştir.



Görsel 2.6: Mikrodenetleyici kart çeşitleri ve USB arabirimleri

Bu kart çeşitlerinden Görsel 2.6'da A ile gösterilen kart **klon kart** olarak tanınır. Diğer karttan farkı USB arabirimi olarak CH340 çipini kullanmasıdır. Uygulamalarınızda Görsel 2.6'da B ile gösterilen mikrodenetleyici kartı kullanacaksanız bu mikrodenetleyici IDE programı için herhangi bir ek sürücü kurulmasına gerek yoktur.

Uygulamalarınızda **klon kart** kullanacaksanız CH341Ser isminde bir sürücünün Görsel 2.7'de görüldüğü gibi yüklenmesi gerekir.



Görsel 2.7: Klon mikrodenetleyiciler için CH341Ser sürücü kurulumu

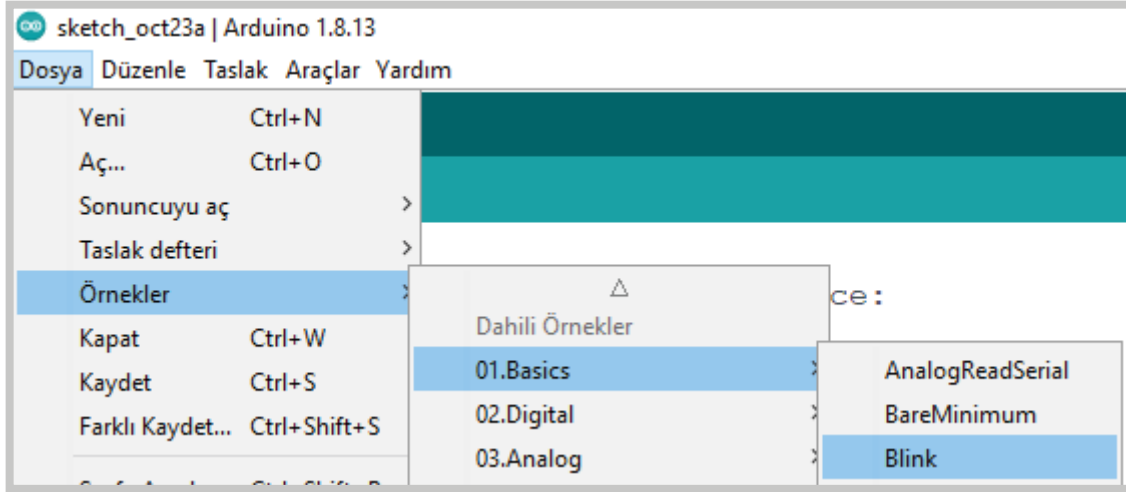
### 2.2. MİKRODENETLEYİCİ KARTIN BİLGİSAYAR BAĞLANTISI VE ÖRNEK PROGRAM YÜKLENMESİ

Mikrodenetleyici IDE programının kurulumu yapıldıktan sonra, mikrodenetleyici kartın çalışıp çalışmadığını denemek amacıyla kullanımı kolay ve mikrodenetleyici IDE programı içinde kodları hazır olan **Blink (göz kırpma)** programı kullanılacaktır.

#### 2. Uygulama

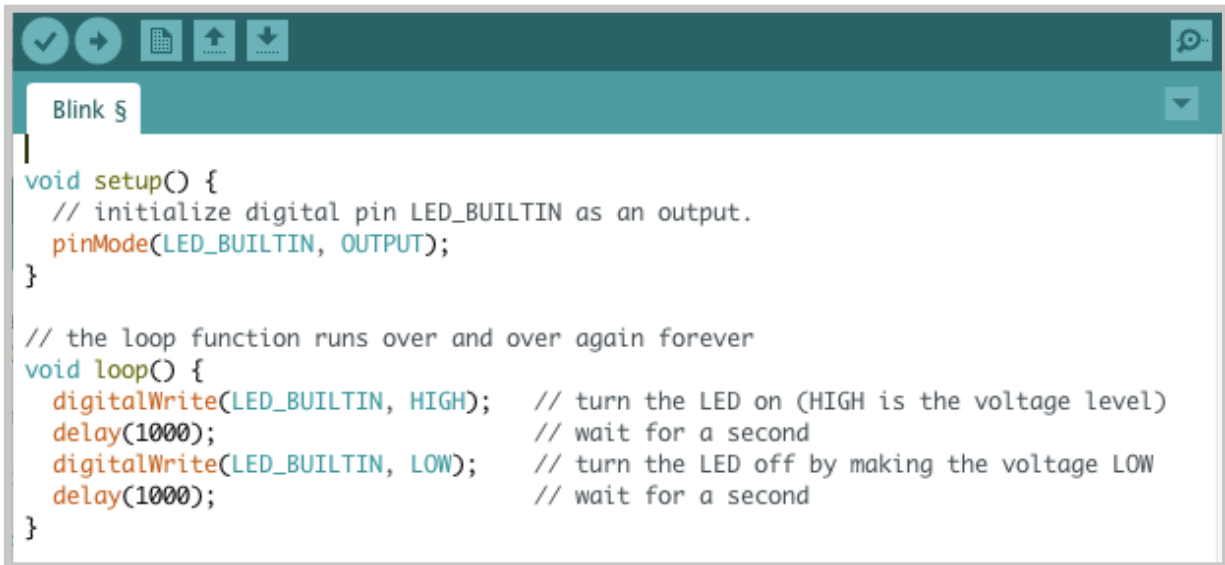
Mikrodenetleyici kartınızın çalışıp çalışmadığını kontrol etmek için aşağıdaki adımları takip ediniz.

**1. Adım:** Mikrodenetleyici IDE programı çalıştırıldıktan sonra Görsel 2.8'deki gibi Blink programı açılır.



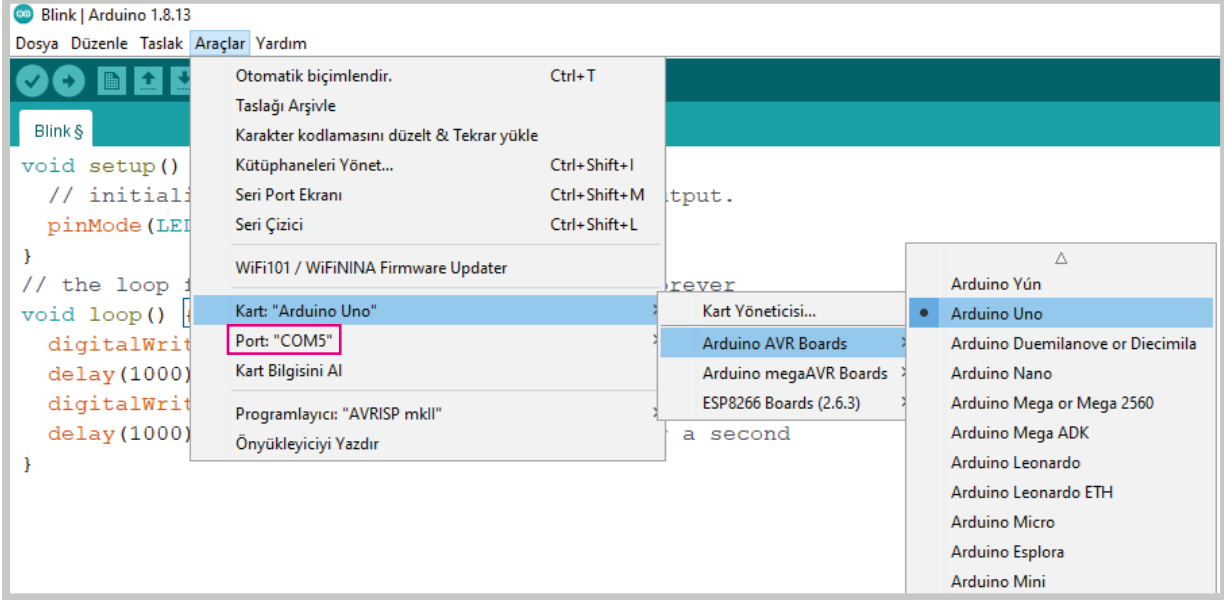
Görsel 2.8: Blink örnek programının açılması

**2. Adım:** Blink programına ait bir uygulamayı birinci üniteye bilgisayarlı çizim programı kullanarak denemiştiniz. Blink programı için mikrodenetleyici karta ekstra bir devre elemanı takılmasına gerek yoktur. Blink programının amacı, Görsel 2.6'daki mikrodenetleyici kart üzerinde bulunan 13 numaralı giriş çıkış portuna bağlı LED'in 1 saniye aralıkla yanıp sönmelerini sağlamaktır. Blink programının kodları Görsel 2.9'da gösterilmiştir.



Görsel 2.9: Blink örnek program kodları

**3. Adım:** Mikrodenetleyici kartın Blink programını tanıyıp çalıştırması için USB kablusuyla mikrodenetleyici kart bilgisayara donanımsal olarak bağlanır. Bağlantı sağlandıktan sonra Görsel 2.10'da olduğu gibi **Araçlar** menüsünden kart türü seçilir. Programda varsayılan kart türü olarak **Arduino Uno** seçilmiştir. Farklı bir kart türünün kullanımında bu kart türü adı kullanılacak kart türü ile değiştirilmelidir.



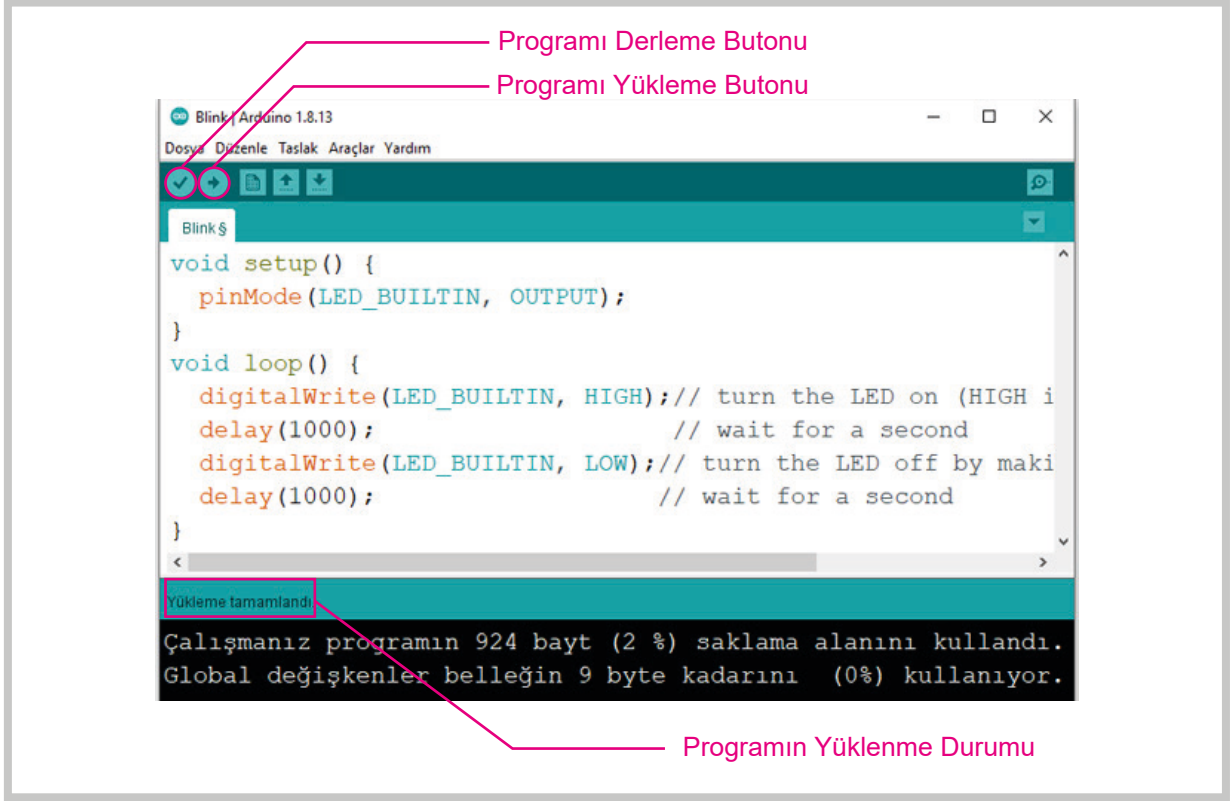
Görsel 2.10: Mikrodenetleyici kart çeşidi ve bağlantı portunun seçilmesi

**4. Adım:** Program yüklenirken diğer bir önemli konu da **port** numarasıdır. Görsel 2.10'da görüldüğü gibi mikrodenetleyici IDE programı mikrodenetleyici kartın bilgisayara bağlandığı USB portunu **COM5** olarak tanımıştır. Bu port numarası kullanılan bilgisayara ve farklı USB bağlantı noktalarına göre (COM1, COM2, COM3 vb.) değişiklik gösterebilir.



Port bölümünde mutlaka bir port numarası olmalıdır. Port numarası bilgisayarda hiç gözüküyorsa bilgisayar kartı tanımamış veya kartın USB arabirimi arızalanmış demektir.

**5. Adım:** Bu adıma kadar işlemler tamamlandıysa Görsel 2.11'de belirtilen yükleme butonuna basılır. Program yüklenmeden önce derleme işlemi yapar ve hata yoksa yükleme aşamasına geçilir. Yükleme tamamlandığında karşınıza "**Yükleme tamamlandı**" şeklinde bilgi ekranı gelir. Mikrodenetleyici kart üzerindeki Blink LED'in 1 saniye aralıklarla yanıp söndüğü görülecektir.

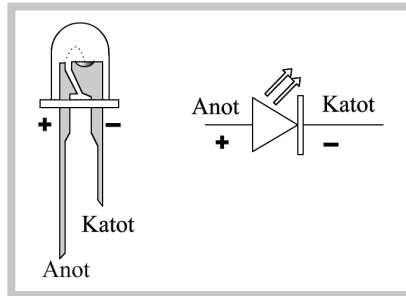


Görsel 2.11: Mikrodenetleyici karta programın yüklenmesi

### 2.3. MİKRODENETLEYİCİ KART İLE LED UYGULAMALARI

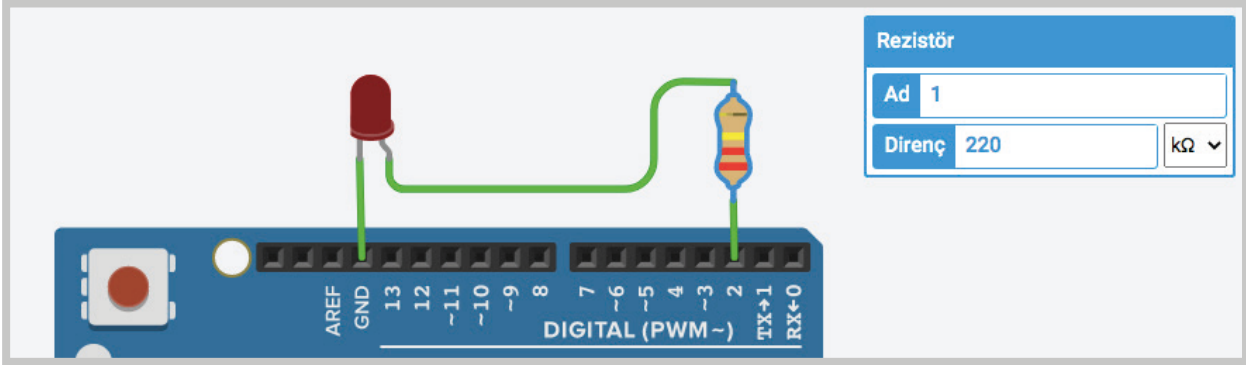
LED'lerin kullanımı kolay olduğundan, mikrodenetleyici kart üzerindeki giriş çıkış portlarının çalışmasını kontrol eden uygulamalarda sıklıkla kullanılır. LED kısaca **ışık veren diyot** (yalnızca bir yönde akım geçiren devre elemanı) anlamına gelir. Bir mikrodenetleyici kartın çıkış portuna LED bağlandığında LED ışık veriyorsa port çıkış 1 (çalışıyor), ışık vermiyorsa port çıkış 0 (çalışmıyor) anlamına gelir.

LED'lerin Görsel 2.12'de görüldüğü gibi **anot (+)** ve **katot (-)** uçları vardır. LED'in ışık vermesi için **katot (-)** ucu mikrodenetleyicinin GND (topraklama) portuna, **anot (+)** ucu da D2'den D13'e kadar herhangi bir çıkış portuna bağlanabilir. LED'in uçları ters bağlarsa LED ışık vermez ama zarar da görmez.



Görsel 2.12: LED uçlarının tespit edilmesi

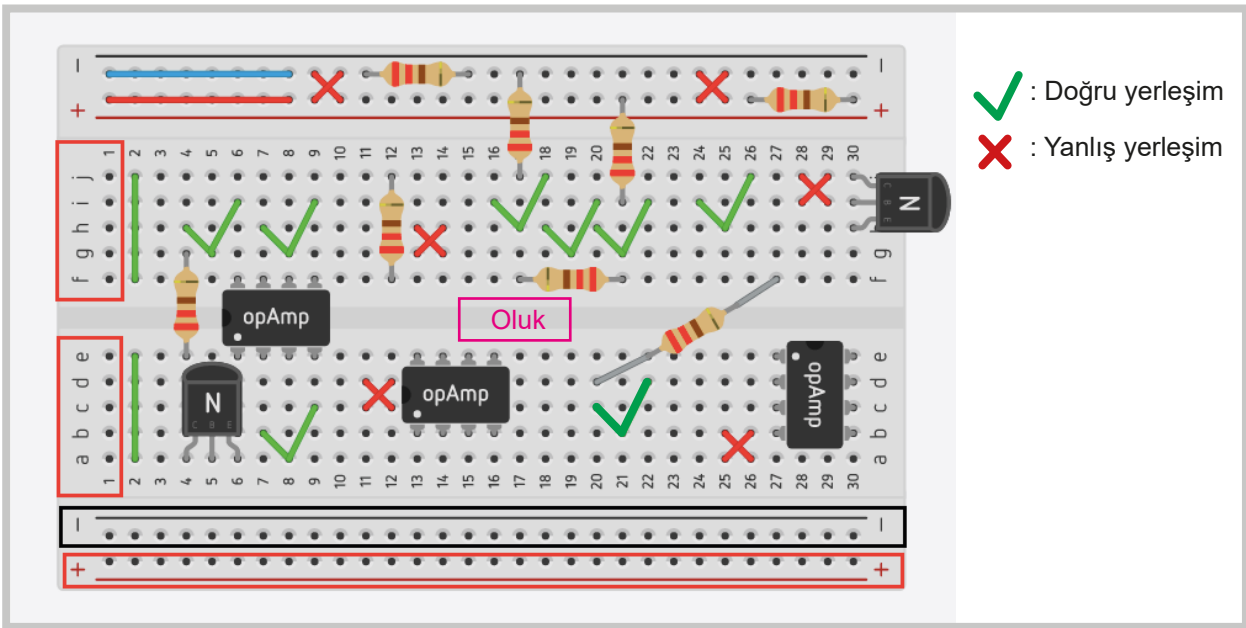
LED'e voltaj fazla geldiğinde LED'in zarar görmemesi için LED'in **anot (+)** ucu Görsel 2.13'te görüldüğü gibi bir direnç ile bağlanmalıdır.



Görsel 2.13: LED'in mikrodenetleyici karta doğru bir şekilde bağlanması

Mikrodenetleyici kartı bilgisayarınıza tanıttıktan ve LED'in görevini öğrendikten sonra devre elemanlarının takılacağı breadboardun kullanımı bilinmelidir. Breadboard, kurulan devrelerin lehim işlemine gerek kalmadan test edilmesini sağlar. Devre elemanları tak-çıkartır şeklinde kullanılabilir. Başka uygulamalarda da tekrar kullanılmasına olanak tanır.

Breadboard bileşeninde Görsel 2.14'te görüldüğü gibi devre elemanlarının bacaklarının takılması için birbirine bağlantılı küçük deliklerden oluşan paralel hatlar bulunur. Dikey kırmızı ve mavi hatlar genellikle gerilim bağlantıları için kullanılır. Kırmızı hatta +, mavi hatta ise GND (toprak) hattı bağlanır. Kırmızı çizgi boyunca uzanan her delik kısa devredir. Orta bölümde bulunan 5'li (**abcde** ve **fg hij** ile ifade edilen) delikler de oluğa kadar kendi içinde dikey olarak birbirine bağlıdır. Breadboard bileşeninin üzerine yerleştirilen devre elemanlarının doğru ve yanlış kullanımları Görsel 2.14'te görülmektedir.



Görsel 2.14: Breadboard kullanımı

### 2.3.1. Mikrodenetleyici Kart ve Bir Adet LED'in Kullanımı

Önceki uygulamada mikrodenetleyicinin çalışması Blink programıyla kontrol edilmiş ve mikrodenetleyici üzerindeki LED'in yanıp sönmesiyle mikrodenetleyicinin çalıştığı anlaşılmıştı. Aynı programlama kodları kullanılarak yapılacak uygulamada bu defa LED'in 1 saniye aralıkla yanıp sönmesi sağlanacaktır.

### 3. Uygulama

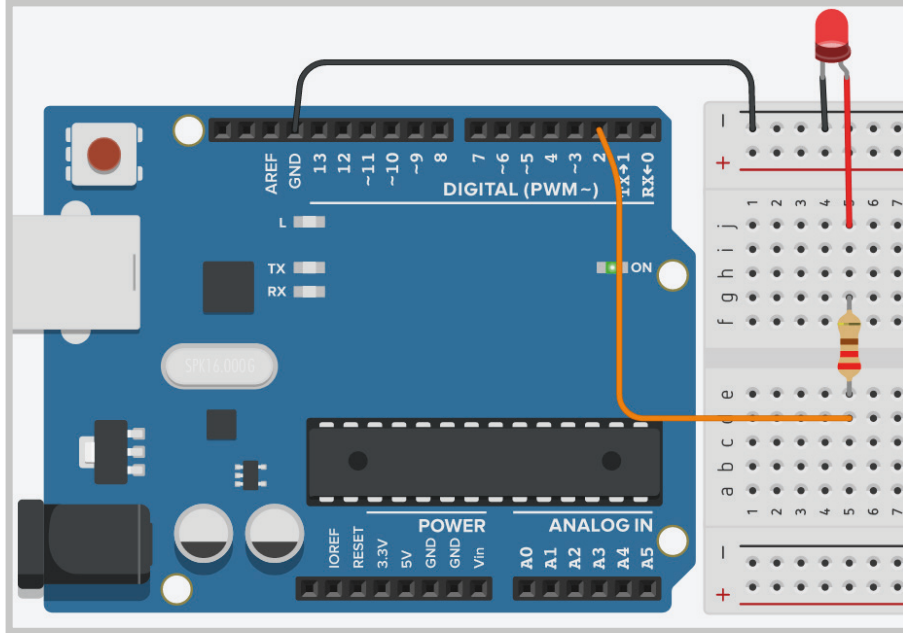
LED'in bir saniye aralıkla yanıp sönmesini sağlayacak uygulamayı aşağıdaki adımları takip ederek yapınız.

#### Gerekli Devre Elemanları

- 1 adet mikrodenetleyici kart
- 1 adet breadboard
- 1 adet LED
- 1 adet 220Ω (kırmızı, kırmızı, kahverengi, altın rengi) direnç
- Bağlantı kabloları

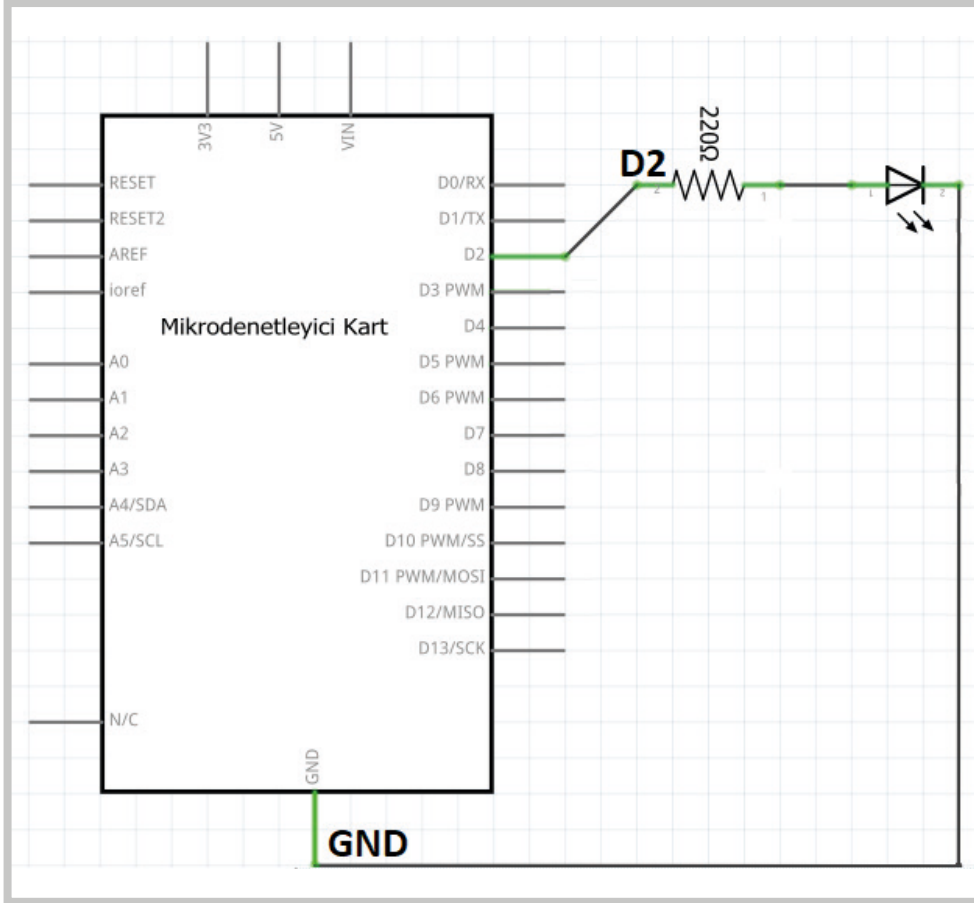
**1. Adım:** Görsel 2.15'te görüldüğü gibi mikrodenetleyicinin GND ucundan bir kablo ile (Görsel 2.15'teki siyah kablo) breadboardun mavi hattına bağlanmalıdır. LED'in Görsel 2.15'teki gibi uçları tespit edildikten sonra katot ucu breadboardun mavi hattına, anot ucu da direncin bir ucuna bağlanır. LED için bağlantı yönü çok fark eder ancak direnç bağlamada yön önemli değildir.

Dirençin bir ucu LED'in anot ucuna diğer ucu ise breadboard oluşunun alt bölümündeki deliklerden birine bağlanmalıdır. Mikrodenetleyici kartın 2 No.lu portundan bir kabloyla (Görsel 2.15'teki sarı kablo) direncin boşta kalan diğer ucuna bağlanmalıdır.



Görsel 2.15: Mikrodenetleyici kart ve breadboard ile LED uygulama devresi

Devre elemanlarının mikrodenetleyiciye olan bağlantıları Görsel 2.16'daki elektriksel devre gösteriminde daha net görülmektedir.



Görsel 2.16: Mikrodenetleyici kart ile LED uygulamasının elektriksel devresi

**2. Adım:** Gerekli devre bağlantıları yapıldıktan sonra mikrodenetleyici IDE programından Blink uygulama kodları tekrar açılır ve devrenin bağlantılarına göre aşağıdaki gibi düzeltilir.

```
void setup() //Tanımlama fonksiyonu
{
  pinMode(2, OUTPUT); // 2 numaralı portu çıkış yap
}
void loop() // Ana program fonksiyonu
{
  digitalWrite(2, 1); // 2 numaralı portu 1 yap
  delay(1000); // 1000 milisaniye (1 saniye) bekle
  digitalWrite(2, 0); // 2 numaralı portu 0 yap
  delay(1000); // 1000 milisaniye (1 saniye) bekle
}
```

## 2. Öğrenme Birimi: Mikrodenetleyici Kart Programlama



### Sıra Sizde

Görsel 2.15 veya Görsel 2.16'daki devreyi breadboard üzerine kurunuz. Mikrodenetleyici IDE programını açarak yukarıdaki kodları yazınız. Tüm işlemler bittikten sonra programı mikrodenetleyici karta yükleyerek devrenin çalışmasını gözlemleyiniz.



### Değerlendirme

Çalışmalarınız öğrenme biriminin sonunda yer alan Kontrol Listesi kullanılarak değerlendirilecektir. Çalışmalarınızı yaparken değerlendirme ölçütlerini dikkate alınız.

### 2.3.2. Mikrodenetleyici Kart ile 2 LED'in Kullanımı

Bu uygulamada amaç, Görsel 2.17 ve Görsel 2.18'de görüldüğü gibi mikrodenetleyici kartın 2 ve 3 numaralı portlarına bağlı 2 farklı LED'in sırasıyla yanmasını sağlamaktır.



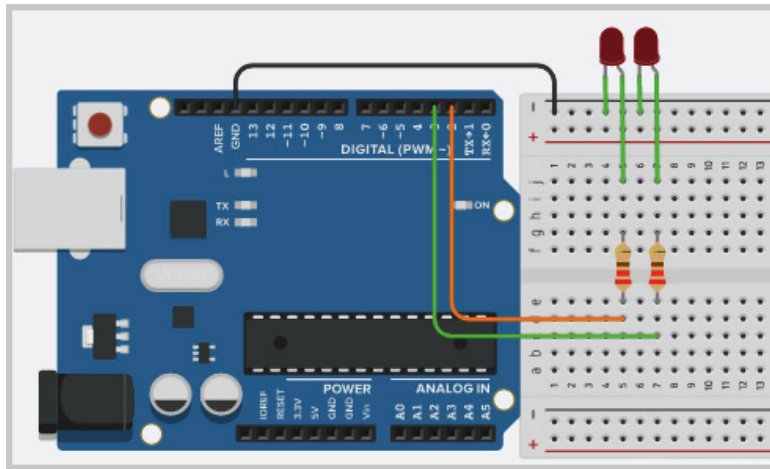
### 4. Uygulama

Mikrodenetleyici kartın 2 ve 3 numaralı portlarına bağlı 2 farklı LED'in sırasıyla yanmasını sağlamak için aşağıdaki adımları takip ediniz.

#### Gerekli Devre Elemanları

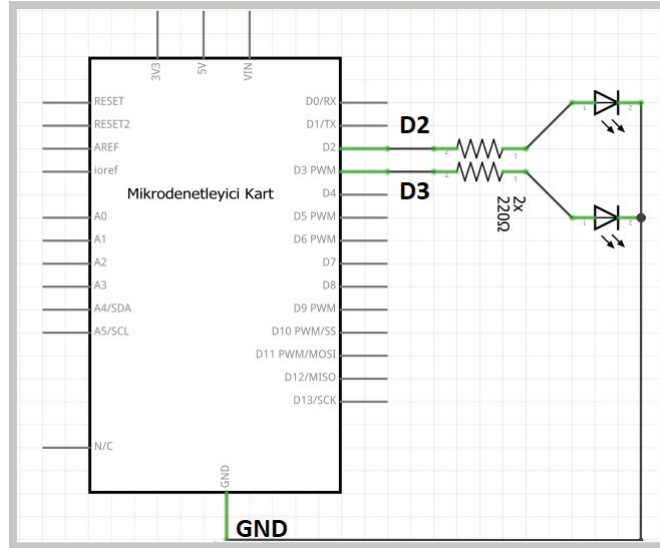
- 1 adet mikrodenetleyici kart
- 1 adet breadboard
- 2 adet LED
- 2 adet 220Ω (kırmızı, kırmızı, kahverengi, altın rengi) direnç
- Bağlantı kabloları

Görsel 2.17'deki uygulama devresi, 1 adet mikrodenetleyici kart, 1 adet breadboard, 2 adet LED, 2 adet 220Ω direnç ve bağlantı kablolarıyla Görsel 2.18'deki elektriksel devre bağlantılarına bakılarak kurulur.



Görsel 2.17: Mikrodenetleyici kart ve breadboard ile 2 LED'li uygulama devresi





Görsel 2.18: Mikrodenetleyici kart ile 2 LED'li uygulamanın elektriksel devresi

**2. Adım:** Devre kurulumu bittikten sonra mikrodenetleyici IDE programından Blink uygulama kodları tekrar açılır ve devrenin yeni durumuna göre kodlar aşağıdaki gibi düzenlenerek program çalıştırılır. Yeniden düzenlenmiş 2 LED'li program kodları aşağıdaki gibidir.

```
void setup() // Tanımlama fonksiyonu
{
    pinMode(2, OUTPUT); // 2 numaralı portu çıkış yap
    pinMode(3, OUTPUT); // 3 numaralı portu çıkış yap
}
void loop() // Ana program fonksiyonu
{
    digitalWrite(2, 1); // 2 numaralı portu 1 yap
    digitalWrite(3, 0); // 3 numaralı portu 0 yap
    delay(1000); // 1000 milisaniye (1 saniye) Bekle
    digitalWrite(2, 0); // 2 numaralı portu 0 yap
    digitalWrite(3, 1); // 3 numaralı portu 1 yap
    delay(1000); // 1000 milisaniye (1 saniye) Bekle
}
```



Sıra Sizde

Görsel 2.17 ve Görsel 2.18'deki devreyi breadboard üzerine kurunuz. Mikrodenetleyici IDE programını açıp yeniden düzenlenmiş 2 LED'li uygulamanın kodlarını yazınız. Tüm işlemler bittikten sonra programı mikrodenetleyici karta yükleyip, devrenin çalışmasını gözlemleyiniz.



Değerlendirme

Çalışmalarınız öğrenme biriminin sonunda yer alan Kontrol Listesi kullanılarak değerlendirilecektir. Çalışmalarınızı yaparken değerlendirme ölçütlerini dikkate alınız.



<http://kitap.eba.gov.tr/KodSor.php?KOD=20779>

### 2.3.3. Mikrodenetleyici Kart ile 5 LED'in Kullanımı

Bu uygulamada amaç, Görsel 2.19 ve Görsel 2.20'de görüldüğü gibi mikrodenetleyici kartın 2, 3, 4, 5 ve 6 numaralı portlarına bağlı 5 farklı LED'in belirli bir sıraya göre yanmasını sağlamaktır.

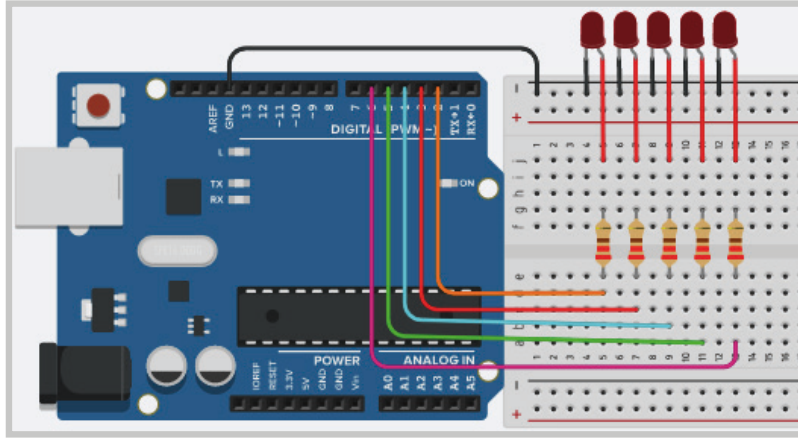
#### 5. Uygulama

Mikrodenetleyici kartın 2, 3, 4, 5 ve 6 numaralı portlarına bağlı 5 farklı LED'in belirli bir sıraya göre yanmasını sağlamak için aşağıdaki adımları takip ediniz.

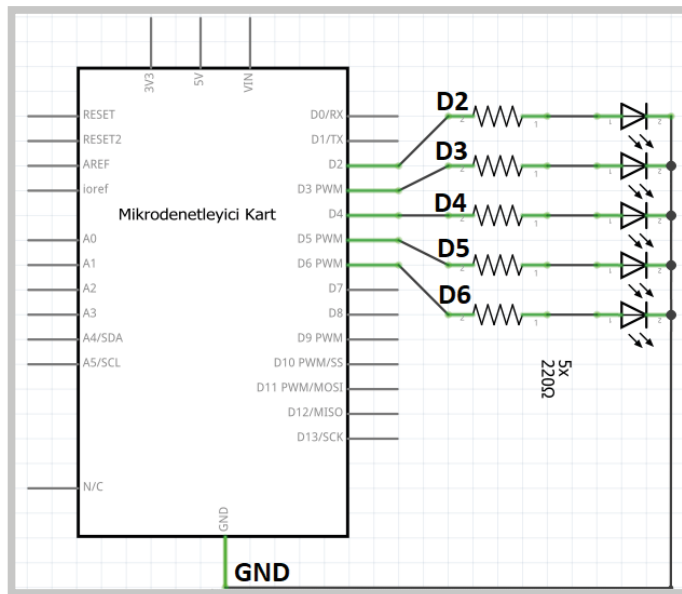
#### Gerekli Devre Elemanları

- 1 adet mikrodenetleyici kart
- 1 adet breadboard
- 5 adet LED
- 5 adet 220Ω (kırmızı, kırmızı, kahverengi, altın rengi) direnç
- Bağlantı kabloları

**1. Adım:** Elemanlar mikrodenetleyici karta sırasıyla Görsel 2.19 ve Görsel 2.20'deki gibi takılır.



Görsel 2.19: Mikrodenetleyici kart ve breadboard ile 5 LED'li uygulama devresi



Görsel 2.20: Mikrodenetleyici kart ile 5 LED'li uygulamanın elektriksel devresi

**2. Adım:** Devre kurulumu yapıldıktan sonra Blink programı tekrar açılır ve devrenin 5 LED'li yeni durumuna göre kodlar değiştirilir. Program aşağıdaki algoritmik sıraya göre düzenlenir.

- LED'ler ilk aşamada sönmük durumda iken 1. LED'den itibaren 5. LED'e kadar 1 saniye aralıklarla sırayla yanar.
- Tüm LED'ler yandıktan sonra 1. LED'den itibaren 5. LED'e kadar tüm LED'ler 1'er saniye aralıklarla sırayla söner.
- LED'lerin yanıp sönmeye olayı, mikrodenetleyici kartın enerjisi kesilene kadar devam eder.

Algoritmik sıraya göre oluşturulan 5 LED'li program kodları aşağıdaki gibidir.

```
void setup() {  
  pinMode(2, OUTPUT); pinMode(3, OUTPUT);  
  pinMode(4, OUTPUT); pinMode(5, OUTPUT);  
  pinMode(6, OUTPUT);  
}  
void loop() {  
  digitalWrite(2, 1); delay(1000);  
  digitalWrite(3, 1); delay(1000);  
  digitalWrite(4, 1); delay(1000);  
  digitalWrite(5, 1); delay(1000);  
  digitalWrite(6, 1); delay(1000);  
  
  digitalWrite(2, 0); delay(1000);  
  digitalWrite(3, 0); delay(1000);  
  digitalWrite(4, 0); delay(1000);  
  digitalWrite(5, 0); delay(1000);  
  digitalWrite(6, 0); delay(1000);  
}
```



### Sıra Sizde

Görsel 2.19 veya Görsel 2.20'de devresi verilen 5 LED'li uygulamanın elemanlarını temin ederek breadboard üzerine kurunuz. Yukarıda verilen program kodlarını mikrodenetleyici IDE programında yazınız. Tüm işlemler bittikten sonra programı mikrodenetleyici karta yükleyip, devrenin çalışmasını gözlemleyiniz.



### Değerlendirme

Çalışmalarınız öğrenme biriminin sonunda yer alan Kontrol Listesi kullanılarak değerlendirilecektir. Çalışmalarınızı yaparken değerlendirme ölçütlerini dikkate alınız.



### Sıra Sizde

Görsel 2.19 veya Görsel 2.20'de devresi verilen 5 LED'li uygulama devresini breadboard üzerine kurunuz. Mikrodenetleyici IDE programını açıp aşağıdaki algoritmik sırayla LED'leri çalıştıran uygulamanın kodlarını yazınız.

- 5. LED'den itibaren sırasıyla tüm LED'ler 1'er saniye aralıklarla yanmaya başlayacak.
- Yanan LED'ler 1. LED yanana kadar sönmeyecek.
- Tüm LED'ler yandıktan sonra sırasıyla 5. LED'den itibaren sönmeye başlayacak.
- Tüm LED'ler söndükten sonra program başa dönüp tekrar 5. LED'den 1. LED'e kadar LED'lerin yanmasını sağlayacak.
- Bu şekilde mikrodenetleyici kartın enerjisi kesilene kadar LED yanma döngüsü devam edecektir.

Algoritmanın programını yazdıktan sonra programı mikrodenetleyici karta yükleyip devrenin çalışmasını gözlemleyiniz.



### Değerlendirme

Çalışmalarınız öğrenme biriminin sonunda yer alan Kontrol Listesi kullanılarak değerlendirilecektir. Çalışmalarınızı yaparken değerlendirme ölçütlerini dikkate alınız.

### 2.3.4. Mikrodenetleyici Kart ile 7 LED'in Kullanımı (for döngüsü ile)

Bu uygulamadaki amaç, Görsel 2.21 ve Görsel 2.22'de görüldüğü gibi mikrodenetleyici kartın 2, 3, 4, 5, 6, 7 ve 8 numaralı portlarına bağlı 7 farklı LED'in belirli bir sıraya göre **for döngüsü** ile yanmasını sağlamaktır.



### 6. Uygulama

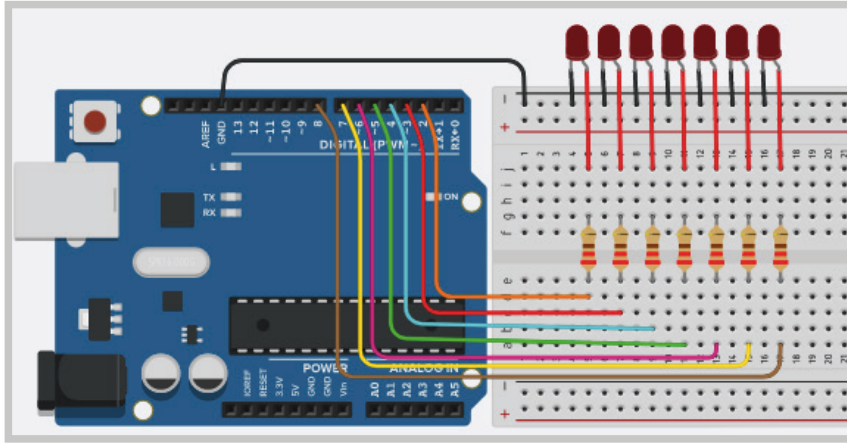
Mikrodenetleyici kartın 2, 3, 4, 5, 6, 7 ve 8 numaralı portlarına bağlı 7 farklı LED'in belirli bir sıraya göre **for döngüsü** ile yanmasını sağlamak için aşağıdaki adımları takip ediniz.

#### Gerekli Devre Elemanları

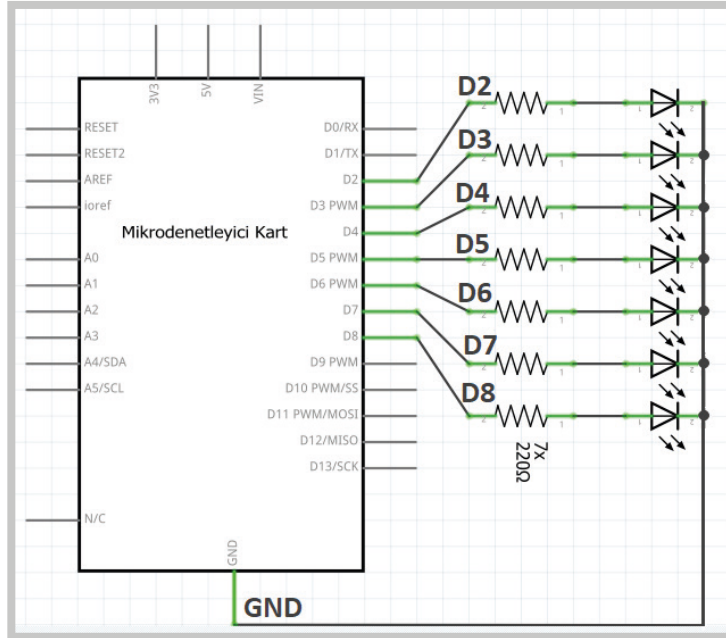
- 1 adet mikrodenetleyici kart
- 1 adet breadboard
- 7 adet LED
- 7 adet 220Ω (kırmızı, kırmızı, kahverengi, altın rengi) direnç
- Bağlantı kabloları

**1. Adım:** Mikrodenetleyici IDE programında kod yazılırken `digitalWrite(port_no, 1);` komutuyla 1 ile 3 arası portu kontrol etmek bir karmaşıklık oluşturmaz. Ancak port sayısı arttıkça hem satır sayısı artacağından hem de program karmaşıklaşacağından **döngü komutlarını** kullanmak gerekli olacaktır.

Görsel 2.19'da 5 LED kullanılarak gerçekleştirilen bir uygulama, Görsel 2.21'de 7 LED'e çıkarıldığında aynı kodları tekrardan yazıp program satırlarını artırmak yerine, döngü komutları kullanılarak satır sayısı sadeleştirilir.



Görsel 2.21: Mikrodenetleyici kart ve breadboard ile 7 LED'li uygulama devresi



Görsel 2.22: Mikrodenetleyici kart ile 7 LED'li uygulamanın elektriksel devresi

**2. Adım:** Görsel 2.21 ve Görsel 2.22'de devresi verilen 7 LED'li uygulamanın program kodları yazılırken 3 kez **for döngüsü** kullanılmalıdır. Programın çalışma mantığı aşağıdaki gibidir.

- Program kodlarını yazarken ilk for döngüsü `void setup()` fonksiyonu içindeki `pinMode(i, OUTPUT)`; komutuyla mikrodenetleyici kartın 2. portundan 8. portuna kadar olan tüm portları çıkış yapar.
- `void loop()` fonksiyonu içindeki 1. for döngüsü `digitalWrite(i, 1)`; komutuyla mikrodenetleyici kartın 2. portundan 8. portuna kadar olan tüm portların çıkış değerlerini 1 saniye aralıklarla 1 yapar.
- Fonksiyon içinde kullanılan 2. for döngüsü `digitalWrite(i, 0)`; komutuyla mikrodenetleyici kartın 2. portundan 8. portuna kadar olan tüm portların çıkış değerlerini 1 saniye aralıklarla 0 yapar.
- Böylece `void loop()` fonksiyonu içinde 14 satırla yazılabilecek komutlar 2 tane for döngüsü kullanılarak 2 satırda yazılmış olur.

## 2. Öğrenme Birimi: Mikrodenetleyici Kart Programlama

**3. Adım:** Mikrodenetleyici kart ile 7 LED'li uygulamanın kodları aşağıdaki gibidir.

```
int i;
void setup() {
  for (i=2;i<=8;i++) pinMode(i, OUTPUT);
}
void loop() {
  for (i=2;i<=8;i++) { digitalWrite(i, 1); delay(1000); }
  for (i=2;i<=8;i++) { digitalWrite(i, 0); delay(1000); }
}
```



### Sıra Sizde

Görsel 2.21 veya Görsel 2.22'de devresi verilen 7 LED'li uygulama devre elemanlarını breadboard üzerine yerleştiriniz. Mikrodenetleyici kart ile 7 LED'li uygulamanın program kodlarını mikrodenetleyici IDE programında yazınız. Tüm işlemler bittikten sonra programı mikrodenetleyici karta yükleyip, devrenin çalışmasını gözlemleyiniz.



http://kitap.eba.gov.tr/KodSor.php?KOD=20781

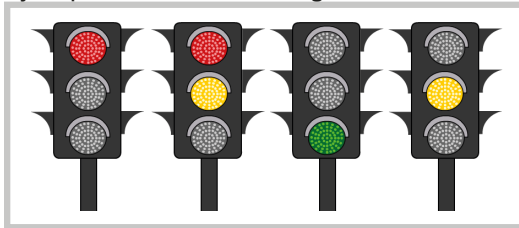


### Değerlendirme

Çalışmalarınız öğrenme biriminin sonunda yer alan Kontrol Listesi kullanılarak değerlendirilecektir. Çalışmalarınızı yaparken değerlendirme ölçütlerini dikkate alınız.

### 2.3.5. Mikrodenetleyici Kart ile Trafik Lambası Kontrolü

Bu bölümde trafik lambası uygulaması yapılacaktır. Kavşaklarda ve yaya geçitlerinde trafikteki geçişlerin düzenli bir şekilde olmasını sağlamak için trafik lambaları kullanılır. Görsel 2.23'te bir trafik ışığı sinyalizasyon devresinde trafik ışıklarının yanıp sönme sıralaması görülmektedir.



Görsel 2.23: Trafik ışıklarının yanıp sönme sıralaması



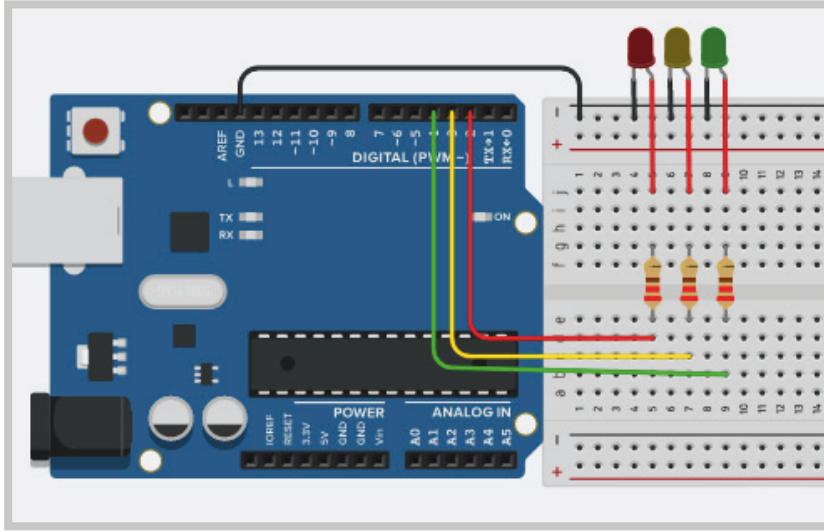
### 7. Uygulama

Trafik ışıklarının yanıp sönme sıralamasını aşağıdaki adımları takip ederek yapınız.

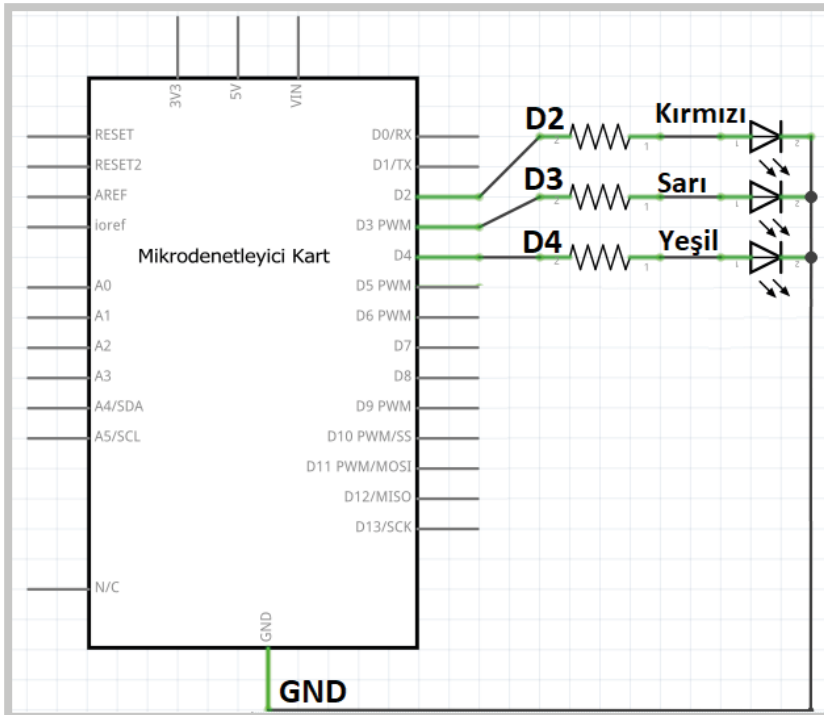
#### Gerekli Devre Elemanları

- 1 adet mikrodenetleyici kart
- 1 adet breadboard
- 3 adet LED
- 3 adet 220Ω (kırmızı, kırmızı, kahverengi, altın rengi) direnç
- Bağlantı kabloları

**1. Adım:** Devrenin kurulumu Görsel 2.24 ve Görsel 2.25'te gösterildiği gibi yapılır.



Görsel 2.24: Trafik sinyalizasyonu uygulama devresi



Görsel 2.25: Trafik sinyalizasyonu uygulamasının elektriksel devresi

**2. Adım:** Görsel 2.24 ve Görsel 2.25'te devresi verilen trafik sinyalizasyonu uygulamasının program kodları yazılırken aşağıdaki hususlara dikkat edilmelidir.

- İlk olarak her LED için değişken tanımlaması yapılır.
- Programda `void setup ()` fonksiyonu içinde `pinMode (k, OUTPUT)` komutu ile portlar çıkış yapılır.
- Programda `void loop ()` fonksiyonu içinde aşağıda verilen trafik ışıklarının (LED'lerin) yanıp sönmeye sıralamasına göre program kodları yazılır.

## 2. Öğrenme Birimi: Mikrodenetleyici Kart Programlama

Trafik sinyalizasyonu program kodları şunlardır:

```
int k=2; //port numarasını deęiřkene aktarır.
int s=3; //port numarasını deęiřkene aktarır.
int y=4; //port numarasını deęiřkene aktarır.
void setup()
{
    pinMode(k, OUTPUT); //k portunu ıkıř yapar.
    pinMode(s, OUTPUT); //s portunu ıkıř yapar.
    pinMode(y, OUTPUT); //y portunu ıkıř yapar.
}
void loop()
{
    digitalWrite(k, 1);
    digitalWrite(s, 0);
    digitalWrite(y, 0);
    delay(5000);
    digitalWrite(s, 1);
    delay(1000);
    digitalWrite(k, 0);
    digitalWrite(s, 0);
    digitalWrite(y, 1);
    delay(5000);
    digitalWrite(s, 1);
    digitalWrite(y, 0);
    delay(1000);
}
```



### Sıra Sizde

Görsel 2.24 veya Görsel 2.25'te devresi verilen trafik sinyalizasyonu uygulaması devre elemanlarını breadboard üzerine yerleřtiriniz. Trafik sinyalizasyonu program kodlarını mikrodenetleyici IDE programında yazınız. Tüm iřlemler bittikten sonra programı mikrodenetleyici karta yükleyip, devrenin alıřmasını gözlemleyiniz.



### Deęerlendirme

alıřmalarınız öęrenme biriminin sonunda yer alan Kontrol Listesi kullanılarak deęerlendirilecektir. alıřmalarınızı yaparken deęerlendirme ölçütlerini dikkate alınız.



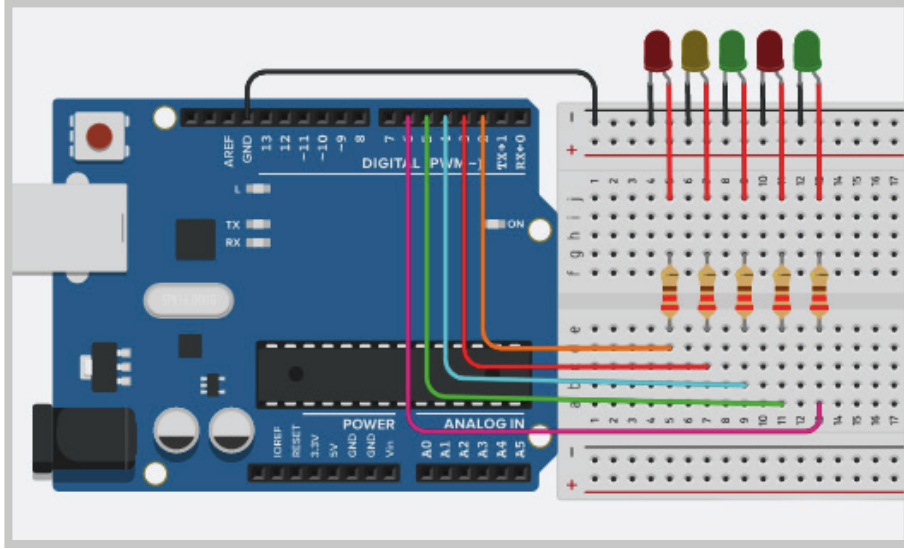
<http://kitap.eba.gov.tr/KodSor.php?KOD=20782>



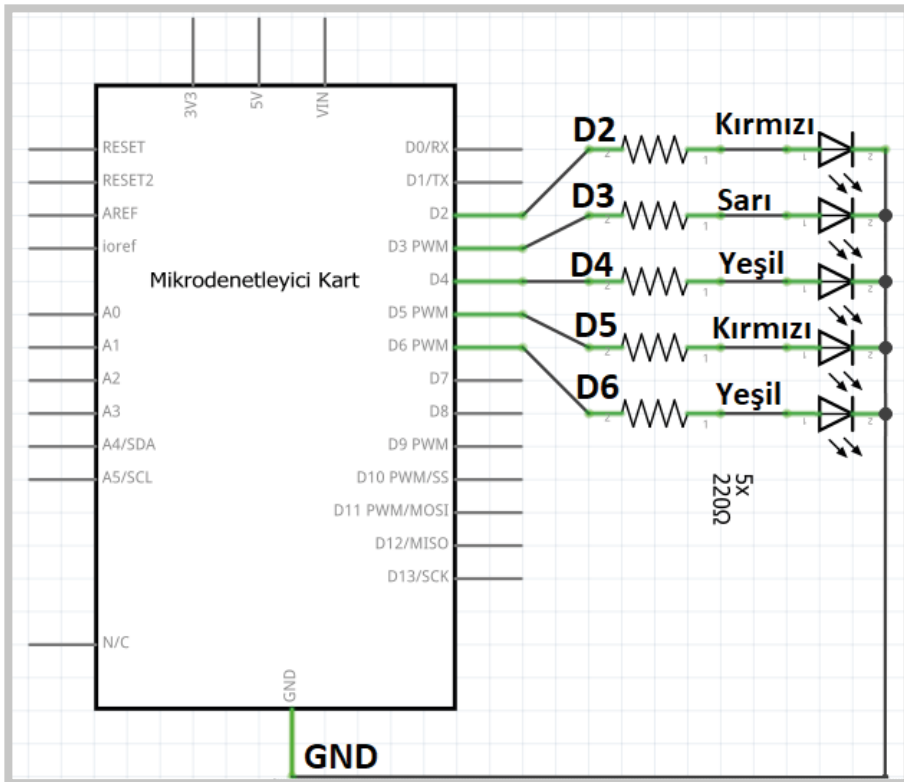
### 8. Uygulama

Trafik ışıkları uygulaması biraz daha geliştirilip, yayalar için sinyalizasyon eklenmiş uygulama yapılacaktır.

**1. Adım:** Görsel 2.26 ve Görsel 2.27’de araçlara yeşil yandığı zaman yayalar için kırmızı, araçlara kırmızı yandığı zaman ise yayalar için yeşil LED yanmasını sağlayacak fazladan 2 LED eklenmiş uygulama devresi görülmektedir.



Görsel 2.26: Yayalar ve araçlar için trafik sinyalizasyonu uygulama devresi



Görsel 2.27: Yayalar ve araçlar için trafik sinyalizasyonu uygulamasının elektriksel devresi

**2. Adım:** Yayalar ve araçlar için trafik sinyalizasyonu program kodları şunlardır:

```
int ak=2; //Araçlara kırmızı ışık için kullanılır.
int as=3; //Araçlara sarı ışık için kullanılır.
int ay=4; //Araçlara yeşil ışık için kullanılır.
int yk=5; //Yayalara kırmızı ışık için kullanılır.
int yy=6; //Yayalara yeşil ışık için kullanılır.

void setup()
{
    pinMode(ak, OUTPUT);
    //ak değişkeni ile belirtilen portu çıkış yapar.
    pinMode(as, OUTPUT);
    //as değişkeni ile belirtilen portu çıkış yapar.
    pinMode(ay, OUTPUT);
    //ay değişkeni ile belirtilen portu çıkış yapar.
    pinMode(yk, OUTPUT);
    //yk değişkeni ile belirtilen portu çıkış yapar.
    pinMode(yy, OUTPUT);
    //yy değişkeni ile belirtilen portu çıkış yapar.
}

void loop()
{
    digitalWrite(ak, 1); //Araçlara kırmızı ışık yakar.
    digitalWrite(as, 0);
    digitalWrite(ay, 0);
    digitalWrite(yk, 0);
    digitalWrite(yy, 1); //Yayalara yeşil ışık yakar.
    delay(5000);
    digitalWrite(as, 1); //Araçlara sarı ışık yakar.
    digitalWrite(yk, 1); //Yayalara kırmızı ışık yakar.
    digitalWrite(yy, 0);
    delay(1000);
    digitalWrite(ak, 0);
    digitalWrite(as, 0);
    digitalWrite(ay, 1); //Araçlara yeşil ışık yakar.
    delay(5000);
    digitalWrite(as, 1); //Araçlara sarı ışık yakar.
    digitalWrite(ay, 0);
    delay(1000);
}
```



### Sıra Sizde

Görsel 2.26 veya Görsel 2.27'de devresi verilen yayalar ve araçlar için trafik sinyalizasyonu uygulaması için devre elemanlarını breadboard üzerine yerleştiriniz. Yayalar ve araçlar için trafik sinyalizasyonu program kodlarını mikrodenetleyici IDE programında yazınız. Tüm işlemler bittikten sonra programı mikrodenetleyici karta yükleyip, devrenin çalışmasını gözlemleyiniz.



### Değerlendirme

Çalışmalarınız öğrenme biriminin sonunda yer alan Kontrol Listesi kullanılarak değerlendirilecektir. Çalışmalarınızı yaparken değerlendirme ölçütlerini dikkate alınız.

## 2.4. MİKRODENETLEYİCİ KART İLE DİJİTAL GİRİŞ UYGULAMALARI

Mikrodenetleyici kartta kullanılan dijital portlar, çıkış olarak kullanılabileceği gibi buton, anahtar veya bir sensörden gelen dijital sinyali kontrol etmek için giriş olarak da kullanılabilir. Mikrodenetleyici kartın dijital portlarını giriş olarak tanımlayabilmek için `void setup()` fonksiyonu içindeki `pinMode(port_no, INPUT)` ; komutu kullanılır.

### 2.4.1. Anahtar Kullanımı

Anahtarlar, elektrik devrelerinde ve elektronik devrelerde bir devreye enerji vermek veya devreden enerji kesmek için kullanılabildiği gibi birden fazla işlemin seçilmesini sağlamak için de kullanılır. Görsel 2.28'de elektronik devrelerde kullanılan anahtar çeşitleri görülmektedir.



Görsel 2.28: Elektronik devrelerde kullanılan anahtar çeşitleri

### 9. Uygulama

Anahtar kullanımını göstermek amacıyla bir uygulama yapılacaktır.

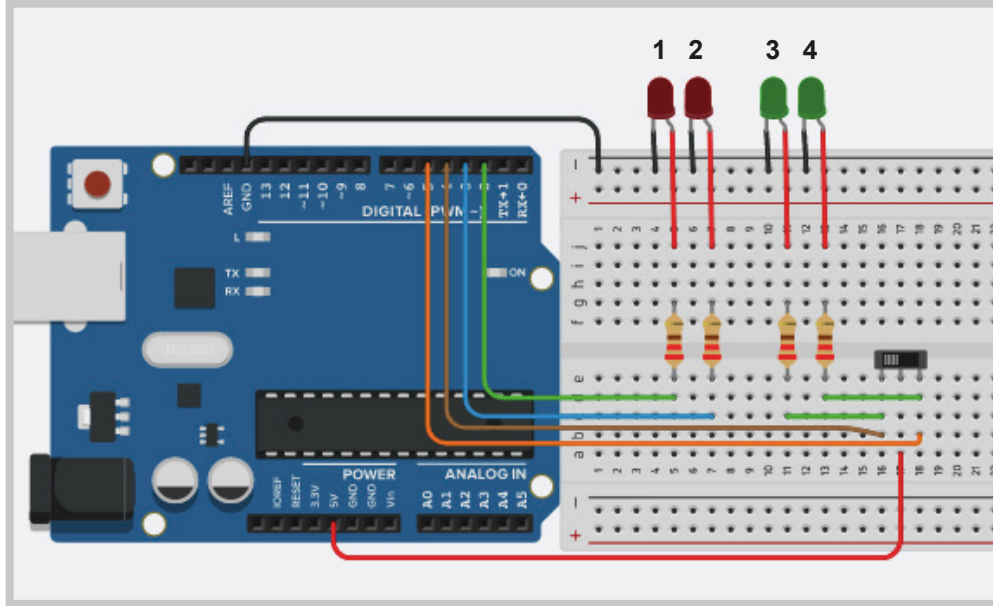
#### Gerekli Devre Elemanları

- 1 adet mikrodenetleyici kart
- 1 adet breadboard
- 1 adet anahtar
- 4 adet LED
- 4 adet 220Ω (kırmızı, kırmızı, kahverengi, altın rengi) direnç
- Bağlantı kabloları

**1. Adım:** Görsel 2.29 ve Görsel 2.30'da gösterilen devrede kullanılan anahtarın ortak ucu, mikrodenetleyici kartın 5V ucuna bağlanmıştır. Anahtarın diğer uçları ise hem yeşil LED'lere hem de mikrodenetleyici kartın D4 ve D5 portlarına bağlanmıştır.

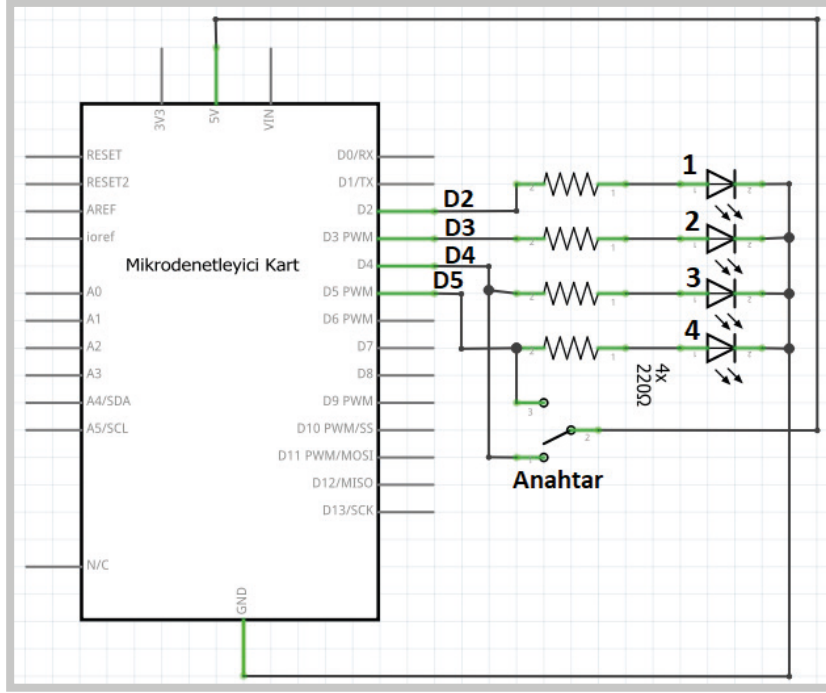
Uygulamanın algoritmik çalışma şekli şu şekildedir:

- Mikrodenetleyici karta herhangi bir program yüklenmeden önce devreye enerji verildiğinde anahtar durumlarına göre D4 ve D5 portlarına bağlı olan LED'ler yanar.
- Anahtar sürgüsü sola çekildiğinde 4. LED, sağa çekildiğinde 5. LED yanar.
- Anahtar ile dijital giriş uygulamasının program kodları yüklendikten sonra anahtar sürgüsü sola çekildiğinde 4. ve 1. LED, sağa çekildiğinde 5. ve 2. LED yanar.



Görsel 2.29: Anahtar ile dijital giriş uygulama devresi

Uygulamada 1. ve 2. LED'lerin durumu mikrodenetleyici kartın giriş portlarına anahtar vasıtasıyla gelen dijital veriye göre değişir. Anahtar ile dijital giriş uygulamasının program kodları incelendiğinde anahtarın durumu `void loop()` fonksiyonu içindeki `if(digitalRead(4)==1` ve `if(digitalRead(5)==1)` komutlarıyla kontrol edilmektedir.



Görsel 2.30: Anahtar ile dijital giriş uygulamasının elektriksel devresi

**2. Adım:** Anahtar ile dijital giriş uygulamasının program kodları aşağıdaki gibidir.

```
void setup ()
{
  pinMode (2, OUTPUT) ; pinMode (3, OUTPUT) ;
  pinMode (4, INPUT) ;   pinMode (5, INPUT) ;
}

void loop ()
{
  if (digitalRead (4)==1)
    { digitalWrite (2, 1) ;
    }
  else digitalWrite (2, 0) ;

  if (digitalRead (5)==1)
    { digitalWrite (3, 1) ;
    }
  else digitalWrite (3, 0) ;
}
```



Sıra Sizde

Görsel 2.29 veya Görsel 2.30'da devresi verilen anahtar ile dijital giriş uygulamasını breadboard üzerine kurunuz. Anahtar ile dijital giriş uygulamasının program kodlarını mikrodenetleyici IDE programında yazınız. Tüm işlemler bittikten sonra programı mikrodenetleyici karta yükleyip, devrenin çalışmasını gözlemleyiniz.

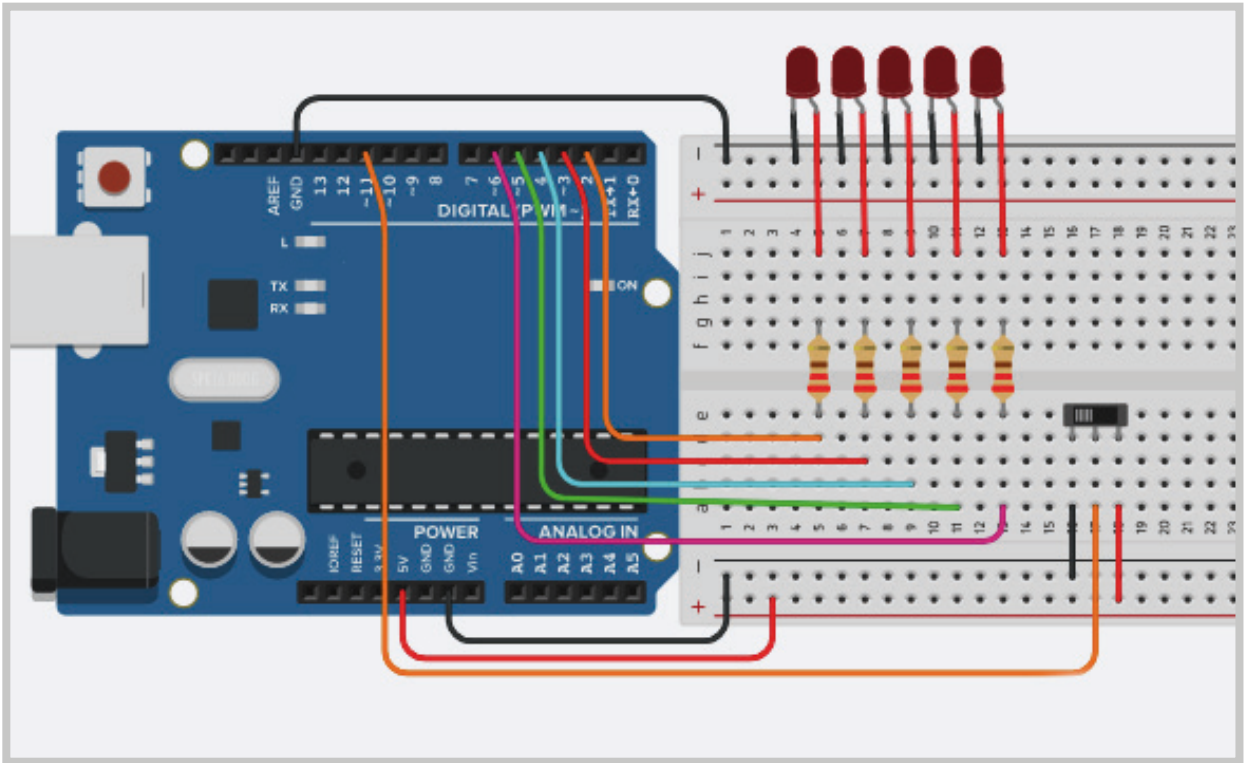
### Değerlendirme

Çalışmalarınız öğrenme biriminin sonunda yer alan Kontrol Listesi kullanılarak değerlendirilecektir. Çalışmalarınızı yaparken değerlendirme ölçütlerini dikkate alınız.

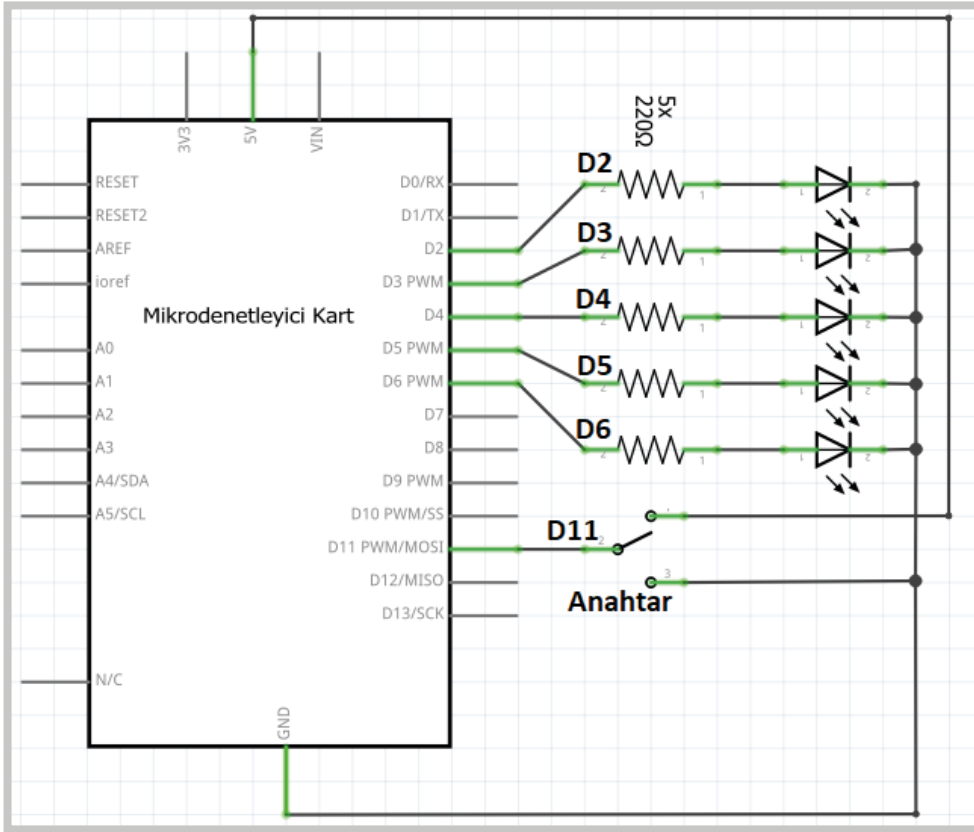
### 10. Uygulama

Bu uygulamada LED'lerin yanma sırasının anahtar vasıtasıyla değiştirilmesi amaçlanmaktadır.

**1. Adım:** Anahtar kullanımıyla ilgili uygulama Görsel 2.31 ve Görsel 2.32'de görülmektedir. Uygulamanın devresi incelendiğinde anahtarın ortak ucu mikrodenetleyici kartın D11 portuna bağlanmış, anahtarın sağda ve solda kalan uçları ise GND ve 5V pinlerine bağlanmıştır. Anahtar sürgüsü sola çekildiğinde LED'ler sola doğru yanarak gitmekte, anahtar sürgüsü sağa çekildiğinde ise LED'ler sağa doğru yanarak gitmektedir.



Görsel 2.31: Anahtar ile farklı LED animasyonu seçme uygulama devresi



Görsel 2.32: Anahtar ile farklı LED animasyonu seçme uygulamasının elektriksel devresi

**2. Adım:** Anahtar ile farklı LED animasyonu seçme uygulamasının program kodları aşağıda verilmiştir.

```

int i;
void setup() {
  for(i=2;i<=6;i++) pinMode(i, OUTPUT);
  pinMode(11, INPUT);
}
void loop() {
  //Anahtar sürgüsü sola çekildiğinde
  if (digitalRead(11)==1) {
    for(i=2;i<=6;i++) { digitalWrite(i, 1); delay(1000); }
    for(i=2;i<=6;i++) { digitalWrite(i, 0); delay(1000); }
  } //if için
  //Anahtar sürgüsü sağa çekildiğinde
  if (digitalRead(11)==0) {
    for(i=6;i>=2;i--) { digitalWrite(i, 1); delay(1000); }
    for(i=6;i>=2;i--) { digitalWrite(i, 0); delay(1000); }
  } //if için
} //loop için

```



### Sıra Sizde

Görsel 2.31 veya Görsel 2.32'de devresi verilen anahtar ile dijital giriş uygulaması devre elemanlarını breadboarda yerleştiriniz. Anahtar ile farklı LED animasyonu seçme uygulamasının program kodlarını mikrodenetleyici IDE programında yazarak programı mikrodenetleyici karta yükleyip, devrenin çalışmasını gözlemleyiniz.



### Değerlendirme

Çalışmalarınız öğrenme biriminin sonunda yer alan Kontrol Listesi kullanılarak değerlendirilecektir. Çalışmalarınızı yaparken değerlendirme ölçütlerini dikkate alınız.



<http://kitap.eba.gov.tr/KodSor.php?KOD=20783>

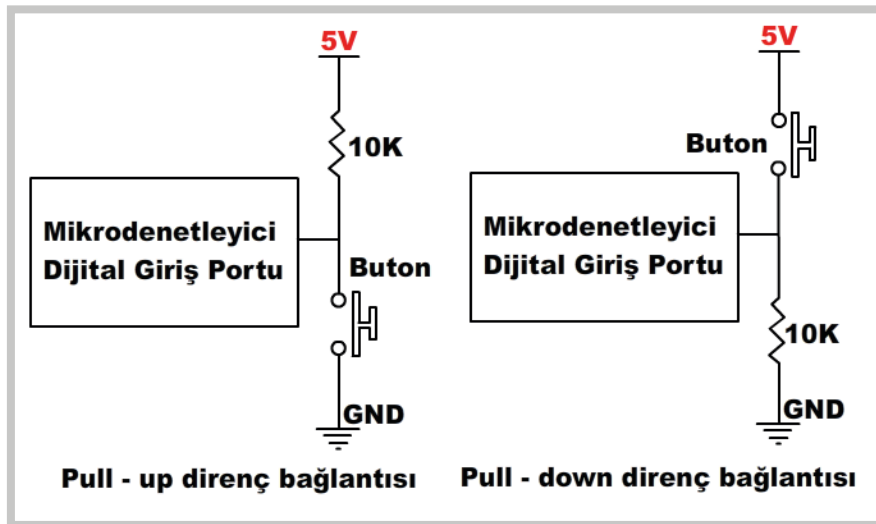
### 2.4.2. Buton Kullanımı

Butonlar aynı anahtarlar gibi elektrik ve elektronik devrelerde iki noktayı birbirine bağlayarak iletimi sağlar. Butonların anahtarlardan farkı, üzerine basıldığında veya basılı tutulduğunda iletimi sağlamasıdır. Görsel 2.33'te elektronik devrelerde kullanılan buton çeşitleri görülmektedir.



Görsel 2.33: Elektronik devrelerde kullanılan buton çeşitleri

Butonlar kullanılırken üzerine basılmadığında açık devre özelliği göstereceğinden mikrodenetleyici giriş portu boşa kalır. Mikrodenetleyici giriş portunun boşa kalması, mikrodenetleyicinin kararsız çalışmasına sebep olur. Bunu önlemek için Görsel 2.34'te görüldüğü gibi **pull up** veya **pull down** dirençleri kullanılır.



Görsel 2.34: Pull-Up direnç ve Pull-Down direnç kullanımı



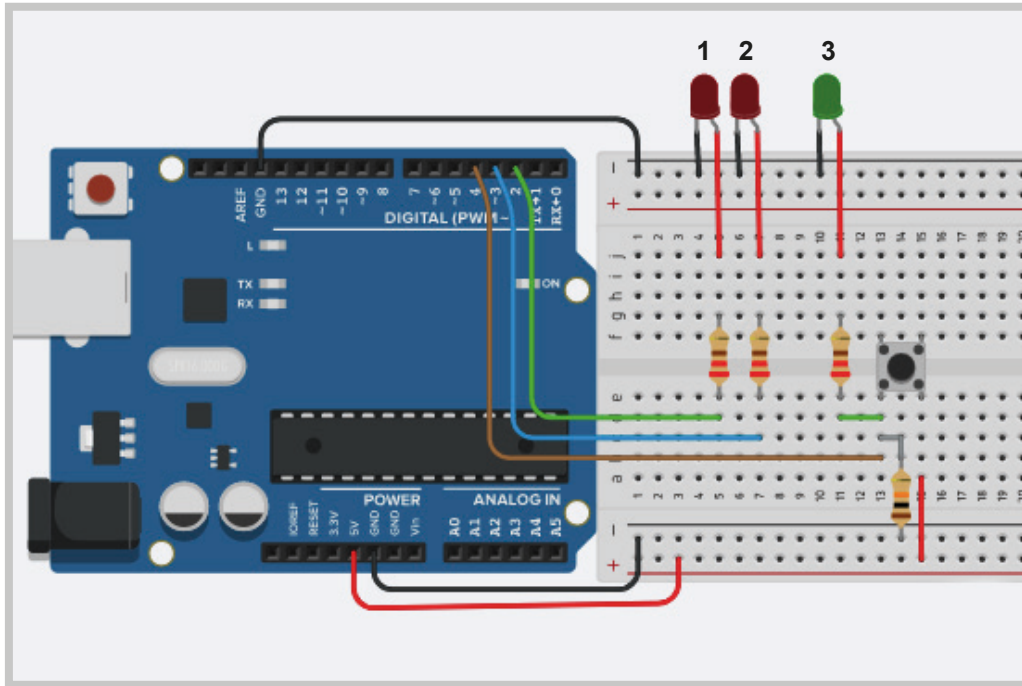
### 11. Uygulama

Buton kullanımıyla ilgili uygulama Görsel 2.35 ve Görsel 2.36'da görülmektedir. Bu uygulamada LED'lerin yanma durumu buton ile değişmektedir.

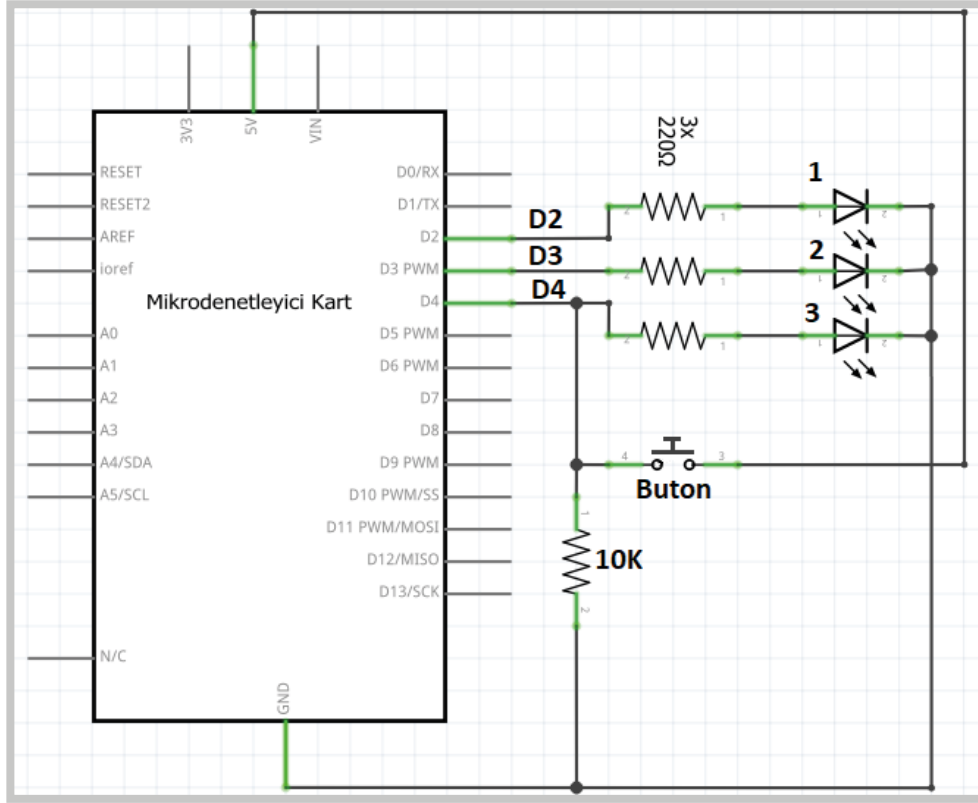
#### Gerekli Devre Elemanları

- 1 adet mikrodenetleyici kart
- 1 adet breadboard
- 1 adet anahtar
- 3 adet LED
- 3 adet 220Ω (kırmızı, kırmızı, kahverengi, altın rengi) direnç
- 1 adet 10KΩ (kahverengi, siyah, sarı, altın rengi) direnç
- Bağlantı kabloları

**1. Adım:** Görsel 2.35'te devreye 3. bir LED eklenmiştir. Eklenen 3. LED'in görevi, butona basılıp basılmadığını yanıp sönerek göstermektir. Devredeki 1. ve 2. LED'ler mikrodenetleyici kartın D2 ve D3 portlarına bağlanmıştır. Buton ile dijital giriş uygulamasının program kodları incelendiğinde, butona basılması durumunda 1. LED yanmakta, aksi durumda 2. LED yanmaktadır.



Görsel 2.35: Buton ile dijital giriş uygulama devresi



Görsel 2.36: Buton ile dijital giriş uygulamasının elektriksel devresi

**2. Adım:** Buton ile dijital giriş uygulamasının program kodları aşağıda gösterilmektedir.

```
void setup ()
{
  pinMode (2, OUTPUT) ; //2. portu çıkış yap.
  pinMode (3, OUTPUT) ; //3. portu çıkış yap.
  pinMode (4, INPUT) ; //4. portu giriş yap.
}
void loop ()
{
  if (digitalRead (4) == 1) // Eğer butona basıldı ise
  {
    digitalWrite (2, 1) ; // 2. porta bağlı LED' i yak.
    digitalWrite (3, 0) ; // 3. porta bağlı LED' i söndür.
  }
  else { digitalWrite (2, 0) ; // 2. porta bağlı LED' i söndür.
        digitalWrite (3, 1) ; // 3. porta bağlı LED' i yak.
  }
}
```

### Sıra Sizde

Görsel 2.35 veya Görsel 2.36'da devresi verilen buton ile dijital giriş uygulamasını breadboard üzerine kurunuz. Buton ile dijital giriş uygulamasının program kodlarını mikrodenetleyici IDE programında yazınız. Tüm işlemler bittikten sonra programı mikrodenetleyici karta yükleyip, devrenin çalışmasını gözlemleyiniz.

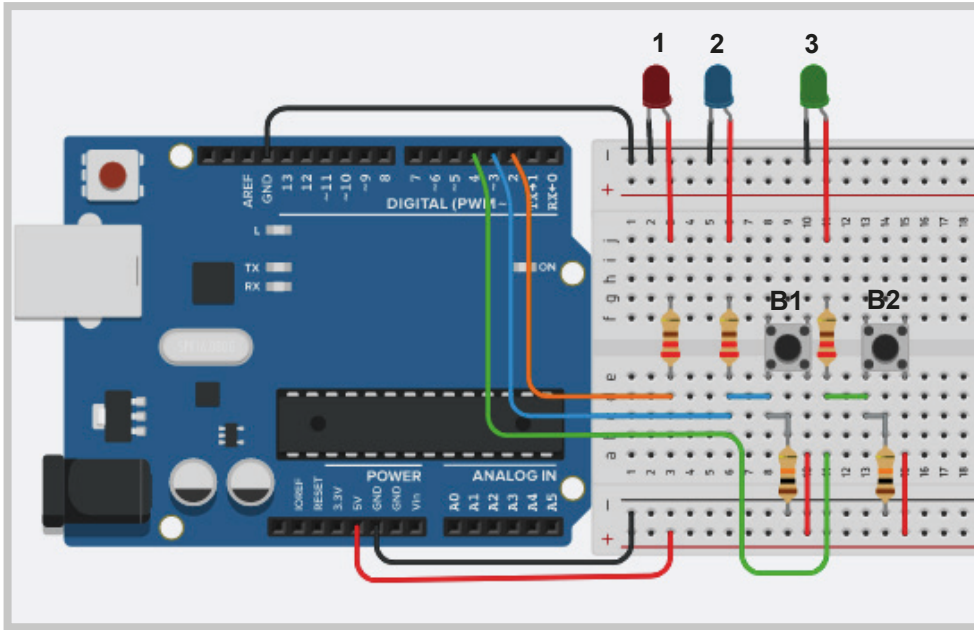
### Değerlendirme

Çalışmalarınız öğrenme biriminin sonunda yer alan Kontrol Listesi kullanılarak değerlendirilecektir. Çalışmalarınızı yaparken değerlendirme ölçütlerini dikkate alınız.

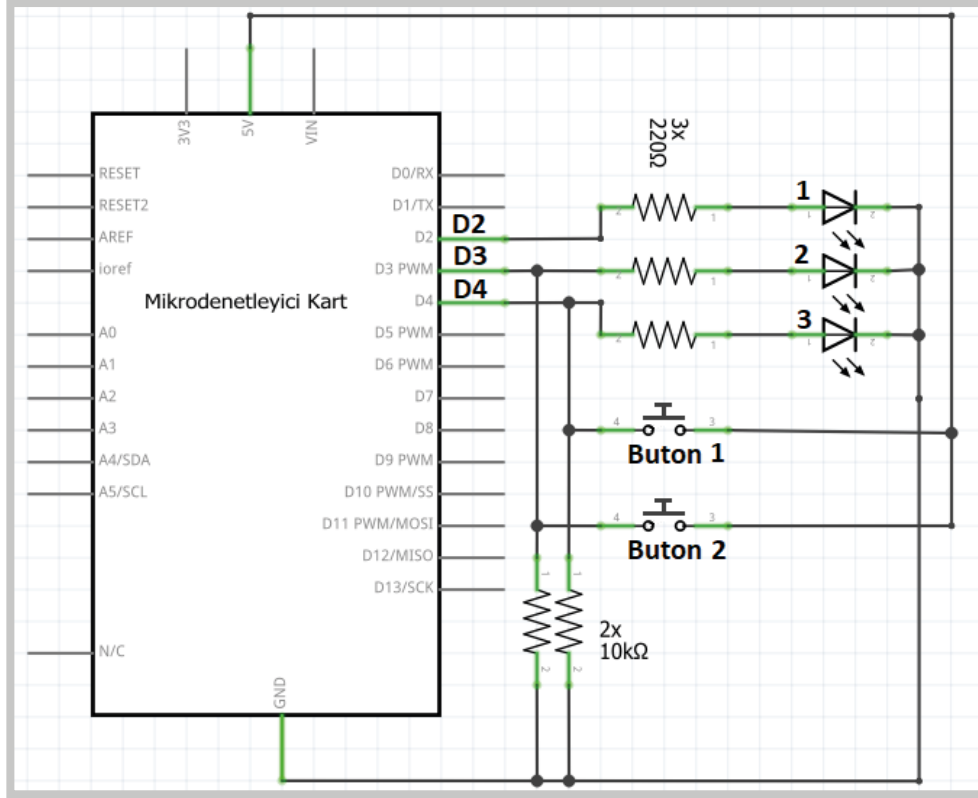
### 12. Uygulama

Buton kullanımıyla ilgili ikinci uygulama Görsel 2.37 ve Görsel 2.38'de görülmektedir. Bu uygulamada LED'lerin yanma durumu iki farklı buton ile değişmektedir. Uygulamanın devresi incelendiğinde 2. LED, B1 buton girişine; 3. LED ise B2 buton girişine bağlanmıştır. Dolayısıyla 2. LED sadece B1 butonuna basılıp basılmadığını, 3. LED ise B2 butonuna basılıp basılmadığını gösterir.

**1. Adım:** Devredeki 1. LED, mikrodenetleyici kartın D2 portuna bağlanmıştır. İki buton ile dijital giriş uygulamasının program kodları incelendiğinde, B1 butonuna basılması durumunda, 1. LED'i yandı, B2 butonuna basılması durumunda 1. LED'in söndüğü gözlemlenmiştir.



Görsel 2.37: İki buton ile dijital giriş uygulaması devresi



Görsel 2.38: İki buton ile dijital giriş uygulamasının elektriksel devresi

**2. Adım:** İki buton ile dijital giriş uygulamasının program kodları aşağıda verilmiştir.

```
void setup()  
{  
  pinMode(2, OUTPUT); //2. portu çıkış yap  
  pinMode(3, INPUT); //3. portu giriş yap  
  pinMode(4, INPUT); //4. portu giriş yap  
}  
void loop()  
{  
  if (digitalRead(3)==1) // Eğer 1. butona basıldı ise  
    digitalWrite(2, 1); // 2. porta bağlı LED' i yak  
  if (digitalRead(4)==1) // Eğer 2. butona basıldı ise  
    digitalWrite(2, 0); // 2. porta bağlı LED' i söndür  
}
```



### Sıra Sizde

Görsel 2.37 veya Görsel 2.38’de devresi verilen iki buton ile dijital giriş uygulamasını bread-board üzerine kurunuz. İki buton ile dijital giriş uygulamasının program kodlarını mikrodenetleyici IDE programında yazınız. Tüm işlemler bittikten sonra programı mikrodenetleyici karta yükleyip, devrenin çalışmasını gözlemleyiniz.



<http://kitap.eba.gov.tr/KodSor.php?KOD=20784>



### Değerlendirme

Çalışmalarınız öğrenme biriminin sonunda yer alan Kontrol Listesi kullanılarak değerlendirilecektir. Çalışmalarınızı yaparken değerlendirme ölçütlerini dikkate alınız.

## 2.5. MİKRODENETLEYİCİ KART İLE SERİ PORT UYGULAMALARI

Seri port ekranı mikrodenetleyici IDE programında yapılan işlemleri görüntülemeye yarayan bir penceredir. Seri port ekranı, mikrodenetleyici kart ile çeşitli sensör ve diğer elektronik bileşenlerin birbirleri arasındaki veri alışverişini izler.




### 13. Uygulama

Seri port ekranında “Robotik ve Kodlama dersini çok seviyorum” mesajını yazdıran programı aşağıdaki adımları takip ederek gerçekleştiriniz.

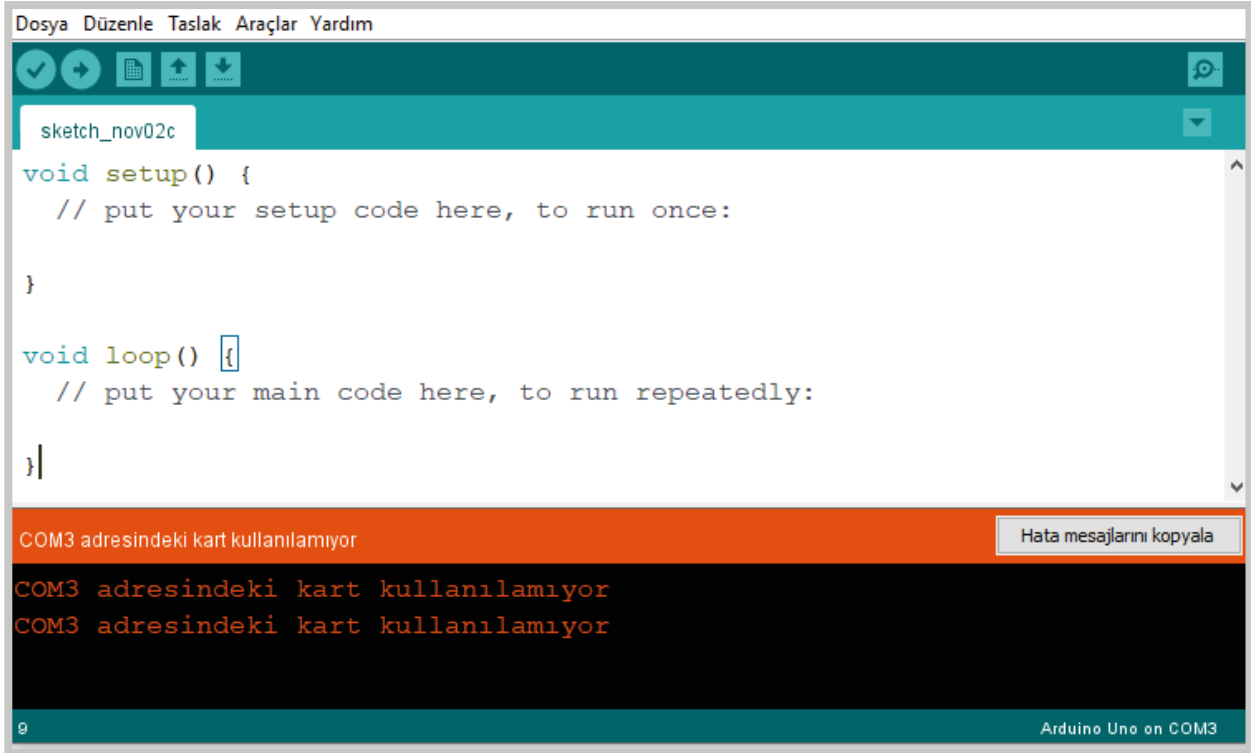
#### Gerekli Devre Elemanları

- 1 adet mikrodenetleyici kart
- Bağlantı kabloları
- Mikrodenetleyici IDE programının hatasız yüklü olması

**1. Adım:** Seri portu başlatabilmek için aşağıdaki ayarlar yapılmalıdır.

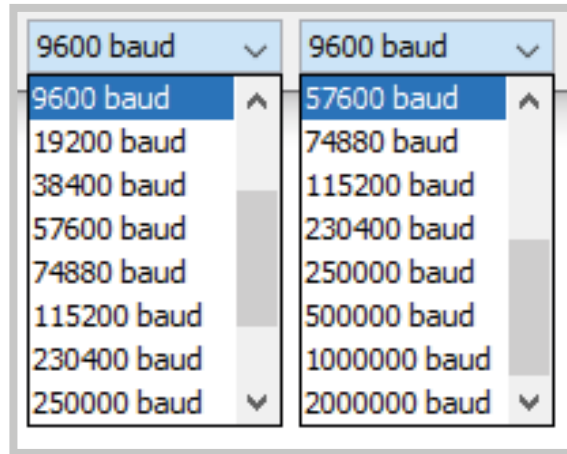
Seri port ekranını açmak için **Ctrl+Shift+M** tuşlarına ya da sağ üst köşede bulunan  simgesine tıklanır. Uygulamalar için yazılan programlara da bazı seri port komutları eklenir. Seri port yazılımsal olarak bir nesnedir ve birçok fonksiyona sahiptir.

Programda seri portu kullanmak için mikrodenetleyici kartın, mikrodenetleyici IDE programıyla bağlantısı sağlanmalıdır. Aksi hâlde program derlenecek ama Görsel 2.39’daki gibi seri port ile bağlantı kurulamadığına dair bir hata alınacaktır.



Görsel 2.39: Seri port bağlantı hatası

- Mikrodenetleyici kart bağlantısından sonra seri haberleşmeyi başlatabilmek için `void setup()` fonksiyonu içinde “`Serial.begin(speed);`” komutu kullanılmalıdır. `Begin` fonksiyonu için Görsel 2.40’taki hız parametrelerinden biri verilir.



Görsel 2.40: Begin fonksiyonu için hız parametreleri

- Seri port ekranında çıktıları doğru alabilmek için mikrodenetleyiciye yüklenen program ile bilgisayardaki seri port ekranının aynı baud hızında olması gerekir. Hızlar farklı olursa karakterler bozuk çıkar. Baud hızı genellikle 9600 olarak kullanılır.

```
void setup()  
{  
  Serial.begin(9600); // veri transfer hızını 9600 olarak ayarla  
}
```

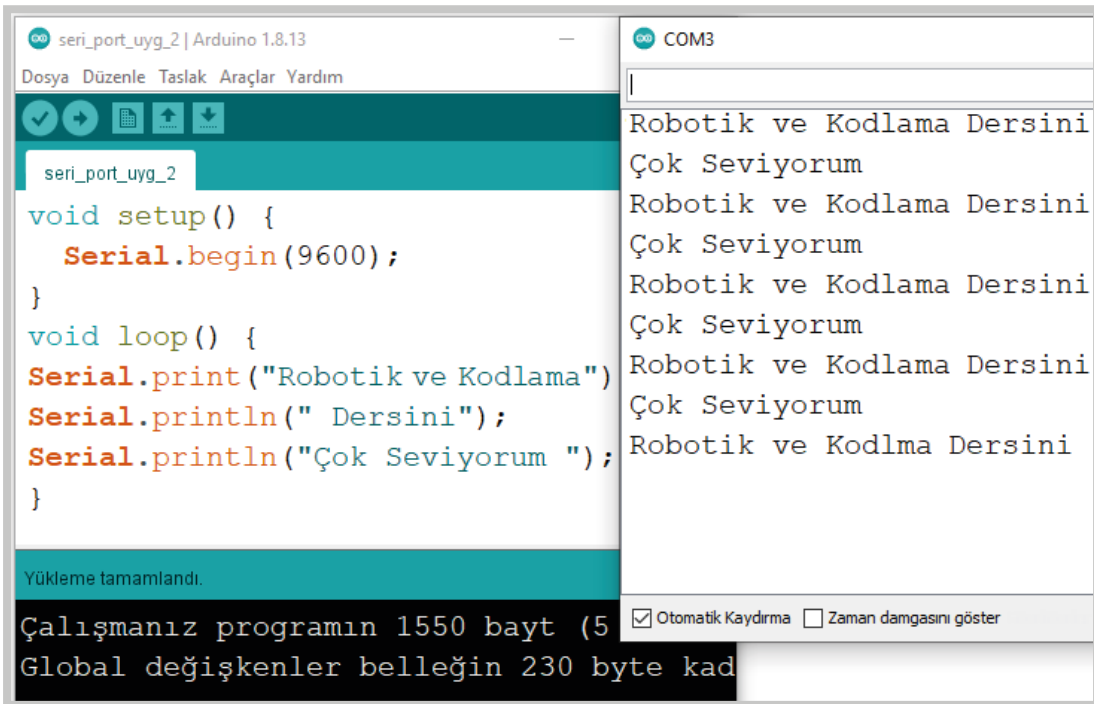
- Seri porttan gönderilen verileri kullanıcının okuyabileceği biçimde ekrana yazdırmak için

`Serial.print("Metinsel ifade");` veya `Serial.println("Metinsel ifade");` komutları kullanılır.

`Serial.print("Metinsel ifade");` komutu yazma işlemi bittikten sonra imleci kaldığı yerde bırakır.

`Serial.println("Metinsel ifade");` komutu ise ekrana yazdırma işlemi bittikten sonra imleci bir alt satırın başına getirir.

**2. Adım:** Bu komutlarla ilgili örnek program ve ekran çıktısı Görsel 2.41’de verilmiştir.



The screenshot shows the Arduino IDE interface. The left pane displays the code for a program named 'seri\_port\_uyg\_2'. The code is as follows:

```
void setup() {  
  Serial.begin(9600);  
}  
void loop() {  
  Serial.print("Robotik ve Kodlama");  
  Serial.println(" Dersini");  
  Serial.println("Çok Seviyorum ");  
}
```

The right pane shows the serial monitor output for COM3, displaying the following text:

```
Robotik ve Kodlama Dersini  
Çok Seviyorum  
Robotik ve Kodlama Dersini  
Çok Seviyorum  
Robotik ve Kodlama Dersini  
Çok Seviyorum  
Robotik ve Kodlama Dersini  
Çok Seviyorum  
Robotik ve Kodlama Dersini
```

At the bottom of the IDE, a status bar indicates 'Yükleme tamamlandı.' (Upload completed.) and 'Çalışmanız programın 1550 bayt (5 Global değişkenler belleğin 230 byte kad' (Your program is 1550 bytes (5 Global variables, memory 230 bytes).

Görsel 2.41: Örnek kodun ekran çıktısı



### Sıra Sizde

Görsel 2.41’de verilen program kodlarını mikrodeneleyici IDE programında yazınız. Tüm işlemler bittikten sonra programı mikrodeneleyici karta yükleyip, seri port ekranını açınız. Seri port ekranında yazılanları inceleyiniz.



### Değerlendirme

Çalışmalarınız öğrenme biriminin sonunda yer alan Kontrol Listesi kullanılarak değerlendirilecektir. Çalışmalarınızı yaparken değerlendirme ölçütlerini dikkate alınız.

### 14. Uygulama

Seri porttan gelen verilerin değişken türlerini değiştirmek için bir uygulama yapılacaktır. Seri porttan gelen verinin her karakterini ayrı ayrı okumak için `Serial.read()` komutu kullanılır. `Serial.read()` komutu integer değerler döndürür. Bu integer değerlerin türleri değiştirilerek (`string`, `char`, `float`.) istenilen veri türünde gösterilebilir. Bu komutla ilgili aşağıda örnek programın kodları ve Görsel 2.42'de ekran çıktısı görülmektedir.

**1. Adım:** `Serial.read()` komutuyla ilgili örneğin program kodları aşağıdaki gibidir. Uygulamanın ekran çıktısı Görsel 2.42'dedir.

```
int girilen_deger=0;
// Alınacak değeri saklamak için kullanılan değişken
void setup() {
  Serial.begin(9600);
}
void loop() {
if( Serial.available() ) {
  // Seri portta girdi olup olmadığını kontrol eder.
  // Eğer girdi varsa okur ve yazdırır.
  girilen_deger= Serial.read(); //Seri porttan değer okur.
  Serial.print("Girilen Değer: ");
  Serial.print(girilen_deger);
  //int olarak alınan değeri yazar.
  Serial.print(" - Char Olarak Girilen Değer: ");
  Serial.println((char)girilen_deger);
  //Char tipine dönüştürülen değeri yaz.
}
}
```



Görsel 2.42: `Serial.read()` komutuyla ilgili örneğin ekran çıktısı





## Sıra Sizde

`Serial.read()` komutuyla ilgili uygulamanın kodlarını yazarak programı mikrodenetleyici karta yükleyip, seri port ekranını açınız. Seri port ekranında yazılanları inceleyiniz.



## Değerlendirme

Çalışmalarınız öğrenme biriminin sonunda yer alan Kontrol Listesi kullanılarak değerlendirilecektir. Çalışmalarınızı yaparken değerlendirme ölçütlerini dikkate alınız.

## 2.6. MİKRODENETLEYİCİ KART İLE DİZİ UYGULAMALARI

Program yazımında değişkenler yalnızca bir değer tutar. Bir değişkenin içinde sıralı bir şekilde birden fazla değer tutulması için **dizi** adı verilen değişken türleri kullanılır. Diziler bir lokomotifin vagonlarına benzetilebilir. Lokomotifin her vagonunda bir değişken değeri tutulur. Örnek olarak Tablo 2.1'de görüldüğü gibi sıcaklık, nem, ışık, renk ve mesafe değerlerinden oluşan 5 elemanlı bir sensör dizisi 5 adet farklı değişkeni hafızasında tutabilir.

Tablo 2.1: 5 Elemanlı Sensörler Dizisi ve Dizi Elemanlarının Programsal Gösterimi

5 Elemanlı Sensörler Dizisi		
İndis No.	Atanan Değer	Programsal Gösterimi
0	Sıcaklık	<code>Sensor[0]="Sıcaklık";</code>
1	Nem	<code>Sensor[1]="Nem";</code>
2	Işık	<code>Sensor[2]="Işık";</code>
3	Renk	<code>Sensor[3]="Renk";</code>
4	Mesafe	<code>Sensor[4]="Mesafe";</code>

Mikrodenetleyici IDE programında dizi tanımlamak için önce dizinin değişken tipini (int, string, char, long, boolean, byte vb.) belirlemek gerekir. Sonra dizi ismi, köşeli parantez içinde dizinin eleman sayısı verilir.

**Kullanımı:** `Veri_tipi Dizi_Adı[indis_sayısı];`

**Örnek:** `int sensor[5]; // sensor adında int tipinde 5 elemanlı bir dizi oluşturuldu.`

Yukarıdaki örnek kodda "sensor" adında 5 elemanlı bir dizi oluşturulmuştur. "int" komutu ile dizi içinde tutacağı veri türü tam sayı olarak belirlenmiştir. Dizi isminden sonra gelen köşeli parantezler içinde dizinin alabileceği eleman sayısı girilmektedir. Buna göre "sensor" dizisi 5 elemana sahip olur.

## 2. Öğrenme Birimi: Mikrodenetleyici Kart Programlama

**Örnek:** Dizi oluşturulurken değer atanacaksa dizi eleman sayısının girilmesine gerek yoktur.

```
// değerleri önceden verilen int tipinde 5 elemanlı bir dizi oluşturuldu.  
int voltaj[]={0,3,5,9,12};
```

Yukarıda oluşturulan “voltaj” isimli 5 elemanlı dizide küme parantezi içinde eleman sayıları girilebilir. Bu şekilde dizi elemanları tanımlanırken girilirse köşeli parantez içinde dizinin eleman sayısının girilmesine gerek yoktur.

**Örnek:** Dizinin elemanlarına erişmek için dizi adı yazılıp köşeli parantez içinde erişilmek istenen elemanın indis numarasının girilmesi gerekir. Dizilerdeki indis numaraları sıfırdan başlar. Dizi elemanları tanımlandıktan sonra da değiştirilebilir.

```
voltaj[0]=3; // "voltaj" dizisinin 0. elemanına 3 değeri atanır.  
voltaj[1]=4; // "voltaj" dizisinin 1. elemanına 4 değeri atanır.  
voltaj[2]=6; // "voltaj" dizisinin 2. elemanına 6 değeri atanır.  
voltaj[3]=10; // "voltaj" dizisinin 3. elemanına 10 değeri atanır.  
voltaj[4]=15; // "voltaj" dizisinin 4. elemanına 15 değeri atanır.
```

**Örnek:** İstenilirse dizi elemanları farklı değişkenlere de atanabilir.

```
int yeni=voltaj[2];  
  
// "voltaj" dizisinin 2. elemanı "yeni" isimli değişkene aktarıldı.
```

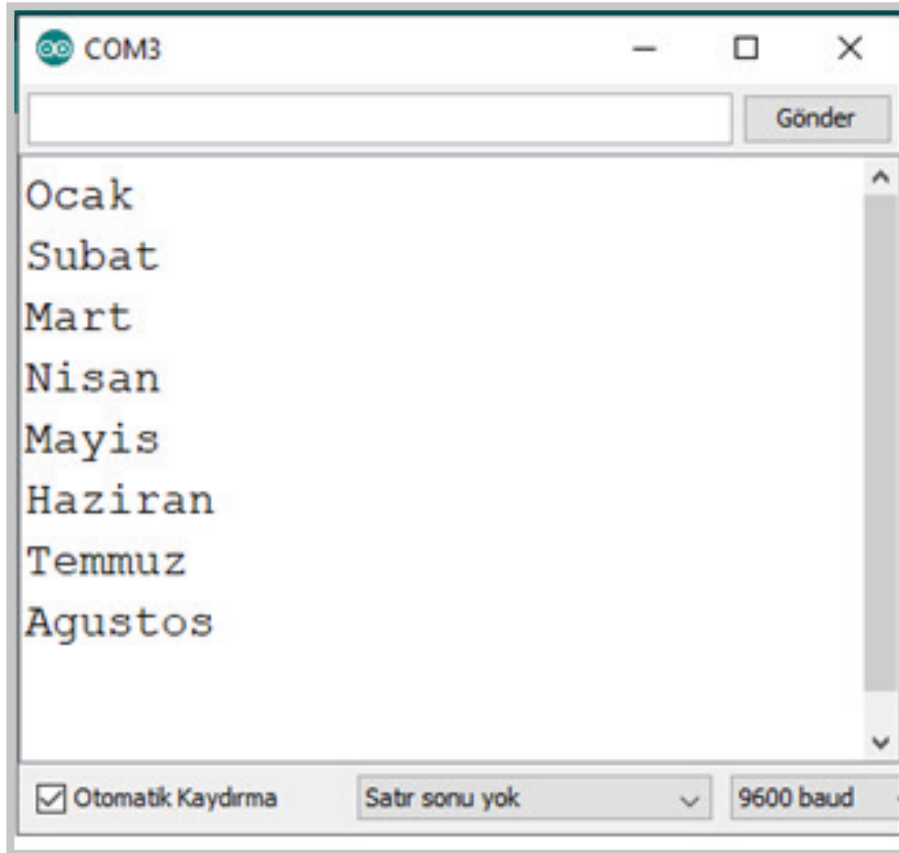
**Örnek:** Bir dizinin tüm elemanlarını listelemek için döngü komutları kullanılır. Aşağıdaki örnekte kullanılan `for (i=0;i<=5;i++)` komutu “portlar” isimindeki 6 elemanlı bir dizinin tüm elemanlarını okutmak için 0'dan 5'e kadar bir döngü oluşturur. `Serial.println` komutu ile her döngüde “i” değişkeni, değerine karşılık gelen dizi elemanını seri porta yazdırır.

```
int i;  
  
int portlar[]={3,5,6,9,10,11}; // "portlar" isminde bir dizi tanımlandı.  
  
for (i=0;i<=5;i++) Serial.println(portlar[i]);  
  
// "portlar" isimindeki dizinin tüm elemanlarını ekrana yazdırır.
```

## 15. Uygulama

Aşağıda verilen string dizi tanımlama programının kodları, mikrodenetleyici IDE programında gerçekleştirildiğinde Görsel 2.43'teki gibi çıktısı elde edilmektedir.

```
void setup() {
  Serial.begin(9600);
}
void loop() {
  String aylar[8]={"Ocak","Subat","Mart","Nisan","Mayis"};
  //Dizinin ilk 5 elemanı dizi tanımlaması yapılırken belirlendi.
  aylar[5]="Haziran"; // Dizinin 5. elemanı "Haziran" olarak atandı.
  aylar[6]="Temmuz"; // Dizinin 6. elemanı "Temmuz" olarak atandı.
  aylar[7]="Agustos"; // Dizinin 7. elemanı "Agustos" olarak atandı.
  int i;
  for (i=0;i<=7;i++)
  // Bloktaki kodları 0'dan 7'ye kadar 8 kez çalışacak döngü oluşur.
  Serial.println(aylar[i]);
  // Dizi elemanlarını i sayısı her arttığında bir defa yazdırır.
  delay(1000);
}
```



Görsel 2.43: String dizi tanımlama programının ekran çıktısı

## 2. Öğrenme Birimi: Mikrodenetleyici Kart Programlama



### Sıra Sizde

String dizi tanımlama programının kodlarını mikrodenetleyici IDE programında yazınız. Tüm işlemler bittikten sonra programı mikrodenetleyici karta yükleyip, seri port ekranını açınız. Seri port ekranında yazılanları inceleyiniz.



### Değerlendirme

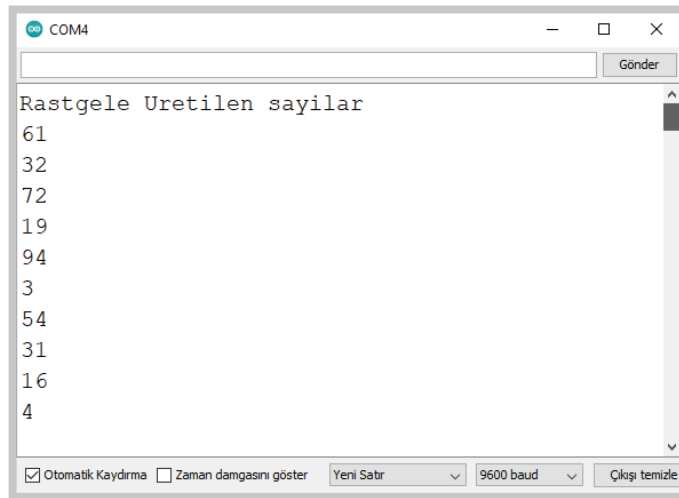
Çalışmalarınız öğrenme biriminin sonunda yer alan Kontrol Listesi kullanılarak değerlendirilecektir. Çalışmalarınızı yaparken değerlendirme ölçütlerini dikkate alınız.



### 16. Uygulama

Aşağıda verilen program kodları, on adet rasgele sayı üretecek ve diziyeye aktaracak şekilde, IDE programında oluşturulmuştur.

```
void setup() {  
    Serial.begin(9600);  
}  
void loop() {  
    int i;  
    int say[10];  
    for (i=0;i<=9;i++)  
        say[i] = random(0,100); // rastgele 10 adet sayı üret.  
    Serial.println("Rastgele Üretilen sayılar");  
    for (i=0;i<=9;i++) Serial.println(say[i]); // Üretilen sayıları yazdır.  
    delay(1000);  
}
```



Görsel 2.44: Rastgele 10 adet sayı üreten ve diziyeye aktaran programın ekran çıktısı



### Sıra Sizde

Rastgele 10 adet sayı üreten ve diziye aktaran program kodlarını mikrodenetleyici IDE programında yazınız. Tüm işlemler bittikten sonra programı mikrodenetleyici karta yükleyip, seri port ekranını açınız. Seri port ekranında yazılanları inceleyiniz.



### Değerlendirme

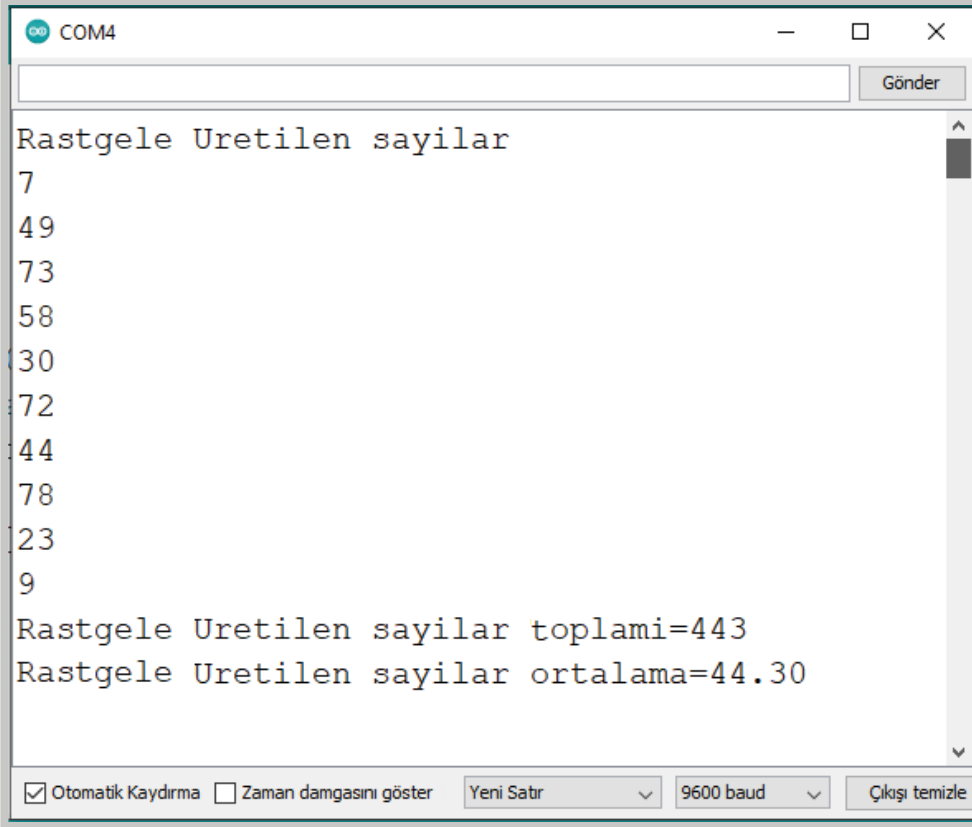
Çalışmalarınız öğrenme biriminin sonunda yer alan Kontrol Listesi kullanılarak değerlendirilecektir. Çalışmalarınızı yaparken değerlendirme kriterlerini dikkate alınız.



### 17. Uygulama

Rastgele 10 adet sayının toplamını ve ortalamasını bulan program kodları aşağıda verilmiş olup, programın çıktısı Görsel 2.45'te gösterilmiştir.

```
void setup() {  
    Serial.begin(9600);  
}  
void loop() {  
    int i;  
    int say[10];  
    for (i=0;i<=9;i++)  
        say[i] = random(0,100); // Rastgele 10 adet sayı üretir.  
    Serial.println("Rastgele Üretilen sayılar");  
    for (i=0;i<=9;i++)  
        Serial.println(say[i]); // Üretilen sayıları yazdır.  
    int toplam=0;  
    for (i=0;i<=9;i++)  
        toplam=toplam+say[i]; // Üretilen sayıları toplar.  
    Serial.print("Rastgele Üretilen sayıların Toplamı=");  
    Serial.println(toplam); // Üretilen sayıların toplamını yazdır.  
    float ort=toplam/10.00;  
    // Üretilen sayıların ortalamasını alır.  
    Serial.print("Rastgele Üretilen sayıların ortalaması=");  
    Serial.println(ort); // Üretilen sayıların ortalamasını yazdır.  
    delay(1000);  
}
```



```
COM4
Rastgele Üretilen sayılar
7
49
73
58
30
72
44
78
23
9
Rastgele Üretilen sayılar toplamı=443
Rastgele Üretilen sayılar ortalama=44.30
 Otomatik Kaydırma  Zaman damgasını göster Yeni Satır 9600 baud Çıkışı temizle
```

Görsel 2.45: Rastgele 10 adet sayının toplamını ve ortalamasını bulan programın çıktısı



### Sıra Sizde

Rastgele 10 adet sayının toplamını ve ortalamasını bulan program kodlarını mikrodenetleyici IDE programında yazınız. Tüm işlemler bittikten sonra programı mikrodenetleyici karta yükleyip, seri port ekranını açınız. Seri port ekranında yazılanları inceleyiniz.



### Değerlendirme

Çalışmalarınız öğrenme biriminin sonunda yer alan Kontrol Listesi kullanılarak değerlendirilecektir. Çalışmalarınızı yaparken değerlendirme ölçütlerini dikkate alınız.

Dizi tanımlamada diğer bir konu da çok boyutlu diziler yani matrislerdir. Çok boyutlu diziler, dizi içinde dizi oluşturmaya benzer. Örneğin 2 boyutlu bir dizi için aşağıdaki gibi bir tanımlama yapılır.

```
Veri_tipi Dizi_Adı[indis_sayısı1][indis_sayısı2] = { , , , };
```

Dizi eğer iki boyutlu ise köşeli parantez içinde 2 indisi olur ve dizinin eleman sayısı 2 indisin çarpımı şeklindedir. Dizi elemanları yazılırken ilk tanımlamada küme parantezi ile yazılabilir.

**Örnek:** Aşağıdaki örnekte 2x2'lik 4 elemanlı bir dizi tanımlanması ve değer ataması görülmektedir.

```
int dizi [2] [2] ;
```

```
dizi [0] [0]=1;//Diziye 1. eleman atanır.
```

```
dizi [0] [1]=2;//Diziye 2. eleman atanır.
```

```
dizi [1] [0]=3;//Diziye 3. eleman atanır.
```

```
dizi [1] [1]=4;//Diziye 4. eleman atanır.
```

**Örnek:** Çok boyutlu diziler tanımlanırken değerler ilk başta atanabilir. Tablo 2.2 ve Tablo 2.3'te 3x5'lik 15 elemanlı bir dizi tanımlaması ve değer ataması görülmektedir.

```
int sayilar [3] [5]={ 1,2,3,4,5,  
                    2,4,6,8,10,  
                    3,6,9,12,15};
```

Tablo 2.2: 15 Elemanlı 3x5'lik Sayılar Dizisinin Programsal Gösterimi

İndis	0	1	2	3	4
0	Dizi [0] [0]	Dizi [0] [1]	Dizi [0] [2]	Dizi [0] [3]	Dizi [0] [4]
1	Dizi [1] [0]	Dizi [1] [1]	Dizi [1] [2]	Dizi [1] [3]	Dizi [1] [4]
2	Dizi [2] [0]	Dizi [2] [1]	Dizi [2] [2]	Dizi [2] [3]	Dizi [2] [4]

Tablo 2.3: 15 Elemanlı 3x5'lik Sayılar Dizisi Elemanlarının Gösterimi

İndis	0	1	2	3	4
0	1	2	3	4	5
1	2	4	6	8	10
2	3	6	9	12	15

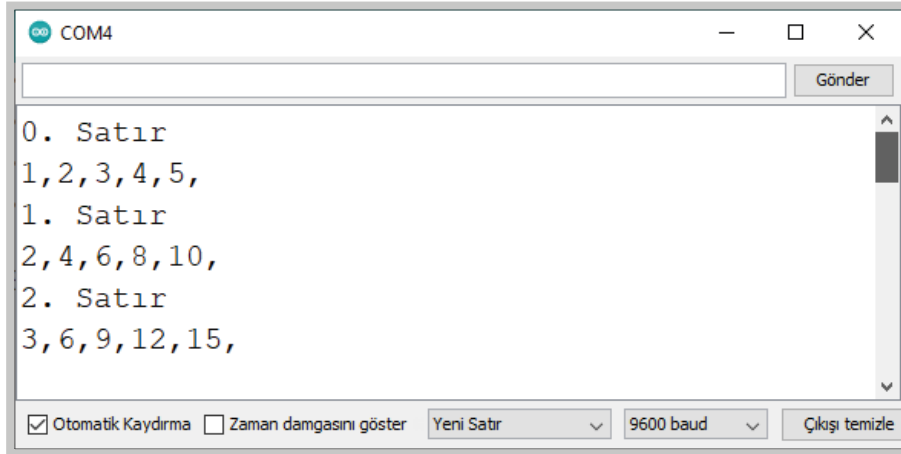
Çok boyutlu dizilerin elemanlarını listelemek için birden fazla döngü kullanılır. Sayılar dizisinin 15 elemanını listeleyebilmek için 2 adet döngü kullanılmalıdır.



### 18. Uygulama

İki boyutlu 3x5'lik sayılar dizisinin elemanlarını yazdıran program kodları aşağıdaki gibidir. İki boyutlu 3x5'lik sayılar dizisinin elemanlarını yazdıran programın ekran çıktısı ise Görsel 2.46'daki gibidir.

```
void setup() {  
    Serial.begin(9600);  
}  
void loop() {  
  
    int sayilar[3][5]={1,2,3,4,5,  
        2,4,6,8,10,  
        3,6,9,12,15};  
  
    int x,y;  
    for (x=0;x<=2;x++){//Sütunlar için kullanılan döngü  
        Serial.print(x); Serial.println(". Satır");  
        for (y=0;y<=4;y++){//Satırlar için kullanılan döngü  
            Serial.print(sayilar[x][y]); //Dizi elemanlarını yazdırır.  
            Serial.print(",");  
        }  
        Serial.println("");  
    }  
}
```



Görsel 2.46: İki boyutlu 3x5'lik sayılar dizisinin elemanlarını yazdıran programın ekran çıktısı



### Sıra Sizde

İki boyutlu 3x5'lik sayılar dizisinin elemanlarını yazdıran program kodlarını mikrodenetleyici IDE programında yazınız. Tüm işlemler bittikten sonra programı mikrodenetleyici karta yükleyip, seri port ekranını açınız. Seri port ekranında yazılanları inceleyiniz.



### Değerlendirme

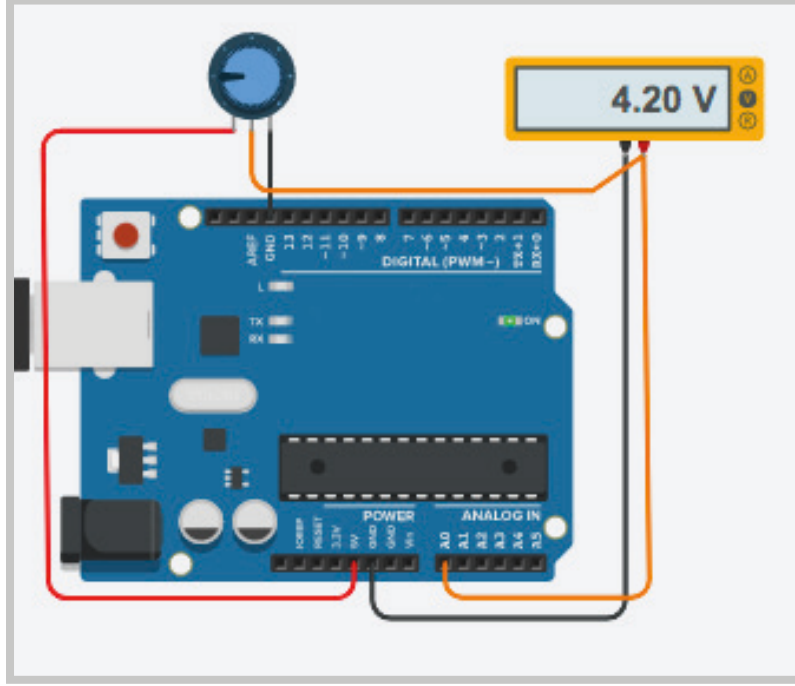
Çalışmalarınız öğrenme biriminin sonunda yer alan Kontrol Listesi kullanılarak değerlendirilecektir. Çalışmalarınızı yaparken değerlendirme ölçütlerini dikkate alınız.



## 2.7. MİKRODENETLEYİCİ KART İLE ANALOG GİRİŞ UYGULAMALARI

Mikrodenetleyici kartlarda dijital ve analog olmak üzere iki farklı giriş portu vardır. Dijital girişlerde 1 ve 0 olmak üzere iki farklı değer bulunur. Mikrodenetleyicilerde 1 değeri için 3V ve 5V gerilim değerleri vardır. Dijital giriş portu 3V değerinde de olsa 5V değerinde de olsa çıkış değeri 1 olur.

Analog girişlerde ise durum farklıdır. Analog girişler, 0 ile 5V arasında bir değer alabilir. Analog girişler mikrodenetleyicinin yapısında bulunan Analog-Dijital dönüştürücüler sayesinde girişten aldığı 0 ile 5V arasındaki voltaj değerini 0 ile 1023 arasında sayısal bir değere çevirebilmektedir.



Görsel 2.47: Potansiyometre ile analog giriş uygulamasının simülasyon devresi

### 19. Uygulama

Analog giriş ile ilgili uygulamalardan biri de Görsel 2.47’de görüldüğü gibi mikrodenetleyici kartın analog girişine elle ayarlanabilen direnç (potansiyometre) bağlanarak gerçekleştirilen uygulamadır.

Potansiyometrenin 3 adet bağlantı ucu bulunur. Potansiyometrenin orta bölümdeki bağlantı ucu ortak uçtur ve mikrodenetleyici kartın analog girişine bağlanır. Potansiyometrenin sağda ve solda bulunan bağlantı uçları 5V ve GND uçlarına bağlanır. Potansiyometrenin Görsel 2.47’deki bağlantısına **gerilim bölücü** denir.

#### Gerekli Devre Elemanları

- 1 adet mikrodenetleyici kart
- 1 adet breadboard
- 1 adet potansiyometre
- 1 adet voltmetre
- Bağlantı kabloları



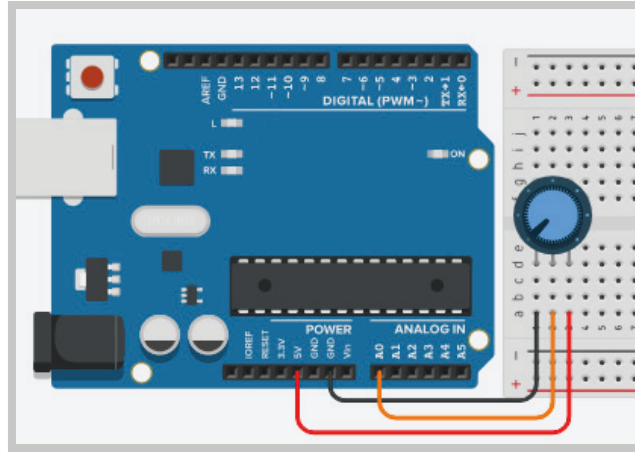
### 20. Uygulama

Potansiyometre ile analog girişin sağlanacağı bir uygulama yapılacaktır.

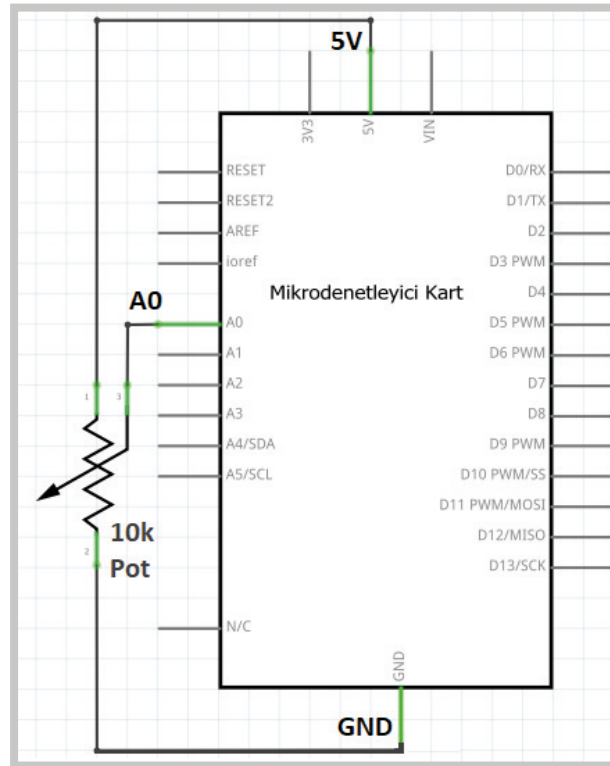
#### Gerekli Devre Elemanları

- 1 adet mikrodenetleyici kart
- 1 adet breadboard
- 1 adet potansiyometre
- Bağlantı kabloları

**1. Adım:** Görsel 2.49 ve Görsel 2.50'de görüldüğü gibi potansiyometre ile analog giriş uygulamasının fiziksel devresi oluşturulur.



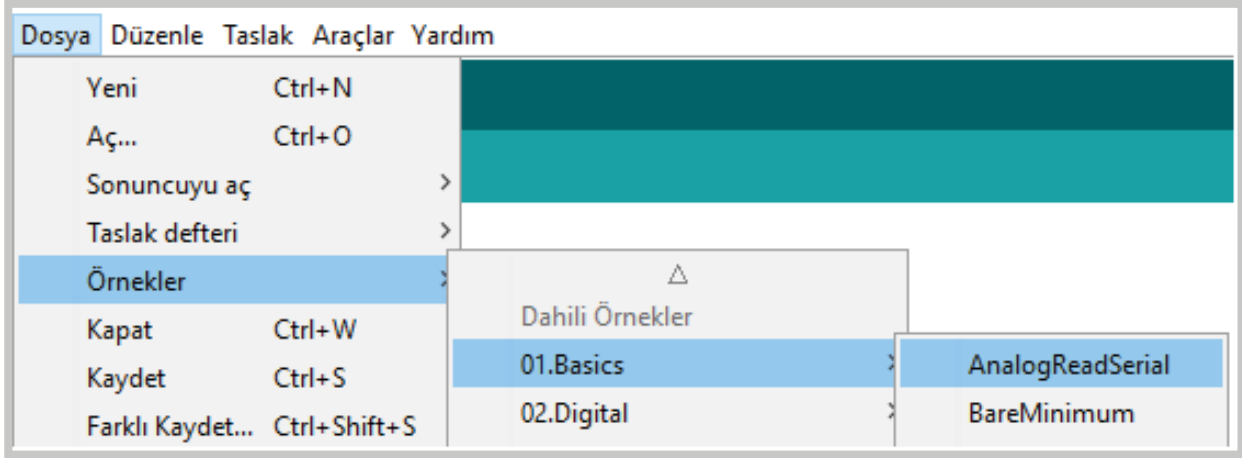
Görsel 2.49: Potansiyometre ile analog giriş uygulama devresi



Görsel 2.50: Potansiyometre ile analog giriş uygulamasının elektriksel devresi

## 2. Öğrenme Birimi: Mikrodenetleyici Kart Programlama

**2. Adım:** Görsel 2.51’de gösterildiği gibi **Dosya** menüsündeki **01.Basics** seçeneğinden **AnalogReadSerial** seçeneği işaretlenir.



Görsel 2.51: Analog giriş örnek uygulamasının açılması

**3. Adım:** Analog giriş örnek uygulamasının kodları şunlardır:

```
void setup () {  
    Serial.begin (9600) ;  
}  
void loop () {  
    int sensorValue = analogRead (A0) ;  
    Serial.println (sensorValue) ;  
    delay (1) ;  
}
```



Sıra Sizde

Görsel 2.49 veya Görsel 2.50’de devresi verilen potansiyometre ile analog giriş uygulamasını breadboard üzerine kurunuz. Görsel 2.51’deki gibi analog giriş örnek uygulamasını mikrodenetleyici IDE programında açınız. Program kodlarında bir değişiklik yapmadan mikrodenetleyici karta yükleyiniz. Tüm işlemler bittikten sonra seri port ekranını açıp potansiyometreyi çevirirken üretilen sayısal değerleri inceleyiniz. Seri port ekranını açmak için “Araçlar->Seri port ekranı”na ya da sağ üst köşedeki büyüteç işaretine tıklayabilirsiniz.

### Değerlendirme

Çalışmalarınız öğrenme biriminin sonunda yer alan Kontrol Listesi kullanılarak değerlendirilecektir. Çalışmalarınızı yaparken değerlendirme ölçütlerini dikkate alınız.

### 2.7.1. Potansiyometre ile LED'lerin Yanıp Sönme Hızının Ayarlanması

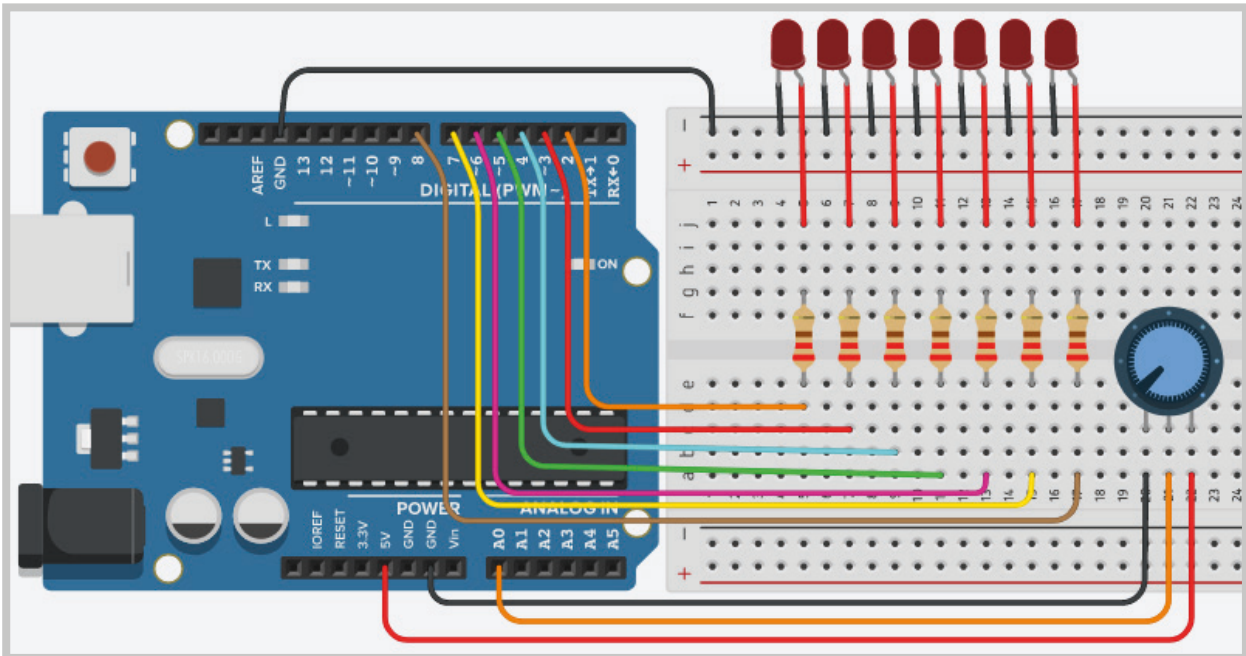
#### 21. Uygulama

Potansiyometre ile LED'lerin yanıp sönme hızının ayarlandığı bir uygulama yapılacaktır.

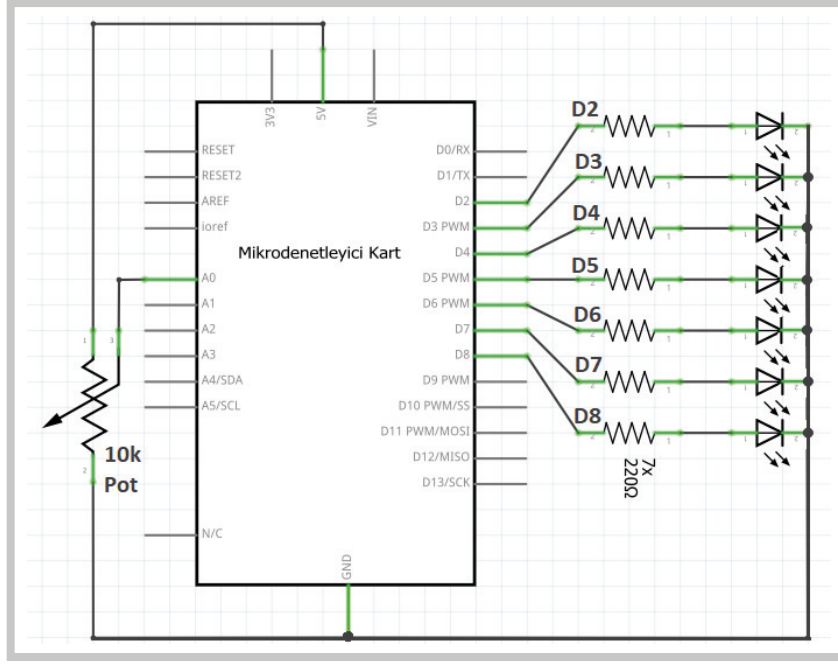
#### Gerekli Devre Elemanları

- 1 adet mikrodenetleyici kart
- 1 adet breadboard
- 1 adet potansiyometre
- 7 adet LED
- 7 adet 220Ω (kırmızı, kırmızı, kahverengi, altın rengi) direnç
- Bağlantı kabloları

**1. Adım:** Görsel 2.52 ve Görsel 2.53'te gösterildiği gibi potansiyometre ile LED'lerin yanma hızını ayarlayan devre oluşturulur.



Görsel 2.52: Potansiyometre ile LED'lerin yanma hızını ayarlayan uygulama devresi



Görsel 2.53: Potansiyometre ile LED'lerin yanma hızını ayarlayan uygulamanın elektriksel devresi

Görsel 2.52 ve Görsel 2.53'te verilen potansiyometre ile LED'lerin yanma hızını ayarlayan uygulama, for döngüsüyle sırayla yanan LED'ler uygulamasına benzemektedir. Bu uygulamadaki farklılık, LED'lerin yanma hızının potansiyometreden okunan değerle ayarlanabilmesidir.

**2. Adım:** Potansiyometre ile LED'lerin yanma hızını ayarlayan uygulamanın program kodları aşağıdaki gibidir.

```
int i; // For döngüleri için değişken tanımlar.
void setup() {
    for (i=2; i<=8; i++)
        pinMode(i, OUTPUT); // 2' den 8' e kadar portları çıkış yapar.
}
void loop() {
    int pot=analogRead(A0);
    for (i=2; i<=8; i++) {digitalWrite(i, 1); delay(pot); }
    for (i=2; i<=8; i++) {digitalWrite(i, 0); delay(pot); }
}
```

Uygulamanın program kodlarında, `pot=analogRead(A0)`; komutuyla potansiyometreden okunan değerler pot değişkenine aktarılır. For döngü bloku içindeki `delay(pot)`; komutuyla LED'lerin yanma hızı olarak milisaniye cinsinden "pot" değişkeninde tutulan değer kullanılır.



Sıra Sizde

Görsel 2.52 veya Görsel 2.53'te devresi verilen potansiyometre ile LED'lerin yanma hızını ayarlayan uygulamayı breadboard üzerine kurunuz. Bu uygulamanın program kodlarını mikrodenetleyici IDE programında yazınız. Tüm işlemler bittikten sonra programı mikrodenetleyici karta yükleyip, devrenin çalışmasını gözlemleyiniz.



http://kitap.eba.gov.tr/  
KodSor.php?KOD=20786

 Değerlendirme

Çalışmalarınız öğrenme biriminin sonunda yer alan Kontrol Listesi kullanılarak değerlendirilecektir. Çalışmalarınızı yaparken değerlendirme ölçütlerini dikkate alınız.

## 2.7.2. Potansiyometre Seviyesinin LED ile Gösterilmesi

 22. Uygulama

LED'lerin yanma hızını ayarlayan uygulama yapılacaktır.

### Gerekli Devre Elemanları

- 1 adet mikrodenetleyici kart
- 1 adet breadboard
- 1 adet potansiyometre
- 7 adet LED
- 7 adet 220Ω (kırmızı, kırmızı, kahverengi, altın rengi) direnç
- Bağlantı kabloları

**1. Adım:** Görsel 2.52 ve Görsel 2.53'te verilen potansiyometre ile LED'lerin yanma hızını ayarlayan uygulamanın devresi kullanılarak, potansiyometre seviyesini LED ile gösteren bir uygulama yapılacaktır. Devrede bir değişiklik yapılmadan program yeniden düzenlenecektir.

**2. Adım:** Bu uygulamada potansiyometreden okunan 0 ile 1023 arasındaki değerler, `map(pot,0,1023,1,7)` komutuyla yeniden örneklenerek, 0 ile 7 seviyesinde LED ile gösterilecek şekilde ayarlanmaktadır. Yeni program kodlarında potansiyometre değerleri ve 1 ile 7 arasında yeniden örneklenen değerler, Görsel 2.54'teki gibi seri port ekranında görülmektedir. Potansiyometre seviyesini LED ile gösteren uygulama kodları aşağıdaki gibidir.

```
int i; // For döngüleri için değişken tanımlar.
void setup() {
    Serial.begin(9600);
    for (i=2;i<=8;i++)
        pinMode(i, OUTPUT); // 2'den 8'e kadar portları çıkış yapar.
}
void loop() {

    int pot = analogRead(A0);
    int seviye=map(pot,0,1023,1,7);
    Serial.print("Pot=");
    Serial.print(pot);

    Serial.print("\tSeviye=");
    Serial.println(seviye);

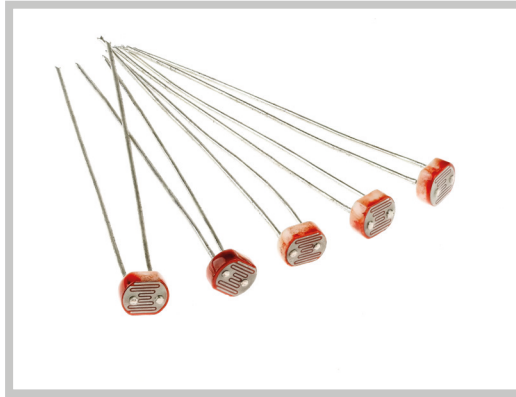
    for (i=2;i<=seviye;i++) digitalWrite(i,1);
    for (i=seviye+1;i<=7;i++) digitalWrite(i,0);
    delay(1);
}
```



Görsel 2.54: Potansiyometre seviyesini LED ile gösteren uygulamanın ekran çıktısı

## 2.8. IŞIK ETKİLİ DİRENÇ (LDR) UYGULAMALARI

LDR, İngilizce Light Dependent Resistor (Işığa Bağlı Direnç) kelimelerinin kısaltmasından oluşmuştur. LDR'nin diğer bir adı da **Foto dirençtir**. LDR'ler, üzerine ışık geldiğinde direnç değeri değişen yarı iletken maddelerden oluşmuş devre elemanlarıdır. Görsel 2.55'te LDR görülmektedir.



Görsel 2.55: LDR

### 23. Uygulama

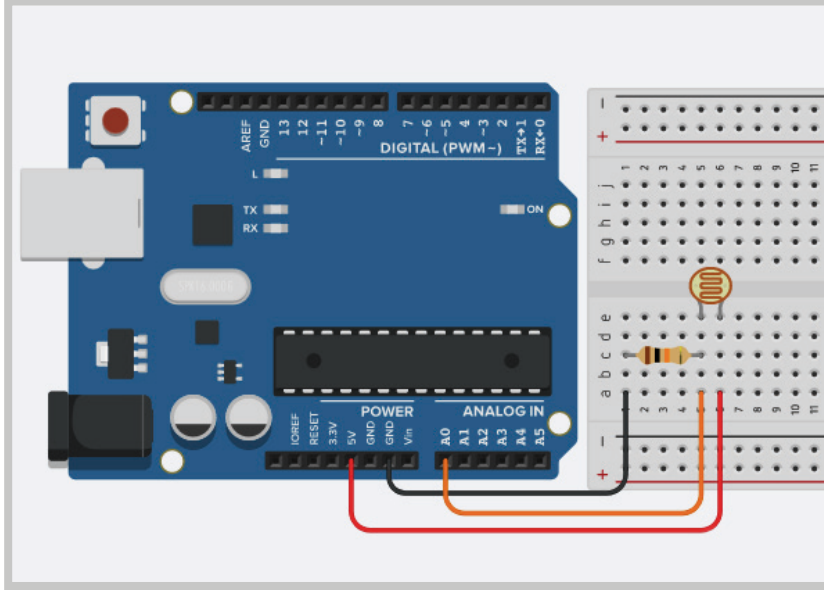
Bu uygulamadaki amaç, LDR devre elemanının ışık algılama sensörü olarak nasıl kullanıldığını göstermektir.

#### Gerekli Devre Elemanları

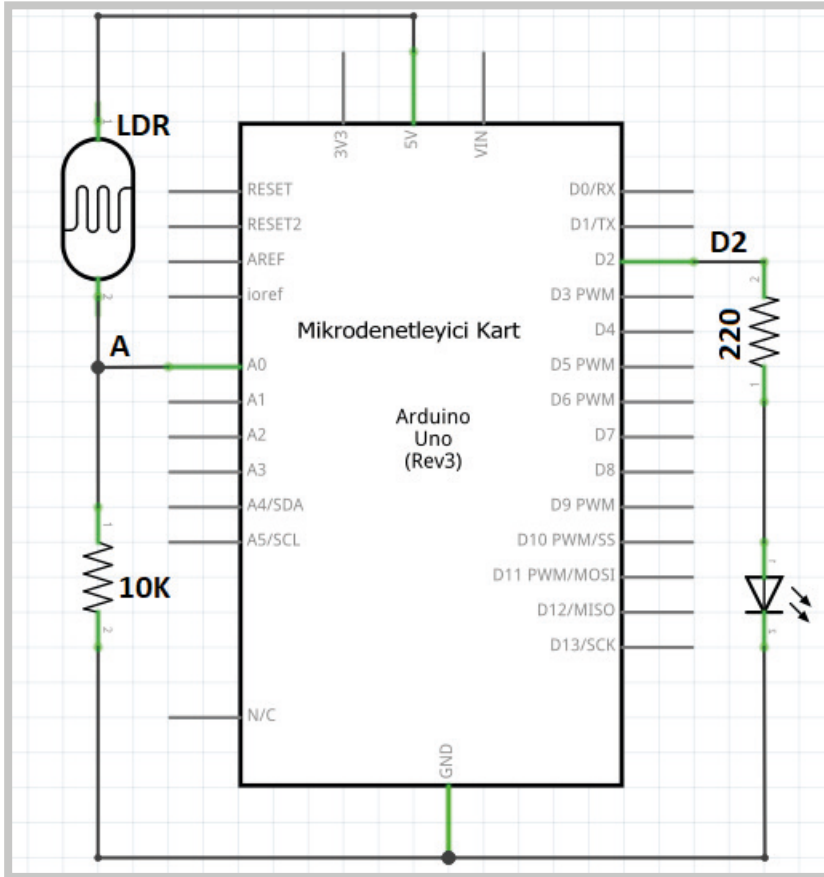
- 1 adet mikrodenetleyici kart
- 1 adet breadboard
- 1 adet LDR
- 1 adet 10K  $\Omega$  (kırmızı, kırmızı, kahverengi, altın rengi) direnç
- Bağlantı kabloları



**1. Adım:** LDR devre elemanlarını Görsel 2.55'te görüldüğü gibi ışık algılama sensörü olarak kullanabilmek için LDR devre elemanlarının elektronik devre kartına lehimli olarak üretilmiş çeşitleri bulunmaktadır. Mikrodenetleyici kart içinde analog girişler olduğu için Görsel 2.56 ve Görsel 2.57'de görüldüğü gibi 10KΩ (kahve-rengi, siyah, turuncu, altın rengi) bir direnç kullanılarak bir gerilim bölücü devresi yapılarak da kullanılabilir.



Görsel 2.56: LDR ile analog giriş uygulama devresi



Görsel 2.57: LDR ile analog giriş elektriksel devresi

## 2. Öğrenme Birimi: Mikrodenetleyici Kart Programlama

**2. Adım:** LDR ile analog giriş uygulamasının program kodları aşağıdaki gibidir.

```
void setup() {  
    Serial.begin(9600);  
}  
  
void loop() {  
    int LDR = analogRead(A0);  
  
    Serial.println("LDR=");  
    Serial.println(LDR);  
    delay(1);  
}
```

LDR ile analog giriş uygulamasının program kodlarına bakıldığında, `int LDR = analogRead(A0);` komutu ile mikrodenetleyici kartın A0 analog girişine bağlanan LDR'nin ışığa bağlı olarak gönderdiği elektriksel sinyalin sayısal değerini, LDR isimli değişkenine aktarmaktadır. `Serial.println(LDR);` komutu ile de LDR değişkeni içindeki sayısal veri seri port ekranına aktarılmaktadır.



### Sıra Sizde

Aşağıdaki sıralamaya göre uygulamayı yapınız.

- Görsel 2.56 veya Görsel 2.57'de devresi verilen LDR ile analog giriş uygulamasını breadboard üzerine kurunuz. LDR ile analog giriş uygulamasının program kodlarını mikrodenetleyici IDE programında yazınız.
- Programı mikrodenetleyici karta yükleyip, seri port ekranını açınız.
- Tüm işlemler bittikten sonra LDR üzerine gelen ışığın miktarını kapatıp açarak oluşan sayısal değişimleri seri port ekranından gözlemleyiniz.
- Seri port ekranını açmak için "Araçlar->Seri port ekranı"na ya da sağ üst köşedeki büyüteç işaretine tıklayabilirsiniz.



### Değerlendirme

Çalışmalarınız öğrenme biriminin sonunda yer alan Kontrol Listesi kullanılarak değerlendirilecektir. Çalışmalarınızı yaparken değerlendirme ölçütlerini dikkate alınız.

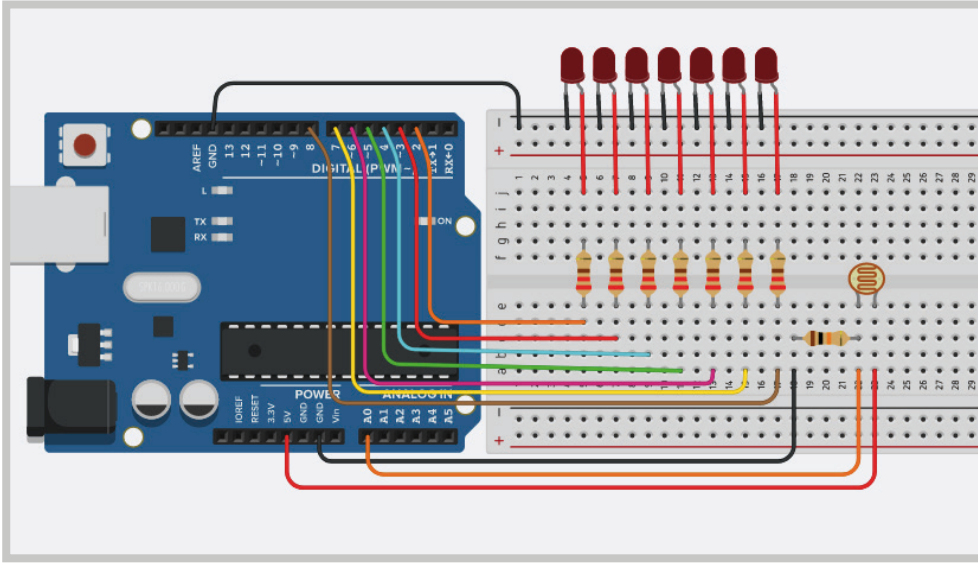
## 24. Uygulama

Bu uygulamadaki amaç, LDR ile ışık seviyesinin ölçülüp LED'ler ile gösterilmesini sağlamaktır.

### Gerekli Devre Elemanları

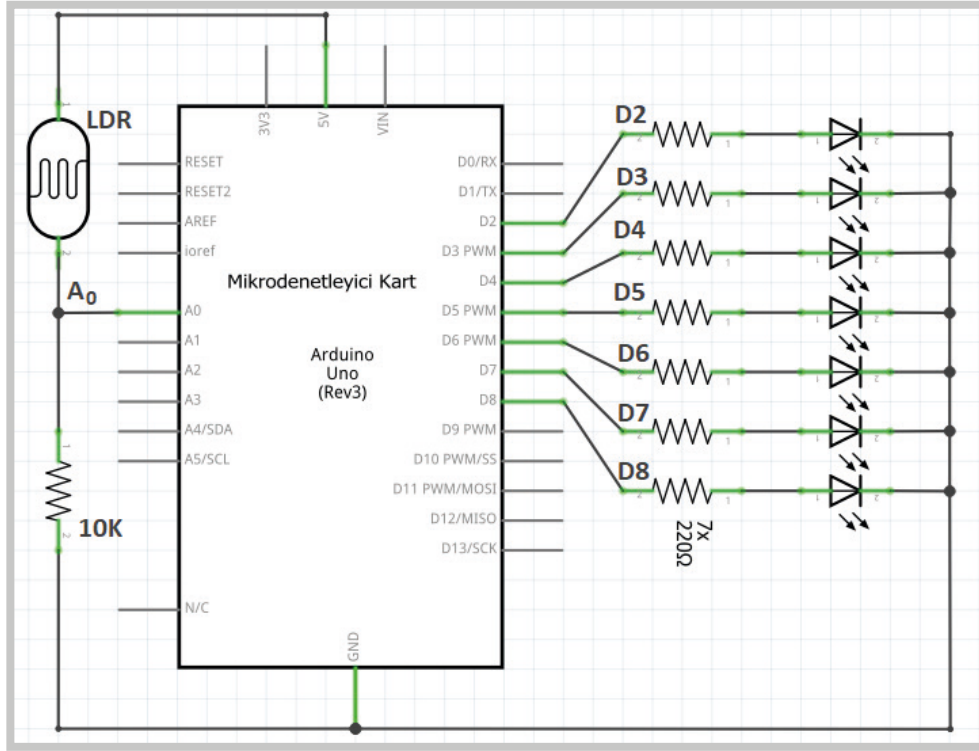
- 1 adet mikrodenetleyici kart
- 1 adet breadboard
- 1 adet LDR
- 7 adet LED
- 7 adet 220Ω (kırmızı, kırmızı, kahverengi, altın rengi) direnç
- 1 adet 10KΩ (kahverengi, siyah, sarı, altın rengi) direnç
- Bağlantı kabloları

**1. Adım:** Bu uygulamada (Görsel 2.58), LDR ile oluşturulan gerilim bölücü devresinden okunan 150 ile 600 arasındaki değerler, `map(LDR,150,600,1,7)` ; komutuyla yeniden örneklenerek 0 ile 7 seviyesinde LED ile gösterilecek şekilde ayarlanmaktadır.



**Görsel 2.58:** LDR ile ışık seviyesinin ölçülüp LED'ler ile gösterilen uygulama devresi

Normalde potansiyometre kullanıldığında okunan analog değer 0 ile 1023 arasında olabilirken, LDR yapısından ve üretiminden dolayı 150 ile 600 arasında bir değer alabilmektedir. Bu değerler, her LDR için farklılık gösterebilir. Program yüklenip seri porttan okunduktan sonra, kullanılan LDR'nin minimum ve maksimum değerleri alınarak, `map(LDR, minimum, maksimum, 1, 7)` ; komutunda değişiklik yapmak gerekir.



Görsel 2.59: LDR ile ışık seviyesinin ölçülüp LED'ler ile gösterilen uygulamanın elektriksel devresi

**2. Adım:** LDR ile ışık seviyesinin ölçülüp LED'ler ile gösteren uygulamanın program kodları aşağıdaki gösterilmektedir. Programın aşamaları aşağıdaki gibidir.

- Programın `void setup()` fonksiyonu içinde 2 adet for döngüsü kullanılmıştır.
- `for (i=2;i<=seviye;i++) digitalWrite(i,1);` komutuyla mikrodenetleyici kartın 2. portundan, seviye değişkeninde tutulan değer gösterdiği porta kadar 1 yapar.
- `for (i=2;i<=seviye;i++) digitalWrite(i,1);` komutuyla da mikrodenetleyici kartın seviye değişkeninde tutulan değer gösterdiği porttan, 8. porta kadar 0 yapar.
- Bu şekilde LDR'nin ürettiği değerler LED'ler ile ifade edilmiş olur.
- Programda aynı zamanda LDR değerleri ve 1 ile 7 arasında yeniden örneklenen değerler Görsel 2.60'ta gösterildiği gibi seri port ekranında görülmektedir.

```

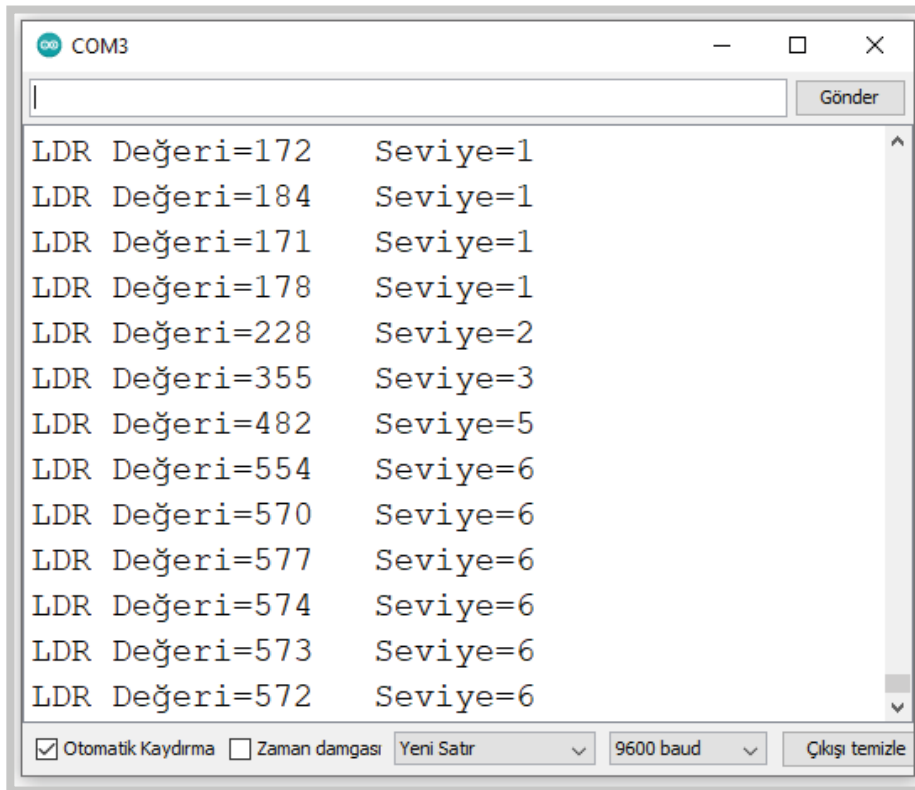
int i; // For döngüleri için değişken tanımlar.
void setup() {
  Serial.begin(9600);
  for (i=2;i<=8;i++)
    pinMode(i, OUTPUT); // 2'den 8'e kadar portları çıkış yapar.
}
void loop() {

  int LDR = analogRead(A0);
  int seviye=map(LDR,160,600,1,7);
  Serial.print("LDR Değeri=");
  Serial.print(LDR); // seviye değişkenini seri porta yazdırır.

  Serial.print("\tSeviye=");
  Serial.println(seviye); // seviye değişkenini seri porta yazdırır.

  for (i=2;i<=seviye;i++) digitalWrite(i,1);
  for (i=seviye+1;i<=8;i++) digitalWrite(i,0);
  delay(1);
}

```



Görsel 2.60: LDR ile ışık seviyesinin ölçülüp LED'ler ile gösterilen uygulamanın ekran çıktısı



### Sıra Sizde

Görsel 2.58 veya Görsel 2.59'da devresi verilen LDR ile ışık seviyesinin ölçülüp LED'ler ile gösterilen uygulamayı breadboard üzerine kurunuz. Bu uygulamaya ait program kodlarını mikrodenetleyici IDE programında yazınız. Programı mikrodenetleyici karta yükleyip, seri port ekranını açınız. LDR üzerine gelen ışığın miktarını, kapatıp açarak oluşan sayısal değişimleri seri port ekranından gözlemleyiniz.

LDR ışığını kapattığınızda oluşan minimum değeri ve LDR'e ışık geldiğinde ürettiği maksimum değeri `map(LDR, minimum, maksimum, 1, 7)` ; komutunda yerine yazınız. Programı mikrodenetleyici karta tekrar yükleyip seri port ekranını açınız. Tüm işlemler bittikten sonra LDR üzerine gelen ışığın miktarını, kapatıp açarak oluşan sayısal değişimleri hem seri port ekranından hem de LED'ler üzerinden gözlemleyiniz.

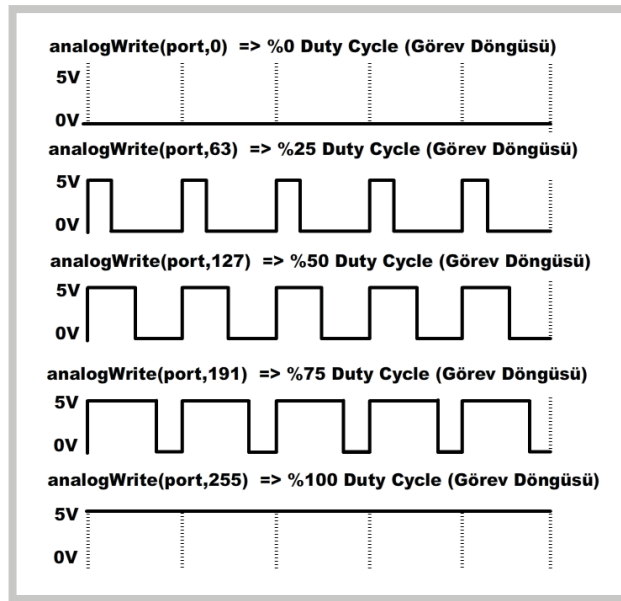


### Değerlendirme

Çalışmalarınız öğrenme biriminin sonunda yer alan Kontrol Listesi kullanılarak değerlendirilecektir. Çalışmalarınızı yaparken değerlendirme ölçütlerini dikkate alınız.

## 2.9. MİKRODENETLEYİCİ KART İLE ANALOG ÇIKIŞ (PWM) UYGULAMALARI

PWM, İngilizce Pulse Width Modulation (Darbe Genişliği Modülasyonu) kelimelerinin kısaltmasından oluşmuştur. Bazı devrelerde analog sinyal kullanılması gerekir ancak mikrodenetleyicilerde analog sinyal çıkışı yerine PWM adı verilen ve belirli periyotlarla kesik kesik enerji vermeyi sağlayan bir teknik kullanılır. PWM sinyalleri, LED parlaklık kontrolünden motor devir kontrolüne kadar birçok devrede kullanılabilir. Mikrodenetleyici kartın belirli portları PWM sinyal üretebilir. Bu portlar kart üzerinde (~) işaretiyle gösterilmiş olup D3, D5, D6, D9, D10, D11 olmak üzere 6 adettir. Mikrodenetleyici kartın PWM sinyali üretebilmesi için `analogWrite(port_no, deger)` ; komutu kullanılır. Komutta "port\_no" yerine PWM için kullanılan port numarası, "deger" yerine de 0 ile 255 arasında bir sayı yazılır. Görsel 2.61'de mikrodenetleyici kartın ürettiği PWM sinyalleri, komutta kullanılan değere göre verilmiştir.



Görsel 2.61: Mikrodenetleyici kartın ürettiği PWM sinyalleri



<http://kitap.eba.gov.tr/KodSor.php?KOD=20787>

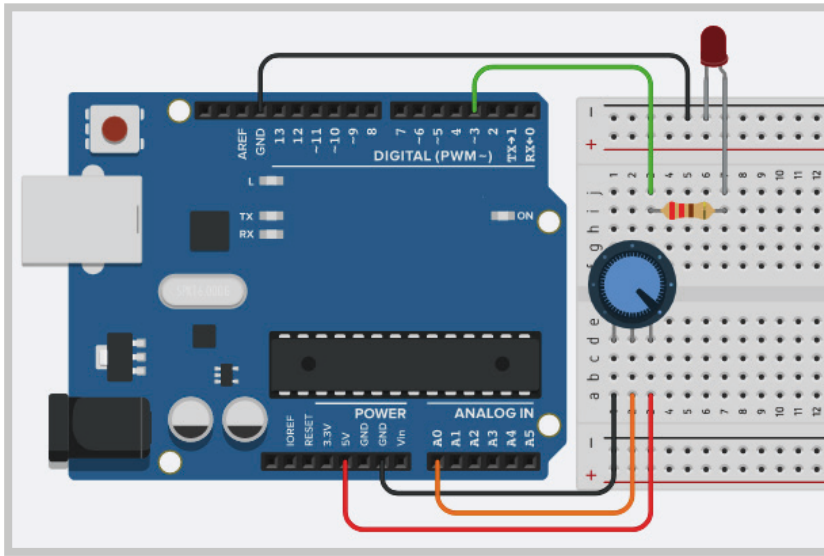
### 25. Uygulama

Bu uygulamadaki amaç, mikrodenetleyici kart ile örnek bir analog çıkış (PWM) uygulaması yapmaktır.

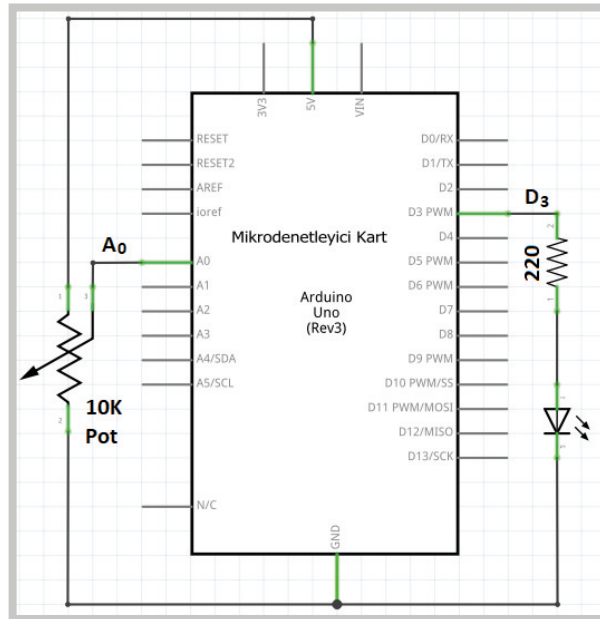
#### Gerekli Devre Elemanları

- 1 adet mikrodenetleyici kart
- 1 adet breadboard
- 1 adet potansiyometre
- 1 adet LED
- 1 adet 220Ω (kırmızı, kırmızı, kahverengi, altın rengi) direnç
- Bağlantı kabloları

**1. Adım:** Görsel 2.62 ve Görsel 2.63'te analog çıkış (PWM) uygulaması görülmektedir. Bu uygulamada, potansiyometreden alınan analog bilgiyle analog çıkış üzerinden LED parlaklığı ayarlanmaktadır.



Görsel 2.62: Mikrodenetleyici kart ile analog çıkış (PWM) uygulama devresi



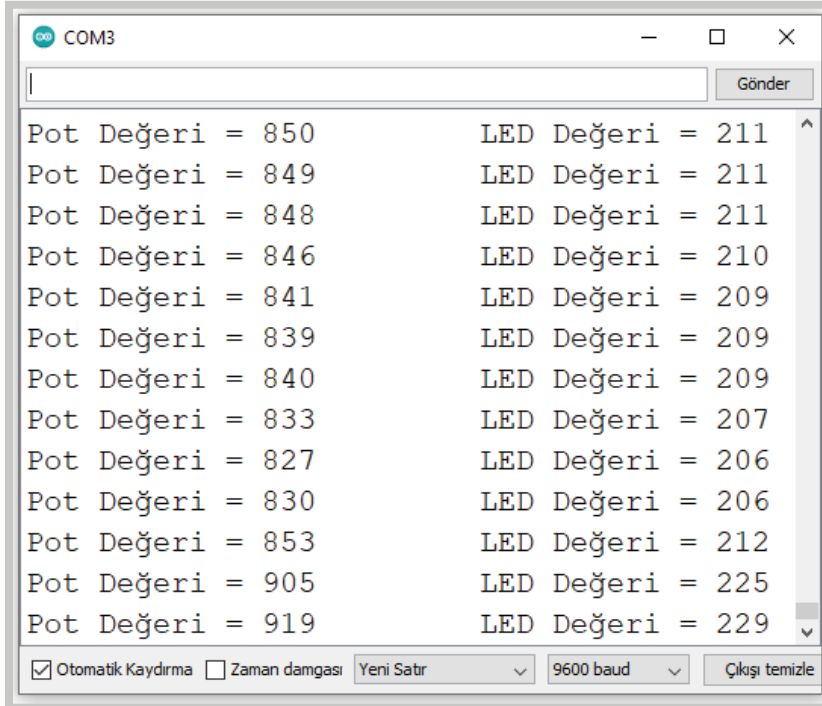
Görsel 2.63: Mikrodenetleyici kart ile analog çıkış (PWM) uygulamasının elektriksel devresi

## 2. Öğrenme Birimi: Mikrodenetleyici Kart Programlama

**2. Adım:** Aşağıda uygulamanın kodlarına bakıldığında;

- `int LED = map(pot, 0, 1023, 0, 255);` komutu potansiyometreden alınan 0 – 1023 arasındaki analog bilgiyi, 0 – 255 arasında yeniden örnekleyerek LED değişkenine aktarmaktadır.
- `analogWrite(3, LED);` komutu ile LED değişkenindeki bilgiyi, mikrodenetleyici kartın D3 portuna bağlı LED'e göndermektedir.
- LED'e gönderilen sayısal değer 0 ise LED sönmüş, 255 ise LED parlak yanmaktadır.
- Ayrıca potansiyometreden alınan analog bilgi Görsel 2.64'te görüldüğü gibi seri port ekranında da görüntülenebilmektedir.

```
void setup() {  
    Serial.begin(9600);  
    pinMode(3, OUTPUT);  
}  
void loop() {  
    int pot = analogRead(A0);  
    int LED = map(pot, 0, 1023, 0, 255);  
    analogWrite(3, LED); // pot değişkenini 3 numaralı porta gönderir.  
    Serial.print("Pot Değeri = ");  
    Serial.print(pot); // pot değişkenini seri porta yazdırır.  
    Serial.print("\tLED Değeri = ");  
    Serial.println(LED); // LED değişkenini seri porta yazdırır.  
    delay(2);  
}
```



Görsel 2.64: Mikrodenetleyici kart ile analog çıkış (PWM) uygulamasının ekran çıktısı





## Sıra Sizde

Görsel 2.62 veya Görsel 2.63'te verilen uygulamayı breadboard üzerine kurunuz. Potansiyometre seviyesini LED ile gösteren uygulamanın program kodlarını mikrodenetleyici IDE programında yazınız. Programı mikrodenetleyici karta yükleyip, seri port ekranını açınız. Tüm işlemler bittikten sonra potansiyometre düğmesini çevirerek seri port ekranında oluşan sayısal değişimleri gözlemleyiniz. Seri port ekranını açmak için "Araçlar->Seri port ekranı"na ya da sağ üst köşedeki büyüteç işaretine tıklayabilirsiniz.



<http://kitap.eba.gov.tr/KodSor.php?KOD=20788>

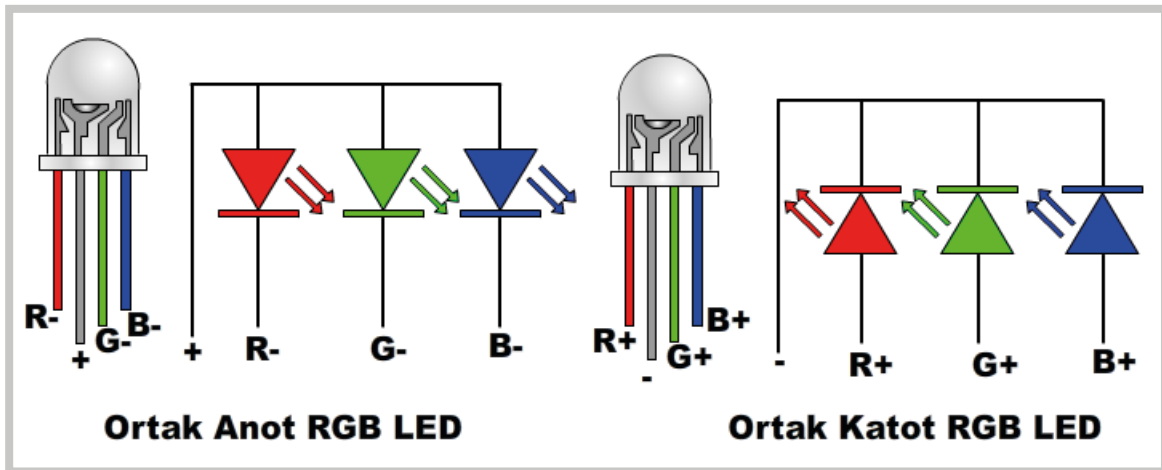


## Değerlendirme

Çalışmalarınız öğrenme biriminin sonunda yer alan Kontrol Listesi kullanılarak değerlendirilecektir. Çalışmalarınızı yaparken değerlendirme ölçütlerini dikkate alınız.

## 2.10. MİKRODENETLEYİCİ KART İLE RGB LED UYGULAMALARI

RGB, İngilizce Red Green Blue (Kırmızı, Yeşil, Mavi) kelimelerinin kısaltmasından oluşmuştur. RGB LED'ler, aynı kılıf içinde kırmızı, yeşil ve mavi renk ışık veren 3 farklı LED'in birleşiminden oluşur. RGB LED'ler, Görsel 2.65'te gösterildiği gibi ortak uçların bağlantı yöntemlerine göre ortak anot ve ortak katot olmak üzere ikiye ayrılır.



Görsel 2.65: RGB LED çeşitleri



## 26. Uygulama

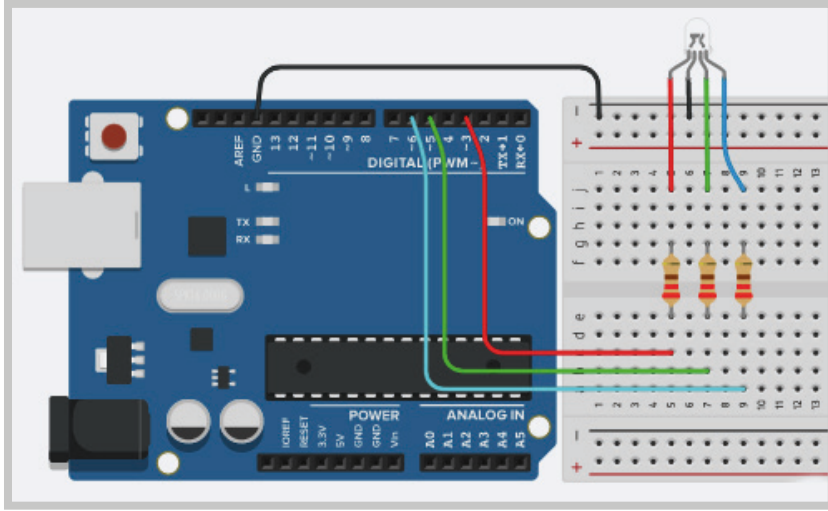
Bu uygulamadaki amaç, ortak katotlu RGB LED uygulaması yapmaktır.

### Gerekli Devre Elemanları

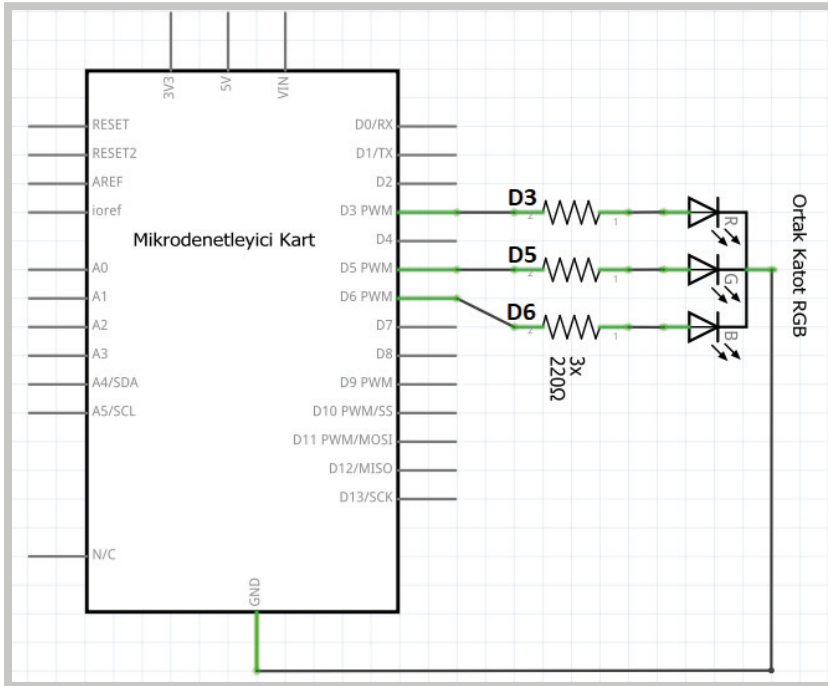
- 1 adet mikrodenetleyici kart
- 1 adet breadboard
- 1 adet Ortak Katot LED
- 3 adet direnç
- Bağlantı kabloları

## 2. Öğrenme Birimi: Mikrodenetleyici Kart Programlama

**1. Adım:** Görsel 2.66 ve Görsel 2.67’de ortak katotlu RGB LED uygulaması görülmektedir. Uygulamada eğer ortak anotlu RGB LED kullanılacaksa LED’in ortak ucu mikrodenetleyici kartın GND pini yerine, 5V pinine bağlanır. Uygulamada kullanılan RGB LED çeşidine göre program kodları da değişmektedir. Uygulamada ortak katot RGB LED kullanılırsa renkli LED’leri yakmak için port çıkışlarını 1 yapmak gerekir. Uygulamada ortak anot RGB LED kullanılırsa renkli LED’leri yakmak için port çıkışlarını 0 yapmak gerekir. Ortak katotlu RGB LED uygulamasının program kodları incelendiğinde, öncelikle R,G,B isiminde mikrodenetleyici kartın 3, 5, 6 numaralı portlarını temsil eden 3 adet değişken tanımlanmıştır. `void setup()` fonksiyonu içinde `pinMode()` komutuyla mikrodenetleyici kartın 3, 5, 6 numaralı portları çıkış yapılır. `void loop()` fonksiyonu içinde `digitalWrite()` komutuyla RGB LED’in tüm renklerini ve renk karışımlarını 1 saniye aralıklarla yakan kodlar yazılmıştır.



Görsel 2.66: Ortak katotlu RGB LED uygulama devresi



Görsel 2.67: Ortak katotlu RGB LED uygulamasının elektriksel devresi

**2. Adım:** Ortak katot RGB LED uygulamasının program kodları şunlardır:

```
int R=3;
int G=5;
int B=6;
void setup() {
  pinMode(R, OUTPUT);
  pinMode(G, OUTPUT);
  pinMode(B, OUTPUT);
}
void loop() {
  digitalWrite(R, 1);digitalWrite(G, 0);
  digitalWrite(B, 0); delay(1000);//Kırmızı
  digitalWrite(R, 0);digitalWrite(G, 1);
  digitalWrite(B, 0); delay(1000);//Yeşil
  digitalWrite(R, 0);digitalWrite(G, 0);
  digitalWrite(B, 1); delay(1000);//Mavi
  digitalWrite(R, 1);digitalWrite(G, 1);
  digitalWrite(B, 0); delay(1000);//Sarı
  digitalWrite(R, 1);digitalWrite(G, 0);
  digitalWrite(B, 1); delay(1000);//Mor (Magenta)
  digitalWrite(R, 0);digitalWrite(G, 1);
  digitalWrite(B, 1); delay(1000);//Turkuaz (Cyan)
  digitalWrite(R, 1);digitalWrite(G, 1);
  digitalWrite(B, 1); delay(1000);//Beyaz
  digitalWrite(R, 0);digitalWrite(G, 0);
  digitalWrite(B, 0); delay(1000);
}
```

**NOT:** Uygulama Ortak anot RGB LED ile yapılırsa program kodları şunlardır:

```
int R=3;
int G=5;
int B=6;
void setup() {
  pinMode(R, OUTPUT);
  pinMode(G, OUTPUT);
  pinMode(B, OUTPUT);
}
void loop() {
  digitalWrite(R, 0);digitalWrite(G, 1);
  digitalWrite(B, 1); delay(1000);//Kırmızı
  digitalWrite(R, 1);digitalWrite(G, 0);
  digitalWrite(B, 1); delay(1000);//Yeşil
  digitalWrite(R, 1);digitalWrite(G, 1);
  digitalWrite(B, 0); delay(1000);//Mavi
  digitalWrite(R, 0);digitalWrite(G, 0);
  digitalWrite(B, 1); delay(1000);//Sarı
  digitalWrite(R, 0);digitalWrite(G, 1);
  digitalWrite(B, 0); delay(1000);//Mor (Magenta)
  digitalWrite(R, 1);digitalWrite(G, 0);
  digitalWrite(B, 0); delay(1000);//Turkuaz (Cyan)
  digitalWrite(R, 0);digitalWrite(G, 0);
  digitalWrite(B, 0); delay(1000);//Beyaz
  digitalWrite(R, 1);digitalWrite(G, 1);
  digitalWrite(B, 1); delay(1000);
}
```



### Sıra Sizde

Görsel 2.66 veya Görsel 2.67’de devresi verilen ortak katotlu RGB LED uygulamasının uygulamasını breadboard üzerine kurunuz. Bu uygulamaya ait program kodlarını mikrodenetleyici IDE programında yazınız. Tüm işlemler bittikten sonra programı mikrodenetleyici karta yükleyip, devrenin çalışmasını gözlemleyiniz.



<http://kitap.eba.gov.tr/KodSor.php?KOD=20789>



### Değerlendirme

Çalışmalarınız öğrenme biriminin sonunda yer alan Kontrol Listesi kullanılarak değerlendirilecektir. Çalışmalarınızı yaparken değerlendirme ölçütlerini dikkate alınız.



### 27. Uygulama

Bu uygulamadaki amaç, 3 adet potansiyometre ile ortak katotlu RGB LED ile renklerin parlaklığını kontrol eden uygulamayı gerçekleştirmektir.

#### Gerekli Devre Elemanları

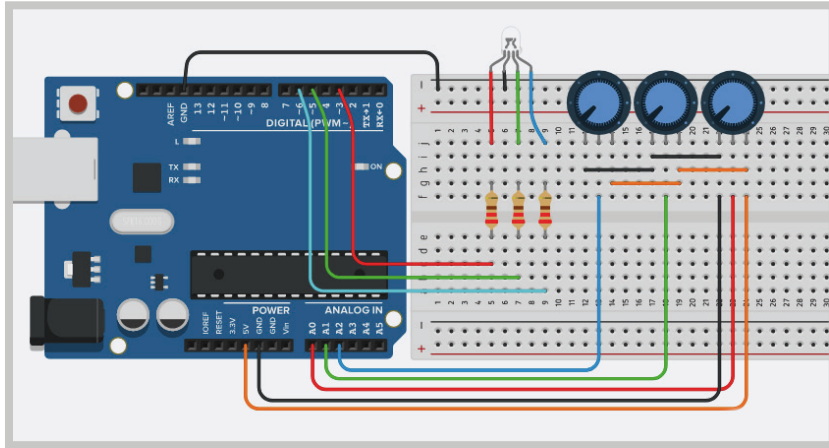
- 1 adet mikrodenetleyici kart
- 1 adet breadboard
- 3 adet potansiyometre
- 1 adet ortak katotlu RGB LED
- 3 adet 220Ω (kırmızı, kırmızı, kahverengi, altın rengi) direnç
- Bağlantı kabloları

**1. Adım:** Görsel 2.68 ve Görsel 2.69’da 3 adet potansiyometre ile ortak katotlu RGB LED ile renklerin parlaklığını kontrol eden uygulama görülmektedir. Uygulamada, ortak anotlu RGB LED kullanılacaksa LED’in ortak ucu, mikrodenetleyici kartın GND pini yerine 5V pinine bağlanır.

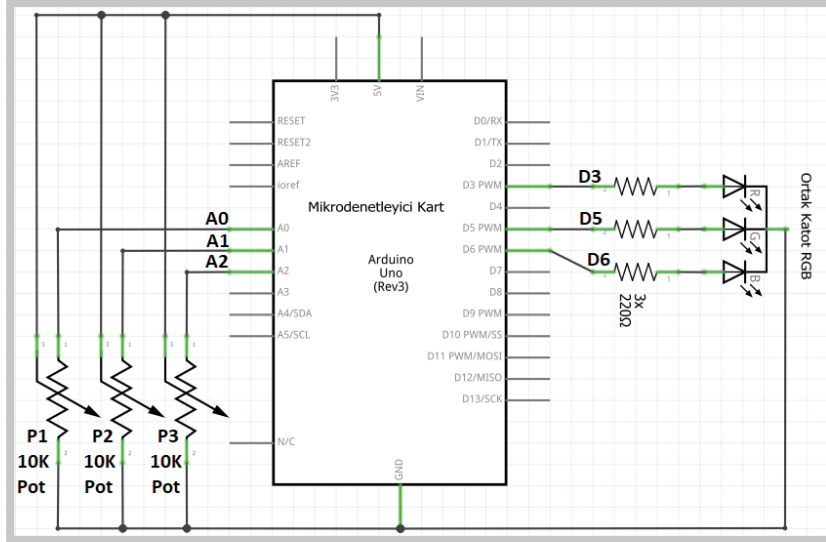


### Önemli

Uygulamada kullanılan RGB LED çeşidine göre program kodları değişmez.



Görsel 2.68: Ortak katotlu RGB LED'in potansiyometre ile kontrolü uygulama devresi



Görsel 2.69: Ortak katotlu RGB LED'in potansiyometre ile kontrolü uygulamasının elektriksel devresi

**2. Adım:** RGB LED'in potansiyometre ile kontrolü uygulamasının aşağıdaki program kodları incelendiğinde;

- `void setup ()` fonksiyonu içinde `pinMode ()` komutuyla mikrodenetleyici kartın 3, 5, 6 numaralı portları çıkış yapılmıştır.
- `void loop ()` fonksiyonu içinde `int Rpot = analogRead (A0)` ; koduyla kırmızı LED için mikrodenetleyici kartın A0 girişine bağlı potansiyometreden alınan analog bilgi, Rpot değişkenine aktarılır.
- `int Rled = map (Rpot, 0, 1023, 0, 255)` ; komutu potansiyometreden alınan 0 – 1023 arasındaki analog bilgiyi, 0 – 255 arasında yeniden örnekleyerek Rled değişkenine aktarmaktadır.
- `analogWrite (3, Rled)` ; komutu ile Rled değişkenindeki bilgiyi, mikrodenetleyici kartın D3 portuna bağlı RGB'nin kırmızı LED'ine göndermektedir.
- Bu işlem aynı şekilde yeşil ve mavi LED için de yapılır.

**3. Adım:** RGB LED'in potansiyometre ile kontrolü uygulamasının program kodları aşağıdaki gibidir.

```
void setup () {
  pinMode (3, OUTPUT) ; //Kırmızı LED portu
  pinMode (5, OUTPUT) ; //Yeşil LED portu
  pinMode (6, OUTPUT) ; //Mavi LED portu
}
void loop () {
  int Rpot = analogRead (A0) ;
  int Rled = map (Rpot, 0, 1023, 0, 255) ;
  int Gpot = analogRead (A1) ;
  int Gled = map (Gpot, 0, 1023, 0, 255) ;
  int Bpot = analogRead (A2) ;
  int Bled = map (Bpot, 0, 1023, 0, 255) ;
  analogWrite (3, Rled) ; // Rpot değişkenini 3. porta gönderir.
  analogWrite (5, Gled) ; // Gpot değişkenini 5. porta gönderir.
  analogWrite (6, Bled) ; // Bpot değişkenini 6. porta gönderir.
}
```



### Sıra Sizde

Görsel 2.68 veya Görsel 2.69'da devresi verilen ortak katotlu RGB LED'in potansiyometre ile kontrolü uygulamasını breadboard üzerine kurunuz. Bu uygulamaya ait program kodlarını mikrodenetleyici IDE programında yazınız. Tüm işlemler bittikten sonra programı mikrodenetleyici karta yükleyip, devrenin çalışmasını gözlemleyiniz.



### Değerlendirme

Çalışmalarınız öğrenme biriminin sonunda yer alan Kontrol Listesi kullanılarak değerlendirilecektir. Çalışmalarınızı yaparken değerlendirme ölçütlerini dikkate alınız.



### 28. Uygulama

Bu uygulamadaki amaç, RGB LED ile rastgele renkler üreten bir uygulama gerçekleştirmektir.

#### Gerekli Devre Elemanları

- 1 adet mikrodenetleyici kart
- 1 adet breadboard
- 3 adet potansiyometre
- 1 adet ortak katotlu RGB LED
- 3 adet 220Ω (kırmızı, kırmızı, kahverengi, altın rengi) direnç
- Bağlantı kabloları

**1. Adım:** Bu uygulama için Görsel 2.68 ve Görsel 2.69'daki gibi devrenin tekrar kurulması gerekir. Rastgele sayılar üretmek için `random(0,255)` ; komutu kullanılır. Kullanılan bu komutta 0 ile 255 arasında rastgele bir sayı üretilir. Üretilen rastgele sayılar "RRandom", "GRandom", "BRandom" değişkenlerine aktarılır. `analogWrite(3, RRandom)` ; komutu ile de üretilen bu rastgele sayılar mikrodenetleyici kartın ilgili portuna gönderilir.

**2. Adım:** RGB LED ile rastgele renk üreten uygulamanın program kodları aşağıdaki gibidir.

```
int RRandom;
int GRandom;
int BRandom;
void setup()
{
  pinMode(3, OUTPUT); //Kırmızı LED portu
  pinMode(5, OUTPUT); //Yeşil LED portu
  pinMode(6, OUTPUT); //Mavi LED portu
}
void loop()
{
```

```
// Bu aşamada her adımda 0 ile 255 arasında rastgele sayı üretir.
RRandom = random(0,255);
GRandom = random(0,255);
BRandom = random(0,255);
// Seçilen rastgele renk değerleri
// ayrı ayrı analog olarak renk pinlerine
analogWrite(3, RRandom); // RRandom değişkenini 3. porta gönderir.
analogWrite(5, GRandom); // GRandom değişkenini 5. porta gönderir.
analogWrite(6, BRandom); // BRandom değişkenini 6. porta gönderir.

delay(1000);
}
```



### Sıra Sizde

Görsel 2.68 veya Görsel 2.69'da devresi verilen ortak katotlu RGB LED'in potansiyometre ile kontrolü uygulamasını breadboard üzerine kurunuz. Bu uygulamaya ait program kodlarını mikrodenetleyici IDE programında yazınız. Tüm işlemler bittikten sonra programı mikrodenetleyici karta yükleyip, devrenin çalışmasını gözlemleyiniz.



### Değerlendirme

Çalışmalarınız öğrenme biriminin sonunda yer alan Kontrol Listesi kullanılarak değerlendirilecektir. Çalışmalarınızı yaparken değerlendirme ölçütlerini dikkate alınız.



### 29. Uygulama

Bu uygulamadaki amaç, RGB LED ile tüm renklerin birbirine geçişlerini sağlayan uygulamayı gerçekleştirmektir.

#### Gerekli Devre Elemanları

- 1 adet mikrodenetleyici kart
- 1 adet breadboard
- 3 adet potansiyometre
- 1 adet ortak katotlu RGB LED
- 3 adet 220Ω (kırmızı, kırmızı, kahverengi, altın rengi) direnç
- Bağlantı kabloları

**1. Adım:** Uygulamanın devresi Görsel 2.68 ve Görsel 2.69'daki ile aynıdır; sadece programında değişiklik vardır. Uygulamada eğer ortak anotlu RGB LED kullanılacaksa LED'in ortak ucu mikrodenetleyici kartın GND pini yerine, 5V pinine bağlanır.



Uygulamada kullanılan RGB LED çeşidine göre program kodları da değişmez.

**2. Adım:** RGB LED ile tüm renklerin birbirine geçişlerini sağlayan uygulamanın program kodları incelendiğinde;

- Red, Green, Blue isiminde mikrodenetleyici kartın 3, 5, 6 numaralı portlarını temsil eden 3 adet değişken tanımlanmıştır.
- `void setup()` fonksiyonu içinde `pinMode()` komutuyla mikrodenetleyici kartın 3, 5, 6 numaralı portları çıkış yapılır.
- `void loop()` fonksiyonu içinde 3 adet for döngüsü kullanılmıştır.
- Bu döngülerden ilkinde `analogWrite(Red, (255-i))`; komutuyla kırmızı LED parlaklığı 255'ten 0'a doğru azaltılırken `analogWrite(Green, i)`; komutuyla yeşil LED parlaklığı 0'dan 255'e kadar artırılır.
- Bu sayede RGB LED'de kırmızıdan yeşile dönüşen bir renk geçişi sağlanır. Aynı şekilde 2. for döngüsünde RGB LED'de yeşilden maviye, 3. for döngüsünde RGB LED'de maviden kırmızıya dönüşen bir renk geçişi sağlanır.

RGB LED ile tüm renklerin birbirine geçişlerini sağlayan uygulamanın program kodları aşağıdaki gibidir.

```
int Red = 3;
int Green = 5;
int Blue = 6;
void setup()
{
    pinMode(Red, OUTPUT);
    pinMode(Green, OUTPUT);
    pinMode(Blue, OUTPUT);
}
void loop()
{ int i;

for (i=0;i<=255;i++)
{ // Kırmızıdan Yeşile Geçiş
analogWrite(Red, (255-i));
analogWrite(Green, i);
delay(10);
}

for (i=0;i<=255;i++)
{ // Yeşilden Maviye Geçiş
analogWrite(Green, (255-i));
analogWrite(Blue, i);
delay(10);
}

for (i=0;i<=255;i++)
{ // Maviden Kırmızıya Geçiş
analogWrite(Blue, (255-i));
analogWrite(Red, i);
delay(10);
}
}
```





### Sıra Sizde



<http://kitap.eba.gov.tr/KodSor.php?KOD=20790>

Görsel 2.68 veya Görsel 2.69'daki devreyi breadboard üzerine kurunuz. RGB LED ile tüm renklerin birbirine geçişlerini sağlayan uygulamanın program kodlarını mikrodenetleyici IDE programında yazınız. Tüm işlemler bittikten sonra programı mikrodenetleyici karta yükleyip, devrenin çalışmasını gözlemleyiniz.



### Değerlendirme

Çalışmalarınız öğrenme biriminin sonunda yer alan Kontrol Listesi kullanılarak değerlendirilecektir. Çalışmalarınızı yaparken değerlendirme ölçütlerini dikkate alınız.

## 2.11. MİKRODENETLEYİCİ KART İLE BUZZER UYGULAMALARI

Buzzer devre elemanları, temel olarak silindirik şeklindeki metal plaka ve üzerindeki piezo seramik tabakadan oluşur. Piezo seramik diske enerji verildiğinde metal plaka üzerinde titreşim oluşturularak ses dalgaları üretir. Görsel 2.70'te görüldüğü gibi **aktif buzzer** ve **pasif buzzer** olmak üzere iki çeşidi vardır. Aktif buzzer'ların içinde ses sinyali üreten elektronik bir devre bulunur.



Görsel 2.70: Buzzer yapısı ve çeşitleri

Buzzer bağlantısı yapılırken üzerinde + işareti bulunan uç, mikrodenetleyici kartın 5V pinine, diğer uç da GND pinine bağlandığında bip sesi üretilir. Pasif buzzer içinde ses sinyali üreten elektronik bir devre bulunmadığı için ses sinyali üreten devrelerle kullanılabilir.



### 30. Uygulama

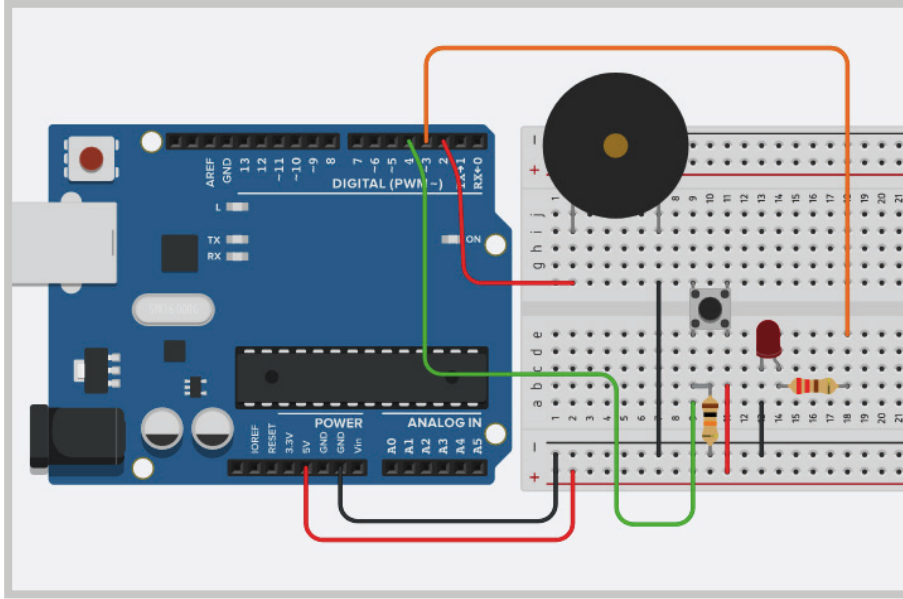
Aktif buzzer ile ilgili bir uygulama yapılacaktır.

#### Gerekli Devre Elemanları

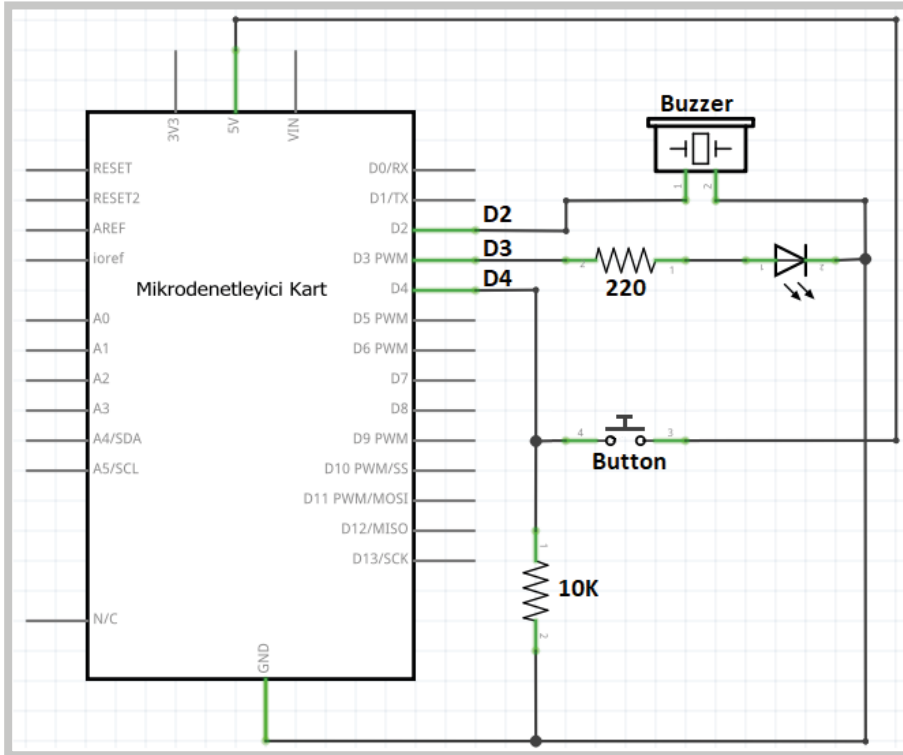
- 1 adet mikrodenetleyici kart
- 1 adet breadboard
- 1 adet aktif buzzer
- Bağlantı kabloları

## 2. Öğrenme Birimi: Mikrodenetleyici Kart Programlama

**1. Adım:** Görsel 2.71 ve Görsel 2.72'de aktif buzzer ile ilgili bir uygulama görülmektedir. Bu uygulamada mikrodenetleyici kartın D4 portuna bağlı botuna basıldığında hem D3 portuna bağlı LED yanar hem de D2 portuna bağlı buzzer bip sesi üretir.



Görsel 2.71: Aktif buzzer uygulama devresi



Görsel 2.72: Aktif buzzer uygulamasının elektriksel devresi

**2. Adım:** Aktif buzzer uygulamasının program kodları aşağıdaki gibidir.

```
void setup() {  
    pinMode(2, OUTPUT); //Buzzer' ın bağlı olacağı pini çıkış yapar.  
    pinMode(3, OUTPUT); //LED' in bağlı olacağı pini çıkış yapar.  
    pinMode(4, INPUT); //Butonun bağlı olacağı pini giriş yapar.  
}  
  
void loop() {  
    if (digitalRead(4)==1) { //Eğer Butona basıldı ise;  
        digitalWrite(2,1); //Buzzer' ı çalıştır.  
        digitalWrite(3,1); //LED' i yak.  
    } else {  
        digitalWrite(2,0);  
        digitalWrite(3,0);  
    }  
}
```



### Sıra Sizde

Görsel 2.71 veya Görsel 2.72'deki aktif buzzer uygulamasını breadboard üzerine kurunuz. Bu uygulamaya ait program kodlarını mikrodenetleyici IDE programında yazınız. Tüm işlemler bittikten sonra programı mikrodenetleyici karta yükleyip, devrenin çalışmasını gözlemleyiniz.



### Değerlendirme

Çalışmalarınız öğrenme biriminin sonunda yer alan Kontrol Listesi kullanılarak değerlendirilecektir. Çalışmalarınızı yaparken değerlendirme ölçütlerini dikkate alınız.



### 31. Uygulama

Mikrodenetleyici kart ile melodi çalma uygulaması yapılacaktır.

#### Gerekli Devre Elemanları

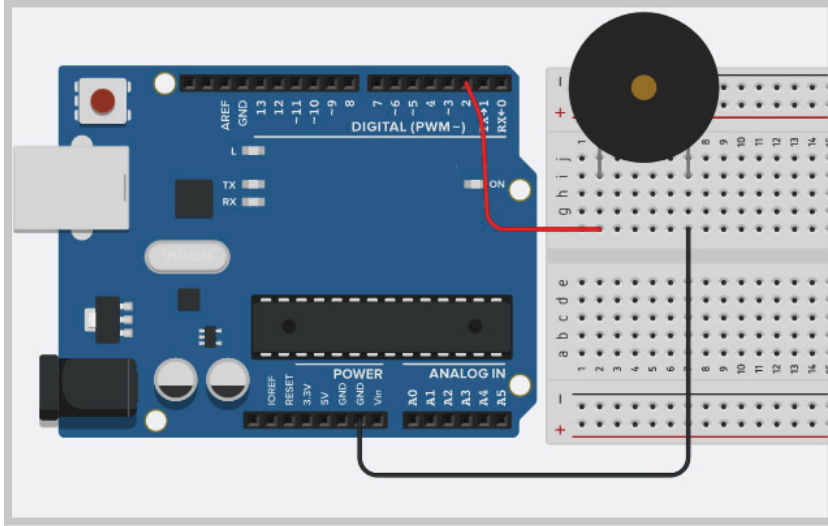
- 1 adet mikrodenetleyici kart
- 1 adet breadboard
- 1 adet Buzzer
- 1 adet LED
- 1 adet 220Ω (kırmızı, kırmızı, kahverengi, altın rengi) direnç
- 1 adet 10KΩ (kahverengi, siyah, sarı, altın rengi) direnç
- Bağlantı kabloları

## 2. Öğrenme Birimi: Mikrodenetleyici Kart Programlama

**1. Adım:** Melodi uygulaması oluşturmak için Görsel 2.73'teki gibi mikrodenetleyici kartın D2 numaralı portuna pasif buzzer bağlamak gerekir. Melodi devresinde aktif buzzer da bağlanabilir ancak daha temiz bir ses elde edebilmek için pasif buzzer tercih edilmelidir. Buzzer ile melodi oluşturma uygulamasının program kodları incelendiğinde, öncelikle notaların ses frekanslarını tanımlamak gerekir. Notaların ses frekans değerleri için <https://www.arduino.cc/en/tutorial/melody> adresinden ya da Tablo 2.4'ten yararlanılabilir.

Tablo 2.4: Notaların Ses Frekans Değerleri

Nota	Harf gösterimi	Ses Frekansı
Do	c	261 Hz
Re	d	294 Hz
Mi	e	329 Hz
Fa	f	349 Hz
Sol	g	392 Hz
La	a	440 Hz
Si	b	493 Hz
Do	C	523 Hz



Görsel 2.73: Buzzer ile melodi oluşturma uygulama devresi

**2. Adım:** Görsel 2.73'te görülen melodi uygulaması programında;

- **tone(buzzer, frekans)** ; komutu kullanılır. Bu komutta "buzzer" değişkeni buzzer'ın bağlı bulunduğu mikrodenetleyici portunu, "frekans" değişkeni ise buzzer'ın üreteceği ses frekansını belirlemek için kullanılır.
- **noTone(buzzer)** ; komutu ise üretilen sesi sonlandırır. Bir müzik parçasının sıralı bir şekilde çalınabilmesi için **notalar[]** isminde bir dizi oluşturulmuştur.
- Dizideki notaları okumak için **for** döngüsü kullanılır.
- Program içinde 2 adet **delay()** ; komutu kullanılır. Bunlardan **delay(500)** ; komutu her notanın çalma süresini, **delay(50)** ; komutu ise notalar arasındaki bekleme süresini belirlemek için kullanılır.

**3. Adım:** Buzzer ile melodi oluşturma uygulamasının program kodları aşağıdaki gibidir.

```
int buzzer=2; //Buzzer' ın bağlı olacağı port numarası yazılır.

int Do=261;

int Re=294;

int Mi=329;

int Fa=349;

int Sol=392;

int La=440;

int Si=493;

int Do2=523;

int notalar[]={Do,Re,Mi,Sol,La,Si,Do2}; //Müzik parçasının notaları
yazılır.

void setup() {
    pinMode(buzzer,OUTPUT);

    //Buzzer' ın bağlı olacağı pini çıkış yapar.
}

void loop() {
    int i;

    for (i=0;i< sizeof(notalar)/sizeof(int);i++)
        // Dizi sayısı kadar döngü oluşturur.
        {
            tone(buzzer,notalar[i]); // Nota seslerini üretir.

            delay(500); // Her notanın çalma süresi

            noTone(buzzer);

            delay(50); // Notalar arasındaki bekleme süresi
        }

    noTone(buzzer); //Sesi kapatır.
}
```

## 2. Öğrenme Birimi: Mikrodenetleyici Kart Programlama



### Sıra Sizde



HTTP://KITAP.EBA.GOV.TR/KODSOR.PHP?KOD=20791

Görsel 2.73'teki buzzer ile melodi oluşturma uygulamasını breadboard üzerine kurunuz. Bu uygulamaya ait program kodlarını mikrodenetleyici IDE programında yazınız. Tüm işlemler bittikten sonra programı mikrodenetleyici karta yükleyip, devrenin çalışmasını gözlemleyiniz.

```
int i;
void setup() {
  for(i=2;i<=6;i++) pinMode(i, OUTPUT);
  pinMode(11, INPUT);
}
void loop() {
  //Anahtar sürgüsü sola çekildiğinde
  if (digitalRead(11)==0) {
    for(i=2;i<=6;i++) { digitalWrite(i, 1); delay(1000); }
    for(i=2;i<=6;i++) { digitalWrite(i, 0); delay(1000); }
  } //if için
  //Anahtar sürgüsü sağa çekildiğinde
  if (digitalRead(11)==1) {
    for(i=6;i>=2;i--) { digitalWrite(i, 1); delay(1000); }
    for(i=6;i>=2;i--) { digitalWrite(i, 0); delay(1000); }
  } //if için
} //loop için
```



### Değerlendirme

Çalışmalarınız öğrenme biriminin sonunda yer alan Kontrol Listesi kullanılarak değerlendirilecektir. Çalışmalarınızı yaparken değerlendirme ölçütlerini dikkate alınız.



### Sıra Sizde

Görsel 2.73'teki buzzer ile melodi oluşturma uygulamasını breadboard üzerine kurunuz. Bu uygulamaya ait program kodlarını mikrodenetleyici IDE programında yazınız. notalar[] dizisine aşağıdaki müzik parçasının notalarını ekleyiniz. Tüm işlemler bittikten sonra programı mikrodenetleyici karta yükleyip, devrenin çalışmasını gözlemleyiniz.

### Müzik Parçasının Notaları

Do, Do, Sol, Sol, La, La, Sol, Fa, Fa, Mi, Mi, Re, Re, Do, Do, Do, Sol, Sol, La, La, Sol, Fa, Fa, Mi, Mi, Re, Re, Do, Sol, Sol, Fa, Fa, Mi, Mi, Re, Sol, Sol, Fa, Fa, Mi, Mi, Re, Do, Do, Sol, Sol, La, La, Sol, Fa, Fa, Mi, Mi, Re, Re, Do



### Değerlendirme

Çalışmalarınız öğrenme biriminin sonunda yer alan Kontrol Listesi kullanılarak değerlendirilecektir. Çalışmalarınızı yaparken değerlendirme ölçütlerini dikkate alınız.

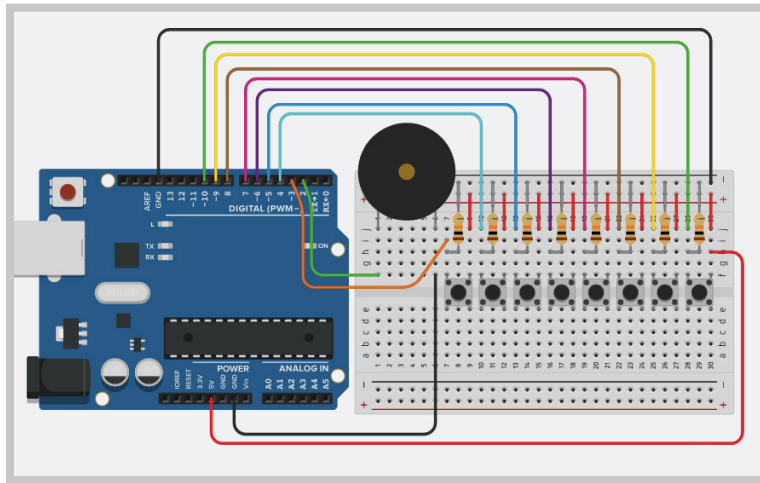
### 32. Uygulama

Bu uygulamanın amacı buzzer ile dijital piyano uygulaması oluşturmaktır.

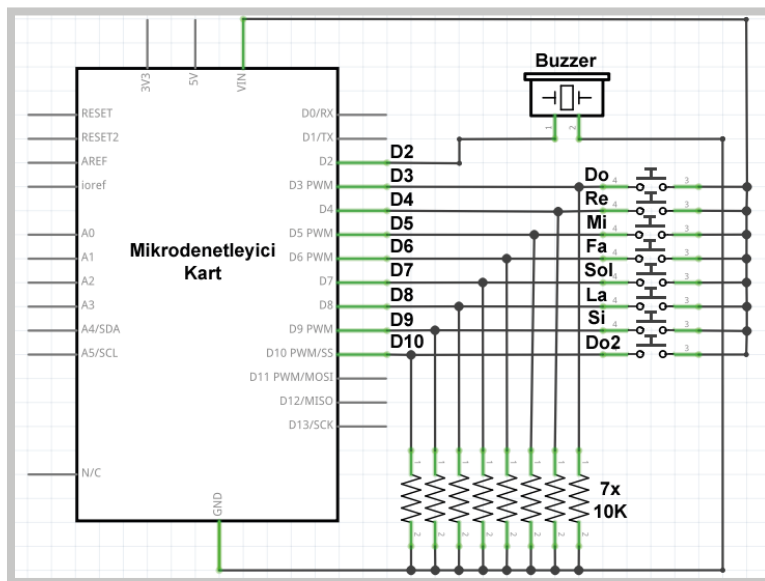
#### Gerekli Devre Elemanları

- 1 adet mikrodenetleyici kart
- 1 adet breadboard
- 1 adet buzzer
- 8 adet buton
- 8 adet 10KΩ (kahverengi, siyah, sarı, altın rengi) direnç
- Bağlantı kabloları

**1. Adım:** Görsel 2.74 ve Görsel 2.75'te müzik notalarını butonlar yardımıyla çalan buzzer uygulaması görülmektedir. Bu uygulamada mikrodenetleyici kartın D3 ile D10 arasındaki portlarına bağlı butonlara basıldığında ilgili butona ait ses frekansı buzzer yardımıyla üretilir.



Görsel 2.74: Buzzer ile dijital piyano uygulama devresi



Görsel 2.75: Buzzer ile dijital piyano uygulamasının elektriksel devresi

**2. Adım:** Buzzer ile dijital piyano uygulamasının kodları aşağıdaki gibidir.

```
int buzzer=2; //Buzzer' ın bağlı olacağı pin numarası

int Do=261;

int Re=294;

int Mi=329;

int Fa=349;

int Sol=392;

int La=440;

int Si=493;

int Do2=523;

void setup() {

    pinMode(buzzer,OUTPUT);

    //Buzzer' ın bağlı olacağı pini çıkış yapar.

    for (int i=3;i<=10;i++) pinMode(i,INPUT);

    pinMode(13,OUTPUT);

}

void loop() {

    noTone(buzzer);

    if (digitalRead(3)==1) {

        tone(buzzer,Do);

        delay(100); // her notanın çalma süresi

        noTone(buzzer);

        delay(50); // notalar arasındaki bekleme süresi

    }

    if (digitalRead(4)==1) {

        tone(buzzer,Re);

        delay(100); // her notanın çalma süresi

        noTone(buzzer);

        delay(50); // notalar arasındaki bekleme süresi

    }

    if (digitalRead(5)==1) {

        tone(buzzer,Mi);

        delay(100); // her notanın çalma süresi
```



```
noTone (buzzer) ;
delay (50) ;// notalar arasındaki bekleme süresi
}
if (digitalRead (6)==1) {
    tone (buzzer ,Fa) ;
    delay (100) ; // her notanın çalma süresi
    noTone (buzzer) ;
    delay (50) ;// notalar arasındaki bekleme süresi
}
if (digitalRead (7)==1) {
    tone (buzzer ,Sol) ;
    delay (100) ; // her notanın çalma süresi
    noTone (buzzer) ;
    delay (50) ;// notalar arasındaki bekleme süresi
}
if (digitalRead (8)==1) {
    tone (buzzer ,La) ;
    delay (100) ; // her notanın çalma süresi
    noTone (buzzer) ;
    delay (50) ;// notalar arasındaki bekleme süresi
}
if (digitalRead (9)==1) {
    tone (buzzer ,Si) ;
    delay (100) ; // her notanın çalma süresi
    noTone (buzzer) ;
    delay (50) ;// notalar arasındaki bekleme süresi
}
if (digitalRead (10)==1) {
    tone (buzzer ,Do2) ;
    delay (100) ; // her notanın çalma süresi
    noTone (buzzer) ;
    delay (50) ;// notalar arasındaki bekleme süresi
}
}
```



### Sıra Sizde

Görsel 2.74 veya Görsel 2.75'teki aktif buzzer uygulamasını breadboard üzerine kurunuz. Bu uygulamaya ait program kodlarını mikrodenetleyici IDE programında yazınız. Tüm işlemler bittikten sonra programı mikrodenetleyici karta yükleyip, devrenin çalışmasını gözlemleyiniz.

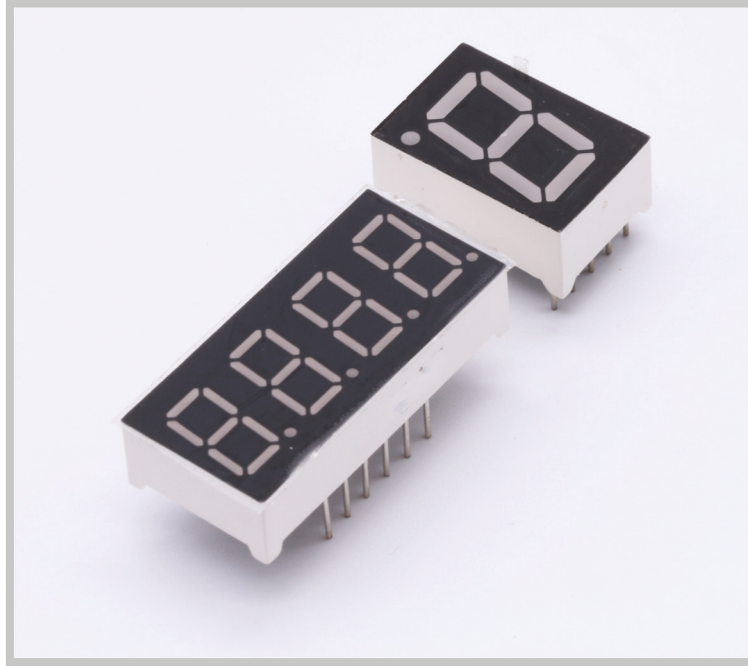


### Değerlendirme

Çalışmalarınız öğrenme biriminin sonunda yer alan Kontrol Listesi kullanılarak değerlendirilecektir. Çalışmalarınızı yaparken değerlendirme ölçütlerini dikkate alınız.

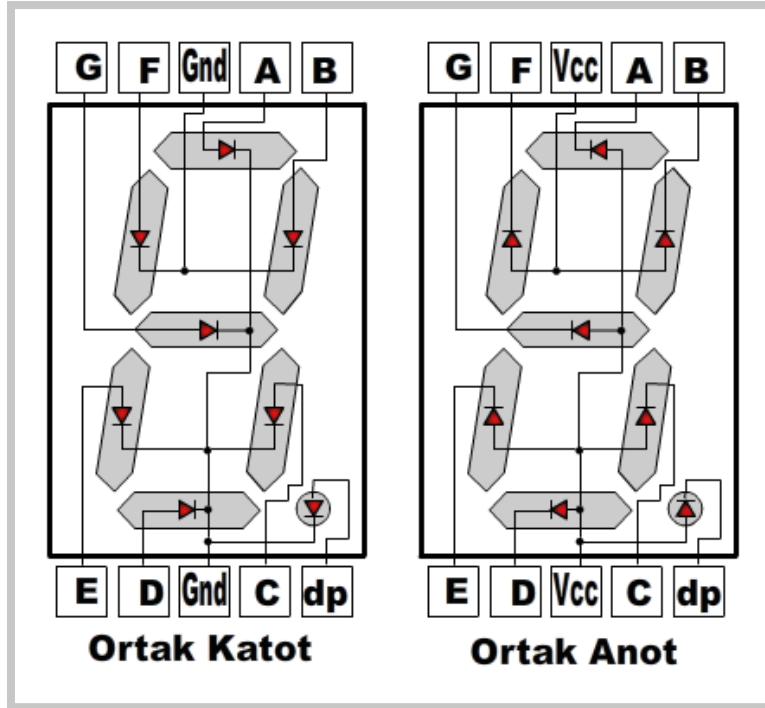
## 2.12. MİKRODENETLEYİCİ KART İLE 7 SEGMENT DİSPLAY UYGULAMALARI

İngilizcesi 7 Segment Display (7 Parçalı Gösterge) olan bu elektronik elemanlar ölçü aleti, saat, zamanlayıcı gibi elektronik devrelerde sayısal göstergeler için kullanılır. Görsel 2.76'da farklı büyüklük ve renklerde 7 Segment Display çeşitleri görülmektedir.



Görsel 2.76: 7 Segment display çeşitleri

Görsel 2.77'de görüldüğü gibi 7 tanesi sayıları göstermek, 1 tanesi de dip nokta için olmak üzere toplam 8 adet LED'den oluşur. İçinde bulunan LED'lerin ortak uçlarının bağlantı şekline göre ortak katot ve ortak anot olmak üzere iki çeşidi vardır.



Görsel 2.77: Ortak katot, ortak anot display içyapısı ve bağlantı uçları

Ortak katot display ile sayı değerlerini göstermek için öncelikle display ortak ucu, mikrodenetleyici kartın GND pinine bağlanmalıdır. Diğer display uçlarına da Tablo 2.5'te görüldüğü gibi mikrodenetleyici kartın portlarından 1 veya 0 vermek gerekir.

Ortak anot display ile sayı değerlerini göstermek için öncelikle display ortak ucu, mikrodenetleyici kartın 5V pinine bağlanmalıdır. Diğer display uçlarına da Tablo 2.6'da görüldüğü gibi mikrodenetleyici kartın portlarından 1 veya 0 vermek gerekir.

Tablo 2.5: Ortak Katot Display İçin Sayıları Gösteren LED Durumları

Gösterilecek Sayılar	Ortak katot display için LED durumları						
	a	b	c	d	e	f	g
0 sayısı	1	1	1	1	1	1	0
1 sayısı	0	1	1	0	0	0	0
2 sayısı	1	1	0	1	1	0	1
3 sayısı	1	1	1	1	0	0	1
4 sayısı	0	1	1	0	0	1	1
5 sayısı	1	0	1	1	0	1	1
6 sayısı	1	0	1	1	1	1	1
7 sayısı	1	1	1	0	0	0	0
8 sayısı	1	1	1	1	1	1	1
9 sayısı	1	1	1	1	0	1	1

Tablo 2.6: Ortak Anot Display İçin Sayıları Gösteren LED Durumları

Gösterilecek Sayılar	Ortak anot display için LED durumları						
	a	b	c	d	e	f	g
0 sayısı	0	0	0	0	0	0	1
1 sayısı	1	0	0	1	1	1	1
2 sayısı	0	0	1	0	0	1	0
3 sayısı	0	0	0	0	1	1	0
4 sayısı	1	0	0	1	1	0	0
5 sayısı	0	1	0	0	1	0	0
6 sayısı	0	1	0	0	0	0	0
7 sayısı	0	0	0	1	1	1	1
8 sayısı	0	0	0	0	0	0	0
9 sayısı	0	0	0	0	1	0	0

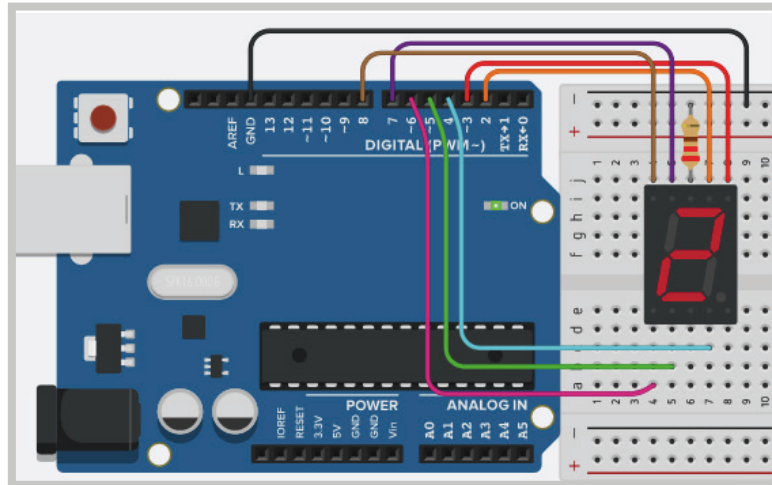
### 33. Uygulama

#### Gerekli Devre Elemanları

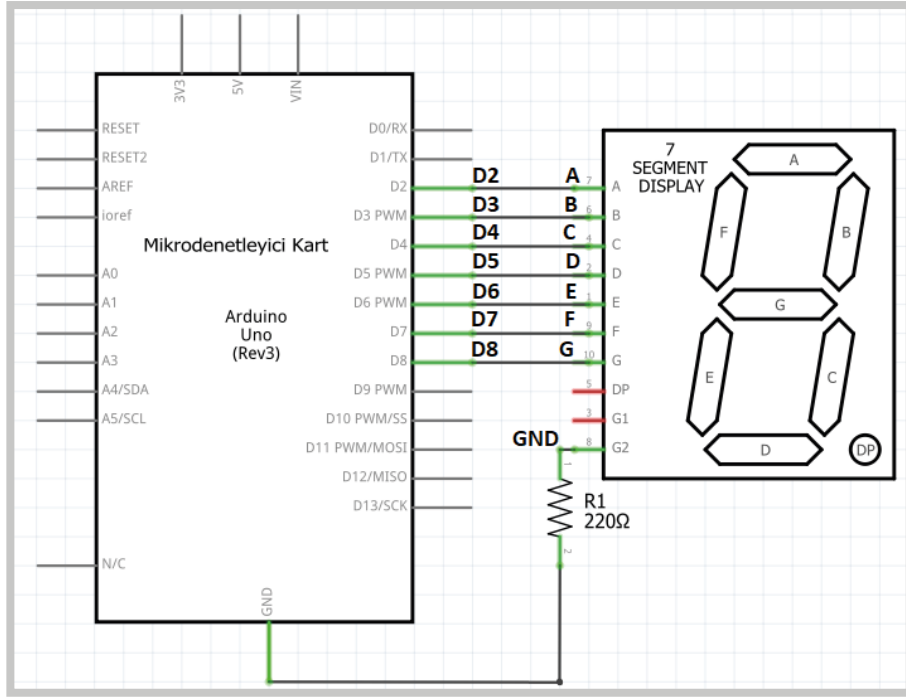
- 1 adet mikrodenetleyici kart
- 1 adet breadboard
- 1 adet 220Ω (kırmızı, kırmızı, kahverengi, altın rengi) direnç
- 1 adet ortak katot display
- Bağlantı kabloları

**1. Adım:** Görsel 2.78 ve Görsel 2.79'da ortak katot display uygulamasının devresi verilmiştir. Bu devrede ortak katot display kullanılmıştır.

- a, b, c, d, e, f, g bağlantı uçları sırasıyla mikrodenetleyici kartın D2, D3, D4, D5, D6, D7, D8 portlarına bağlanmıştır.
- Display ortak ucu ise 220Ω dirençle mikrodenetleyici kartın GND pinine bağlanmıştır.



Görsel 2.78: Ortak katot display uygulama devresi



Görsel 2.79: Ortak katot display uygulamasının elektriksel devresi

**2. Adım:** Ortak katot display uygulamasının aşağıdaki kodları incelendiğinde;

- Her bir sayı değerini display ekranında gösterebilmek için bir fonksiyon tanımlanmalıdır.

Fonksiyon tanımlama aşağıda gösterilmiştir.



- Fonksiyondan değer döndürülmeyecekse veri tipi olarak `void` kullanılır.
- Display ekranında her sayının gösterimi için oluşturulan fonksiyonlar `void loop()` ana fonksiyonu içinde çağrılır.
- Bu şekilde 1 saniye aralıklarla 0'dan 9'a kadar yukarı doğru sayan bir sayıcı elde edilir.

**3. Adım:** Ortak katot display uygulamasının program kodları aşağıdaki gibidir.

```
void setup () {
  for (int i=2; i<=8; i++) pinMode (i, OUTPUT);
}
```

```
void loop () {
  sifir (); delay (1000);
  bir (); delay (1000);
  iki (); delay (1000);
  uc (); delay (1000);
  dort (); delay (1000);
  bes (); delay (1000);
  alti (); delay (1000);
  yedi (); delay (1000);
  sekiz (); delay (1000);
  dokuz (); delay (1000);
}
```

```
}  
void sifir() {  
    digitalWrite(2, 1); //a  
    digitalWrite(3, 1); //b  
    digitalWrite(4, 1); //c  
    digitalWrite(5, 1); //d  
    digitalWrite(6, 1); //e  
    digitalWrite(7, 1); //f  
    digitalWrite(8, 0); //g  
}  
  
void bir() {  
  
    digitalWrite(2, 0); //a  
    digitalWrite(3, 1); //b  
    digitalWrite(4, 1); //c  
    digitalWrite(5, 0); //d  
    digitalWrite(6, 0); //e  
    digitalWrite(7, 0); //f  
    digitalWrite(8, 0); //g  
}  
  
void iki() {  
    digitalWrite(2, 1); //a  
    digitalWrite(3, 1); //b  
    digitalWrite(4, 0); //c  
    digitalWrite(5, 1); //d  
    digitalWrite(6, 1); //e  
    digitalWrite(7, 0); //f  
    digitalWrite(8, 1); //g  
}  
  
void uc() {  
  
    digitalWrite(2, 1); //a  
    digitalWrite(3, 1); //b  
    digitalWrite(4, 1); //c  
    digitalWrite(5, 1); //d  
    digitalWrite(6, 0); //e  
    digitalWrite(7, 0); //f  
    digitalWrite(8, 1); //g  
}  
  
void dort() {  
    digitalWrite(2, 0); //a  
    digitalWrite(3, 1); //b  
    digitalWrite(4, 1); //c  
    digitalWrite(5, 0); //d  
    digitalWrite(6, 0); //e  
    digitalWrite(7, 1); //f  
    digitalWrite(8, 1); //g
```

```
}  
void bes () {  
    digitalWrite (2, 1); //a  
    digitalWrite (3, 0); //b  
    digitalWrite (4, 1); //c  
    digitalWrite (5, 1); //d  
    digitalWrite (6, 0); //e  
    digitalWrite (7, 1); //f  
    digitalWrite (8, 1); //g  
  
}  
  
void alti () {  
    digitalWrite (2, 1); //a  
    digitalWrite (3, 0); //b  
    digitalWrite (4, 1); //c  
    digitalWrite (5, 1); //d  
    digitalWrite (6, 1); //e  
    digitalWrite (7, 1); //f  
    digitalWrite (8, 1); //g  
  
}  
  
void yedi () {  
    digitalWrite (2, 1); //a  
    digitalWrite (3, 1); //b  
    digitalWrite (4, 1); //c  
    digitalWrite (5, 0); //d  
    digitalWrite (6, 0); //e  
    digitalWrite (7, 0); //f  
    digitalWrite (8, 0); //g  
  
}  
  
void sekiz () {  
    digitalWrite (2, 1); //a  
    digitalWrite (3, 1); //b  
    digitalWrite (4, 1); //c  
    digitalWrite (5, 1); //d  
    digitalWrite (6, 1); //e  
    digitalWrite (7, 1); //f  
    digitalWrite (8, 1); //g  
  
}  
  
void dokuz () {  
    digitalWrite (2, 1); //a  
    digitalWrite (3, 1); //b  
    digitalWrite (4, 1); //c  
    digitalWrite (5, 1); //d  
    digitalWrite (6, 0); //e  
    digitalWrite (7, 1); //f  
    digitalWrite (8, 1); //g  
}
```



### Sıra Sizde

Görsel 2.78 veya Görsel 2.79'daki ortak katot display uygulamasını breadboard üzerine kurunuz. Bu uygulamaya ait program kodlarını mikrodenetleyici IDE programında 0'dan 9'a kadar yukarı sayacak şekilde yazınız. Tüm işlemler bittikten sonra programı mikrodenetleyici karta yükleyip, devrenin çalışmasını gözlemleyiniz.



### Değerlendirme

Çalışmalarınız öğrenme biriminin sonunda yer alan Kontrol Listesi kullanılarak değerlendirilecektir. Çalışmalarınızı yaparken değerlendirme ölçütlerini dikkate alınız.



### Sıra Sizde

Görsel 2.78 veya Görsel 2.79'daki ortak katot display uygulamasını breadboard üzerine kurunuz. Bu uygulamaya ait program kodlarını mikrodenetleyici IDE programında 9'dan 0'a kadar aşağı sayacak şekilde yazınız. Tüm işlemler bittikten sonra programı mikrodenetleyici karta yükleyip, devrenin çalışmasını gözlemleyiniz.



### Değerlendirme

Çalışmalarınız öğrenme biriminin sonunda yer alan Kontrol Listesi kullanılarak değerlendirilecektir. Çalışmalarınızı yaparken değerlendirme ölçütlerini dikkate alınız.



### 34. Uygulama

Bu uygulamadaki amaç, ortak katot display ile yukarı aşağı sayıcı uygulamasını gerçekleştirmektir.

#### Gerekli Devre Elemanları

- 1 adet mikrodenetleyici kart
- 1 adet breadboard
- 1 adet 220Ω (kırmızı, kırmızı, kahverengi, altın rengi) direnç
- 1 adet ortak katot display
- 1 adet anahtar
- Bağlantı kabloları

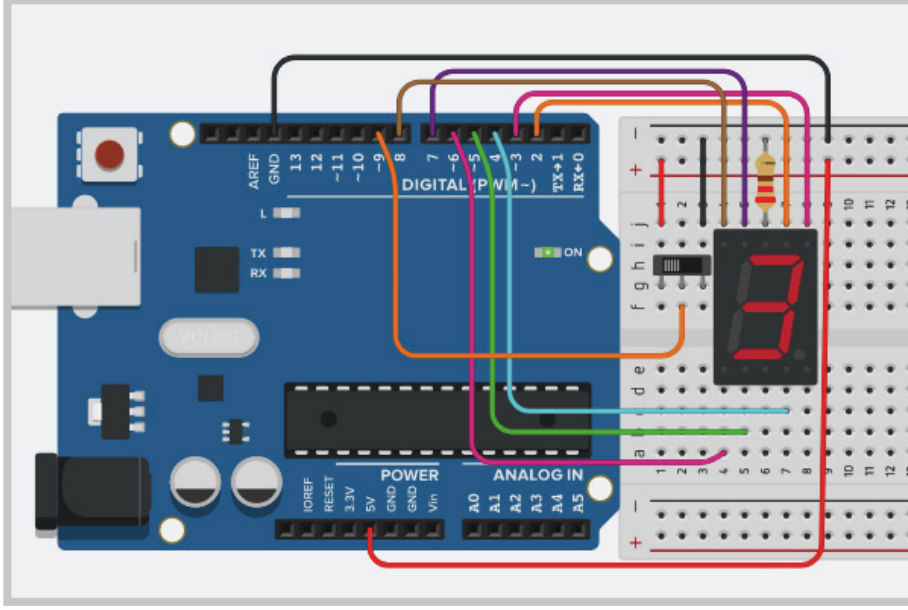
**1. Adım:** Görsel 2.80 ve Görsel 2.81'de ortak katot display ile bir anahtara bağlı olarak yukarı veya aşağı sayma işlemi yapan uygulamanın devresi görülmektedir.

- Bu devrede ortak katot display kullanılmış olup bağlantı uçları a, b, c, d, e, f, g sırasıyla mikrodenetleyici kartın D2, D3, D4, D5, D6, D7, D8 portlarına bağlanmıştır.
- Display ortak ucu ise 220Ω dirençle mikrodenetleyici kartın GND pinine bağlanmıştır.
- Ayrıca mikrodenetleyici kartın D9 portuna da bir anahtarın ortak ucu bağlanmıştır.
- Anahtarın sağda kalan ucu mikrodenetleyici kartın 5V pinine, solda kalan ucu GND pinine bağlanmıştır.

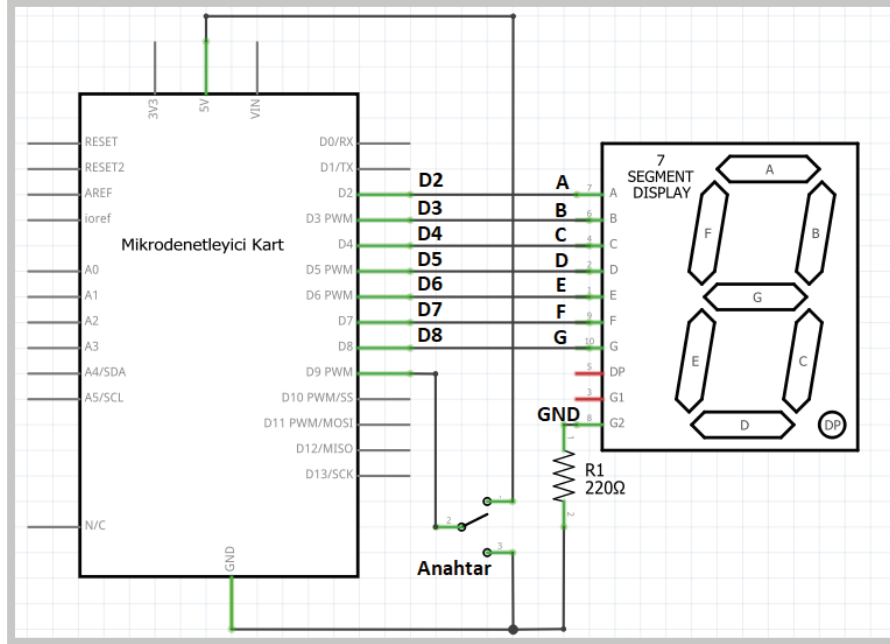


**2. Adım:** Bu uygulamanın program kodları incelendiğinde her bir sayı değerini display ekranında gösterebilmek için bir fonksiyon tanımlanmasının yanında yukarı ve aşağı sayması içinde ayrı bir fonksiyon tanımlanmıştır.

- Programda yukarı veya aşağı sayması, `if (digitalRead(9)==1) yukarisay();` komutuyla anahtar durumunu okuyan porta göre belirlenmektedir.
- Anahtar sağa çekildiğinde mikrodenetleyici kart D9 portunu 1 olarak algıladığı için yukarı sayacak, sola çekildiğinde ise D9 portunu 0 olarak algıladığı için aşağı sayacaktır.



GörSEL 2.80: Ortak katot display ile yukarı aşağı sayıcı uygulama devresi



GörSEL 2.81: Ortak katot display ile yukarı aşağı sayıcı uygulamasının elektriksel devresi

**3. Adım:** Ortak katot display ile yukarı aşağı sayıcı uygulamasının program kodları aşağıdaki gibidir.

```
void setup() {
  for (int i=2;i<=8;i++) pinMode(i, OUTPUT);

  pinMode(9, INPUT); //1 ileri 0 geri seçme anahtarı
}

void loop() {

  if (digitalRead(9)==1) yukarisay();
  if (digitalRead(9)==0) asagisay();
}

void yukarisay() {
  sifir(); delay(1000);
  bir(); delay(1000);
  iki(); delay(1000);
  uc(); delay(1000);
  dort(); delay(1000);
  bes(); delay(1000);
  alti(); delay(1000);
  yedi(); delay(1000);
  sekiz(); delay(1000);
  dokuz(); delay(1000);

}

void asagisay() {
  dokuz(); delay(1000);
  sekiz(); delay(1000);
  yedi(); delay(1000);
  alti(); delay(1000);
  bes(); delay(1000);
  dort(); delay(1000);
  uc(); delay(1000);
  iki(); delay(1000);
  bir(); delay(1000);
  sifir(); delay(1000);
}

void sifir() {
  digitalWrite(2, 1); //a
  digitalWrite(3, 1); //b
  digitalWrite(4, 1); //c
  digitalWrite(5, 1); //d
  digitalWrite(6, 1); //e
  digitalWrite(7, 1); //f
  digitalWrite(8, 0); //g
}
```

```
}  
  
void bir() {  
  
    digitalWrite(2, 0); //a  
    digitalWrite(3, 1); //b  
    digitalWrite(4, 1); //c  
    digitalWrite(5, 0); //d  
    digitalWrite(6, 0); //e  
    digitalWrite(7, 0); //f  
    digitalWrite(8, 0); //g  
  
}  
  
void iki() {  
    digitalWrite(2, 1); //a  
    digitalWrite(3, 1); //b  
    digitalWrite(4, 0); //c  
    digitalWrite(5, 1); //d  
    digitalWrite(6, 1); //e  
    digitalWrite(7, 0); //f  
    digitalWrite(8, 1); //g  
  
}  
  
void uc() {  
  
    digitalWrite(2, 1); //a  
    digitalWrite(3, 1); //b  
    digitalWrite(4, 1); //c  
    digitalWrite(5, 1); //d  
    digitalWrite(6, 0); //e  
    digitalWrite(7, 0); //f  
    digitalWrite(8, 1); //g  
  
}  
void dort() {  
    digitalWrite(2, 0); //a  
    digitalWrite(3, 1); //b  
    digitalWrite(4, 1); //c  
    digitalWrite(5, 0); //d  
    digitalWrite(6, 0); //e  
    digitalWrite(7, 1); //f  
    digitalWrite(8, 1); //g  
  
}  
void bes() {  
    digitalWrite(2, 1); //a
```

```
digitalWrite(3, 0); //b
digitalWrite(4, 1); //c
digitalWrite(5, 1); //d
digitalWrite(6, 0); //e
digitalWrite(7, 1); //f
digitalWrite(8, 1); //g

}

void alti () {
digitalWrite(2, 1); //a
digitalWrite(3, 0); //b
digitalWrite(4, 1); //c
digitalWrite(5, 1); //d
digitalWrite(6, 1); //e
digitalWrite(7, 1); //f
digitalWrite(8, 1); //g

}

void yedi () {
digitalWrite(2, 1); //a
digitalWrite(3, 1); //b
digitalWrite(4, 1); //c
digitalWrite(5, 0); //d
digitalWrite(6, 0); //e
digitalWrite(7, 0); //f
digitalWrite(8, 0); //g

}

void sekiz () {
digitalWrite(2, 1); //a
digitalWrite(3, 1); //b
digitalWrite(4, 1); //c
digitalWrite(5, 1); //d
digitalWrite(6, 1); //e
digitalWrite(7, 1); //f
digitalWrite(8, 1); //g

}

void dokuz () {
digitalWrite(2, 1); //a
digitalWrite(3, 1); //b
digitalWrite(4, 1); //c
digitalWrite(5, 1); //d
digitalWrite(6, 0); //e
digitalWrite(7, 1); //f
digitalWrite(8, 1); //g

}
```



### Sıra Sizde

Görsel 2.80 veya Görsel 2.81'deki ortak katot display ile yukarı aşağı sayıcı uygulamasını breadboard üzerine kurunuz. Bu uygulamaya ait program kodlarını mikrodenetleyici IDE programında 9'dan 0'a kadar aşağı sayacak şekilde yazınız. Tüm işlemler bittikten sonra programı mikrodenetleyici karta yükleyip, devrenin çalışmasını gözlemleyiniz.



<http://kitap.eba.gov.tr/KodSor.php?KOD=20792>



### Değerlendirme

Çalışmalarınız öğrenme biriminin sonunda yer alan Kontrol Listesi kullanılarak değerlendirilecektir. Çalışmalarınızı yaparken değerlendirme ölçütlerini dikkate alınız.



### 35. Uygulama

Bu uygulamadaki amaç, dizi kullanarak kaldığı yerden devam eden ileri geri sayıcı uygulamayı oluşturmaktır.

#### Gerekli Devre Elemanları

- 1 adet mikrodenetleyici kart
- 1 adet breadboard
- 1 adet 220Ω (kırmızı, kırmızı, kahverengi, altın rengi) direnç
- 1 adet ortak katot display
- Bağlantı kabloları

**1. Adım:** Görsel 2.80 ve Görsel 2.81'deki ortak katot display ile yukarı aşağı sayıcı uygulamasında anahtar sağa çekilip sonra sola çekildiğinde normalde önce yukarı sayarken kaldığı yerden aşağı saymaya devam etmesi gerekir. Ancak yukarı sayma işlemi bitmeden aşağı saymaya başlamaz. Bu sorunu çözmek ve program kodlarını kısaltmak için dizi kullanılabilir.

**2. Adım:** Aşağıda dizi kullanılarak kaldığı yerden devam eden ileri geri sayıcı program kodları incelendiğinde,

- Display ekranında sayıların gösterilmesi için gerekli display LED durumlarının `say[10][7]` dizisi içinde tanımlanması gerekir.
- `say[10][7]` dizisi içindeki "10" değeri 0 ile 9'a kadar olan rakamları, "7" sayısı ise display LED durumlarını ifade eder.
- Programda `int s=digitalRead(9)` ; komutu ile D9 portuna bağlı anahtardan okunan değer "s" değişkenine aktarılır.
- `if (s==1) i++` ; komutu ile eğer anahtar 1 konumunda ise 1 saniye aralıklı olarak yukarı saydırmaya, `if (s==0)i--` ; komutu ile de 1 saniye aralıklı olarak kaldığı yerden aşağı saydırmaya başlar.
- `if (i>9) i=0` ; komutu artırma işleminde "i" değişkeni son durumunun dokuzdan büyük olduğu durumlar için kullanılır.
- `if (i<0) i=9` ; komutu azaltma işleminde "i" değişkeninin son durumunun sıfırdan küçük olmaması için kullanılır.

**3. Adım:** Dizi kullanarak kaldığı yerden devam eden ileri geri sayıcı program kodları şunlardır:

```
int i=0;
int say[10][7]={
1,1,1,1,1,1,0,
0,1,1,0,0,0,0,
1,1,0,1,1,0,1,
1,1,1,1,0,0,1,
0,1,1,0,0,1,1,
1,0,1,1,0,1,1,
1,0,1,1,1,1,1,
1,1,1,0,0,0,0,
1,1,1,1,1,1,1,
1,1,1,1,0,1,1,
};
void setup() {
for (int p=2;p<=8;p++) pinMode(p, OUTPUT);

pinMode(9, INPUT); //1 ileri 0 geri seçme anahtarı
}

void loop() {

int s=digitalRead(9);
digitalWrite(2, say[i][0]); //a
digitalWrite(3, say[i][1]); //b
digitalWrite(4, say[i][2]); //c
digitalWrite(5, say[i][3]); //d
digitalWrite(6, say[i][4]); //e
digitalWrite(7, say[i][5]); //f
digitalWrite(8, say[i][6]); //g
delay(1000);
if (s==1) i++;
if (s==0) i--;
if (i>9) i=0; if (i<0) i=9;
}
```

### Sıra Sizde

Görsel 2.80 veya Görsel 2.81'deki devreyi breadboard üzerine kurunuz. Ortak katot display ile dizi kullanarak yukarı aşağı sayıcı uygulamasına ait program kodlarını mikrodenetleyici IDE programında yazınız. Tüm işlemler bittikten sonra programı mikrodenetleyici karta yükleyip, devrenin çalışmasını gözlemleyiniz.

### Değerlendirme

Çalışmalarınız öğrenme biriminin sonunda yer alan Kontrol Listesi kullanılarak değerlendirilecektir. Çalışmalarınızı yaparken değerlendirme ölçütlerini dikkate alınız.

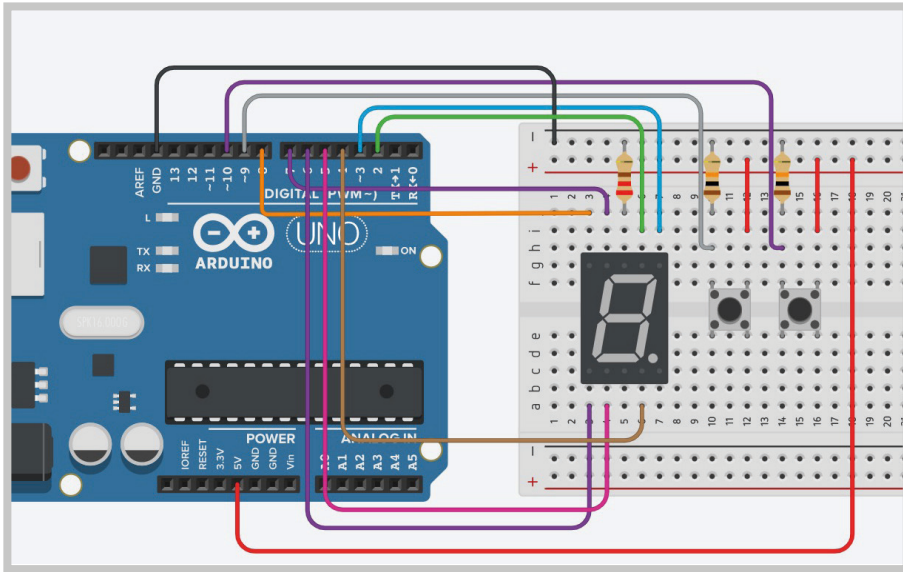
### 36. Uygulama

Bu uygulamadaki amaç, butonlar ile yukarı aşağı sayıcı uygulamasını gerçekleştirmektir.

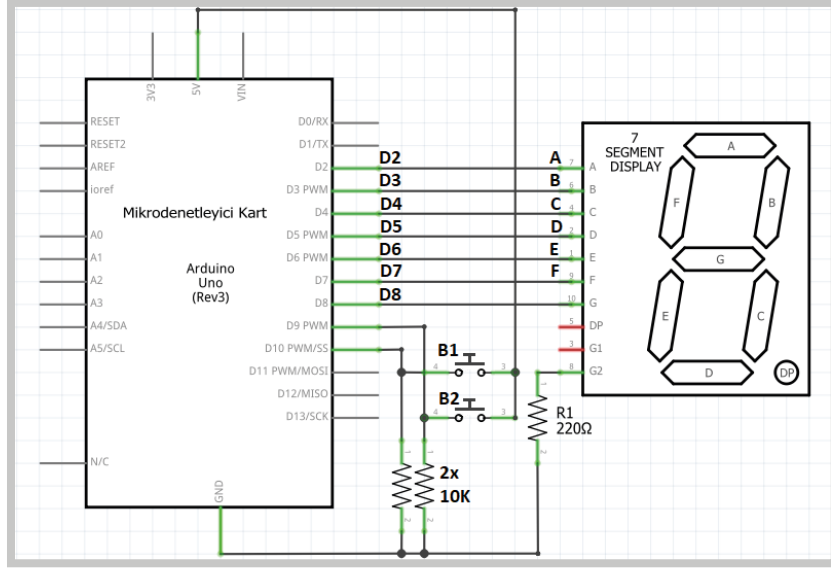
#### Gerekli Devre Elemanları

- 1 adet mikrodenetleyici kart
- 1 adet breadboard
- 1 adet 220Ω (kırmızı, kırmızı, kahverengi, altın rengi) direnç
- 2 adet 10KΩ (kahverengi, siyah, sarı, altın rengi) direnç
- 1 adet ortak katot display
- 2 adet buton
- Bağlantı kabloları

**1. Adım:** Görsel 2.82 ve Görsel 2.83'te butonlar ile yukarı aşağı sayıcı uygulamasının breadbord devresi ve elektriksel devresi verilmiştir.



Görsel 2.82: Buton ile yukarı aşağı sayıcı uygulama devresi



Görsel 2.83: Butonlar ile yukarı aşağı sayıcı uygulamasının elektriksel devresi

**2. Adım:** Butonlar ile yukarı aşağı sayıcı uygulamasının program kodları aşağıda gösterildiği gibidir.

```

int i=0;
int say[10][7]={
  1,1,1,1,1,1,0,
  0,1,1,0,0,0,0,
  1,1,0,1,1,0,1,
  1,1,1,1,0,0,1,
  0,1,1,0,0,1,1,
  1,0,1,1,0,1,1,
  1,0,1,1,1,1,1,
  1,1,1,0,0,0,0,
  1,1,1,1,1,1,1,
  1,1,1,1,0,1,1,
};
void setup() {
  for (int p=2;p<=8;p++) pinMode(p, OUTPUT);
  pinMode(9, INPUT);
  pinMode(10, INPUT);
}
void loop() {
  digitalWrite(2, say[i][0]); //a
  digitalWrite(3, say[i][1]); //b
  digitalWrite(4, say[i][2]); //c
  digitalWrite(5, say[i][3]); //d
  digitalWrite(6, say[i][4]); //e
  digitalWrite(7, say[i][5]); //f
  digitalWrite(8, say[i][6]); //g
  delay(500);
  if (digitalRead(9)==1) i++;
  if (digitalRead(10)==1) i--;
  if (i>9) i=0;
  if (i<0) i=9;
}

```





### Sıra Sizde

Görsel 2.82 veya Görsel 2.83'teki buton ile yukarı aşağı sayıcı uygulamasını breadboard üzerine kurunuz. Bu uygulamaya ait program kodlarını mikrodenetleyici IDE programında yazınız. Tüm işlemler bittikten sonra programı mikrodenetleyici karta yükleyip, devrenin çalışmasını gözlemleyiniz.

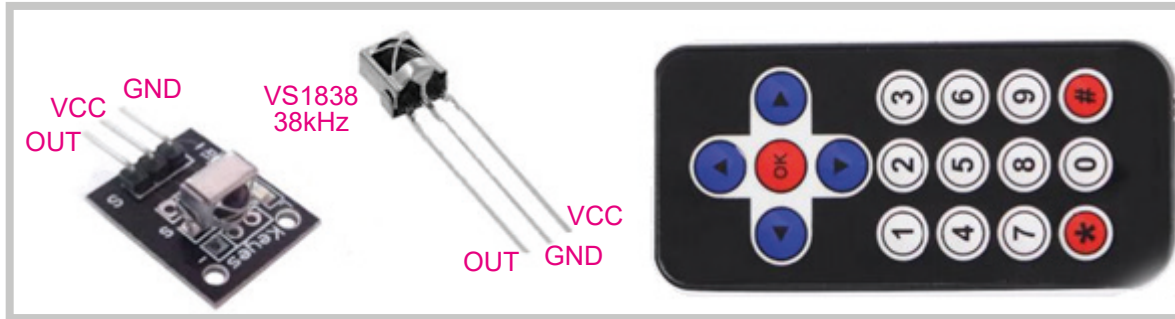


### Değerlendirme

Çalışmalarınız öğrenme biriminin sonunda yer alan Kontrol Listesi kullanılarak değerlendirilecektir. Çalışmalarınızı yaparken değerlendirme ölçütlerini dikkate alınız.

## 2.13. MİKRODENETLEYİCİ KART İLE IR ALICI UYGULAMALARI

IR, İngilizce Infra Red (kızılötesi) kelimelerinin kısaltmasından oluşmuştur. Alıcı-verici devrelerinde kullanılan bu teknoloji, insan gözünün göremediği kızılötesi sinyal gönderen bir LED yardımıyla verinin gönderilmesi, karşı taraftan da foto diyot benzeri bir devre elemanı vasıtasıyla verinin dijital bilgiye dönüştürülmesi mantığına dayanır. Uzaktan kumandalar, Görsel 2.84 c'de görüldüğü gibi üzerinde bulunan tuş kombinasyonuna bağlı olarak belirli kodları kızılötesi LED yardımıyla karşı tarafa gönderir. Uzaktan kumandalar, günlük yaşantıda TV, uydu alıcısı, müzik sistemi gibi birçok elektronik aygıtı kontrol etmek için kullanılır.



a) Uzaktan kumanda alıcısı devresi

b) Uzaktan kumanda alıcısı devre elemanı

c) Uzaktan kumanda

**Görsel 2.84: Uzaktan kumanda alıcısı ve genel amaçlı uzaktan kumanda**

Uzaktan kumandanın gönderdiği sinyalleri çözen devre elemanı Görsel 2.84 a ve b'de görülmektedir. IR alıcı ve kod çözücü devre elemanı, Görsel 2.84 a'da gösterilen şekilde elektronik kart üzerine monte edilebileceği gibi Görsel 2.84 b'deki gibi kart üzerine monte edilmeden de kullanılabilir.

### 2.13.1. Uzaktan Kumandanın Kodunun Çözülmesi

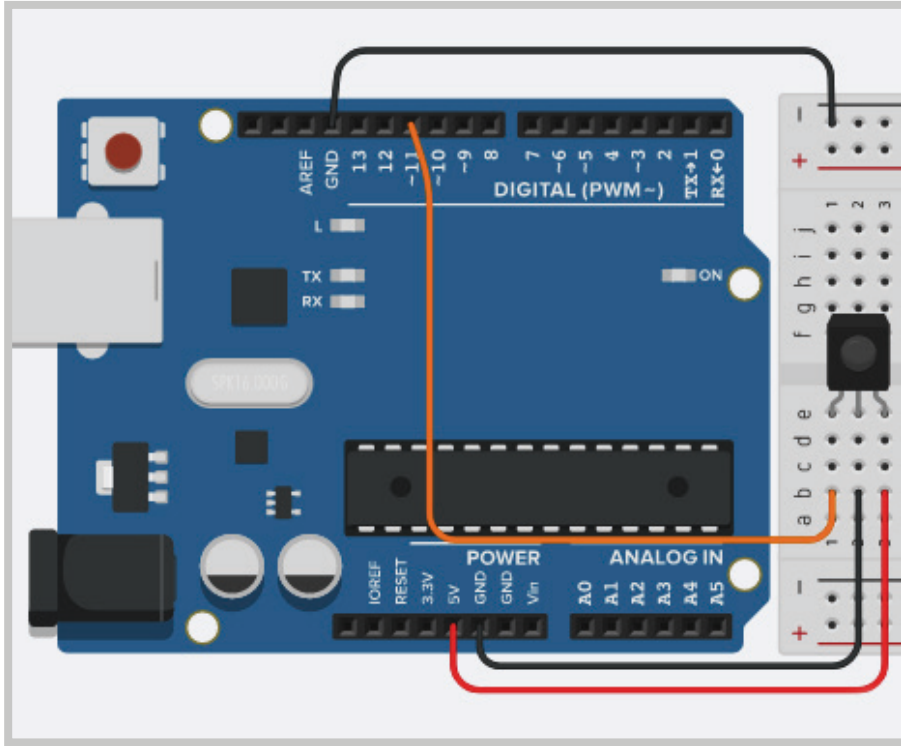
#### 37. Uygulama

Bu uygulamadaki amaç, IR alıcı ile kod çözücü uygulamasını “Bilgisayarlı Çizim Programı”nda gerçekleştirmektir.

#### Gerekli Devre Elemanları

- 1 adet mikrodenetleyici kart
- 1 adet breadboard
- 1 adet IR alıcı
- 1 adet IR verici kumanda
- Bağlantı kabloları

**1. Adım:** IR alıcı ve kod çözücü devre elemanı Görsel 2.85’te görüldüğü gibi breadboard üzerine monte edilir. Bağlantı uçları yapılırken “Vcc” ucu mikrodenetleyici üzerindeki “5V” pinine, “GND” ucu mikrodenetleyici üzerindeki “GND” pinine, “Out” ucu mikrodenetleyici kart üzerindeki “D11” pinine bağlanır.



Görsel 2.85: IR alıcı ve kod çözücü uygulama devresi



Görsel 2.86: Elektronik devre simülasyon programındaki uzaktan kumanda ve tuş kodları

**2. Adım:** On-line elektronik devre simülasyon programında Görsel 2.85'teki gibi kurulan devrenin kod bölümüne uygulamayla ilgili kodlar yazılmalıdır. Programda çalışma alanına Görsel 2.86'daki uzaktan kumanda eklenmelidir. "Simülasyonu Başlat" butonu ile devre çalıştırılıp uzaktan kumandanın tuşlarına basıldığında tuşlara ait kodlar seri port ekranında görülmelidir.

**3. Adım: IR Alıcı** Kod çözücü uygulamasının program kodları aşağıda gösterildiği gibidir.

```
#include <IRremote.h>

int RECV_PIN = 11;
IRrecv irrecv(RECV_PIN);
decode_results results;
void setup()
{
  Serial.begin(9600);
  irrecv.enableIRIn();
}

void loop() {
  if (irrecv.decode(&results)) {
    Serial.println(results.value, HEX);
    irrecv.resume();
  }
  delay(100);
}
```



### Sıra Sizde

Görsel 2.85'teki IR alıcı ve kod çözücü uygulamasını elektronik devre simülatör programında kurunuz. Bu uygulamaya ait program kodlarını elektronik devre simülatör programında “Kod” bölümüne yazınız. Tüm işlemler bittikten sonra “Simülasyonu Başlat” butonu ile devreyi çalıştırıp uzaktan kumandanın tuşlarına basarak tuşlara ait kodları seri port ekranında gözlemleyiniz. Tablo 2.7’de eksik bırakılan uzaktan kumandanın tuş kodları tamamlayınız.



### Değerlendirme

Çalışmalarınız öğrenme biriminin sonunda yer alan Kontrol Listesi kullanılarak değerlendirilecektir. Çalışmalarınızı yaparken değerlendirme ölçütlerini dikkate alınız.

Tablo 2.7: Elektronik Devre Simülatör Programındaki Uzaktan Kumandanın Tuş Kodları

Tuş simgesi	Kodu	Açıklaması
	FD00FF	Açma – kapama fonksiyonu
		Ses artırma fonksiyonu
		Ses azaltma fonksiyonu
		Özel fonksiyonlar
		Geri sardır fonksiyonu
		İleri sardır fonksiyonu
		Durdur fonksiyonu
		Yukarı fonksiyonu
		Aşağı fonksiyonu
		Özel fonksiyonlar
		Özel fonksiyonlar
		0 fonksiyonu
		1 fonksiyonu
		2 fonksiyonu
		3 fonksiyonu
		4 fonksiyonu
		5 fonksiyonu
		6 fonksiyonu
		7 fonksiyonu
		8 fonksiyonu
		9 fonksiyonu

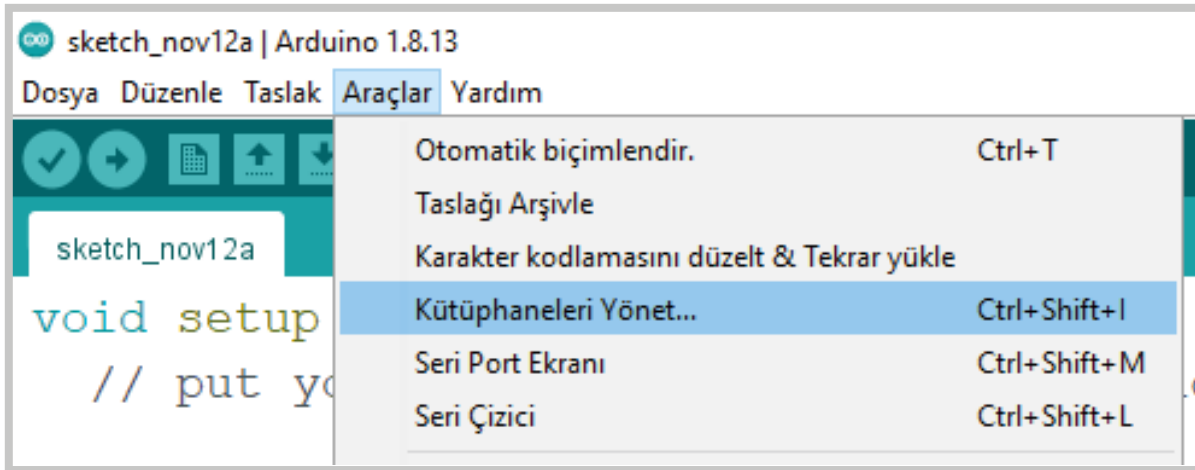
### 38. Uygulama

Bu uygulamadaki amaç, IR alıcı ile kod çözücü uygulamasını mikrodenetleyici IDE programında gerçekleştirmektir.

#### Gerekli Devre Elemanları

- 1 adet mikrodenetleyici kart
- 1 adet breadboard
- 1 adet IR alıcı
- 1 adet IR verici kumanda
- Bağlantı kabloları

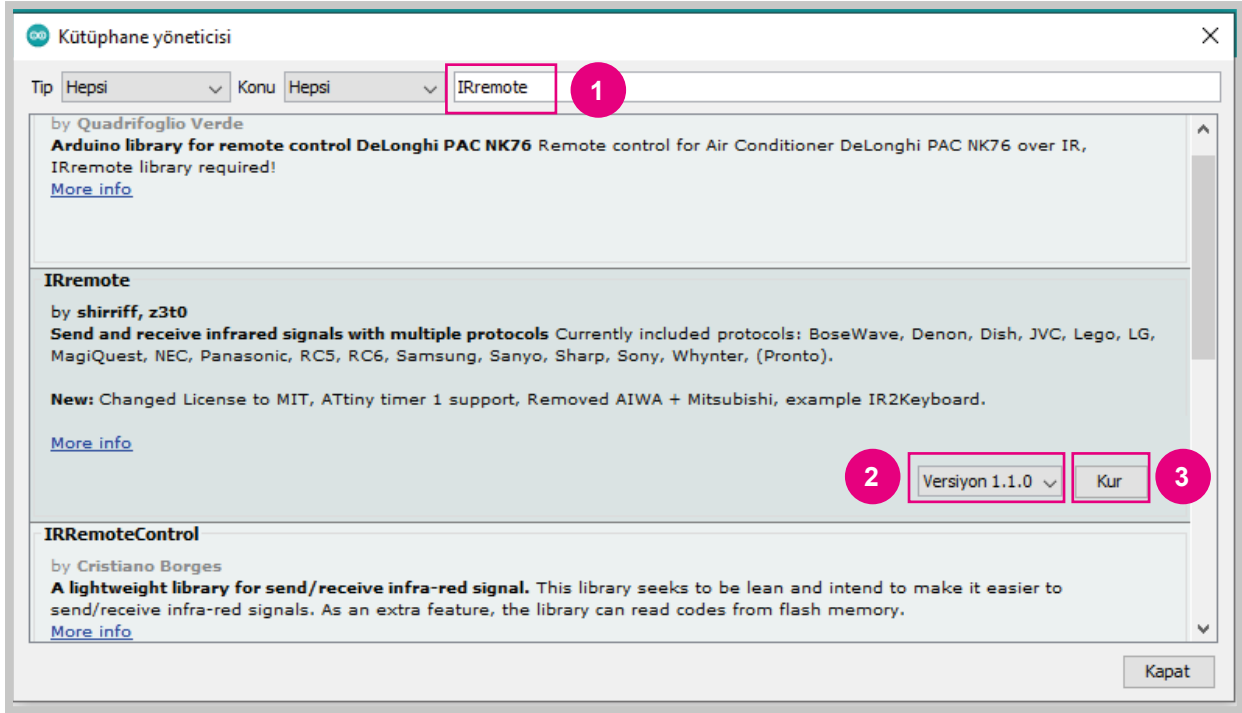
**1. Adım:** Bilgisayarlı çizim programında yapılan uygulamayı, mikrodenetleyici IDE programında yapabilmek için IR alıcı ve kod çözücü devre elemanının bağlantıları fiziksel olarak breadboard üzerinde kurulmalıdır. Sonraki aşamada “**IRremote**” kütüphanesinin mikrodenetleyici IDE programında eklenmesi için Görsel 2.87’deki gibi mikrodenetleyici IDE programında **Araçlar** menüsünden “Kütüphaneleri Yönet” penceresi açılmalıdır.



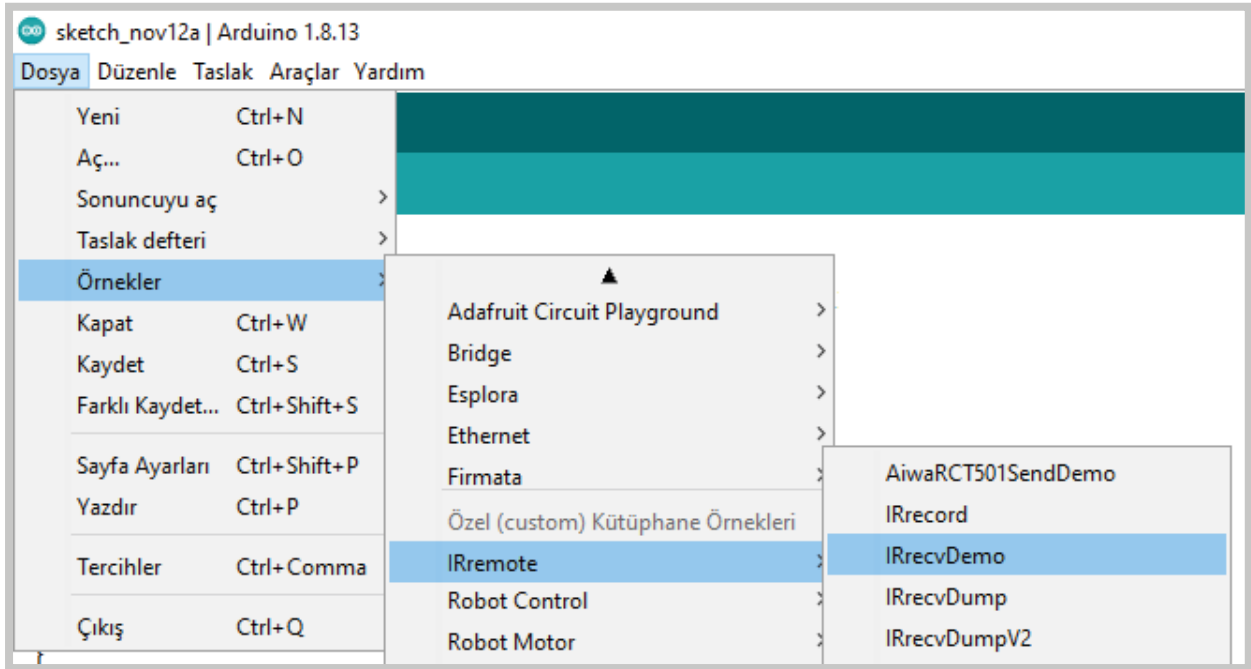
Görsel 2.87: Mikrodenetleyici IDE programında Kütüphaneleri Yönet penceresinin açılması

**2. Adım:** Görsel 2.88’deki kütüphane yöneticisi penceresinde “1” numarayla gösterilen arama alanına kütüphane isminin girilmelidir. IRremote kütüphanesi için farklı kütüphane versiyonları mevcuttur. Bu bölümde kullanılacak olan 1.1.0 versiyonu, daha basit ve kodları anlaşılır düzeyde olan bir versiyondur. Kütüphane versiyonu da seçildikten sonra **kur** butonuna basılması gerekir. Kurma işlemi bittikten sonra Görsel 2.89’daki gibi **IRrecvDemo** programı açılır.

## 2. Öğrenme Birimi: Mikrodenetleyici Kart Programlama

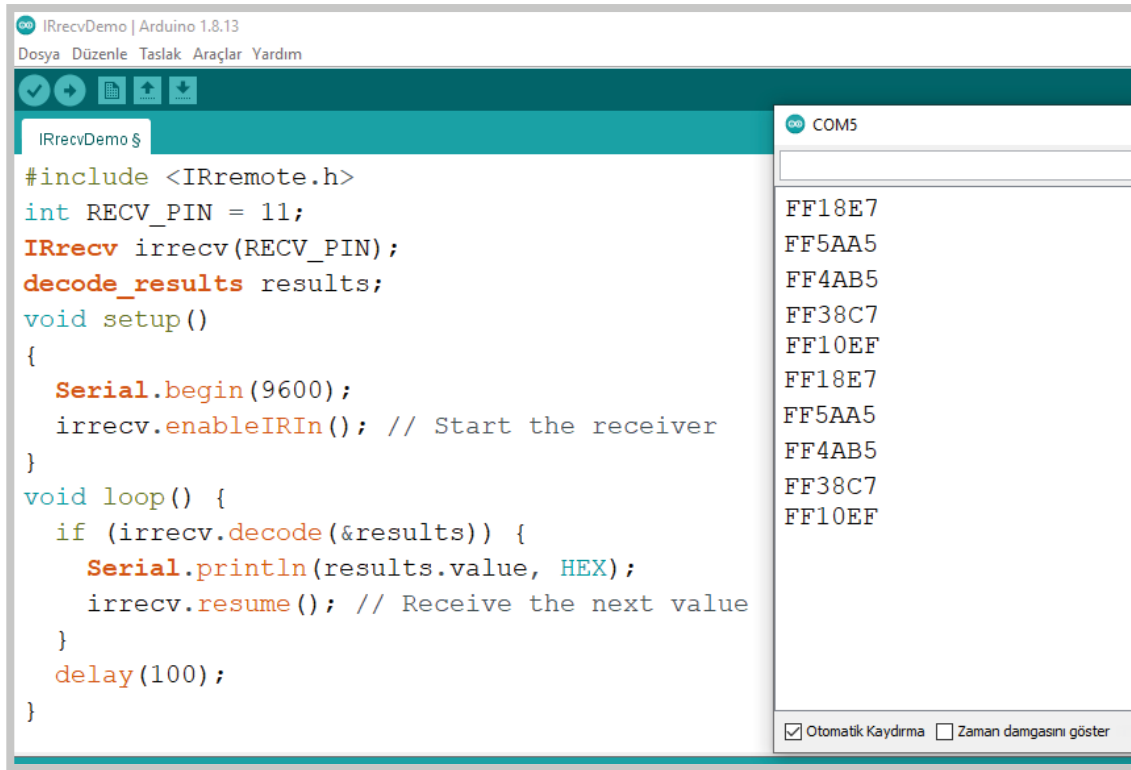


Görsel 2.88: Kütüphane yöneticisi penceresinde program kütüphanesinin aranması



Görsel 2.89: IRremote kütüphanesindeki örnek programlar

**3. Adım:** IRrecvDemo programı açıldıktan sonra program üzerinde hiçbir değişiklik yapılmadan mikrodenetleyici karta yüklenmelidir. Uzaktan kumandanın IR alıcıya gönderdiği kodları görebilmek için "Seri Port" ekranı açılmalıdır. Elektronik devre simülatör programındaki uzaktan kumandaya benzer herhangi bir cihaz kumandasının tuşlarına basıldığında seri port ekranında Görsel 2.90'da görüldüğü gibi kodlar karşınıza gelir.



```

IRrecvDemo | Arduino 1.8.13
Dosya Düzenle Taslak Araçlar Yardım

IRrecvDemo §
#include <IRremote.h>
int RECV_PIN = 11;
IRrecv irrecv(RECV_PIN);
decode_results results;
void setup()
{
  Serial.begin(9600);
  irrecv.enableIRIn(); // Start the receiver
}
void loop() {
  if (irrecv.decode(&results)) {
    Serial.println(results.value, HEX);
    irrecv.resume(); // Receive the next value
  }
  delay(100);
}
COM5
FF18E7
FF5AA5
FF4AB5
FF38C7
FF10EF
FF18E7
FF5AA5
FF4AB5
FF38C7
FF10EF
 Otomatik Kaydırma  Zaman damgasını göster

```

Görsel 2.90: IRrecvDemo programıyla kumanda kodlarının çözülmesi



## Sıra Sizde

Görsel 2.85'teki devreyi breadboard üzerine kurunuz. IR alıcı ve kod çözücü uygulamasının program kodlarını mikrodenetleyici IDE programında yazınız. Tüm işlemler bittikten sonra programı mikrodenetleyici karta yükleyip, seri port ekranını açınız. Çalışır vaziyette temin edilen herhangi bir uzaktan kumandanın tuşlarına basarak tuşlara ait kodları seri port ekranında gözlemleyiniz. Elinizdeki kumanda için Tablo 2.7'dekine benzer bir tablo oluşturarak uzaktan kumandanın tuş kodlarını yazınız.



## Değerlendirme

Çalışmalarınız öğrenme biriminin sonunda yer alan Kontrol Listesi kullanılarak değerlendirilecektir. Çalışmalarınızı yaparken değerlendirme ölçütlerini dikkate alınız.

## 2.13.2. RGB LED ile Uzaktan Kumandanın Kullanımı



## 39. Uygulama

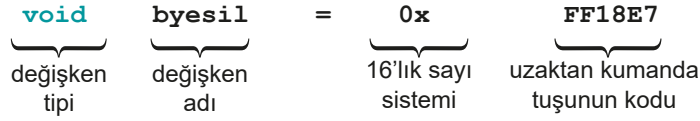
Uzaktan kumandanın kodları çözüldükten sonra her tuşa bir fonksiyon atanıp farklı devreler çalıştırılabilir. Görsel 2.92 ve Görsel 2.93'te IR alıcı ve kod çözücü ile RGB LED uygulaması verilmiştir. Bu uygulamanın amacı, uzaktan kumandanın belirli tuşlarıyla RGB LED'i farklı renklerde yakmaktır.

## 2. Öğrenme Birimi: Mikrodenetleyici Kart Programlama

### Gerekli Devre Elemanları

- 1 adet mikrodenetleyici kart
- 1 adet breadboard
- 1 adet IR alıcı
- 1 adet RGB LED
- 3 adet 220Ω (kırmızı, kırmızı, kahverengi, altın rengi) direnç
- 1 adet IR verici kumanda
- Bağlantı kabloları

**1. Adım:** Uzaktan kumandadan elde edilen kodlar kullanılırken, Görsel 2.91’de olduğu gibi “long” tipinde değişken tanımlanarak yazılmalıdır. Heksadesimal sayıları belirtmek için ise başına “0x” getirerek yazmak gerekir.



Görsel 2.91: IRrecvDemo programında uzaktan kumanda tuş kodlarının kullanılması

**2. Adım:** Son durumda **IRrecvDemo** örnek programını değiştirirken aşağıdaki program kodları eklenir.

```
long bkirmizi = 0xFD08F7; // Kırmızı renk için kullanılan U.K. kodu
long byesil = 0xFD8877; // Yeşil renk için kullanılan U.K. kodu
long bmavi = 0xFD48B7; // Mavi renk için kullanılan U.K. kodu
long bsari = 0xFD28D7; // Sarı renk için kullanılan U.K. kodu
long bmor = 0xFDA857; // Mor renk için kullanılan U.K. kodu
long bturkuaz = 0xFD6897; // Turkuaz renk için kullanılan U.K. kodu
long bbeyaz = 0xFD18E7; // Beyaz renk için kullanılan U.K. kodu
long bsiyah = 0xFD9867; // LED’lerin sönmesi için kullanılan U.K. kodu
```

RGB LED’in bağlı olduğu mikrodenetleyici kartın portları tanımlanırken “`void setup()`” fonksiyonu içine aşağıdaki kodlar eklenir.

```
pinMode(3, OUTPUT); // RGB Kırmızı LED portu
pinMode(5, OUTPUT); // RGB Yeşil LED portu
pinMode(6, OUTPUT); // RGB Mavi LED portu
```

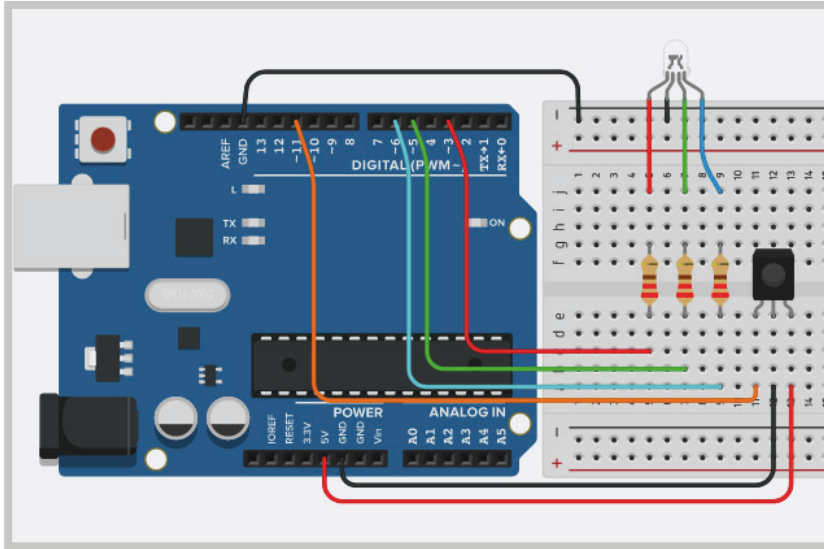
Mikrodenetleyici karta bağlı IR alıcı ve kod çözücünün uzaktan kumandadan aldığı kodları RGB LED’in ilgili renklerini yakabilmesi için “`void loop()`” fonksiyonu içine aşağıdaki kodların eklenmelidir. Burada “`results.value`” değişkeni uzaktan kumandadan anlık olarak aldığı verileri saklayan bir değişkendir.

```
if (results.value == bkirmizi) {
    digitalWrite(3, 1);
    digitalWrite(5, 0);
    digitalWrite(6, 0);
}
```

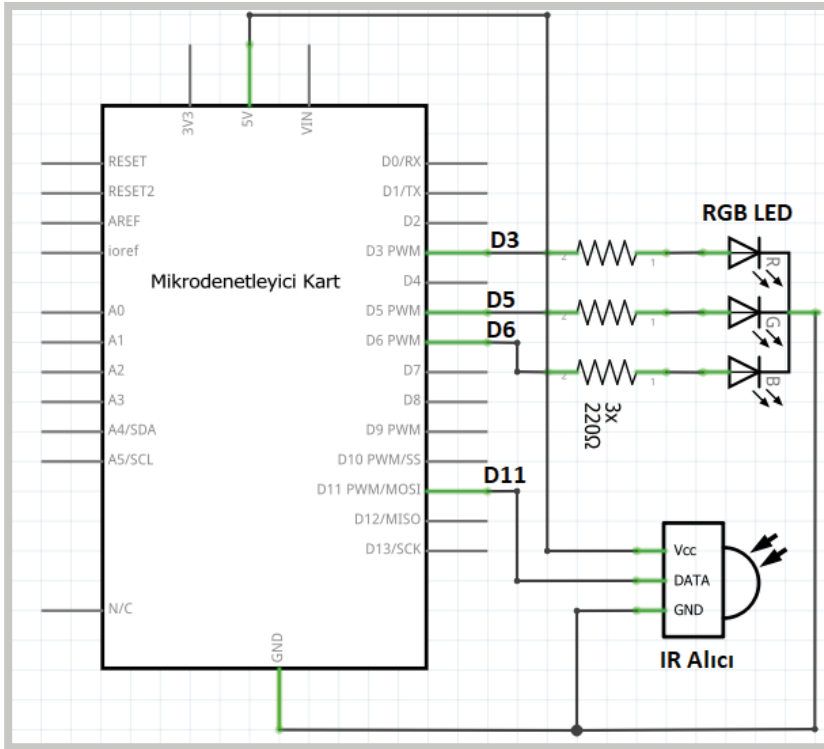


**3. Adım:** Yukarıdaki kodlama adımları sadece kırmızı renk için verilmiştir. Bu kodlama adımlarının aynı yeşil, mavi, mor, turkuaz ve beyaz renkler için de yapılmalıdır.

- **bkırmızı = 0xFF18E7**; program kodunda kırmızı renk için verilen **FF18E7** değeri elektronik devre simülatörü içinde bulunan uzaktan kumandanın “1” tuşuna karşılık gelen değerdir.
- Dolayısıyla IR alıcı ve kod çözücü ile RGB LED uygulamasının program kodları, elektronik devre simülatörü programında yazılıp çalıştırıldığında kumandanın “1” tuşu kırmızı rengi, “2” tuşu yeşil rengi, “3” tuşu mavi rengi, “4” tuşu sarı rengi, “5” tuşu mor rengi, “6” tuşu turkuaz rengi, “7” tuşu beyaz rengi yakar.



GörSEL 2.92: IR alıcı ve kod çözücü ile RGB LED uygulama devresi



GörSEL 2.93: IR alıcı ve kod çözücü ile RGB LED uygulamasının elektriksel devresi

**4. Adım:** IR Alıcı ve kod çözücü ile RGB LED uygulamasının program kodları aşağıda gösterilmektedir.

```
#include <IRremote.h>

int RECV_PIN = 11;

IRrecv irrecv(RECV_PIN);

decode_results results;

long bkirmizi = 0xFD08F7; // Kırmızı renk için kullanılan U.K. kodu
long byesil = 0xFD8877; // Yeşil renk için kullanılan U.K. kodu
long bmavi = 0xFD48B7; // Mavi renk için kullanılan U.K. kodu
long bsari = 0xFD28D7; // Sarı renk için kullanılan U.K. kodu
long bmor = 0xFDA857; // Mor renk için kullanılan U.K. kodu
long bturkuaz = 0xFD6897; // Turkuaz renk için kullanılan U.K. kodu
long bbeyaz = 0xFD18E7; // Beyaz renk için kullanılan U.K. kodu
long bsiyah = 0xFD9867; // LED'lerin sönməsi için kullanılan U.K. kodu

void setup()
{
    pinMode(3, OUTPUT); // Kırmızı
    pinMode(5, OUTPUT); // Yeşil
    pinMode(6, OUTPUT); // Mavi

    Serial.begin(9600);
    irrecv.enableIRIn(); // IR alıcıyı başlatır.
}

void loop() {
    if (irrecv.decode(&results)) {
        Serial.println(results.value, HEX);
        if (results.value == bkirmizi) {
            digitalWrite(3, 1);
            digitalWrite(5, 0);
            digitalWrite(6, 0);
        }
        if (results.value == byesil) {
            digitalWrite(3, 0);
            digitalWrite(5, 1);
            digitalWrite(6, 0);
        }
        if (results.value == bmavi) {
            digitalWrite(3, 0);
            digitalWrite(5, 0);
            digitalWrite(6, 1);
        }
    }
}
```

```

}
if (results.value== bsari) {
    digitalWrite(3,1);
    digitalWrite(5,1);
    digitalWrite(6,0);
}
if (results.value== bmor) {
    digitalWrite(3,1);
    digitalWrite(5,0);
    digitalWrite(6,1);
}
if (results.value== bturkuaz) {
    digitalWrite(3,0);
    digitalWrite(5,1);
    digitalWrite(6,1);
}
if (results.value== bbeyaz) {
    digitalWrite(3,1);
    digitalWrite(5,1);
    digitalWrite(6,1);
}
if (results.value==bsiyah) {
    digitalWrite(3,0);
    digitalWrite(5,0);
    digitalWrite(6,0);
}

irrecv.resume();// Sıradaki değeri algılar.
}
}

```



### Sıra Sizde

Görsel 2.92'deki IR alıcı ve kod çözücü ile RGB LED uygulamasını elektronik devre simülatör programında kurunuz. Bu uygulamaya ait program kodlarını elektronik devre simülatör programında “Kod” bölümüne yazınız. Tüm işlemler bittikten sonra “Simülasyonu Başlat” butonu ile devreyi çalıştırıp uzaktan kumandanın tuşlarına basarak RGB LED'deki değişimleri gözlemleyiniz.



### Değerlendirme

Çalışmalarınız öğrenme biriminin sonunda yer alan Kontrol Listesi kullanılarak değerlendirilecektir. Çalışmalarınızı yaparken değerlendirme ölçütlerini dikkate alınız.



<http://kitap.eba.gov.tr/KodSor.php?KOD=20794>

## 2. Öğrenme Birimi: Mikrodenetleyici Kart Programlama



### Sıra Sizde

Görsel 2.92 veya Görsel 2.93'teki devreyi breadboard üzerine kurunuz. IR alıcı ve kod çözücü ile RGB LED uygulama programındaki uzaktan kumanda tuş kodlarını elinizdeki kumandanın tuş kodlarıyla yeniden düzenleyip mikrodenetleyici IDE programında yazınız. Tüm işlemler bittikten sonra programı mikrodenetleyici karta yükleyip, uzaktan kumandanın tuşlarına basarak RGB LED'deki değişimleri gözlemleyiniz.



### Değerlendirme

Çalışmalarınız öğrenme biriminin sonunda yer alan Kontrol Listesi kullanılarak değerlendirilecektir. Çalışmalarınızı yaparken değerlendirme ölçütlerini dikkate alınız.

### 2.13.3. 7 Segment Display'in Uzaktan Kumanda ile Kullanımı



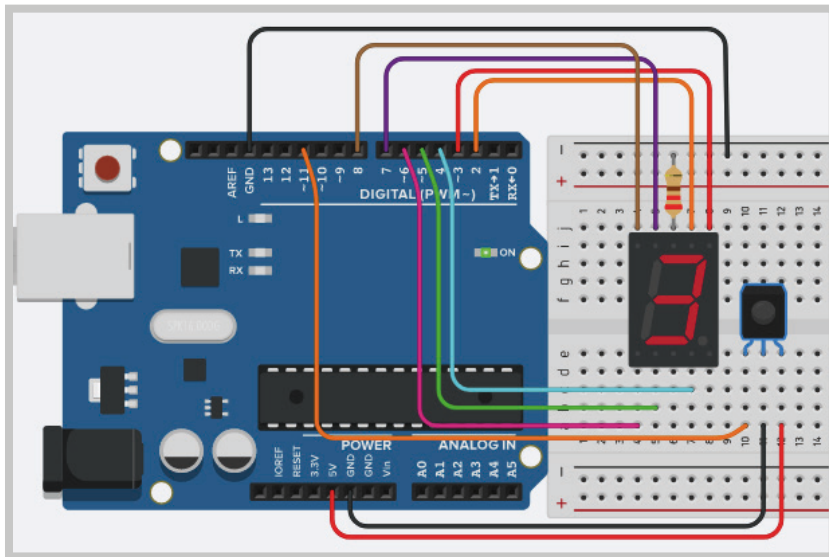
### 40. Uygulama

Uzaktan kumandanın kodları çözüldükten sonra yapılabilecek devrelerden biri de IR alıcı ve kod çözücü ile 7 segment display uygulamasıdır. Bu devrenin amacı, uzaktan kumandanın özellikle sayısal tuşlarına basıldığında display ekranında rakam göstermesidir.

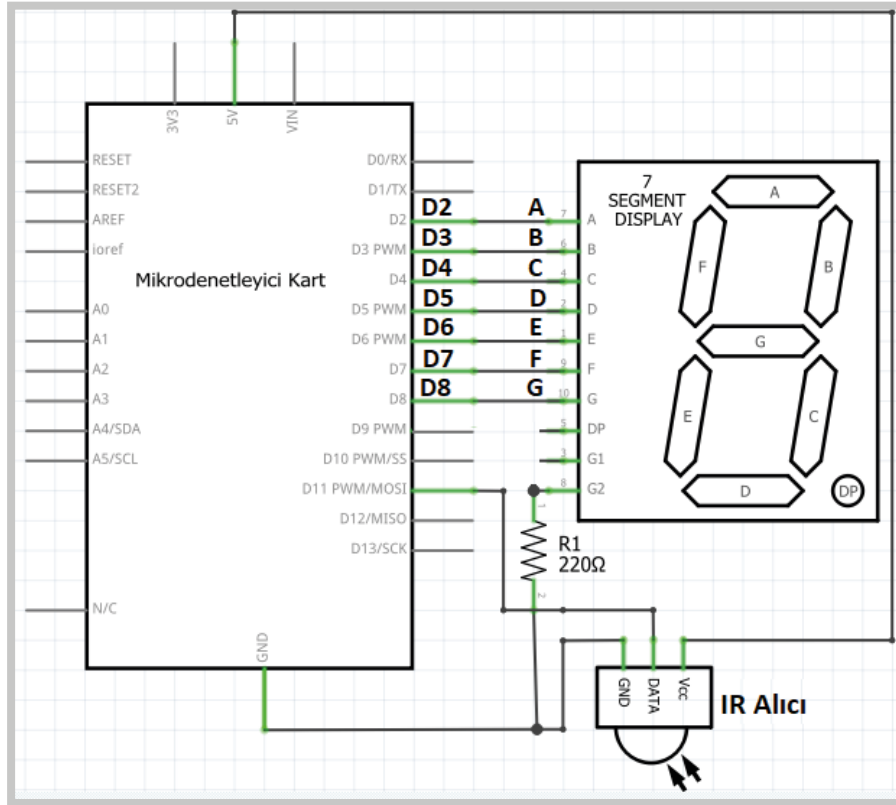
#### Gerekli Devre Elemanları

- 1 adet mikrodenetleyici kart
- 1 adet breadboard
- 1 adet 7 segment display
- 1 adet 220Ω (kırmızı, kırmızı, kahverengi, altın rengi) direnç
- 1 adet IR alıcı
- 1 adet IR verici kumanda
- Bağlantı kabloları

**1. Adım:** IR alıcı ve kod çözücü ile 7 segment display uygulamasının devresi Görsel 2.94 ve Görsel 2.95'te verilmiştir.



Görsel 2.94: IR alıcı ve kod çözücü ile 7 segment display uygulama devresi



Görsel 2.95: IR alıcı ve kod çözücü ile 7 segment display uygulamasının elektriksel devresi

**2. Adım:** IR alıcı ve kod çözücü ile 7 segment display uygulamasının program kodları aşağıda verilmiştir.

```

long b0 = 0xFD30CF;
long b1 = 0xFD08F7;
long b2 = 0xFD8877;
long b3 = 0xFD48B7;
long b4 = 0xFD28D7;
long b5 = 0xFDA857;
long b6 = 0xFD6897;
long b7 = 0xFD18E7;
long b8 = 0xFD9867;
long b9 = 0xFD58A7;
long by = 0xFD50AF;
long ba = 0xFD10EF;

#include <IRremote.h>

int RECV_PIN = 11;

IRrecv irrecv(RECV_PIN);

decode_results results;

void setup()
{
  Serial.begin(9600);

```

```
irrecv.enableIRIn(); // IR alıcısı başlatır.

for (int i=2;i<=8;i++) pinMode(i, OUTPUT);
}

void loop() {
  if (irrecv.decode(&results)) {
    Serial.println(results.value, HEX);
    if (results.value==b1) bir();
    if (results.value==b2) iki();
    if (results.value==b3) uc();
    if (results.value==b4) dort();
    if (results.value==b5) bes();
    if (results.value==b6) alti();
    if (results.value==b7) yedi();
    if (results.value==b8) sekiz();
    if (results.value==b9) dokuz();
    if (results.value==b0) sifir();
    if (results.value==by) ilerisay();
    if (results.value==ba) gerisay();

    irrecv.resume(); // Sıradaki değeri algılar.
  }
}

void ilerisay() {
  sifir();      delay(1000);
  bir();        delay(1000);
  iki();        delay(1000);
  uc();         delay(1000);
  dort();       delay(1000);
  bes();        delay(1000);
  alti();       delay(1000);
  yedi();       delay(1000);
  sekiz();      delay(1000);
  dokuz();      delay(1000);
}

void gerisay() {
  dokuz();      delay(1000);
  sekiz();      delay(1000);
  yedi();       delay(1000);
  alti();       delay(1000);
  bes();        delay(1000);
  dort();       delay(1000);
  uc();         delay(1000);
  iki();        delay(1000);
  bir();        delay(1000);
  sifir();      delay(1000);
}
}
```

```
void sifir() {
digitalWrite(2, 1); //a
digitalWrite(3, 1); //b
digitalWrite(4, 1); //c
digitalWrite(5, 1); //d
digitalWrite(6, 1); //e
digitalWrite(7, 1); //f
digitalWrite(8, 0); //g

}

void bir() {

digitalWrite(2, 0); //a
digitalWrite(3, 1); //b
digitalWrite(4, 1); //c
digitalWrite(5, 0); //d
digitalWrite(6, 0); //e
digitalWrite(7, 0); //f
digitalWrite(8, 0); //g

}

void iki() {
digitalWrite(2, 1); //a
digitalWrite(3, 1); //b
digitalWrite(4, 0); //c
digitalWrite(5, 1); //d
digitalWrite(6, 1); //e
digitalWrite(7, 0); //f
digitalWrite(8, 1); //g

}

void uc() {

digitalWrite(2, 1); //a
digitalWrite(3, 1); //b
digitalWrite(4, 1); //c
digitalWrite(5, 1); //d
digitalWrite(6, 0); //e
digitalWrite(7, 0); //f
digitalWrite(8, 1); //g

}

void dort() {
digitalWrite(2, 0); //a
digitalWrite(3, 1); //b
digitalWrite(4, 1); //c
digitalWrite(5, 0); //d
digitalWrite(6, 0); //e
digitalWrite(7, 1); //f
digitalWrite(8, 1); //g

}

}
```

```
void bes () {
digitalWrite (2, 1); //a
digitalWrite (3, 0); //b
digitalWrite (4, 1); //c
digitalWrite (5, 1); //d
digitalWrite (6, 0); //e
digitalWrite (7, 1); //f
digitalWrite (8, 1); //g

}

void alti () {
digitalWrite (2, 1); //a
digitalWrite (3, 0); //b
digitalWrite (4, 1); //c
digitalWrite (5, 1); //d
digitalWrite (6, 1); //e
digitalWrite (7, 1); //f
digitalWrite (8, 1); //g

}

void yedi () {
digitalWrite (2, 1); //a
digitalWrite (3, 1); //b
digitalWrite (4, 1); //c
digitalWrite (5, 0); //d
digitalWrite (6, 0); //e
digitalWrite (7, 0); //f
digitalWrite (8, 0); //g

}

void sekiz () {
digitalWrite (2, 1); //a
digitalWrite (3, 1); //b
digitalWrite (4, 1); //c
digitalWrite (5, 1); //d
digitalWrite (6, 1); //e
digitalWrite (7, 1); //f
digitalWrite (8, 1); //g

}

void dokuz () {
digitalWrite (2, 1); //a
digitalWrite (3, 1); //b
digitalWrite (4, 1); //c
digitalWrite (5, 1); //d
digitalWrite (6, 0); //e
digitalWrite (7, 1); //f
digitalWrite (8, 1); //g

}
```





### Sıra Sizde

Görsel 2.94'teki IR alıcı ve kod çözücü ile 7 segment display uygulamasını elektronik devre simülâtör programında kurunuz. Bu uygulamaya ait program kodlarını elektronik devre simülâtör programında "Kod" bölümüne yazınız. Tüm işlemler bittikten sonra "Simülasyonu Başlat" butonu ile devreyi çalıştırıp uzaktan kumandanın tuşlarına basarak 7 segment display ekranındaki değişimleri gözlemleyiniz.



<http://kitap.eba.gov.tr/KodSor.php?KOD=20795>



### Değerlendirme

Çalışmalarınız öğrenme biriminin sonunda yer alan Kontrol Listesi kullanılarak değerlendirilecektir. Çalışmalarınızı yaparken değerlendirme ölçütlerini dikkate alınız.



### Sıra Sizde

Görsel 2.94 veya Görsel 2.95'teki devreyi breadboard üzerine kurunuz. IR alıcı ve kod çözücü ile 7 segment display uygulama programındaki uzaktan kumanda tuş kodlarını elinizdeki kumandanın tuş kodlarıyla yeniden düzenleyip mikrodenetleyici IDE programında yazınız. Tüm işlemler bittikten sonra programı mikrodenetleyici karta yükleyip, uzaktan kumandanın tuşlarına basarak 7 segment display ekranındaki değişimleri gözlemleyiniz.

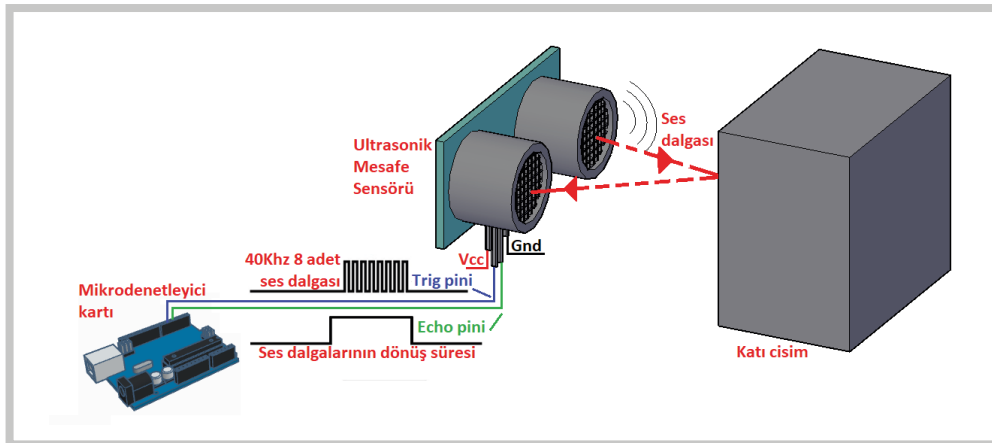


### Değerlendirme

Çalışmalarınız öğrenme biriminin sonunda yer alan Kontrol Listesi kullanılarak değerlendirilecektir. Çalışmalarınızı yaparken değerlendirme ölçütlerini dikkate alınız.

## 2.14. MİKRODENETLEYİCİ KART İLE MESAFE SENSÖRÜ UYGULAMALARI

Elektronik devrelerde veya robotik uygulamalarda engel algılama işlemini gerçekleştirebilmek için mesafe algılama sensörleri kullanılır. Mesafe algılama sensörlerinin birçok çeşidi vardır. Bu bölümde HC-SR04 olarak da bilinen ultrasonik mesafe sensörü kullanılacaktır. Ultrasonik mesafe sensörünün çalışma prensibi Görsel 2.96'da verilmiştir.



Görsel 2.96: Ultrasonik mesafe sensörünün çalışma prensibi

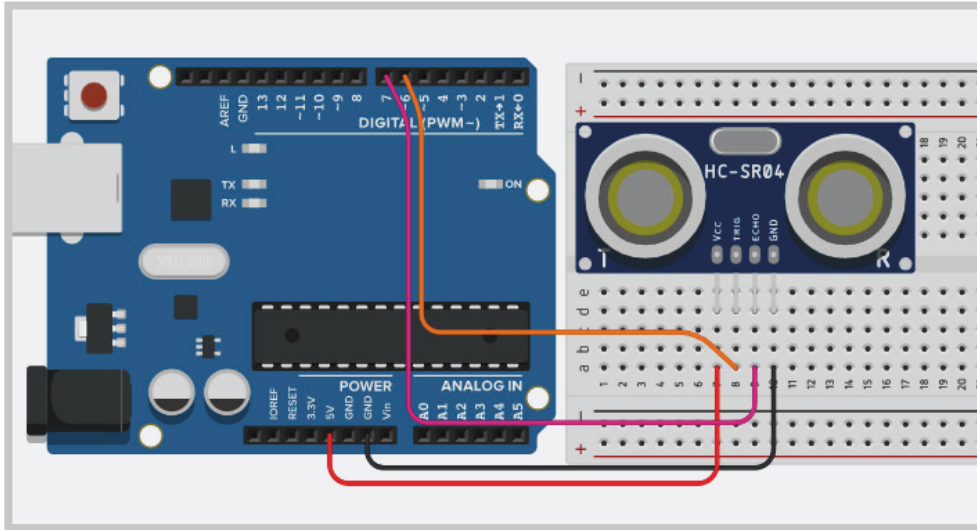
## 2. Öğrenme Birimi: Mikrodenetleyici Kart Programlama

Ultrasonik mesafe sensörü iki adet hoparlörden oluşur. Bunlardan biri karşı tarafa 40 kHz'lik 8 adet ses dalgası gönderir. Diğer hoparlör ise bu ses dalgasının bir cisme çarpıp geri dönmesini bekler. Gönderilen ses sinyali bir cisme çarpıp geri döndüğünde ise sesin gecikme süresi hesaplanır. Bu şekilde sensör ile cisim arasındaki mesafe ölçülmüş olur.

Sesin hızı 0°C'de 331 m/sn.dir. Sıcaklığa bağlı olarak da bu değer değişmektedir. Sıcaklığa göre hızın hesaplanması  $V = 331 \cdot \sqrt{1 + \frac{T}{273}}$  denkleminde elde edilir. Bu formülde V sesin hızı, T ise ortam sıcaklığı için kullanılır. Formülden elde edilen değer m/sn. cinsinden verilir, ancak programda kullanılacak değer  $\frac{1}{V} \cdot 10000$  yapılmaması gerekir. Programda kullanılan 28.965 sabit değeri 24°C'de hesaplanan değerdir. Bu sabit değer yuvarlanıp 28.97 olarak kullanılabilir.



Ultrasonik mesafe sensörü uygulamasında, hassas bir ölçüm yapılması isteniyorsa verilen formül kullanılarak devrenin bulunduğu ortama göre yeniden hesaplanma yapılmalıdır.



GörSEL 2.97: Ultrasonik mesafe sensörü uygulama devresi



Bu uygulamanın amacı ultrasonik mesafe sensörü kullanımının öğrenilmesi için bir örnek yapmaktır (GörSEL 2.97).

### Gerekli Devre Elemanları

- 1 adet mikrodenetleyici kart
- 1 adet breadboard
- 1 adet Ultrasonik mesafe sensörü
- Bağlantı kabloları

**1. Adım:** Ultrasonik mesafe sensöründe Görsel 2.97’de görüldüğü gibi 4 adet bağlantı pini bulunur.

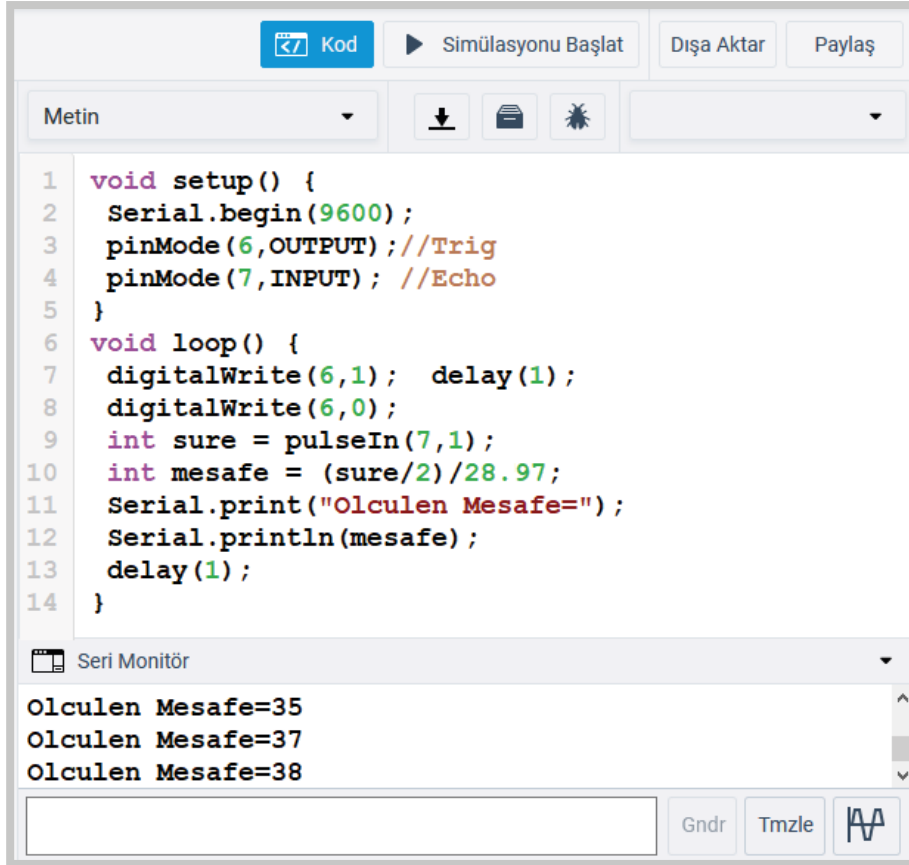
- Bu pinlerden Vcc pini mikrodenetleyici kartın 5V pinine, GND pini ise mikrodenetleyici kartın GND pinine bağlanır.
- Geriye kalan Trig pini (ses sinyali gönderen pin) mikrodenetleyici kartın D6 pinine, Echo pini (ses sinyalini alan pin) ise mikrodenetleyici kartın D7 pinine bağlanır.

**2. Adım:** Ultrasonik mesafe sensörü uygulamasının aşağıdaki program kodları incelendiğinde;

- “`void setup ()`” fonksiyonu içinde `pinMode (6, OUTPUT)` ; komutuyla mikrodenetleyici kartın “trig” pini için kullanılan D6 portu çıkış, `pinMode (7, INPUT)` ; komutuyla mikrodenetleyici kartın “echo” pini için kullanılan D7 portu giriş yapılır.
- Mesafe sensörüyle cisim arasındaki mesafe hesaplanırken, “`void loop ()`” fonksiyonu içinde öncelikle `digitalWrite (6,1)` ; komutuyla karşı tarafa bir ses dalgası gönderilir.
- 1 milisaniye beklenip `digitalWrite (6,0)` ; komutuyla bu ses dalgası kesilir.
- Gönderilen sesi almak için `pulseIn (7,1)` ; komutu kullanılarak “sure” ismindeki değişkene aktarılır.
- Mesafe hesaplaması yapılırken, “mesafe” isminde bir değişken oluşturularak, `int mesafe = (sure/2) / 28.97` ; komutuyla “sure değişkenin” yarısı alınıp 28.97 sabit sayısı ile çarpılır.
- Burada sesin gidip gelme mesafesinden dolayı “sure” değişkenin yarısı alınır. Komutta kullanılan sabit sayı ise ortam sıcaklığına göre hesaplanır.

**3. Adım:** Ultrasonik mesafe sensörü uygulamasının program kodları aşağıda verilmiştir. Programın seri port ekran çıktısı Görsel 2.98’de görülmektedir.

```
void setup () {
    Serial.begin (9600) ;
    pinMode (6, OUTPUT) ; //Trig
    pinMode (7, INPUT) ; //Echo
}
void loop () {
    digitalWrite (6,1) ; delay (1) ;
    digitalWrite (6,0) ;
    int sure=pulseIn (7,1) ;
    int mesafe=(sure/2)/28.97 ;
    Serial.print ("Olculen Mesafe=") ;
    Serial.println (mesafe) ;
    delay (1) ;
}
```



```
1 void setup() {
2   Serial.begin(9600);
3   pinMode(6,OUTPUT); //Trig
4   pinMode(7,INPUT); //Echo
5 }
6 void loop() {
7   digitalWrite(6,1); delay(1);
8   digitalWrite(6,0);
9   int sure = pulseIn(7,1);
10  int mesafe = (sure/2)/28.97;
11  Serial.print("Olculen Mesafe=");
12  Serial.println(mesafe);
13  delay(1);
14 }
```

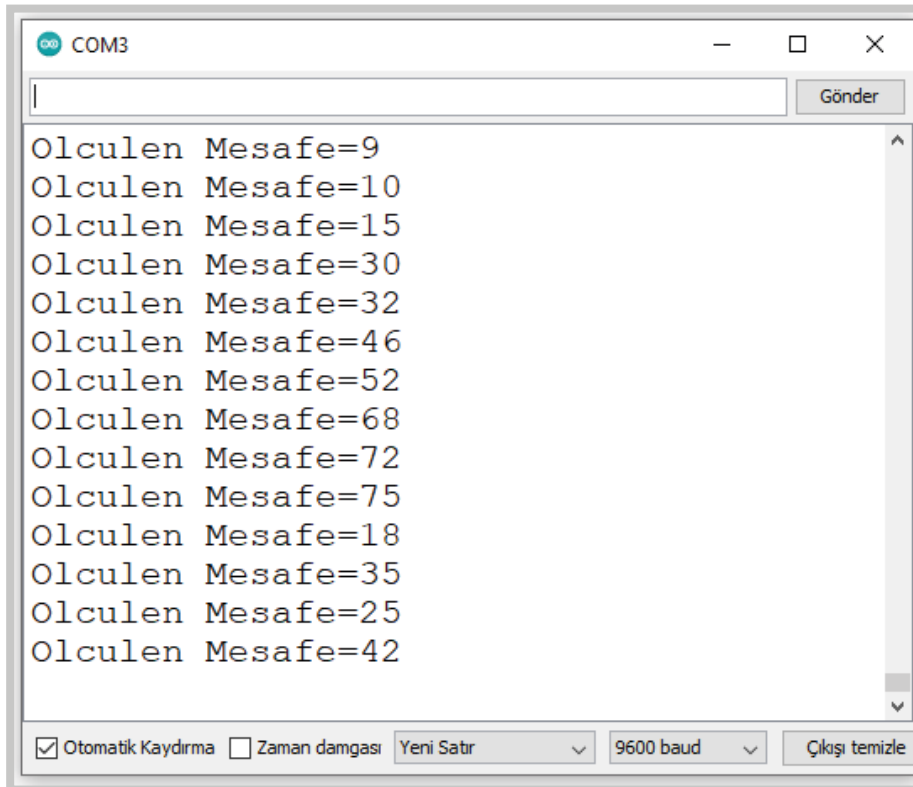
Seri Monitör

Olculen Mesafe=35  
Olculen Mesafe=37  
Olculen Mesafe=38

Gndr Tmzle

Görsel 2.98: Ultrasonik mesafe sensörü uygulamasının bilgisayarlı çizim programındaki ekran çıktısı

Görsel 2.99'daki uygulamanın mikrodenetleyici IDE programında seri port ekran çıktısı görülmektedir.



COM3

Gönder

Olculen Mesafe=9  
Olculen Mesafe=10  
Olculen Mesafe=15  
Olculen Mesafe=30  
Olculen Mesafe=32  
Olculen Mesafe=46  
Olculen Mesafe=52  
Olculen Mesafe=68  
Olculen Mesafe=72  
Olculen Mesafe=75  
Olculen Mesafe=18  
Olculen Mesafe=35  
Olculen Mesafe=25  
Olculen Mesafe=42

Otomatik Kaydırma  Zaman damgası Yeni Satır 9600 baud Çıkışı temizle

Görsel 2.99: Ultrasonik mesafe sensörü uygulamasının mikrodenetleyici IDE programındaki ekran çıktısı



### Sıra Sizde

Görsel 2.97'deki ultrasonik mesafe sensörü uygulamasını bilgisayarlı çizim programında kurunuz. Bu uygulamaya ait program kodlarını bilgisayarlı çizim programında "Kod" bölümüne yazınız. Tüm işlemler bittikten sonra "Simülasyonu Başlat" butonu ile devreyi çalıştırınız. Mesafe sensörünün üzerine tıklayarak engel değerlerini değiştiriniz ve bu değişimleri seri port ekranında gözlemleyiniz.



<http://kitap.eba.gov.tr/KodSor.php?KOD=20796>



### Değerlendirme

Çalışmalarınız öğrenme biriminin sonunda yer alan Kontrol Listesi kullanılarak değerlendirilecektir. Çalışmalarınızı yaparken değerlendirme ölçütlerini dikkate alınız.



### Sıra Sizde

Görsel 2.97'deki ultrasonik mesafe sensörü uygulamasındaki devreyi breadboard üzerine kurunuz. Bu uygulamaya ait program kodlarını mikrodenetleyici IDE programında yazınız. Tüm işlemler bittikten sonra programı mikrodenetleyici karta yükleyiniz. Ultrasonik mesafe sensörünün önüne bir engel yaklaştırıp uzaklaştırarak seri port üzerindeki değişimleri gözlemleyiniz.



### Değerlendirme

Çalışmalarınız öğrenme biriminin sonunda yer alan Kontrol Listesi kullanılarak değerlendirilecektir. Çalışmalarınızı yaparken değerlendirme ölçütlerini dikkate alınız.



### 42. Uygulama

Bu uygulamanın amacı, ultrasonik mesafe sensörünün ölçtüğü değerleri LED ile gösteren bir çalışma gerçekleştirmektir.

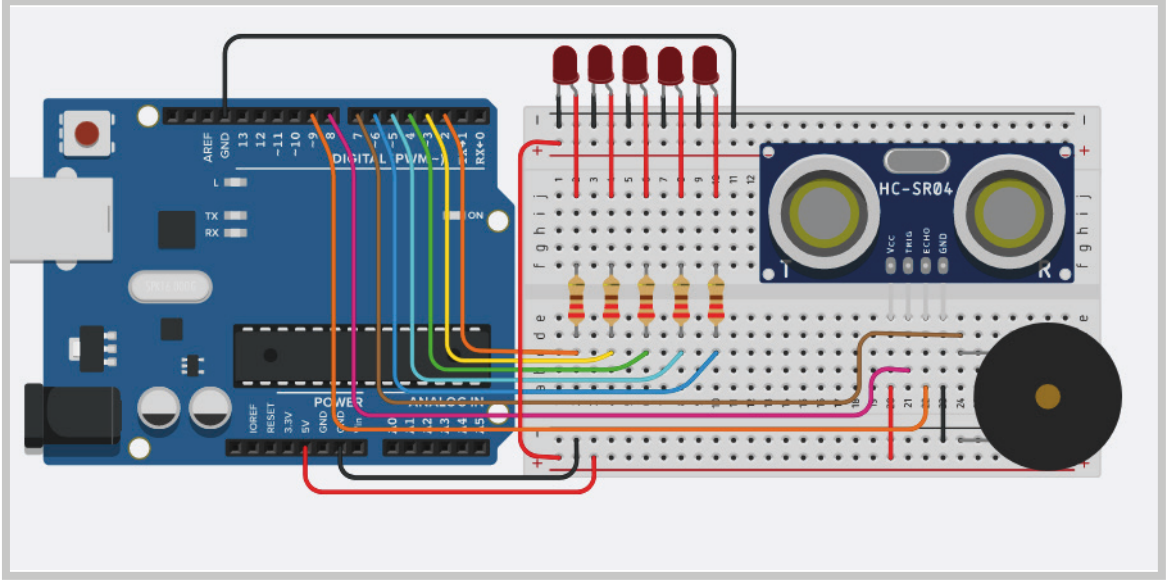
#### Gerekli Devre Elemanları

- 1 adet mikrodenetleyici kart
- 1 adet breadboard
- 5 adet 220Ω (kırmızı, kırmızı, kahverengi, altın rengi) direnç
- 5 adet LED
- 1 adet Ultrasonik mesafe sensörü
- 1 adet buzzer
- Bağlantı kabloları

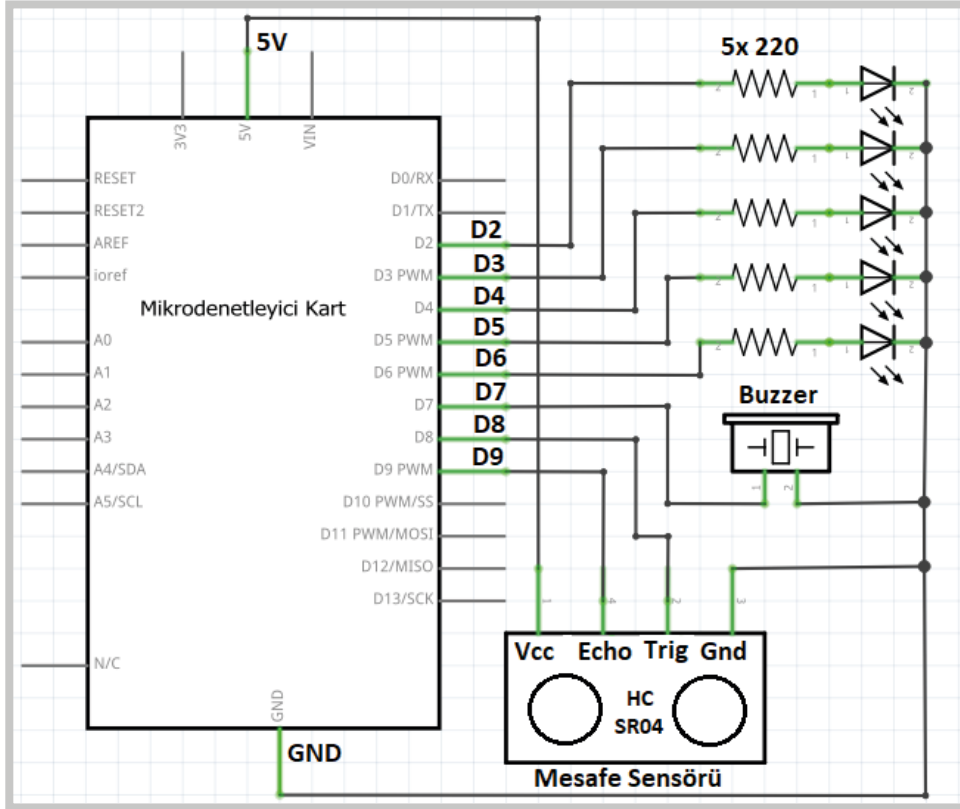
**1. Adım:** Görsel 2.100 ve Görsel 2.101'de ultrasonik mesafe sensörünün ölçtüğü değerleri LED ile gösteren bir uygulama verilmiştir. Bu uygulamada ultrasonik mesafe sensörünün ölçtüğü değerler sırasıyla şöyledir:

- 10 cm'den küçükse buzzer bip sesi üretmektedir.
- 10 cm'ye eşit veya 10 cm'den büyükse mikrodenetleyici kartın D2 portuna bağlı LED yanmaktadır.
- 20 cm'ye eşit veya 20 cm'den büyükse mikrodenetleyici kartın D3 portuna bağlı LED yanmaktadır.
- 30 cm'ye eşit veya 30 cm'den büyükse mikrodenetleyici kartın D4 portuna bağlı LED yanmaktadır.
- 40 cm'ye eşit veya 40 cm'den büyükse mikrodenetleyici kartın D5 portuna bağlı LED yanmaktadır.
- 50 cm'ye eşit veya 50 cm'den büyükse mikrodenetleyici kartın D6 portuna bağlı LED yanmaktadır.

## 2. Öğrenme Birimi: Mikrodenetleyici Kart Programlama



GörSEL 2.100: Ultrasonik mesafe sensörlü LED uygulama devresi



GörSEL 2.101: Ultrasonik mesafe sensörlü LED uygulamasının elektriksel devresi

**2. Adım:** Ultrasonik mesafe sensörlü LED uygulamasının program kodları aşağıdaki gibidir.

```
void setup() {  
    Serial.begin(9600);  
    pinMode(8,OUTPUT); //Trig  
    pinMode(9,INPUT); //Echo  
    for (int i=2;i<=7;i++) pinMode(i,OUTPUT);  
}
```

```
void loop() {  
    digitalWrite(8,1); delay(1);  
    digitalWrite(8,0);  
    int sure=pulseIn(9,1);  
    int mesafe=(sure/2)/28.97;  
    Serial.print("Olculen Mesafe=");  
    Serial.println(mesafe);  
    delay(1);  
  
    if(mesafe<10) digitalWrite(7,1); else digitalWrite(7,0);  
    if(mesafe>=10) digitalWrite(2,1); else digitalWrite(2,0);  
    if(mesafe>=20) digitalWrite(3,1); else digitalWrite(3,0);  
    if(mesafe>=30) digitalWrite(4,1); else digitalWrite(4,0);  
    if(mesafe>=40) digitalWrite(5,1); else digitalWrite(5,0);  
    if(mesafe>=50) digitalWrite(6,1); else digitalWrite(6,0);  
}
```

### Sıra Sizde

Görsel 2.100'deki ultrasonik mesafe sensörlü LED uygulamasını elektronik devre simülatör programında kurunuz. Bu uygulamaya ait program kodlarını elektronik devre simülatör programında "Kod" bölümüne yazınız. Tüm işlemler bittikten sonra "Simülasyonu Başlat" butonu ile devreyi çalıştırınız. Mesafe sensörünün üzerine tıklayarak engel değerlerini değiştiriniz ve LED'lerdeki yanıp sönme durumlarını gözlemleyiniz.

### Değerlendirme

Çalışmalarınız öğrenme biriminin sonunda yer alan Kontrol Listesi kullanılarak değerlendirilecektir. Çalışmalarınızı yaparken değerlendirme ölçütlerini dikkate alınız.

### Sıra Sizde

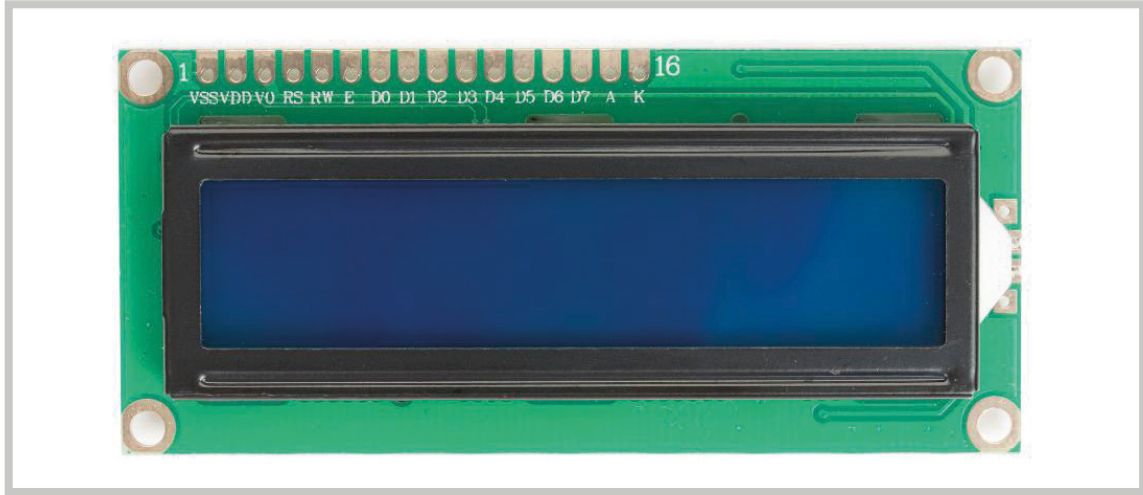
Görsel 2.100 veya Görsel 2.101'deki ultrasonik mesafe sensörlü LED uygulamasındaki devreyi breadboard üzerine kurunuz. Bu uygulamaya ait program kodlarını mikrodenetleyici IDE programında yazınız. Tüm işlemler bittikten sonra programı mikrodenetleyici karta yükleyiniz. Ultrasonik mesafe sensörünün önüne bir engel yaklaştırıp uzaklaştırarak LED'lerdeki yanıp sönme durumlarını gözlemleyiniz.

### Değerlendirme

Çalışmalarınız öğrenme biriminin sonunda yer alan Kontrol Listesi kullanılarak değerlendirilecektir. Çalışmalarınızı yaparken değerlendirme ölçütlerini dikkate alınız.

### 2.15. MİKRODENETLEYİCİ KART İLE LCD UYGULAMALARI

LCD, İngilizce Liquid Crystal Display (Sıvı Kristal Gösterge ) kelimelerinin kısaltmasından oluşmuştur. Genel olarak çalışma prensibi, üzerine elektrik uygulanan sıvı kristallerin belirli bir formatta yazı veya resim oluşturmaya ilkesine dayanır. Çok farklı çeşitleri bulunmaktadır. Bu bölümde elektronik devrelerde yaygın olarak kullanılan 1602 (16 karakter 2 satır) LCD kullanılacaktır. Görsel 2.102'de 1602 LCD ekran ve bağlantı pinleri görülmektedir. Bağlantı pinlerinin açıklaması ve mikrodenetleyici karta bağlantısı Tablo 2.8'de verilmiştir.



Görsel 2.102: 1602 LCD ekran ve bağlantı pinleri

Tablo 2.8: 1602 LCD Ekran ve Bağlantı Pinlerinin Açıklaması

Pin No	Kodu	Mikrodenetleyici Bağlantısı	Açıklaması
1	GND	GND	LCD ekran GND besleme pini
2	Vcc	5V	LCD ekran +5V besleme pini
3	V0	10KΩ Pot	LCD ekran yazılarının görünürliğini ayarlamak için
4	RS	D12	RS (Register Select) pini 0 ise komut, 1 ise veri gönderilir.
5	RW	GND	RW(Read/Write) pini 0 ise ekrana yazma, 1 ise ekrandan okuma
6	E	D11	E (Enable)pini mikrodenetleyiciden alacağı kontrol sinyalleri için
7	DB0	Kullanılmıyor	Veri giriş pini
8	DB1	Kullanılmıyor	Veri giriş pini
9	DB2	Kullanılmıyor	Veri giriş pini
10	DB3	Kullanılmıyor	Veri giriş pini
11	DB4	D5	Veri giriş pini
12	DB5	D4	Veri giriş pini
13	DB6	D3	Veri giriş pini
14	DB7	D2	Veri giriş pini
15	LED+	5V	LCD arka aydınlatma LED'i için anot ucu
16	LED-	GND	LCD arka aydınlatma LED'i için katot ucu



### 2.15.1. LCD Bağlantıları

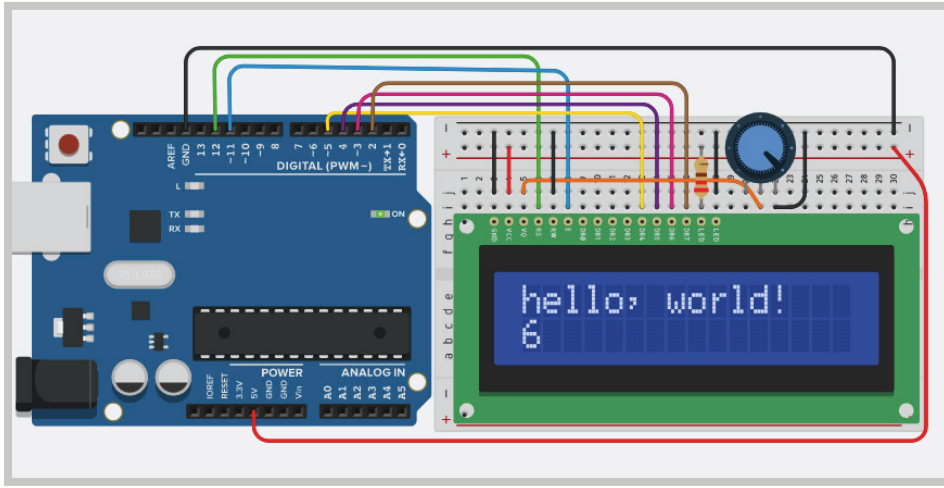
#### 43. Uygulama

Uygulamanın amacı LCD ekran uygulamasını gösteren bir çalışma gerçekleştirmektir.

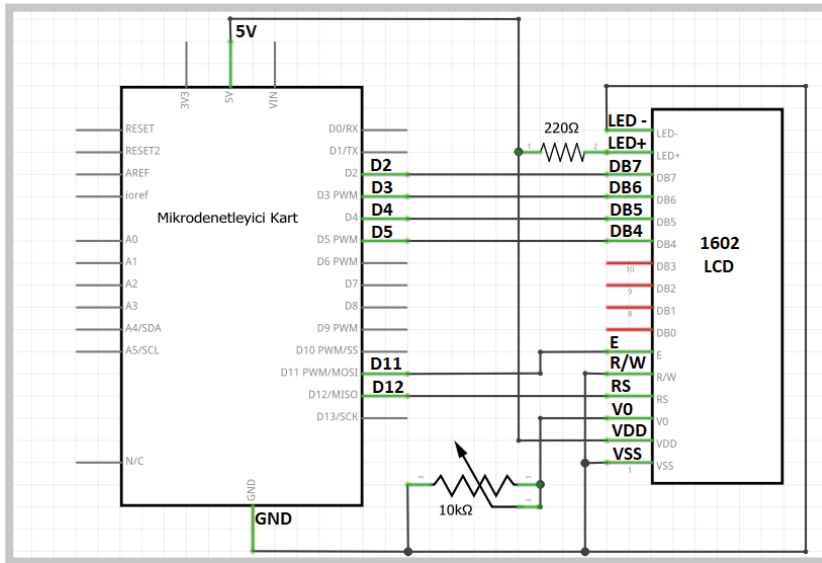
#### Gerekil Devre Elemanları

- 1 adet mikrodenetleyici kart
- 1 adet breadboard
- 1 adet 220Ω (kırmızı, kırmızı, kahverengi, altın rengi) direnç
- 1 adet LCD display
- 1 adet potansiyometre
- Bağlantı kabloları

**1. Adım:** Görsel 2.103 ve Görsel 2.104'te 1602 LCD ekran uygulamasının bir devresi görülmektedir.



Görsel 2.103: 1602 LCD ekran uygulama devresi



Görsel 2.104: 1602 LCD ekran uygulamasının elektriksel devresi

## 2. Öğrenme Birimi: Mikrodenetleyici Kart Programlama

**2. Adım:** Bu uygulamaya ait program kodları incelendiğinde;

- `const int rs = 12, en = 11, d4 = 5, d5 = 4, d6 = 3, d7 = 2;` komutuyla LCD display pinlerinin mikrodenetleyici kartına bağlanan portları için değişkenler tanımlanmıştır.
- `LiquidCrystal lcd(rs, en, d4, d5, d6, d7);` komutuyla da bu bağlantı pinlerini kullanan "LCD" adında "LiquidCrystal" nesnesi tanımlanmıştır.
- `void setup()` fonksiyonu içinde `lcd.begin(16, 2);` komutuyla da 16 karakter 2 satırdan oluşan "LCD" nesnesi kullanıma açılmıştır.
- `lcd.print("hello, world!");` komutu ekrana yazı yazdırma komutudur.
- `void loop()` fonksiyonu içinde `lcd.setCursor(0, 1);` komutuyla LCD ekran kursörü 0. sütun 1.satır konumuna getirilmiştir.
- LCD ekranda yazılacak karakterler ilk satır için `(0,0), (1,0), (2,0), (3,0)...` `(15,0)` şeklinde gider.
- LCD ekranda yazılacak karakterler ikinci satır için ise `(0,1), (1,1), (2,1), (3,1)...` `(15,1)` şeklinde gider.

**3. Adım:** 1602 LCD ekran uygulamasının program kodları aşağıdaki gibidir.

```
#include <LiquidCrystal.h>

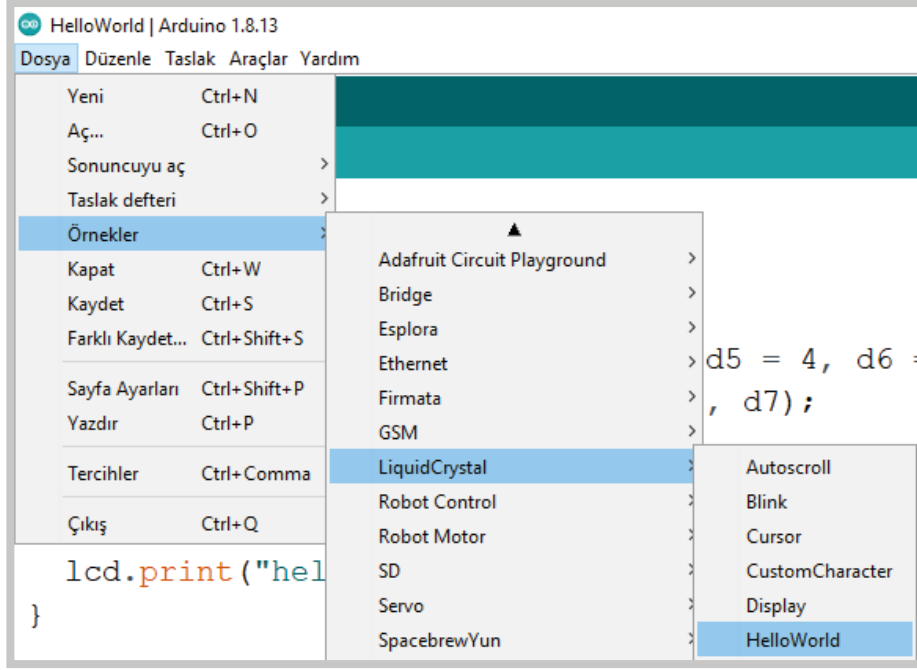
const int rs = 12, en = 11, d4 = 5, d5 = 4, d6 = 3, d7 = 2;

LiquidCrystal lcd(rs, en, d4, d5, d6, d7);

void setup() {
  lcd.begin(16, 2);
  lcd.print("hello, world!");
}

void loop() {
  lcd.setCursor(0, 1);
  lcd.print(millis() / 1000);
}
```

**4. Adım:** 1602 LCD ekran uygulamasının program kodlarına aynı zamanda Görsel 2.107'de görüldüğü gibi mikrodenetleyici IDE programındaki **Dosya** → **Örnekler** → **LiquidCrystal** → **HelloWord** menüsünden de ulaşılabilir.



Görsel 2.105: Mikrodnetleyici IDE programındaki LiquidCrystal örnek programları



### Sıra Sizde

Görsel 2.103'teki 1602 LCD ekran uygulamasını elektronik devre simülasyon programında kurunuz. Bu uygulamaya ait program kodlarını elektronik devre simülasyon programında "Kod" bölümüne yazınız. Tüm işlemler bittikten sonra "Simülasyonu Başlat" butonu ile devreyi çalıştırınız. LCD durumunu gözlemleyiniz.



### Değerlendirme

Çalışmalarınız öğrenme biriminin sonunda yer alan Kontrol Listesi kullanılarak değerlendirilecektir. Çalışmalarınızı yaparken değerlendirme ölçütlerini dikkate alınız.



### Sıra Sizde

Görsel 2.103 veya Görsel 2.104'teki 1602 LCD ekran uygulamasını breadboard üzerine kurunuz. Bu uygulamaya ait program kodlarını mikrodnetleyici IDE programında yazınız. Tüm işlemler bittikten sonra programı mikrodnetleyici karta yükleyiniz ve devrenin çalışmasını gözlemleyiniz.



### Değerlendirme

Çalışmalarınız öğrenme biriminin sonunda yer alan Kontrol Listesi kullanılarak değerlendirilecektir. Çalışmalarınızı yaparken değerlendirme ölçütlerini dikkate alınız.

### 2.15.2. LCD Ekranda Yazılan Yazının Sağa veya Sola Kaydırılması

#### 44. Uygulama

Uygulamanın amacı, LCD ekranda yazılan yazının sağa veya sola kaydırılmasını sağlayan bir çalışma gerçekleştirmektir.

#### Gerekli Devre Elemanları

- 1 adet mikrodenetleyici kart
- 1 adet breadboard
- 1 adet 220Ω (kırmızı, kırmızı, kahverengi, altın rengi) direnç
- 1 adet LCD display
- 1 adet potansiyometre
- Bağlantı kabloları

**1. Adım:** LCD ekranda yazılan yazının sağa veya sola kaydırılması animasyonu, görsellik ve dikkat çekme açısından kullanılan animasyonlardır. Görsel 2.105 ve Görsel 2.106'daki devre kullanılarak LCD ekrandaki yazının soldan sağa kaydırılmasıyla ilgili program yazılır.

**2. Adım:** Yazılan programın kodları incelendiğinde;

- `for (i=0;i<=16;i++)` komutuyla bir döngü oluşturulur.
- `lcd.clear()` ; komutuyla her defasında LCD ekrandaki yazılar silinir.
- `lcd.setCursor(i, 0)` ; komutuyla kursör için sütun sayısı birer birer artırılıp yazının yeniden yazılması sağlanır. Animasyonun çalışma mantığı bu şekildedir.

**3. Adım:** LCD ekranda yazılan yazının for döngüsüyle ile soldan sağa kaydırma uygulamasının program kodları aşağıdaki gibidir.

```
#include <LiquidCrystal.h>
const int rs = 12, en = 11, d4 = 5, d5 = 4, d6 = 3, d7 = 2;
LiquidCrystal lcd(rs, en, d4, d5, d6, d7);

void setup() {
  lcd.begin(16, 2);
}
void loop() {
  int i;
  for (i=0;i<=16;i++)
  {lcd.clear();
  lcd.setCursor(i, 0);
  lcd.print("Robotik ve Kodlama");
  delay(500);
  }
}
```



### Sıra Sizde

Görsel 2.103'teki 1602 LCD ekran uygulamasını elektronik devre simülatör programında kurunuz. LCD ekranda yazılan yazının for döngüsüyle ile soldan sağa kaydırma uygulamasının program kodlarını elektronik devre simülatör programında “Kod” bölümüne yazınız. Tüm işlemler bittikten sonra “Simülasyonu Başlat” butonu ile devreyi çalıştırınız. LCD durumunu gözlemleyiniz.



### Değerlendirme

Çalışmalarınız öğrenme biriminin sonunda yer alan Kontrol Listesi kullanılarak değerlendirilecektir. Çalışmalarınızı yaparken değerlendirme ölçütlerini dikkate alınız.



### 45. Uygulama

Uygulamanın amacı LCD ekranda yazılan yazının sağdan sola ve soldan sağa kaydırılmasıyla ilgili hazır komutları kullanarak bir çalışma gerçekleştirmektir.

#### Gerekli Devre Elemanları

- 1 adet mikrodenetleyici kart
- 1 adet breadboard
- 1 adet 220Ω (kırmızı, kırmızı, kahverengi, altın rengi) direnç
- 1 adet LCD display
- 1 adet potansiyometre
- Bağlantı kabloları
- 

**1. Adım:** 1602 LCD ekranda yazılan yazının sağdan sola ve soldan sağa kaydırılmasıyla ilgili hazır komutlar da vardır. Bu komutlardan `lcd.scrollDisplayRight()` ; komutu LCD de yazılan yazının sağdan sola kaydırmasını, `lcd.scrollDisplayLeft()` ; komutu ise soldan sağa kaydırmasını sağlar.

**2. Adım:** LCD'de yazılan yazının sağdan sola kaydırma uygulamasının program kodları aşağıdaki gibidir.

```
#include <LiquidCrystal.h>
```

```
const int rs = 12, en = 11, d4 = 5, d5 = 4, d6 = 3, d7 = 2;
```

```
LiquidCrystal lcd(rs, en, d4, d5, d6, d7);
```

```
void setup() {  
  lcd.begin(16, 2);  
  lcd.print("Robotik ve Kodlama");  
}  
void loop() {  
  lcd.scrollDisplayRight();  
  delay(250);  
}
```

**3. Adım:** LCD'de yazılan yazının soldan sağa kaydırma uygulamasının program kodları aşağıdaki gibidir.

```
#include <LiquidCrystal.h>  
  
const int rs = 12, en = 11, d4 = 5, d5 = 4, d6 = 3, d7 = 2;  
LiquidCrystal lcd(rs, en, d4, d5, d6, d7);  
  
void setup() {  
  lcd.begin(16, 2);  
  lcd.print("Robotik ve Kodlama");  
}  
void loop() {  
  lcd.scrollDisplayRight();  
  delay(250);  
}
```



### Sıra Sizde

Görsel 2.103'teki 1602 LCD ekran uygulamasını elektronik devre simülasyon programında kurunuz. LCD ekranda yazılan yazının `lcd.scrollDisplayRight()` komutuyla sağdan sola kaydırma uygulamasının program kodlarını elektronik devre simülasyon programında "Kod" bölümüne yazınız. Tüm işlemler bittikten sonra "Simülasyonu Başlat" butonu ile devreyi çalıştırınız. LCD durumunu gözlemleyiniz.



### Değerlendirme

Çalışmalarınız öğrenme biriminin sonunda yer alan Kontrol Listesi kullanılarak değerlendirilecektir. Çalışmalarınızı yaparken değerlendirme ölçütlerini dikkate alınız.

## 2.15.3. LDR ile Ölçülen Işık Şiddetinin LCD Ekranda Gösterilmesi

## 46. Uygulama

Uygulamanın amacı, LDR üzerine gelen ışık şiddetini sayısal değere çevirip yüzdelik değer olarak LCD ekran üzerinde gösteren bir çalışma gerçekleştirmektir.

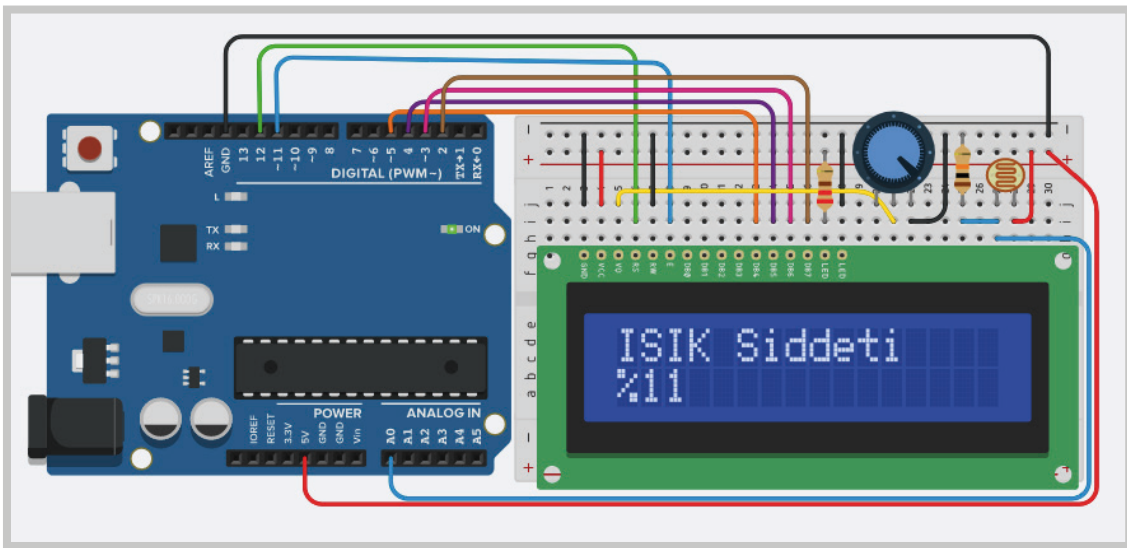
## Gerekli Devre Elemanları

- 1 adet mikrodenetleyici kart
- 1 adet breadboard
- 1 adet 220Ω (kırmızı, kırmızı, kahverengi, altın rengi) direnç
- 1 adet 10KΩ (kahverengi, siyah, turuncu, altın rengi) direnç
- 1 adet LCD display
- 1 adet LDR
- 1 adet potansiyometre
- Bağlantı kabloları

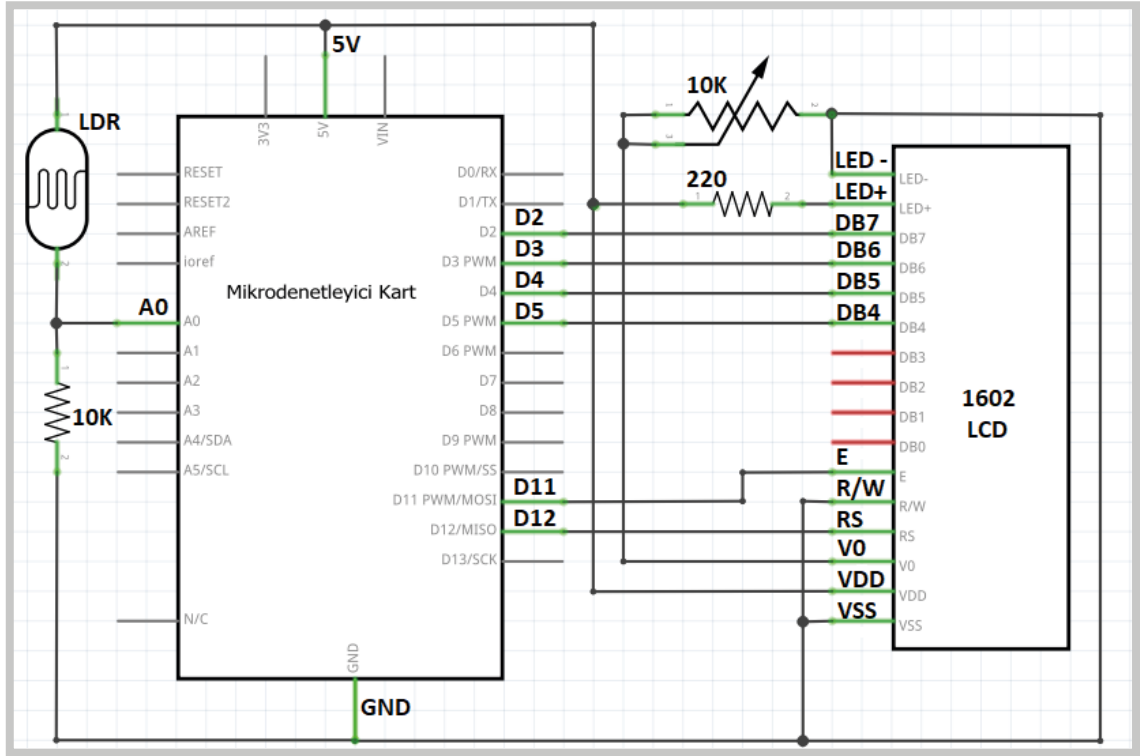
**1. Adım:** Görsel 2.106 ve Görsel 2.107'de, LDR üzerine gelen ışık şiddetini sayısal değere çevirip, yüzdelik değer olarak LCD ekran üzerinde gösteren uygulama görülmektedir.

**2. Adım:** Bu uygulamanın kodları incelendiğinde;

- `int LDR = analogRead(A0)` ; komutu ile mikrodenetleyici kartın A0 analog girişine bağlanan LDR'nin, ışığa bağlı olarak gönderdiği elektriksel sinyalin sayısal değerini, "LDR" isimli değişkenine aktarmaktadır.
- `int seviye=map(LDR,150,600,0,100)` ; komutu ile de "LDR" değişkeni içindeki sayısal veri 0 ile 100 arasında yeniden örneklenerek yüzdelik dilime çevrilmekte ve "seviye" değişkenine aktarılmaktadır.
- `lcd.print(seviye)` ; komutu ile "seviye" değişkeni LCD ekranda yazdırılır.
- LDR ile hassas bir ölçüm yapabilmek için LDR'nin minimum ve maksimum değerlerinin öğrenilip `map(LDR,min,mak,0,100)` ; komutu içindeki "min" ve "mak" ile ifade edilen yerlerde kullanılması gerekir.



Görsel 2.106: 1602 LCD ekran ile LDR uygulama devresi



Görsel 2.107: 1602 LCD ekran ile LDR uygulamasının elektriksel devresi

**3. Adım:** LCD ekran ile LDR uygulamasının program kodları aşağıdaki gibidir.

```
#include <LiquidCrystal.h>

const int rs = 12, en = 11, d4 = 5, d5 = 4, d6 = 3, d7 = 2;
LiquidCrystal lcd(rs, en, d4, d5, d6, d7);

void setup() {

  lcd.begin(16, 2);
  lcd.print("ISIK Siddeti");
}

void loop() {
  int LDR = analogRead(A0);
  int seviye = map(LDR, 150, 600, 0, 100);

  lcd.setCursor(0, 1);
  lcd.print("%");
  lcd.setCursor(1, 1);
  lcd.print(seviye);
}
```





### Sıra Sizde



HTTP://KITAP.EBA.GOV.TR/KODSOR.PHP?KOD=20797

Görsel 2.106'daki LCD ekran ile LDR uygulamasını elektronik devre simülatör programında kurunuz. Bu uygulamaya ait program kodlarını elektronik devre simülatör programında "Kod" bölümüne yazınız. Tüm işlemler bittikten sonra "Simülasyonu Başlat" butonu ile devreyi çalıştırınız. LDR üzerine tıklayarak ışık seviyesini değiştiriniz ve LCD ekran üzerindeki değişimleri gözlemleyiniz.



### Değerlendirme

Çalışmalarınız öğrenme biriminin sonunda yer alan Kontrol Listesi kullanılarak değerlendirilecektir. Çalışmalarınızı yaparken değerlendirme ölçütlerini dikkate alınız.



### Sıra Sizde

Görsel 2.108 veya Görsel 2.109'daki LCD ekran ile LDR uygulamasındaki devreyi breadboard üzerine kurunuz. Bu uygulamaya ait program kodlarını mikrodenetleyici IDE programında yazınız. Tüm işlemler bittikten sonra programı mikrodenetleyici karta yükleyiniz. LDR üzerini kapatıp açarak ışık seviyesini değiştiriniz ve LCD ekran üzerindeki değişimleri gözlemleyiniz.



### Değerlendirme

Çalışmalarınız öğrenme biriminin sonunda yer alan Kontrol Listesi kullanılarak değerlendirilecektir. Çalışmalarınızı yaparken değerlendirme ölçütlerini dikkate alınız.

### 2.15.4. Sıcaklık Sensörü ile Okunan Değerlerin LCD Ekranda Gösterilmesi

Sıcaklık sensörleri, başta dijital termometreler olmak üzere sıcaklığın sayısal değer olarak ifade edilmesi gereken birçok devrede kullanılır. Bu bölümdeki sıcaklık ölçüm uygulamalarında, TMP36 sensörü kullanılacaktır. TMP36 sıcaklık sensörünün katalog değerleri incelendiğinde, -40 ile +125 derece arasında ölçüm yapabildiği görülür.



### 47. Uygulama

Uygulamanın amacı, sıcaklık sensöründen gelen sıcaklık değerini LCD ekran üzerinde gösteren bir çalışma gerçekleştirmektir.

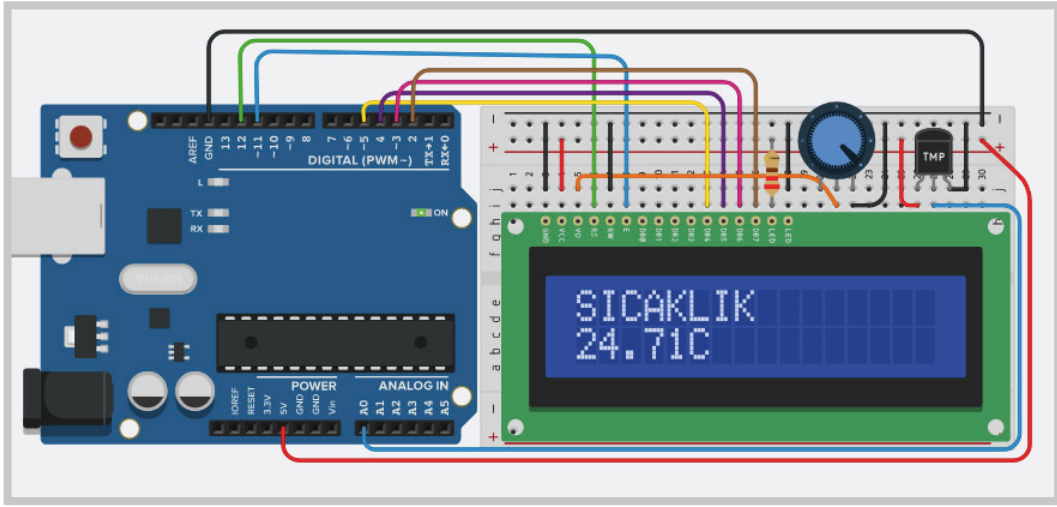
#### Gerekli Devre Elemanları

- 1 adet mikrodenetleyici kart
- 1 adet breadboard

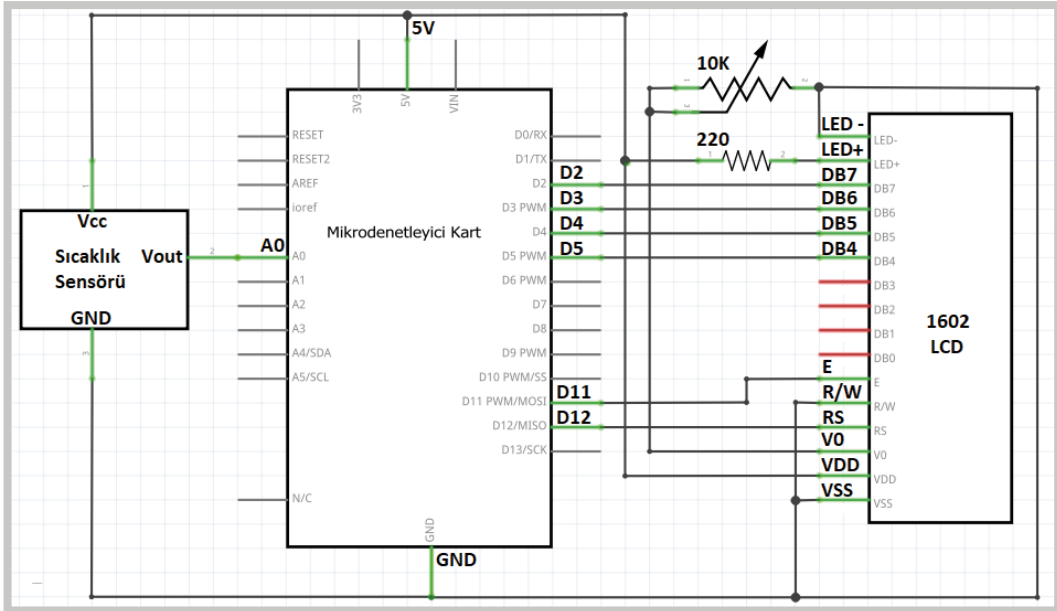
## 2. Öğrenme Birimi: Mikrodenetleyici Kart Programlama

- 1 adet 220Ω (kırmızı, kırmızı, kahverengi, altın rengi) direnç
- 1 adet 10KΩ (kahverengi, siyah, turuncu, altın rengi) direnç
- 1 adet LCD display
- 1 adet sıcaklık sensörü
- 1 adet potansiyometre
- Bağlantı kabloları

**1. Adım:** Görsel 2.108 ve Görsel 2.109'da sıcaklık sensöründen gelen sıcaklık değerini LCD ekran üzerinde gösteren uygulama görülmektedir.



Görsel 2.108: 1602 LCD ekran ile sıcaklık sensörü uygulama devresi



Görsel 2.109: 1602 LCD ekran ile sıcaklık sensörü uygulamasının elektriksel devresi

**2. Adım:** LCD ekran ile sıcaklık sensörü uygulamasının program kodları aşağıdaki gibidir.

```
#include <LiquidCrystal.h>

const int rs = 12, en = 11, d4 = 5, d5 = 4, d6 = 3, d7 = 2;
LiquidCrystal lcd(rs, en, d4, d5, d6, d7);

void setup() {
  lcd.begin(16, 2);
  lcd.print("SICAKLIK");
}

void loop() {
  int sensor = analogRead(A0);
  float analog = sensor * 5.0;
  analog = analog / 1024.0;

  float c = (analog - 0.5) * 100;
  lcd.setCursor(0, 1);
  lcd.print(c);
  lcd.setCursor(5, 1);
  lcd.print("C");
}
```



### Sıra Sizde

Görsel 2.108'deki LCD ekran ile sıcaklık sensörü uygulamasını elektronik devre simülatör programında kurunuz. Bu uygulamaya ait program kodlarını elektronik devre simülatör programında "Kod" bölümüne yazınız. Tüm işlemler bittikten sonra "Simülasyonu Başlat" butonu ile devreyi çalıştırınız. Sıcaklık üzerine tıklayarak sıcaklık seviyesini değiştiriniz ve LCD ekran üzerindeki değişimleri gözlemleyiniz.



### Değerlendirme

Çalışmalarınız öğrenme biriminin sonunda yer alan Kontrol Listesi kullanılarak değerlendirilecektir. Çalışmalarınızı yaparken değerlendirme ölçütlerini dikkate alınız.



### Sıra Sizde

Görsel 2.108 veya Görsel 2.109'daki LCD ekran ile sıcaklık sensörü uygulamasındaki devreyi breadboard üzerine kurunuz. Bu uygulamaya ait program kodlarını mikrodenetleyici IDE programında yazınız. Tüm işlemler bittikten sonra programı mikrodenetleyici karta yükleyiniz. Sıcaklık sensörünü ısıtıp soğutarak sıcaklık seviyesini değiştiriniz ve LCD ekran üzerindeki değişimleri gözlemleyiniz.



### Değerlendirme

Çalışmalarınız öğrenme biriminin sonunda yer alan Kontrol Listesi kullanılarak değerlendirilecektir. Çalışmalarınızı yaparken değerlendirme ölçütlerini dikkate alınız.

## 2.15.5. Mesafe Sensöründe Okunan Değerlerin LCD Ekranda Gösterilmesi



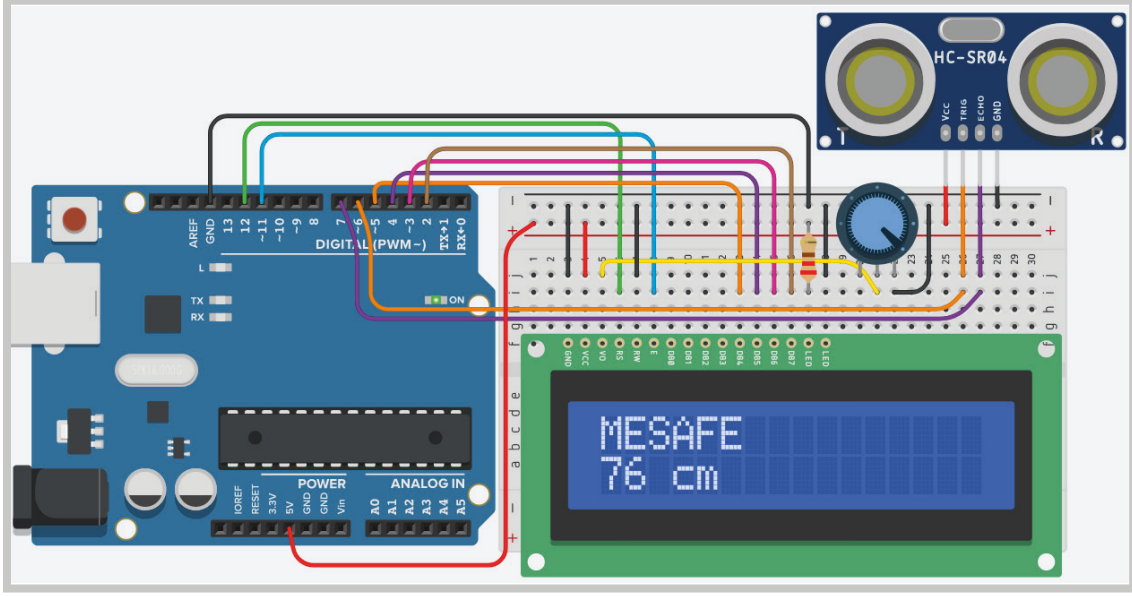
### 48. Uygulama

Uygulamanın amacı, LCD ekran ile mesafe sensörünü kullanan bir çalışma gerçekleştirmektir.

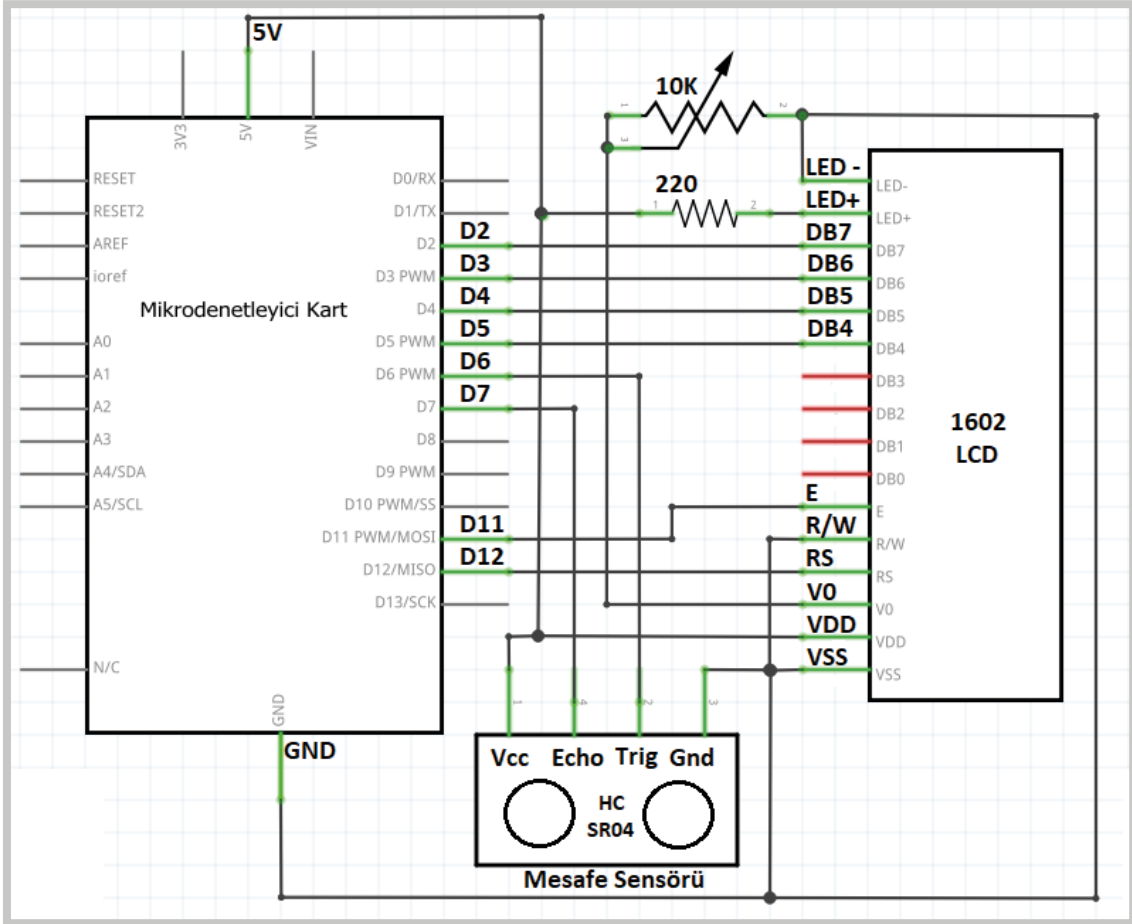
#### Gerekli Devre Elemanları

- 1 adet mikrodenetleyici kart
- 1 adet breadboard
- 1 adet 220Ω (kırmızı, kırmızı, kahverengi, altın rengi) direnç
- 1 adet mesafe sensörü
- 1 adet LCD display
- 1 adet potansiyometre
- Bağlantı kabloları

**1. Adım:** Ultrasonik mesafe sensöründen okunan değerleri LCD ekranda görüntülemek için Görsel 2.110 ve Görsel 2.111'de uygulama devresi verilmiştir.



Görsel 2.110: 1602 LCD ekran ile mesafe sensörü uygulama devresi



Görsel 2.111: 1602 LCD ekran ile mesafe sensörü uygulamasının elektriksel devresi

**2. Adım:** Bu uygulamaya ait program kodları incelendiğinde;

- “`void setup ()`” fonksiyonu içinde `pinMode (6, OUTPUT)` ; komutuyla mikrodenetleyici kartın “trig” pini için kullanılan D6 portu çıkış, `pinMode (7, INPUT)` ; komutuyla “echo” pini için kullanılan D7 portu giriş yapılıır.
- Mesafe sensörüyle cisim arasındaki mesafe hesaplanırken “`void loop ()`” fonksiyonu içinde öncelikle `digitalWrite (6,1)` ; komutuyla karşı tarafa bir ses dalgası gönderilir.
- 1 milisaniye beklenip `digitalWrite (6,0)` ; komutuyla bu ses dalgası kesilir.
- Gönderilen sesi almak için `pulseIn (7,1)` ; komutu kullanılarak “sure” isimindeki değişkene aktarılır.
- Mesafe hesaplaması yapılırken “mesafe” isminde bir değişken oluşturularak `int mesafe = (sure/2) / 28.97` ; komutuyla “sure” değişkenin yarısını alınıp 28.97 sabit sayısıyla çarpılır.
- Burada sesin gidip gelme mesafesinden dolayı “sure” değişkenin yarısı alınır.
- Komutta kullanılan sabit sayı ise ortam sıcaklığına göre hesaplanır.
- `lcd.print(mesafe)` ; komutuyla “mesafe” değişkeni içindeki değer LCD ekrana yazdırılır.

**3. Adım:** LCD kullanılarak yapılan mesafe ölçme uygulamasının program kodları aşağıdaki gibidir.

```
#include <LiquidCrystal.h>

const int rs = 12, en = 11, d4 = 5, d5 = 4, d6 = 3, d7 = 2;
LiquidCrystal lcd(rs, en, d4, d5, d6, d7);

void setup() {

    lcd.begin(16, 2);
    pinMode(6, OUTPUT); //Trig
    pinMode(7, INPUT); //Echo

}

void loop() {
    lcd.clear();
    lcd.setCursor(0, 0);
    lcd.print("MESAFE");
    digitalWrite(6,1); delay(1);
    digitalWrite(6,0);
    int sure=pulseIn(7,1);
    int mesafe=(sure/2)/28.97;
    delay(10);

    lcd.setCursor(0, 1);
    lcd.print(mesafe);
    lcd.setCursor(5, 1);
    lcd.print("cm");
}
```



### Sıra Sizde



<http://kitap.eba.gov.tr/KodSor.php?KOD=20799>

Görsel 2.110'daki LCD ekran ile mesafe sensörü uygulamasını elektronik devre simülör programında kurunuz. Bu uygulamaya ait program kodlarını elektronik devre simülör programında "Kod" bölümüne yazınız. Tüm işlemler bittikten sonra "Simülasyonu Başlat" butonu ile devreyi çalıştırınız. Mesafe sensörünün üzerine tıklayarak engel değerlerini değiştiriniz ve LCD ekrandaki değişimleri gözlemleyiniz.



### Değerlendirme

Çalışmalarınız öğrenme biriminin sonunda yer alan Kontrol Listesi kullanılarak değerlendirilecektir. Çalışmalarınızı yaparken değerlendirme ölçütlerini dikkate alınız.



### Sıra Sizde

Görsel 2.110 veya Görsel 2.111'deki LCD ekran ile mesafe sensörü uygulamasındaki devreyi breadboard üzerine kurunuz. Bu uygulamaya ait program kodlarını mikrodenetleyici IDE programında yazınız. Tüm işlemler bittikten sonra programı mikrodenetleyici karta yükleyiniz. Ultrasonik mesafe sensörünün önüne bir engel yaklaştırıp uzaklaştırarak LCD ekrandaki değişimleri gözlemleyiniz.



### Değerlendirme

Çalışmalarınız öğrenme biriminin sonunda yer alan Kontrol Listesi kullanılarak değerlendirilecektir. Çalışmalarınızı yaparken değerlendirme ölçütlerini dikkate alınız.



### 49. Uygulama

Uygulamanın amacı LCD ekran ile boy ölçme ile ilgili bir çalışma gerçekleştirmektir.

#### Gerekli Devre Elemanları

- 1 adet mikrodenetleyici kart
- 1 adet breadboard
- 1 adet 220Ω (kırmızı, kırmızı, kahverengi, altın rengi) direnç
- 1 adet mesafe sensörü
- 1 adet LCD display
- 1 adet sıcaklık sensörü
- 1 adet potansiyometre
- Bağlantı kabloları

## 2. Öğrenme Birimi: Mikrodenetleyici Kart Programlama

**1. Adım:** Ultrasonik mesafe sensörü ve LCD ekran kullanılarak, boy ölçme uygulaması için Görsel 2.110'daki uygulamanın mevcut program kodlarına, `int boy=220-mesafe`; komutu eklenir. Yukarıya konulan sensörün ölçtüğü değer 220 cm'den çıkarılarak sensörün altına konulan cismin boyu ölçülmüş olur. Uygulama kodlarına göre maksimum 220 cm ölçüm yapar. Daha yüksek cisimler ölçülmek istenirse program kodlarındaki 220 değeri değiştirilmelidir.

**2. Adım:** LCD ekran ile boy ölçme uygulamasının program kodları aşağıdaki gibidir.

```
#include <LiquidCrystal.h>

const int rs = 12, en = 11, d4 = 5, d5 = 4, d6 = 3, d7 = 2;
LiquidCrystal lcd(rs, en, d4, d5, d6, d7);

void setup() {
  lcd.begin(16, 2);
  pinMode(6, OUTPUT); //Trig
  pinMode(7, INPUT); //Echo
}

void loop() {
  lcd.clear();
  lcd.setCursor(0, 0);
  lcd.print("BOYUNUZ");
  digitalWrite(6, 1); delay(1);
  digitalWrite(6, 0);
  int sure=pulseIn(7, 1);
  int mesafe=(sure/2)/28.97;
  int boy=220-mesafe;
  delay(10);

  lcd.setCursor(0, 1);
  lcd.print(boy);
  lcd.setCursor(3, 1);
  lcd.print("cm");
}
```



### Sıra Sizde

Görsel 2.110 veya Görsel 2.111'deki devreyi breadboard üzerine kurunuz. LCD ekran ile sosyal mesafe ölçme uygulamasının program kodlarını mikrodenetleyici IDE programında yazınız. Tüm işlemler bittikten sonra programı mikrodenetleyici karta yükleyiniz. Devreyi duvarda 220 cm yükseklikte tutunuz. Ultrasonik mesafe sensörünün önüne bir cisim koyup boyunun ölçüm değerini LCD ekran üzerinden gözlemleyiniz.



### Değerlendirme

Çalışmalarınız öğrenme biriminin sonunda yer alan Kontrol Listesi kullanılarak değerlendirilecektir. Çalışmalarınızı yaparken değerlendirme ölçütlerini dikkate alınız.



## 50. Uygulama

Uygulamanın amacı LCD ekran ile sosyal mesafe ölçen bir çalışma gerçekleştirmektir.

### Gerekli Devre Elemanları

- 1 adet mikrodenetleyici kart
- 1 adet breadboard
- 1 adet 220Ω (kırmızı, kırmızı, kahverengi, altın rengi) direnç
- 1 adet mesafe sensörü
- 1 adet LCD display
- 1 adet potansiyometre
- Bağlantı kabloları

**1. Adım:** Ultrasonik mesafe sensörü ve LCD ekran kullanılarak, sosyal mesafe ölçme uygulaması için Görsel 2.110'daki uygulamanın mevcut program kodlarına `if(mesafe>=150) lcd.print("S.Mesafe Yeterli")`; komutu eklenir.

**2. Adım:** Burada ölçülen mesafe 150 cm'ye eşit veya 150 cm'den büyükse LCD ekranda "S. Mesafeyi Yeterli." yazısı çıkar. Sosyal mesafe 150 cm'den küçükse LCD ekranda "S.Mesafeyi Artır." yazısı çıkar.

**3. Adım:** LCD ekran ile sosyal mesafe ölçme uygulamasının program kodları aşağıdaki gibidir.

```
#include <LiquidCrystal.h>

const int rs = 12, en = 11, d4 = 5, d5 = 4, d6 = 3, d7 = 2;
LiquidCrystal lcd(rs, en, d4, d5, d6, d7);

void setup() {
  lcd.begin(16, 2);
  pinMode(6, OUTPUT); //Trig
  pinMode(7, INPUT); //Echo
}

void loop() {
  lcd.clear();
  digitalWrite(6, 1); delay(1);
  digitalWrite(6, 0);
  int sure=pulseIn(7, 1);
  int mesafe=(sure/2)/28.97;
  lcd.setCursor(0, 0);
  lcd.print("Mesafe: ");
  lcd.setCursor(10, 0);
  lcd.print(mesafe);
  lcd.setCursor(0, 1);
  if(mesafe>=150) lcd.print("S.Mesafe Yeterli");
  else lcd.print("S.Mesafeyi ARTIR");
  delay(10);
}
```

## 2. Öğrenme Birimi: Mikrodenetleyici Kart Programlama



### Sıra Sizde

Görsel 2.110 veya Görsel 2.111'deki devreyi breadboard üzerine kurunuz. LCD ekran ile boy ölçme uygulamasının program kodlarını mikrodenetleyici IDE programında yazınız. Tüm işlemler bittikten sonra programı mikrodenetleyici karta yükleyiniz. Devreyi duvarda 220 cm yüksekte tutunuz. Ultrasonik mesafe sensörünün önüne bir cisim koyup boy ölçüm değerini LCD ekran üzerinden gözlemleyiniz.



### Değerlendirme

Çalışmalarınız öğrenme biriminin sonunda yer alan Kontrol Listesi kullanılarak değerlendirilecektir. Çalışmalarınızı yaparken değerlendirme ölçütlerini dikkate alınız.



### Sıra Sizde

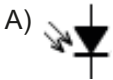
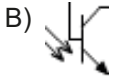
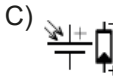
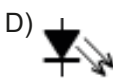
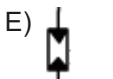
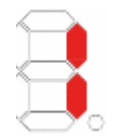
Görsel 2.110 veya Görsel 2.111'deki LCD ekran ile mesafe sensörü uygulamasındaki devreyi breadboard üzerine kurunuz. Bu uygulamaya ait program kodlarını mikrodenetleyici IDE programında yazınız. Tüm işlemler bittikten sonra programı mikrodenetleyici karta yükleyiniz. Ultrasonik mesafe sensörünün önüne bir engel yaklaştırıp uzaklaştırarak LCD ekrandaki değişimleri gözlemleyiniz.



### Değerlendirme

Çalışmalarınız öğrenme biriminin sonunda yer alan Kontrol Listesi kullanılarak değerlendirilecektir. Çalışmalarınızı yaparken değerlendirme ölçütlerini dikkate alınız.

## ÖLÇME VE DEĞERLENDİRME

1. Aşağıdakilerden hangisi değer döndürmeyen fonksiyon tanımlamak için kullanılır?
  - A) Setup
  - B) int
  - C) Loop
  - D) Void
  - E) Type
2. Aşağıdakilerden hangisi 16'lık sayı sisteminin genel adıdır?
  - A) Hexadecimal
  - B) Binary
  - C) Octal
  - D) Decimal
  - E) Double
3. Aşağıdakilerden hangisi mikrodenetleyici IDE programında, virgüllü sayıları tutan değişkenleri tanımlamak için kullanılır?
  - A) long
  - B) byte
  - C) float
  - D) int
  - E) string
4. Aşağıdakilerden hangisi LED devre elemanı sembolüdür?
  - A) 
  - B) 
  - C) 
  - D) 
  - E) 
5. Üzerinde kırmızı-kırmızı-kahverengi-altın renkleri olan direncin değeri kaç ohmdur ( $\Omega$ )?
  - A) 22
  - B) 33
  - C) 220
  - D) 330
  - E) 22000
6. Aşağıdaki direnç çeşitlerinden hangisinin direnç değeri elle ayarlanabilir niteliktedir?
  - A) NTC
  - B) LDR
  - C) VDR
  - D) POT
  - E) PTC
7. Aşağıdaki devre elemanlarından hangisi üzerine her basıldığında sadece bir defalığına aktif olur?
  - A) Anahtar
  - B) Sigorta
  - C) Buton
  - D) Lamba
  - E) Röle
8. Ortak katotlu bir displayde "1" rakamını gösterebilmek için hangi segmentlerin yanması gerekir?
 
  - A) a, b
  - B) a, c
  - C) c, d
  - D) b, c
  - E) d, e

9. **analogWrite()** komutuyla kullanıma en uygun port aşağıdakilerden hangisidir?

- A) D3
- B) D0
- C) A0
- D) A1
- E) D1

10. Aşağıdakilerden hangisi kütüphane çağırmak için kullanılır?

- A) #define
- B) Void
- C) #include
- D) Setup
- E) pinMode

11. Aşağıdakilerden hangisi 1,5 saniye bekleme yapmak için kullanılan komuttur?

- A) delay(1500)
- B) delay(1.5)
- C) delay(150)
- D) delay(15)
- E) delay(15000)

12. Aşağıdaki komutlardan hangisi dijital pinlerin giriş veya çıkış olduğunu belirlemek için kullanılır?

- A) delay
- B) setup
- C) loop
- D) pinMode
- E) digitalWrite

13. Aşağıdakilerden hangisi tam sayı (integer) değişken tanımlamak için kullanılır?

- A) Difane
- B) Cons
- C) int
- D) boolean
- E) long

14. I. Kırmızı  
II. Yeşil  
III. Mavi

**RGB LED kullanılarak beyaz ışık yakılmak istendiğinde yukarıdaki renklerden hangileri yakılmalıdır?**

- A) Yalnız I
- B) I ve II
- C) II ve III
- D) I ve III
- E) I, II ve III

15. Mikrodenetleyici IDE programında 10 numaralı dijital pindeki lojik değeri okuyan komut, aşağıdakilerden hangisinde doğru verilmiştir?

- A) AnalogRead(10);
- B) digitalWrite(10, 1);
- C) digitalRead(10);
- D) pinMode(10, OUTPUT);
- E) pinMode(10, INPUT);

16. Mikrodenetleyici IDE programında değişkenlerin seri port ekranında görüntülenmesini sağlayan komut, aşağıdakilerden hangisidir?

- A) Serial.print
- B) Serial.begin
- C) Serial.write
- D) Serial.show
- E) Serial.read

17. Aşağıdakilerden hangisi Mikrodenetleyici IDE programındaki ana fonksiyon blokudur?

- A) Start();
- B) Begin();
- C) Default();
- D) Loop();
- E) Main();

18. Aşağıdaki dijital pin uçlarından hangisi PWM (analog çıkış ucu) olarak kullanılmaz?

- A) D6
- B) D9
- C) D10
- D) D11
- E) D12

19. Mikrodenetleyici IDE programında yazının LCD ekranında görüntülenmesini sağlayan komut aşağıdakilerden hangisidir?

- A) lcd.print
- B) lcd.begin
- C) lcd.write
- D) lcd.show
- E) lcd.read

20. Aşağıdakilerden hangisi 1602 LCD için bağlantı uçlarından birisi değildir?

- A) Vss
- B) Vdd
- C) Rs
- D) R/W
- E) A0

## SIRA SİZDE KONTROL LİSTESİ

### Kontrol Listesi

Ölçütler	Evet	Hayır
Uygun devre elemanları ve sensörlerin seçimini yapar.		
Devre elemanlarını breadboarda yerleştirir.		
Program kodlarını, simülatör programında uygun bir şekilde yazar.		
Simülasyonu başlatarak yazılan programın çalışmasını gözlemler.		
Devrede kullanılan sensörler ve diğer devre elemanlarının çalışmasını gözlemler.		
Çalışmada iş sağlığı ve güvenliği kurallarına dikkat eder.		
Zamanı verimli kullanır.		

# KONULAR

- 3.1. EĐİTSEL ROBOT BİLEŐENLERİ
- 3.2. EĐİTSEL ROBOTUN DEVRE ŐEMASI
- 3.3. EĐİTSEL ROBOTUN MONTAJI
- 3.4. MOTORLARI SADECE GERİ YÖNDE ÇALIŐTIRMA
- 3.5. EĐİTSEL ROBOTUN PROGRAMLANMASI
- 3.6. EĐİTSEL ROBOTUN UZAKTAN KUMANDA İLE KONTROLÜ
- 3.7. EĐİTSEL ROBOT İLE ENGELDEN KAÇMA
- 3.8. EĐİTSEL ROBOT İLE ÇİZGİ İZLEME
- 3.9. EĐİTSEL ROBOTUN BLUETOOTH KONTROLÜ
- 3.10. SERVO MOTOR İLE ROBOT UYGULAMALARI

## NELER ÖĐRENECEKSİNİZ?

- Eđitsel robot
- Eđitsel robotu oluŐturan bileŐenler
- Eđitsel robotların kontrolü
- Eđitsel robotun uzaktan kumanda ile kontrolü
- Engelden kaçan robotun yapımı
- Çizgi izleyen robotun yapımı
- Eđitsel robotun bluetooth ile kontrolü
- Eđitsel robot gelişmek için gerekli bilgi ve beceriye sahip olma



### Hazırlık Çalışmaları

1. Çevrenizde otonom olarak çalışan ve belirli bir görevi yerine getiren robotlar hangileridir? Açıklayınız.
2. Hayatınızı kolaylaŐtıracak hangi robotlar geliştirilebilir? Bu konuda neler düşünöyorsunuz?



## 3. ÖĞRENME BİRİMİ

Robot Tabanlı  
Proje Geliştirme

### Temel Kavramlar

- Mikrodenetleyici kartı
- Uygulama kartları
- Robotlar
- Robot türleri
- Eğitsel robotlar

#### 3.1. EĞİTSEL ROBOT BİLEŞENLERİ

Temel olarak eğitsel bir robot; bir gövde üzerine tutturulmuş, başta hareket sağlayacak motorlar olmak üzere mikrodenetleyici kartı, sensörler, motor sürücü gibi çeşitli bileşenlerden oluşur. Bu bölümde eğitsel robotta kullanılacak robot bileşenleri tanıtılacaktır. Eğitsel robotta kullanılacak malzeme listesi Görsel 3.1'de gösterildiği gibi aynı zamanda Tablo 3.1'de de ayrıntılı olarak verilmiştir. Robot gövdesini ve sensör aparatlarını 3 boyutlu çizim programında çizmek isteyenler için ilgili ölçüler, kitabın ekler bölümünde verilmiştir.



Görsel 3.1: Eğitsel robotun gövdesi ve gerekli robot bileşenleri

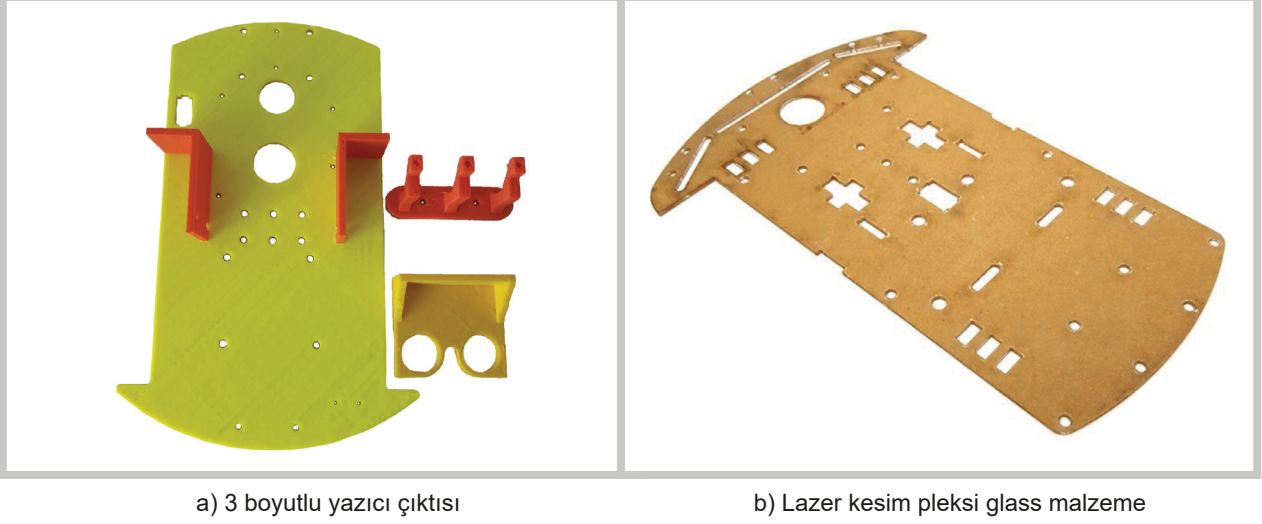
Tablo 3.1: Eğitsel Robotta Kullanılacak Malzeme Listesi

No	Kriterler	Adet	Açıklama
1	Robot gövdesi	1	3 Boyutlu yazıcı çıktısı veya farklı bir materyalden
2	Mikrodenetleyici kartı	1	Arduino Uno veya muadili
3	Motor sürücü	1	L298N veya muadili
4	Robot motorları	2	6-12V 250 rpm plastik redüktörlü motor veya muadili
5	Robot tekerlekleri	2	Çapı: 65 mm, Kalınlık: 30 mm veya muadili
6	Döner robot tekerleği	1	Montaj yüksekliği 35 mm, Tekerlek çapı 24 mm
7	Pil yatağı	1	2x 18650 Lion (3.7V) piller için
8	Açma kapama anahtarı	1	1.3 cm X 0.8 cm küçük anahtar veya muadili
9	M3 Vida ve somun	4	30 cm uzunluğunda (Motorların montajı için)
10	M3 Vida ve somun	4	10 cm uzunluğunda (Döner motor tekerleği montajı için)
11	M3 Vida ve somun	8	20 cm uzunluğunda (Motor sürücü ve Arduino için)
12	M3 Vida ve somun	2	15 cm uzunluğunda (Mesafe sensörü için)
13	M3 Havşa başlı metrik vida	2	6 cm uzunluğunda (Pil yatağı için)
14	M3 Vida ve somun	3	15 cm uzunluğunda (Çizgi izleme sensörleri için)
15	Dişi Dişi jumper kablo	20	20 cm uzunluğunda (Elektrik bağlantıları için)
16	Dişi Erkek jumper kablo	20	20 cm uzunluğunda (Elektrik bağlantıları için)
17	IR alıcı kod çözücü	1	Vs1838 IR alıcı veya muadili
18	Ultrasonik mesafe sensörü	1	HC-Sr04 veya muadili
19	Çizgi izleme sensörü	3	TCRT5000 tekli dijital çıkış sensör veya muadili
20	Bluetooth modülü	1	HC05, Hc06 veya muadili



#### 3.1.1. Robot Gövdesi

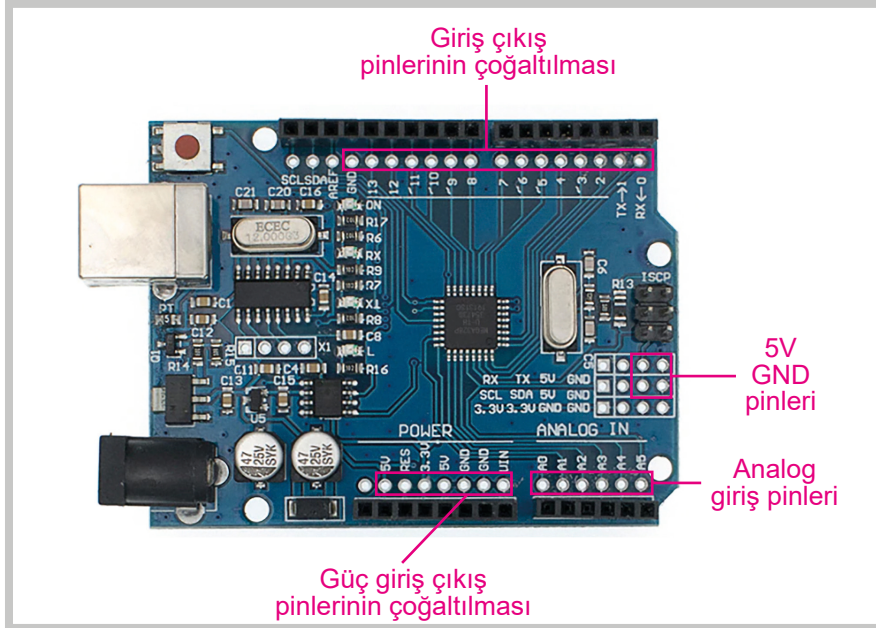
Robot gövdeleri, robotta kullanılan bileşenlerin vida, plastik kelepçe veya sıcak silikon gibi malzemeler kullanılarak üzerine tutturulduğu yüzeylerdir. Robot gövdeleri plastik, pleksi glass, bakır yüzeyli plaket gibi çok çeşitli malzemelerden yapılabilir. Bu öğrenme biriminde, eğitsel robotun parçalarının kolayca montajının yapılabileceği deliklere sahip 3 boyutlu yazıcı çıktısı kullanılacaktır (Görsel 3.2).



Görsel 3.2: Robot gövdesi

#### 3.1.2. Mikrodenetleyici Kartı

Mikrodenetleyici kartları, eğitsel robotta yazılan robot kontrol komutlarını yorumlayıp motor sürücüyü gerekli çalışma sinyallerini gönderen devre kartlarıdır. Eğitsel robotta kullanılacak sensörlerin besleme uçlarını mikrodenetleyici karta bağlantılarını yaparken kart üzerindeki pinlerin sayısı yeterli gelmeyebilir. Mikrodenetleyici kartlar üzerindeki pinler yeterli gelmediğinde ya Görsel 3.3'te görüldüğü gibi ilave pinler lehimlemek ya da Görsel 3.4'te görüldüğü gibi **Sensör Shield** adındaki ilave kartlardan kullanmak gerekir.



Görsel 3.3: Eğitsel robotta kullanılacak mikrodenetleyici kartın pinlerinin çoğaltılması





#### 3.1.4. Tekerlekler

Tekerlekler, robotun motorlarından aldığı dönme hareketini yola ileten robot bileşenleridir. Farklı yapı ve boyutlarda üretilir. Eğitsel robotta Görsel 3.8'deki tekerlekler kullanılacaktır. Tekerlek sayısı olarak bazı robotlarda 4 adet tekerlek kullanılabilir. Eğitsel robotta 2 adet tekerlek kullanılacağı için robotun dengesini sağlamak adına Görsel 3.9'daki döner robot tekerleklerinden (sarhoş tekerlek) kullanılacaktır.



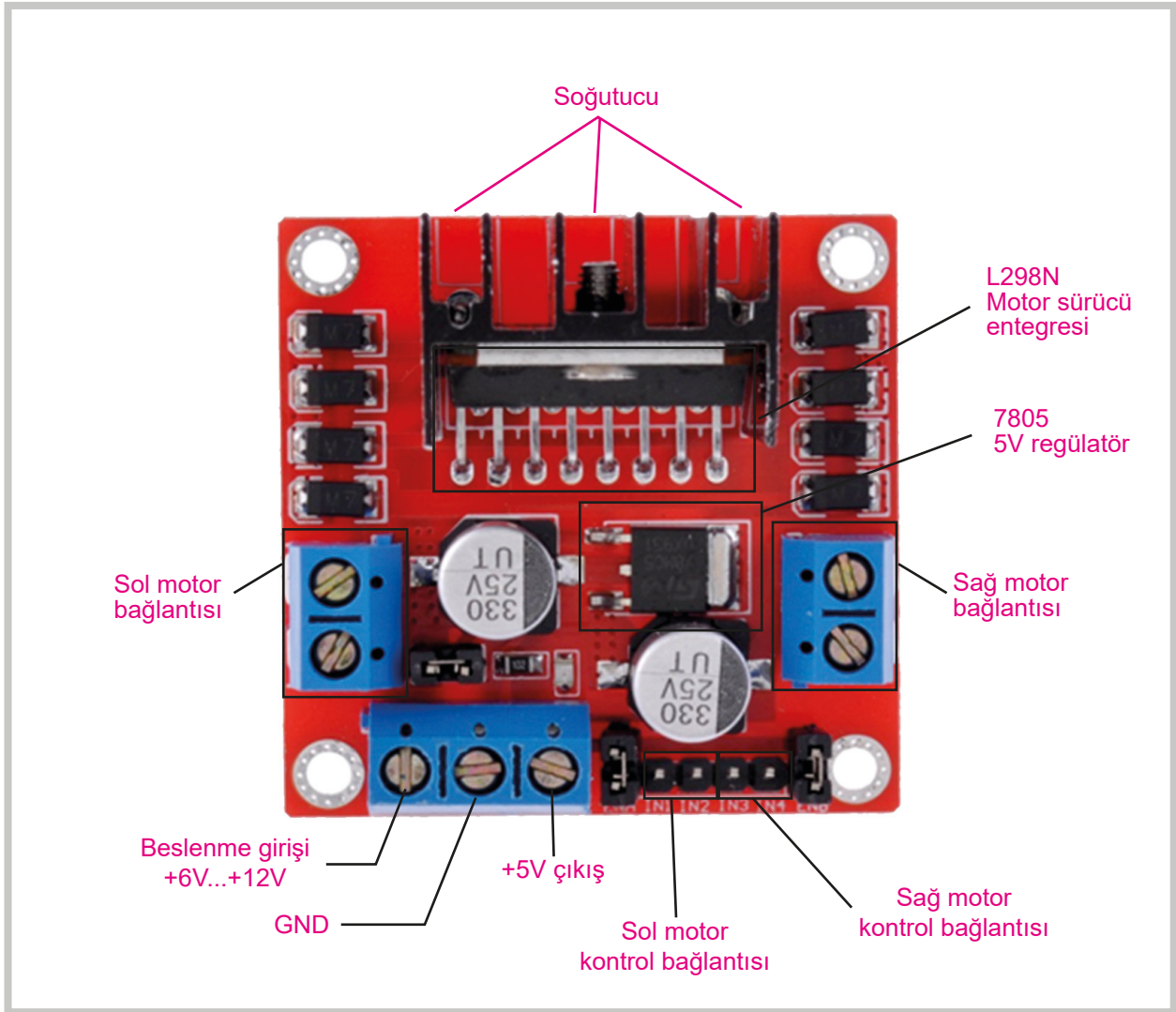
Görsel 3.8: Standart robot tekerlekleri



Görsel 3.9: Döner robot tekerleği (Sarhoş tekerlek)

#### 3.1.5. Motor Sürücü

Motor sürücüler, mikrodenetleyici kartından gelen motor kontrol sinyallerini motora ileten elektronik devrelerdir. Mikrodenetleyici kartının çıkış akımı motoru döndürebilecek seviyede olmadığından ve mikrodenetleyici kartları zarar görmesin diye motor sürücü devrelerine ihtiyaç duyulmuştur. Çok çeşitli özelliklerde (L293, L298N, TB6612 vb.) motor sürücüler mevcuttur. Eğitsel robotta Görsel 3.10'daki L298N motor sürücü devresi kullanılacaktır. Motor sürücüyü ilgili bağlantılar Görsel 3.10'da görülmektedir. Motor kabloları, motor uçları lehimlendikten sonra motor sürücüsü üzerinde bulunan klemenslere bağlanır. Motor sürücüde besleme amacıyla kullanılan 3 adet klemens bulunur. Bunlardan 2 tanesi pil yatağından gelen + ve – uç bağlantılarıdır. Motor sürücüde bulunan diğer besleme ucu ise +5V çıkışıdır. Bu çıkış, devreye fazladan sensör bağlandığında sensörlerin + beslemesini sağlamak için kullanılır. Eğitsel robotta bu çıkış kullanılmayacaktır.



Görsel 3.10: L298N Motor sürücü

#### 3.1.6. Enerji Kaynağı

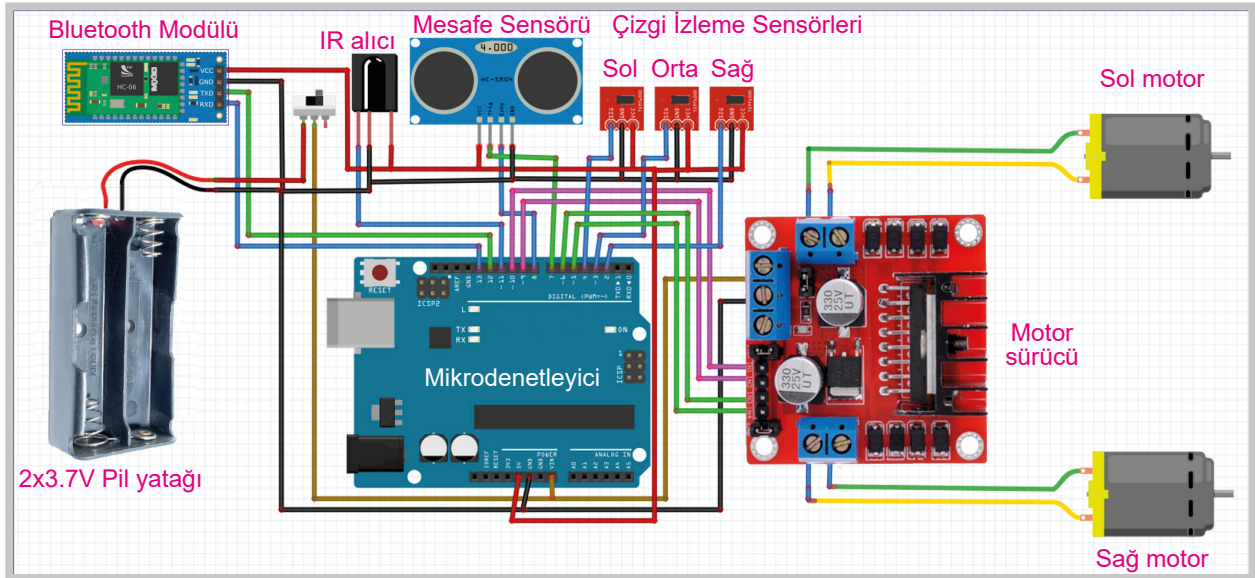
Robotların çalıştırılması için taşınabilir bir enerji kaynağına ihtiyaç vardır. Bu ihtiyacı karşılamak için şarjlı piller kullanmak uygun seçenektir. Robotlarda çok farklı boyut ve özelliklerde şarjlı piller (LiPo, Lion vb.) kullanılmaktadır. Eğitsel robotta GörSEL 3.11'de görülen 2 adet 3.7V 18650 Lion pillerden kullanılacaktır.



GörSEL 3.11: Lion 18650 piller, şarj cihazı ve pil yatağı

#### 3.2. EĞİTSEL ROBOTUN DEVRE ŞEMASI

Eğitsel robotun tüm sensörleri bağlandıktan sonraki devre şeması GörSEL 3.12'de görüldüğü gibidir. Robot bileşenlerinin mikrodenetleyici kartın hangi portlarına bağlanacağı Tablo 3.2'de ayrıntılı bir şekilde verilmiştir. Devre şemasında ve tabloda verilen bileşenlerin bağlantıları, ilk aşamada karmaşık gibi görünse de her bileşenin bağlantısı, sırası geldiğinde adım adım anlatılacaktır.



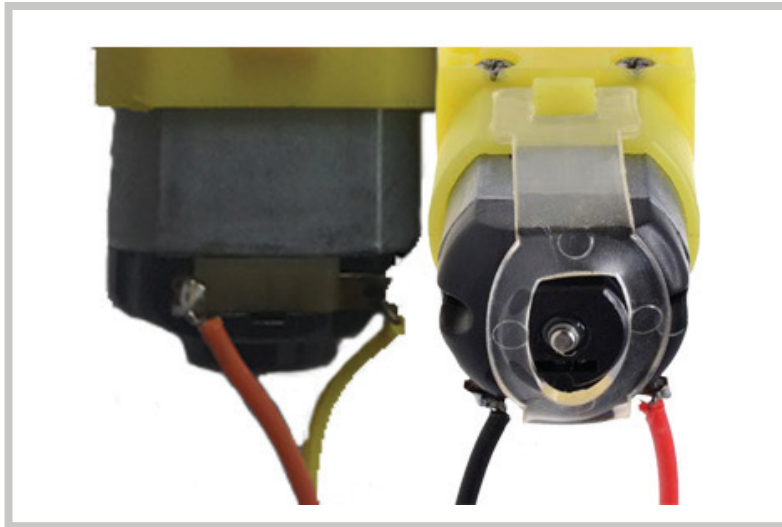
GörSEL 3.12: Eğitsel robotun devre şeması

Tablo 3.2: Mikrodenetleyici Portlarının Robot Bileşenlerine Bağlantısı

Pin No	Robot Bileşenlerinin Bağlantı Noktası
Vin	Pil yatağından anahtara, anahtar çıkışından motor sürücüsünün +12V besleme ucuna
GND	Pil yatağının (-) ucuna, tüm sensörlerin ve motor sürücünün GND ucuna
+5V	Tüm sensörlerin +5V ucuna
D2	Sağ taraftaki çizgi izleme sensörünün çıkış ucuna
D3	Orta taraftaki çizgi izleme sensörünün çıkış ucuna
D4	Sol taraftaki çizgi izleme sensörünün çıkış ucuna
D5	Sağ motor kontrolü için motor sürücüyeye (ileri yön için)
D6	Sağ motor kontrolü için motor sürücüyeye (geri yön için)
D7	Ultrasonik mesafe sensörü trig ucu
D8	Ultrasonik mesafe sensörü echo ucu
D9	Sol motor kontrolü için motor sürücüyeye (ileri yön için)
D10	Sol motor kontrolü için motor sürücüyeye (geri yön için)
D11	IR alıcı çıkış ucu
D12	Bluetooth modülü Tx bağlantısı
D13	Bluetooth modülü Rx bağlantısı

### 3.3. EĞİTSEL ROBOTUN MONTAJI

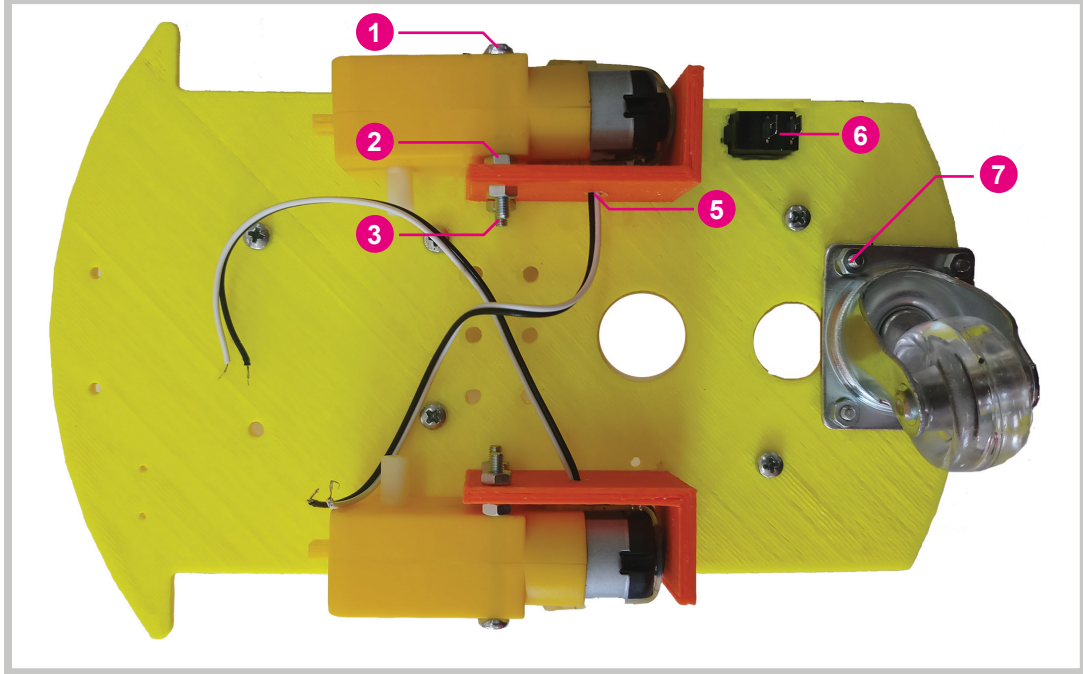
Eğitsel robotun montajı yapılırken, öncelikle motor kablolarının motor uçlarına lehimlenmesi gerekir. Görsel 3.13'te görüldüğü gibi motor kablolarının uçlarının sıyrılarak motor bağlantı uçlarındaki deliklere takılıp lehimlenmesi daha sağlıklı olacaktır.



Görsel 3.13: Eğitsel robotun motor uçlarının lehimlenmesi

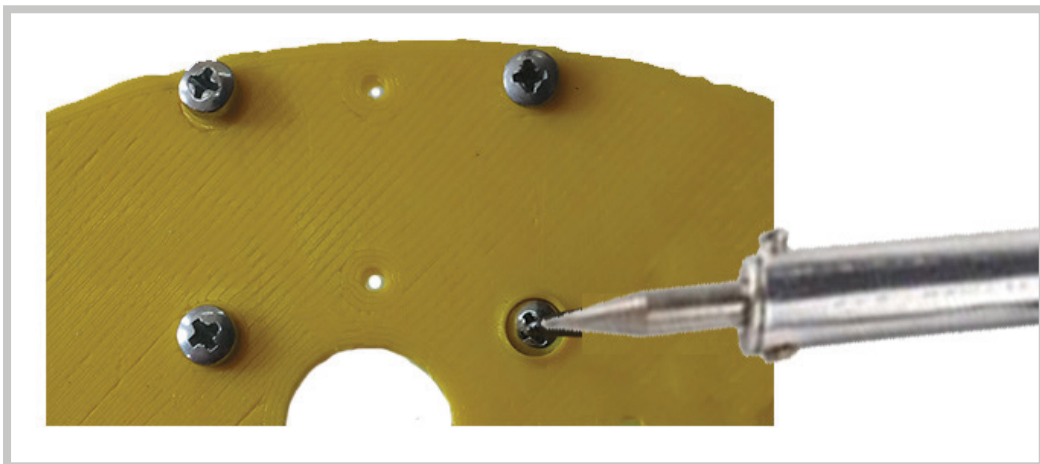
### 3. Öğrenme Birimi: Robot Tabanlı Proje Geliştirme

Motor kabloları lehimlendikten sonra Görsel 3.14'te görüldüğü gibi motorların, robot gövdesinde motorlar için ayrılmış bölüme vidalanması gerekmektedir. Motorları vidalarken 30 cm uzunluğundaki M3 vidalar kullanılır. Her bir motor, robot gövdesine 2 adet vidayla monte edilir. Motorun daha düzgün durması için Görsel 3.14'te 2 numarada gösterildiği gibi fazladan bir somun kullanılır. Motor kabloları 5 numarada gösterildiği gibi delikten geçirilir. Ayrıca 6 numarada gösterilen anahtar, robot gövdesine takılmalıdır.



Görsel 3.14: Motor, anahtar ve döner robot tekerleğinin robot gövdesine montajı

Robot gövdesine döner robot tekerleğini monte ederken 12 cm uzunluğundaki M3 vidalar kullanılır. Gövde üstüne yerleştirilecek pil yatağının montajını daha rahat yapmak için Görsel 3.15'te görüldüğü gibi vida başlarının lehim makinesiyle ısıtılarak gömülmesi gerekir.

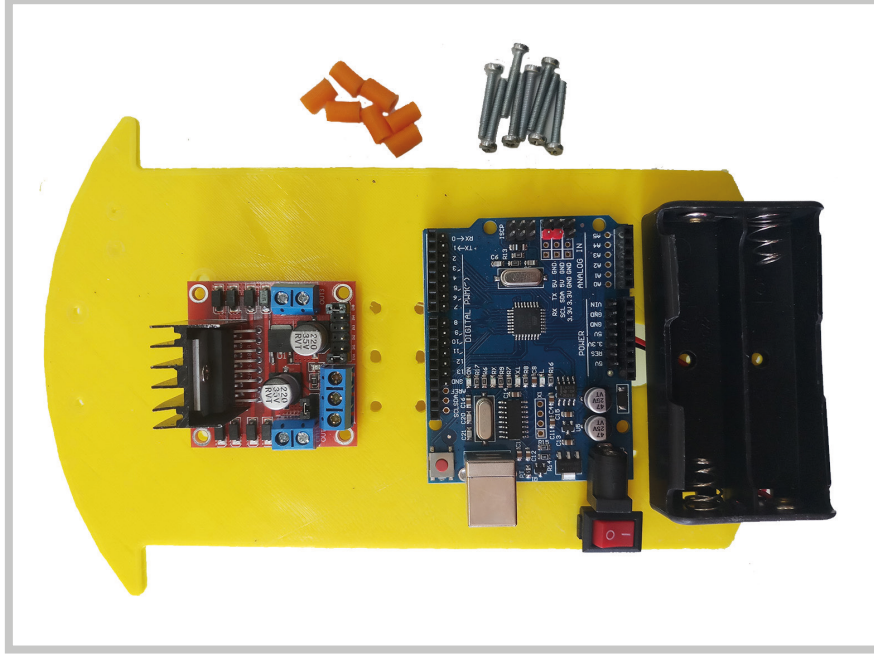


Görsel 3.15: Eğitsel robottaki döner robot tekerleğinin vida başlarının gömülmesi



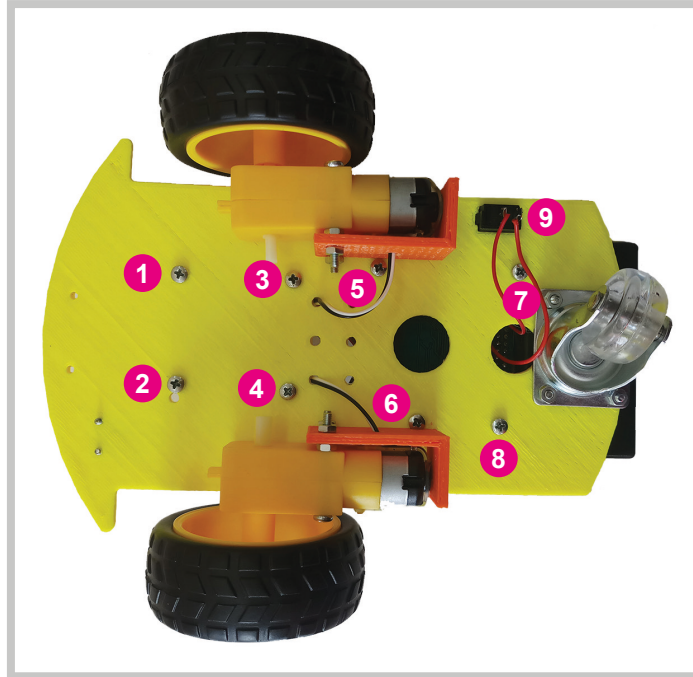
### 3. Öğrenme Birimi: Robot Tabanlı Proje Geliştirme

Eğitsel robot bileşenleri robot gövdesi üzerine Görsel 3.16'daki gibi yerleştirilir. Robot bileşenlerinin, robot gövdesine montajı yapılırken 20 cm uzunluğundaki M3 vidaları kullanılır.

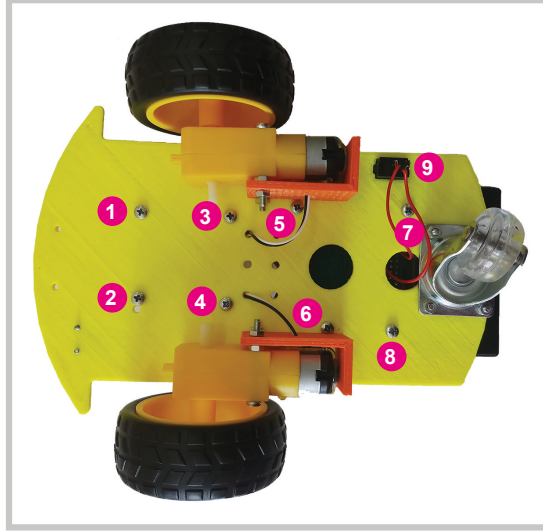


**Görsel 3.16: Eğitsel robot bileşenleri ve gövdeye yerleştirilmesi**

Görsel 3.17'de görüldüğü gibi robot gövdesinin alt kısmında vida delikleri vardır. Bu vida deliklerinden 1, 2, 3 ve 4 numaralı gösterilenler motor sürücü montajı için kullanılır. Yine robot gövdesindeki 5, 6, 7 ve 8 numaralı delikler ise mikrodenetleyici kartının montajı için kullanılır. Motor sürücü ve mikrodenetleyici kartın montajında, elektronik kartları robot gövdesinden belirli bir yükseklikte tutmak için Görsel 3.18'deki gibi yükseltme aparatları kullanılır. Bu şekilde yükseltme yapmak kablo bağlantılarını yaparken büyük kolaylık sağlar.



**Görsel 3.17: Eğitsel robotun alt bölümündeki vidaların gövdeye yerleştirilmesi**



Görsel 3.18: Eğitsel robot bileşenlerinin montajını yaparken kullanılacak yükseltme aparatları



#### Sıra Sizde

Görsel 3.16'daki robot bileşenlerini robot gövdesinin uygun bölümlerine yerleştiriniz. Motor kablolarını lehimleyiniz. Robot bileşenlerini vidayla gövdeye sabitleyiniz. Tekerlekleri uygun bir şekilde motor uçlarına takınız.



#### Değerlendirme

Çalışmalarınız öğrenme biriminin sonunda yer alan Kontrol Listesi kullanılarak değerlendirilecektir. Çalışmalarınızı yaparken değerlendirme ölçütlerini dikkate alınız.

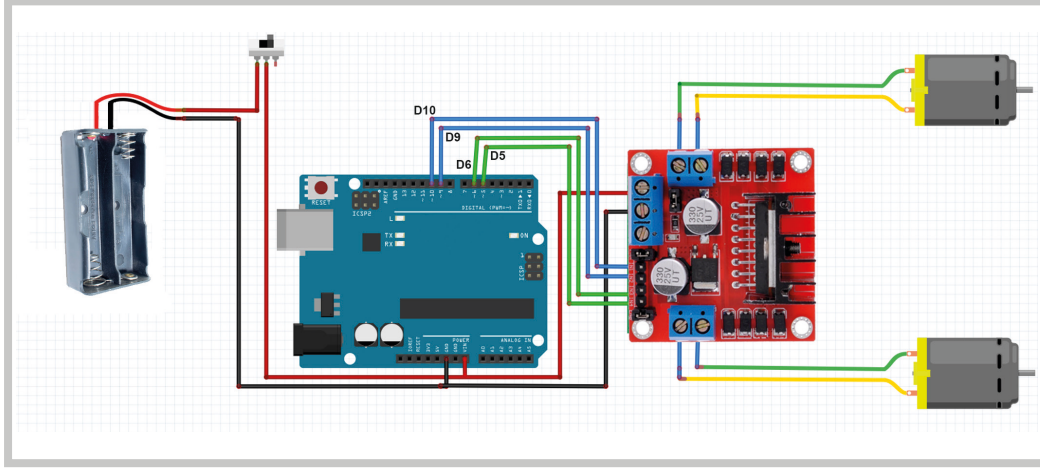
### 3.4. MOTORLARI SADECE GERİ YÖNDE ÇALIŞTIRMA

Robot bileşenlerini, robot gövdesine Görsel 3.19'daki gibi vidalarla sabitledikten sonra Görsel 3.20'deki bağlantı şemasından yararlanarak kablo bağlantıları yapılmalıdır. Bu bağlantılar kısaca şu şekilde açıklanabilir.



Görsel 3.19: Bileşenlerinin montajı yapılmış eğitsel robotun kablo bağlantıları

Pil yatağından gelen kırmızı kablo (pilin + ucu) önce açma kapama anahtarına sonra anahtardan çıkan diğer uç, hem mikroyenetleyici kartın Vin girişine hem de motor sürücünün +12 girişine bağlanır. Pil yatağından çıkan siyah kablo (pilin - ucu) doğrudan hem mikroyenetleyici kartın GND girişine hem de motor sürücünün GND girişine bağlanır. Motor kabloları, motor sürücü üzerindeki sağ ve sol klemenslere bağlanır. Yine motor sürücüdeki 4 adet kontrol girişleri mikroyenetleyici kartın D5, D6 (Sağ motor için) ve D9, D10 (Sol motor için) portlarına bağlanır.



Görsetel 3.20: Eğitsetel robot bileşenlerinin kablo bağlantı şeması



#### Sıra Sizde

Görsetel 3.19'daki kablo bağlantılarını Görsetel 3.20'deki şemayı kullanarak gerçekleştiriniz. Kablo bağlantılarını yaparken pil yatağı kablolarını açma kapama anahtarına uygun bir şekilde lehimleyiniz. Mikroyenetleyici kartından motor sürücüye giden 4 adet kontrol kablosu için dişi-dişi veya erkek-dişi jumper kablo kullanabilirsiniz.

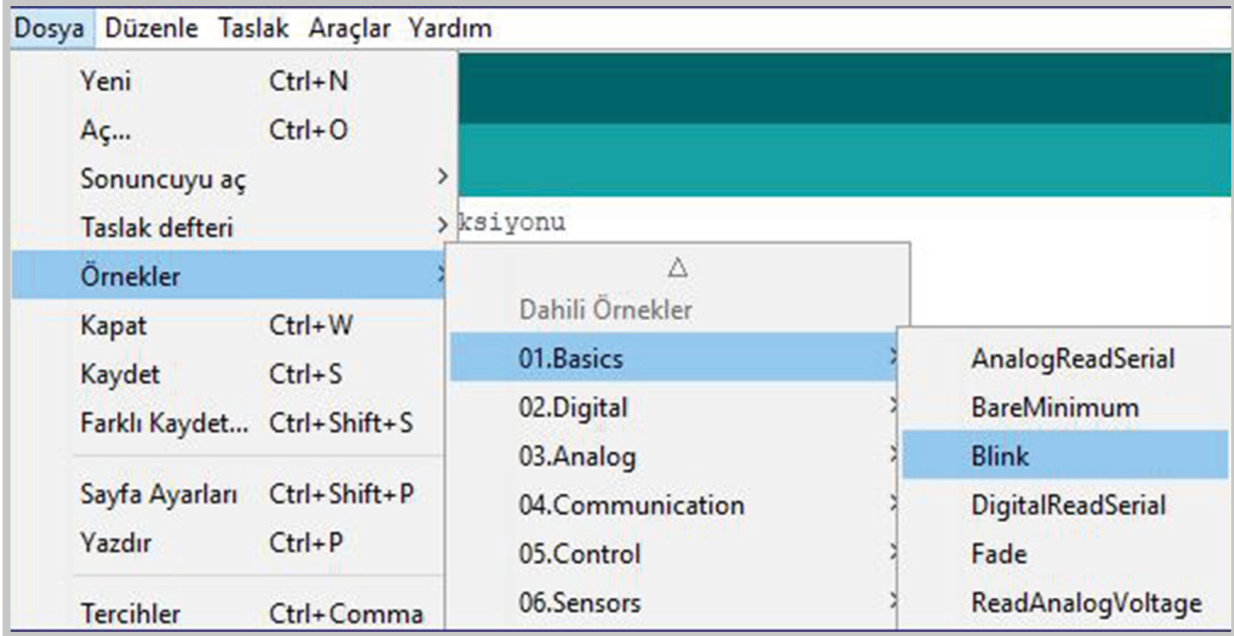


#### Değerlendirme

Çalışmalarınız öğrenme biriminin sonunda yer alan Kontrol Listesi kullanılarak değerlendirilecektir. Çalışmalarınızı yaparken değerlendirme ölçütlerini dikkate alınız.

## 3.5. EĞİTSEL ROBOTUN PROGRAMLANMASI

Eğitsel robotun Görsel 3.19'deki gibi montaj işlemi ve Görsel 3.20'deki şemaya göre kablo bağlantıları yapıldıktan sonra programlama aşamasına geçilir. Öncelikle mikrodenetleyici IDE programı çalıştırılır. Komutları hatırlamak adına Görsel 3.21'deki gibi **Blink** programı açılır.



Görsel 3.21: Blink programının açılması

```
void setup() {  
  pinMode(LED_BUILTIN, OUTPUT);  
}  
  
void loop() {  
  digitalWrite(LED_BUILTIN, HIGH);  
  delay(1000);  
  digitalWrite(LED_BUILTIN, LOW);  
  delay(1000);  
}
```

Yukarıdaki **Blink** programı açıldıktan sonra program içindeki kodların motorlara hareket verecek şekilde yeniden düzenlenmesi gerekir. Programı robota uygun hâle getirirken öncelikle `void setup()` fonksiyonu içindeki `pinMode(LED_BUILTIN, OUTPUT);` komutuyla başlanır. Bu komut içindeki `LED_BUILTIN` yerine uygun mikrodenetleyici port numarası yazılır.

`pinMode` komutu, mikrodenetleyici kartın dijital portlarının giriş veya çıkış olarak kullanılmasına karar verir. Mikrodenetleyici ile motorları kontrol etmek için 4 adet port kullanılır. Bunlardan ikisi sağ motor, diğer ikisi sol motor için kullanılır.

Tablo 3.3'te motorların hareket durumları verilmiştir.

Tablo 3.3: Motorların Hareket Durumları

Mikrodenetleyici Portları	Motorların Durumu
D5 = 0, D6=0	Sağ motor hareket etmez.
D5 = 1, D6=0	Sağ motor ileri yönde hareket eder.
D5 = 0, D6=1	Sağ motor geri yönde hareket eder.
D5 = 1, D6=1	Sağ motor hareket etmez.
D9 = 0, D10=0	Sol motor hareket etmez.
D9 = 1, D10=0	Sol motor ileri yönde hareket eder.
D9 = 0, D10=1	Sol motor geri yönde hareket eder.
D9 = 1, D10=1	Sol motor hareket etmez.

Programı eğitsel robota uygun hâle getirirken kullanılacak 2. komut, `void loop()` fonksiyonu içindeki `digitalWrite(LED_BUILTIN, HIGH)`; komutudur. Bu komut aynı zamanda motorlara enerji vermeyi sağlar. Bu komut içindeki `LED_BUILTIN` yerine, mikrodenetleyici kartın uygun port numarasına yazılır. Ayrıca `HIGH` yerine 1, `LOW` yerine 0 yazılabilir. Motorları ileri veya geri hareket ettirirken Tablo 3.3'teki durumlardan yararlanılabilir. Motorları ileri yönde hareket ettirmek için mikrodenetleyici kart üzerindeki D5 ve D9 numaralı portların "1", D6 ve D10 numaralı portların "0" yapılması gerekir.

Aşağıda verilen program yazılıp mikrodenetleyici karta yüklendiğinde motorlar ileri yönde dönecektir. Eğer motorlardan herhangi biri veya her ikisi birden geri yönde dönüyorsa bu durum iki şekilde düzeltilebilir. Birinci yöntem, ters dönen motorun ilgili mikrodenetleyici kartının kontrol kablosunun yerlerini değiştirmektir. İkinci yöntem ise ters dönen motorun, motor sürücüsü üzerine takılı motor kablolarının yerini değiştirmektir. Örneğin sağ motor ters dönüyorsa, sağ motor için mikrodenetleyici karttan motor sürücüye giden kontrol kablosu, D5 ve D6'nın yerlerini değiştirmek yeterli olur.

```
void setup() {
    pinMode(5, OUTPUT); // sağ motor ileri pini
    pinMode(6, OUTPUT); // sağ motor geri pini
    pinMode(9, OUTPUT); // sol motor ileri pini
    pinMode(10, OUTPUT); //sol motor geri pini
}
void loop() {
    digitalWrite(5,1); // sağ ileri = 1
    digitalWrite(6,0); // sağ geri = 0
    digitalWrite(9,1); // sol ileri = 1
    digitalWrite(10,0); //sol geri = 0
}
```



#### Sıra Sizde

Robot bağlantılarının yapıldığından emin olduktan sonra mikrodenetleyici IDE programını açıp bir önceki sayfada verilen kodları yazınız. Tüm işlemler bittikten sonra programı mikrodenetleyici karta yükleyiniz. Robotun pillerini düzgün bir şekilde takınız ve açma kapama anahtarını açıp, robotun çalışmasını gözlemleyiniz.

### 3. Öğrenme Birimi: Robot Tabanlı Proje Geliştirme



#### Değerlendirme

Çalışmalarınız öğrenme biriminin sonunda yer alan Kontrol Listesi kullanılarak değerlendirilecektir. Çalışmalarınızı yaparken değerlendirme ölçütlerini dikkate alınız.

Motorları geri yönde hareket ettirmek için bu sefer D6 ve D10 numaralı portların “1”, D5 ve D9 numaralı portların “0” yapılması gerekir. Ayrıntılı bilgi için Tablo 3.3'teki durumlardan yararlanılabilir.

```
void setup() {  
    pinMode(5, OUTPUT); // sağ motor ileri pini  
    pinMode(6, OUTPUT); // sağ motor geri pini  
    pinMode(9, OUTPUT); // sol motor ileri pini  
    pinMode(10, OUTPUT); //sol motor geri pini  
}  
void loop() {  
    digitalWrite(5,0); // sağ ileri = 0  
    digitalWrite(6,1); // sağ geri = 1  
    digitalWrite(9,0); // sol ileri = 0  
    digitalWrite(10,1); //sol geri = 1  
}
```



#### Sıra Sizde

Robot bağlantılarının yapıldığından emin olduktan sonra mikrodenetleyici IDE programını açıp bir önceki sayfada verilen kodları yazınız. Tüm işlemler bittikten sonra programı mikrodenetleyici karta yükleyiniz. Robotun pillerini düzgün bir şekilde takınız ve açma kapama anahtarını açıp, robotun çalışmasını gözlemleyiniz.



#### Değerlendirme

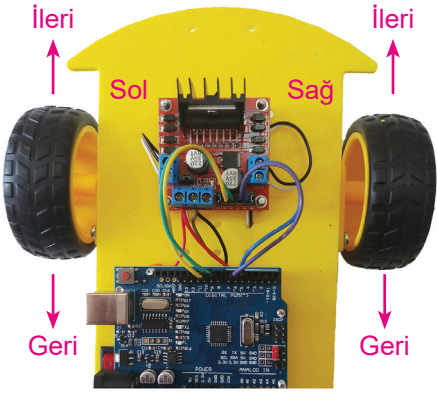
Çalışmalarınız öğrenme biriminin sonunda yer alan Kontrol Listesi kullanılarak değerlendirilecektir. Çalışmalarınızı yaparken değerlendirme ölçütlerini dikkate alınız.

Eğitsel robotun hareketleri, iki motorun hareketine bağlıdır. Tablo 3.3'te görüldüğü gibi mikrodenetleyicinin D5, D6, D9 ve D10 numaralı portlarına gönderilen 0 veya 1 bilgisine göre motorlar hareket etmektedir. Motorların hareketleri de eğitsel robotun hangi doğrultuda hareket etmesi gerektiğini belirlemektedir. Eğitsel robottaki uygulamalarda Tablo 3.4'teki robotun hareket durumlarında, kırmızı ile gösterilenler kullanılacaktır. Mavi ile gösterilenler ise çizgi izleyen robot uygulamalarında kullanılacaktır.

Motorların hareketine göre robotun hareketleri şu şekilde sağlanır.

Tablo 3.4: Motorların Hareketine Göre Robotun Hareket Durumları

Mikrodenetleyici Portları		Motorların Hareket Durumu
Sol Motor	Sağ Motor	
D9 = 0, D10=0	D5 = 0, D6=0	Robot hareket etmez.
D9 = 1, D10=0	D5 = 1, D6=0	Robot ileri yönde hareket eder.
D9 = 0, D10=1	D5 = 0, D6=1	Robot geri yönde hareket eder.
D9 = 0, D10=0	D5 = 1, D6=0	Robot sol teker sabit, sola doğru döner.
D9 = 0, D10=0	D5 = 0, D6=1	Robot sol teker sabit, sağa doğru döner.
D9 = 1, D10=0	D5 = 0, D6=0	Robot sağ teker sabit, sağa doğru döner.
D9 = 0, D10=1	D5 = 0, D6=0	Robot sağ teker sabit, sola doğru döner.
D9 = 0, D10=1	D5 = 1, D6=0	Robot olduğu yerde sola doğru döner.
D9 = 1, D10=0	D5 = 0, D6=1	Robot olduğu yerde sağa doğru döner.



Eğitsel robotu sola doğru hareket ettirebilmek için Tablo 3.4'te gösterildiği gibi mikrodenetleyici kartın D5 ve D10 numaralı portlarına 1 verisini, D6 ve D9 numaralı portlarına 0 verisini göndermek gerekir.

```
void setup() {  
  pinMode(5, OUTPUT); // sağ motor ileri pini  
  pinMode(6, OUTPUT); // sağ motor geri pini  
  pinMode(9, OUTPUT); // sol motor ileri pini  
  pinMode(10, OUTPUT); //sol motor geri pini  
}  
void loop() {  
  digitalWrite(5,1); // sağ ileri = 1  
  digitalWrite(6,0); // sağ geri = 0  
  digitalWrite(9,0); // sol ileri = 0  
  digitalWrite(10,1); //sol geri = 1  
}
```



#### Sıra Sizde

Eğitsel robotun bağlantılarının yapıldığından emin olduktan sonra mikrodenetleyici IDE programını açıp yukarıda verilen kodları yazınız. Tüm işlemler bittikten sonra programı mikrodenetleyici karta yükleyiniz. Robotun pillerini düzgün bir şekilde takınız ve açma kapama anahtarını açıp, robotun çalışmasını gözlemleyiniz.



#### Değerlendirme

Çalışmalarınız öğrenme biriminin sonunda yer alan Kontrol Listesi kullanılarak değerlendirilecektir. Çalışmalarınızı yaparken değerlendirme ölçütlerini dikkate alınız.

### 3. Öğrenme Birimi: Robot Tabanlı Proje Geliştirme

Robotunuzu sağa doğru hareket ettirebilmek için Tablo 3.4'te gösterildiği gibi mikrodenetleyici kartın D6 ve D9 numaralı portlarına 1 verisini, D5 ve D10 numaralı portlarına 0 verisini göndermek gerekir.

```
void setup() {  
  pinMode(5, OUTPUT); // sağ motor ileri pini  
  pinMode(6, OUTPUT); // sağ motor geri pini  
  pinMode(9, OUTPUT); // sol motor ileri pini  
  pinMode(10, OUTPUT); // sol motor geri pini  
}  
void loop() {  
  digitalWrite(5,0); // sağ ileri = 0  
  digitalWrite(6,1); // sağ geri = 1  
  digitalWrite(9,1); // sol ileri = 1  
  digitalWrite(10,0); // sol geri = 0  
}
```



#### Sıra Sizde

Eğitsel robotun bağlantılarının yapıldığından emin olduktan sonra mikrodenetleyici IDE programını açıp yukarıda verilen kodları yazınız. Tüm işlemler bittikten sonra programı mikrodenetleyici karta yükleyiniz. Robotun pillerini düzgün bir şekilde takınız ve açma kapama anahtarını açıp, robotun çalışmasını gözlemleyiniz.



#### Değerlendirme

Çalışmalarınız öğrenme biriminin sonunda yer alan Kontrol Listesi kullanılarak değerlendirilecektir. Çalışmalarınızı yaparken değerlendirme ölçütlerini dikkate alınız.

Eğitsel robotu belli bir duruma veya şarta göre ileri, geri, sağa, sola hareket ettirebilmek için komutları bir defa yazmak gerekir. Aynı hareketleri birden fazla tekrar ettirebilmek içinse aynı komutlar tekrarlı bir şekilde yazılmalıdır. Bu tekrarlı ifadelerden kurtulmak için fonksiyon denilen bir yapı tanımlanmalıdır. Fonksiyon tanımlama aşağıda gösterilmiştir. Fonksiyon, değer döndürmeyecekse veri tipi olarak `void` kullanılır.

<code>void</code>	<code>ileri ()</code>	<code>{komut veya komutları}</code>
fonksiyon veri tipi	fonksiyon adı	süslü parantez içinde komutlar yazılacak

Eğitsel robotun ileri, geri, sağa, sola hareket komutlarını içeren fonksiyonları kullanan örnek bir program yazmak gerekirse aşağıdaki işlemler gerçekleştirilmelidir.

Eğitsel robot 2 saniye ileri, 1 saniye geri, 2 saniye sağa, 1 saniye sola gitsin ve 2 saniye dursun sonra da program başa dönsün.

Programın belirli saniye aralıklarıyla ileri, geri, sağa, sola hareketini içeren fonksiyonlar, "`void loop()`" fonksiyonu içinde çağrılır. Eğitsel robotun ileri, geri, sağa, sola hareket komutlarını içeren fonksiyon tanımlamaları da "`void loop()`" fonksiyonu dışında yapılır.



Robotu ileri, geri, sağa ve sola belirli sürelerde hareket ettiren program kodları şunlardır:

```
void setup() {  
  
    pinMode(5, OUTPUT); // sağ motor ileri pini  
    pinMode(6, OUTPUT); // sağ motor geri pini  
    pinMode(9, OUTPUT); // sol motor ileri pini  
    pinMode(10, OUTPUT); // sol motor geri pini  
}  
  
void loop() {  
  
    ileri(); //ileri fonksiyonunun kullanımı  
    delay(2000);  
    geri(); //geri fonksiyonunun kullanımı  
    delay(1000);  
    sag(); //sağa dönme fonksiyonunun kullanımı  
    delay(2000);  
    sol(); //sola dönme fonksiyonunun kullanımı  
    delay(1000);  
    dur(); //durma fonksiyonunun kullanımı  
    delay(2000);  
}  
  
void ileri(){  
    digitalWrite(5,1);  
    digitalWrite(6,0);  
    digitalWrite(9,1);  
    digitalWrite(10,0);  
}  
void geri(){  
    digitalWrite(5,0);  
    digitalWrite(6,1);  
    digitalWrite(9,0);  
    digitalWrite(10,1);  
}  
void sol(){  
    digitalWrite(5,1);  
    digitalWrite(6,0);  
    digitalWrite(9,0);  
    digitalWrite(10,1);  
}  
void sag(){  
    digitalWrite(5,0);  
    digitalWrite(6,1);  
    digitalWrite(9,1);  
    digitalWrite(10,0);  
}  
void dur(){  
    digitalWrite(5,0);  
    digitalWrite(6,0);  
    digitalWrite(9,0);  
    digitalWrite(10,0);  
}
```

### 3. Öğrenme Birimi: Robot Tabanlı Proje Geliştirme



#### Sıra Sizde

Eğitsel robotun bağlantılarının yapıldığından emin olduktan sonra mikrodenetleyici IDE programını açıp robotu 1 saniye ileri, 1 saniye sağa, 1 saniye geri, 1 saniye sola, 2 saniye ileri ve 2 saniye durduran sonra da programı başa döndüren kodları yazınız. Tüm işlemler bittikten sonra programı mikrodenetleyici karta yükleyiniz. Robotun pillerini düzgün bir şekilde takınız ve açma kapama anahtarını açıp, robotun çalışmasını gözlemleyiniz.



#### Değerlendirme

Çalışmalarınız öğrenme biriminin sonunda yer alan Kontrol Listesi kullanılarak değerlendirilecektir. Çalışmalarınızı yaparken değerlendirme ölçütlerini dikkate alınız.



#### Sıra Sizde

Eğitsel robotun bağlantılarının yapıldığından emin olduktan sonra mikrodenetleyici IDE programını açıp robotu 2 saniye sağa, 1 saniye dur, 2 saniye sola, 1 saniye dur, 1 saniye ileri ve 2 saniye dur kodlarını yazıp sonra da programı başa döndüren kodları yazınız. Tüm işlemler bittikten sonra programı mikrodenetleyici karta yükleyiniz. Robotun pillerini düzgün bir şekilde takınız ve açma kapama anahtarını açıp, robotun çalışmasını gözlemleyiniz.



#### Değerlendirme

Çalışmalarınız öğrenme biriminin sonunda yer alan Kontrol Listesi kullanılarak değerlendirilecektir. Çalışmalarınızı yaparken değerlendirme ölçütlerini dikkate alınız.

Motor sürücünden, mikrodenetleyici karta gerekli bağlantılar yapılırken özellikle sağ motor için D5 ve D6, sol motor için D9 ve D10 portları kullanılacak diye bir şart yoktur. Mikrodenetleyici kartın diğer portları da kullanılabilir. Ancak sağ motor için kullanılan, mikrodenetleyici kartının D5 ve D6 portları ile sol motor için kullanılan, mikrodenetleyici kartının D9 ve D10 portları, motorlarda hız kontrolü yapmaya olanak tanıyan PWM çıkışlarıdır. PWM, sinyalleri motorlara belirli aralıklarla kesik kesik enerji vererek motorun hızlı veya yavaş dönmesini sağlar. Bu sayede eğitsel robotun hızının kontrol edilmesine olanak tanır.

Eğitsel robotun hız kontrolü yapılırken `digitalWrite (5, 1)` ; komutu yerine, `analogWrite (5, h)` ; komutu kullanılır. Komutta kullanılan "h" değişkeni, burada "0" ile "255" arasında değer alabilen bir sayıyı ifade etmektedir. Bu sayı "255" olduğunda motor en hızlı devirde döner, "128" olduğunda yarı hızda döner, "0" olduğunda ise motorlar durur. Daha önce en yüksek hızda çalıştırılan; "robot, 2 saniye ileri, 1 saniye geri, 2 saniye sağa, 1 saniye sola gitsin ve 2 saniye dursun sonra da programı başa dönsün" şeklindeki programı, motorları yarı hızda çalıştırabilmek için yeniden düzenlemek gerekir.

PWM ile motorların hızını kontrol eden program kodları şunlardır:

```
int h=120; // motor hızını belirleyen değişken
void setup() {
    pinMode(5, OUTPUT); // sağ motor ileri pini
    pinMode(6, OUTPUT); // sağ motor geri pini
    pinMode(9, OUTPUT); // sol motor ileri pini
    pinMode(10, OUTPUT); // sol motor geri pini
}
void loop() {

ileri(); delay(2000);
geri(); delay(1000);
sag(); delay(2000);
sol(); delay(1000);
dur(); delay(2000);
}
void ileri(){
analogWrite(5,h);
analogWrite(6,0);
analogWrite(9,h);
analogWrite(10,0);
}
void geri(){
analogWrite(5,0);
analogWrite(6,h);
analogWrite(9,0);
analogWrite(10,h);
}
void sol(){
analogWrite(5,h);
analogWrite(6,0);
analogWrite(9,0);
analogWrite(10,h);
}
void sag(){
analogWrite(5,0);
analogWrite(6,h);
analogWrite(9,h);
analogWrite(10,0);
}
void dur(){
analogWrite(5,0);
analogWrite(6,0);
analogWrite(9,0);
analogWrite(10,0);
}
```



#### Sıra Sizde

Eğitsel robotun bağlantılarının yapıldığından emin olduktan sonra mikrodenetleyici IDE programını açıp hız değişkenine 150 atayınız. Robotu 2 saniye sola, 1 saniye dur, 2 saniye sağa, 1 saniye dur, 3 saniye ileri, 1 saniye ileri ve 2 saniye durdurun sonra da programı başa döndüren kodları yazınız. Tüm işlemler bittikten sonra programı mikrodenetleyici karta yükleyiniz. Robotun pillerini düzgün bir şekilde takınız ve açma kapama anahtarını açıp, robotun çalışmasını gözlemleyiniz.

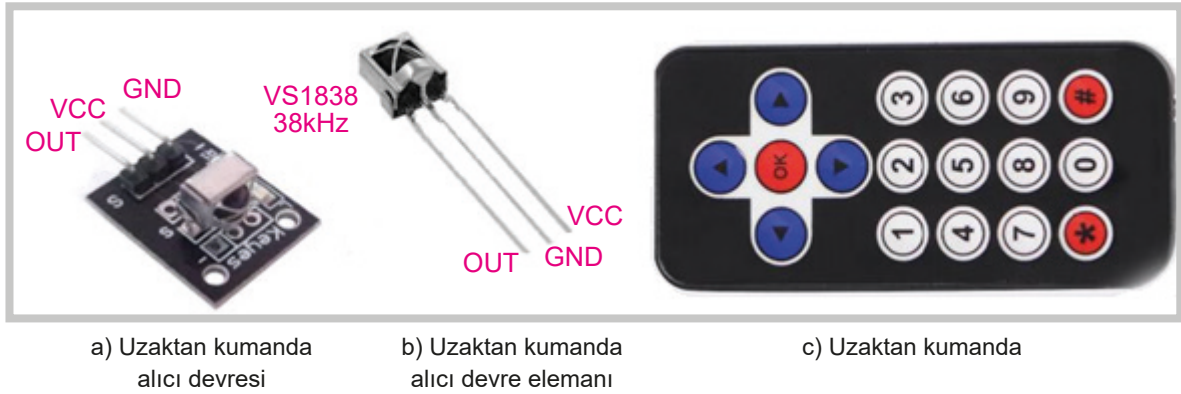


#### Değerlendirme

Çalışmalarınız öğrenme biriminin sonunda yer alan Kontrol Listesi kullanılarak değerlendirilecektir. Çalışmalarınızı yaparken değerlendirme ölçütlerini dikkate alınız.

### 3.6. EĞİTSEL ROBOTUN UZAKTAN KUMANDA İLE KONTROLÜ

Eğitsel robot, şu ana kadar kullanıcı yönlendirmesi olmadan belirli bir program dâhilinde çalıştırıldı. Bundan sonraki aşamada uzaktan kumanda ile eğitsel robot yönlendirilebilir. Uzaktan kumandalar, Görsel 3.22.c'de görüldüğü gibi üzerinde bulunan tuş kombinasyonuna bağlı olarak belirli kodları kızılötesi LED yardımıyla karşı tarafa gönderir. Uzaktan kumandalar, günlük yaşantıda TV, uydu alıcısı, müzik sistemi gibi birçok elektronik aygıtı kontrol etmek için kullanılır. Bu bölümde herhangi bir uzaktan kumanda ile eğitsel robot kontrol edilecektir.

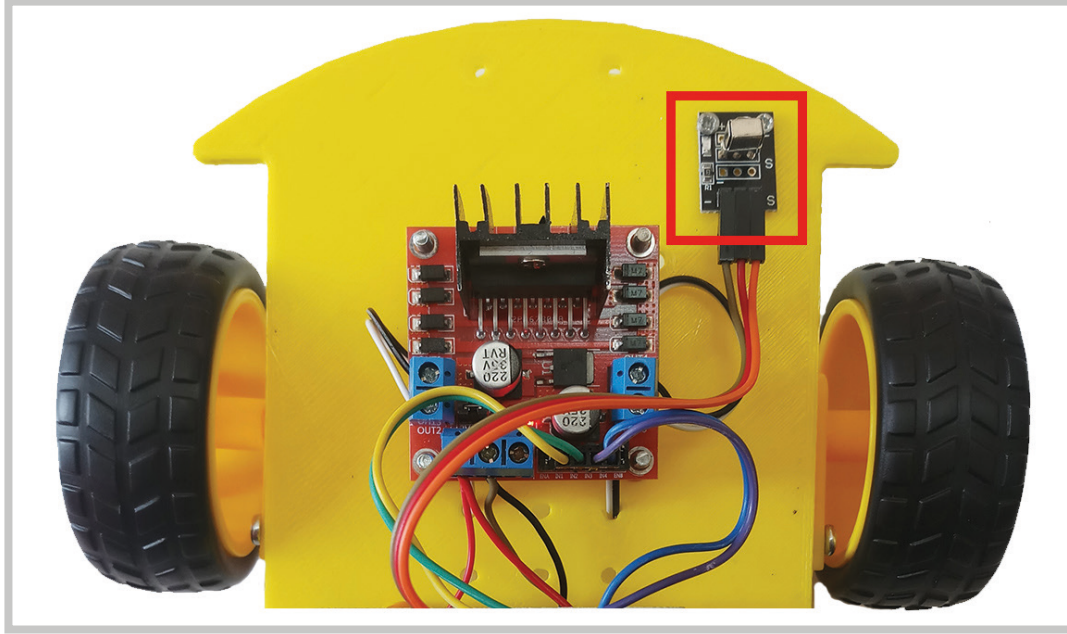


Görsel 3.22: Uzaktan kumanda alıcısı ve genel amaçlı uzaktan kumanda

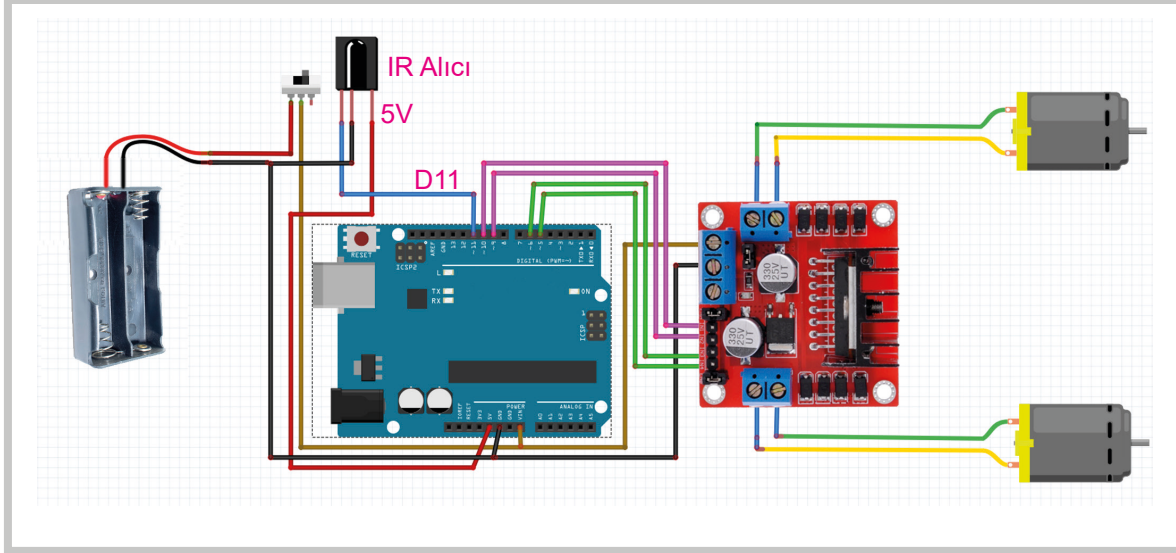
Uzaktan kumandanın gönderdiği sinyalleri çözen devre elemanı, Görsel 3.22.a ve b'de görülmektedir. VS1838 isimli IR alıcı ve kod çözücü devre elemanı, Görsel 3.22.a'da gösterilen şekilde elektronik kart üzerine monte edilir. Görsel 3.22.b'de görüldüğü gibi kart üzerine monte edilmeden de kullanılabilir.

### 3. Öğrenme Birimi: Robot Tabanlı Proje Geliştirme

IR alıcı ve kod çözücü devre elemanı, Görsel 3.23'te görüldüğü gibi robot üzerine monte edilir. Bağlantı uçları yapılırken “Vcc” ucu mikrodenetleyici üzerindeki “5V” pinine, “GND” ucu mikrodenetleyici üzerindeki “GND” pinine, “Out” ucu mikrodenetleyici kart üzerindeki “D11” pinine bağlanır (Görsel 3.24).



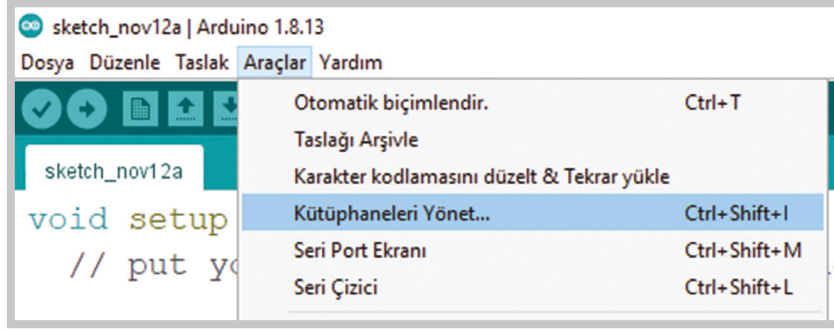
Görsel 3.23: Uzaktan kumanda alıcısının eğitsel robot üzerine montajı



Görsel 3.24: Uzaktan kumanda alıcısının bağlantı şeması

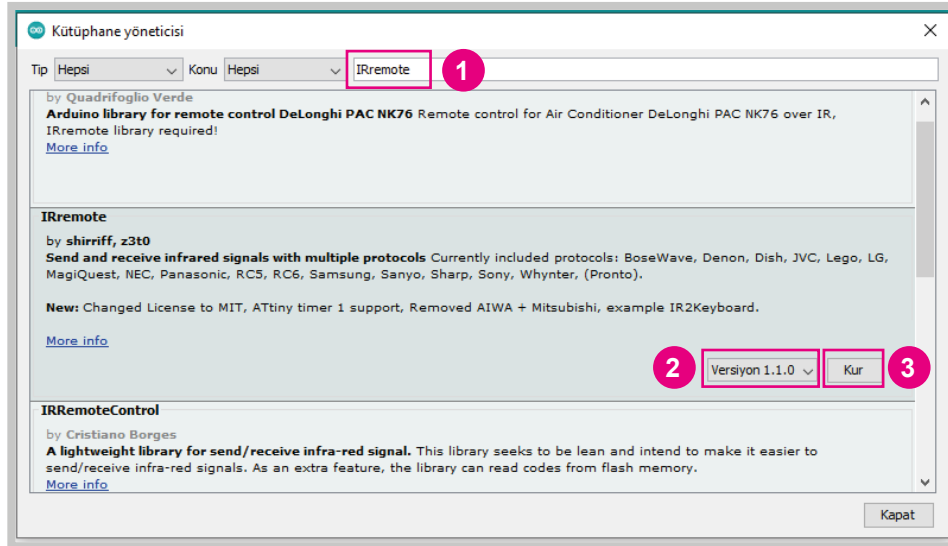
IR alıcı ve kod çözücü devre elemanının bağlantıları yapıldıktan sonra kodlamaya geçebilmek için “IRremote” kütüphanesinin kurulu olmalıdır. Bunun için öncelikle, Görsel 3.25'teki gibi mikrodenetleyici IDE programında **Araçlar** menüsünden **Kütüphaneleri Yönet** penceresinin açılır.

### 3. Öğrenme Birimi: Robot Tabanlı Proje Geliştirme

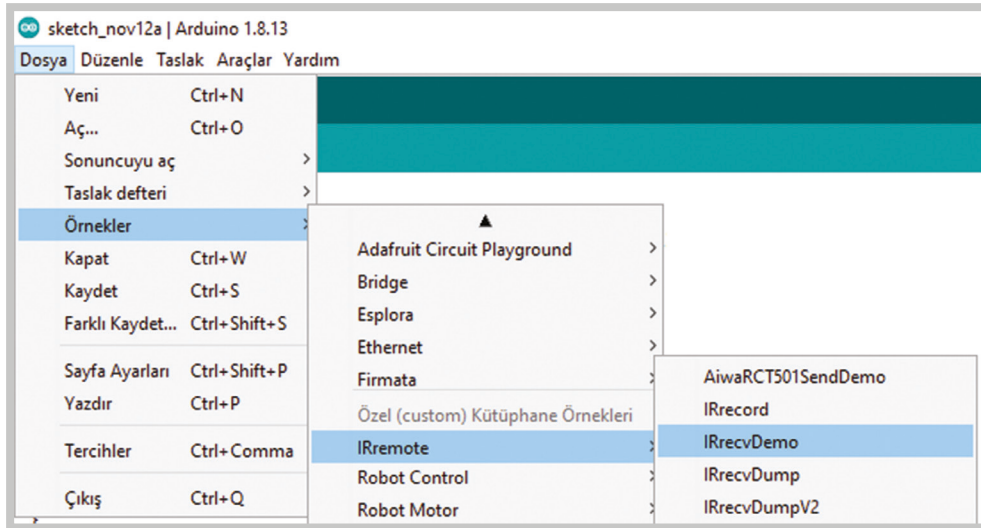


Görsel 3.25: Mikrodenetleyici IDE programında Kütüphaneleri Yönet penceresinin açılması

Görsel 3.26 kütüphane yöneticisi penceresinde “1” numarayla gösterilen arama alanına, kütüphane ismi girilmelidir. IRremote kütüphanesi için farklı kütüphane versiyonları mevcuttur. Bu bölümde kullanılacak olan 1.1.0 versiyonu, daha basit ve kodları anlaşılır düzeyde olan bir versiyondur. Kütüphane versiyonu da seçildikten sonra **kur** butonuna basılmalıdır. Kurma işlemi bittikten sonra Görsel 3.27’deki gibi **IRrecvDemo** programı açılır.



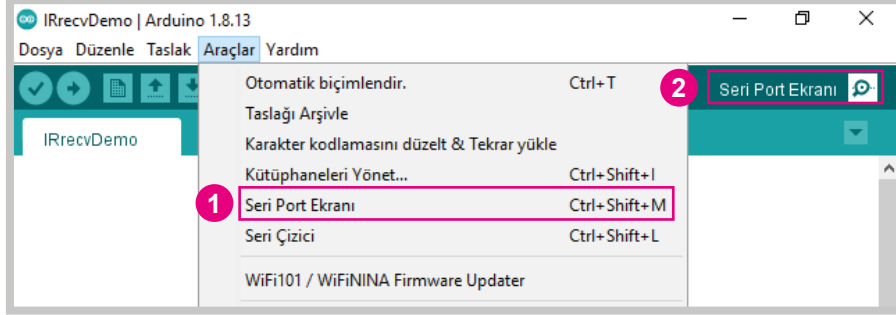
Görsel 3.26: Kütüphane yöneticisi penceresinde program kütüphanesinin aranması



Görsel 3.27: IRremote kütüphanesindeki örnek programlar

### 3. Öğrenme Birimi: Robot Tabanlı Proje Geliştirme

IRrecvDemo programı açıldıktan sonra program üzerinde hiçbir değişiklik yapmadan, eğitsel robottaki mikrodenetleyici karta yüklenmelidir. Uzaktan kumandanın IR alıcıya gönderdiği kodları görebilmek için “Seri Port” ekranı açılmalıdır. Eğitsel robottaki mikrodenetleyici kartın USB bağlantısını çıkarmadan, “Seri Port” ekranının Görsel 3.29’da gösterildiği gibi açılması gerekir. “Seri Port” ekranı “Araçlar” menüsünden, “Ctrl+S-hift+M” kısa yoluyla ya da Görsel 3.28’de pencerenin sağ üst köşesinde gösterilen ikona basılarak açılabilir.



Görsel 3.28: Seri Port ekranının açılması

Görsel 3.29’da örnek bir kumanda gösterilmiştir. Eğitsel robotu hareket ettirmek için uzaktan kumanda üzerinde 5 farklı tuş kullanılır. İstenirse buna benzer farklı şekillerde kumandalar kullanılabilir. Burada asıl önemli olan, kullanılan uzaktan kumandanın tuşlarına karşılık gelen kodları belirlemektir. Görsel 3.29’da olduğu gibi uzaktan kumandanın tuşlarına karşılık gelen kodlar, seri port ekranında görüntülenecektir. Uzaktan kumandanın tuşlarına karşılık gelen kodlar, 16 tabanlı (heksadesimal) sayı sistemine göre görüntülenir. Bu kodların bir yere not edilmelidir.

```
#include <IRremote.h>
int RECV_PIN = 11;
IRrecv irrecv(RECV_PIN);
decode_results results;
void setup()
{
  Serial.begin(9600);
  irrecv.enableIRIn();
}

void loop() {
  if (irrecv.decode(&results)) {
    Serial.println(results.value, HEX);
    irrecv.resume();
  }
  delay(100);
}
```

COM5

FF18E7 İleri  
FFFFFFFF  
FF4AB5 geri  
FF38C7 dur  
FFFFFFFF  
FF10EF sol  
FFFFFFFF  
FF5AA5 sağ  
FFFFFFFF

Otomatik Kaydırma  Zaman damgasını göster

Görsel 3.29: IRrecvDemo programıyla kumanda kodlarının çözülmesi

### 3. Öğrenme Birimi: Robot Tabanlı Proje Geliştirme

`long` `bileri` = `0x` `FF1807;`  
değişken tipi      değişken adı      16'lık sayı sistemi      uzaktan kumanda tuşunun kodu

Görsel 3.30: IRrecvDemo programında uzaktan kumanda tuş kodlarının kullanılması

Uzaktan kumandadan elde edilen kodlar kullanılırken Görsel 3.30'da olduğu gibi “`long`” tipinde değişken tanımlanarak yazılması gerekir. 16 tabanlı (heksadesimal) sayı sisteminde, sayıları belirtmek için başına “0x” getirerek yazılmalıdır. Son durumda IRrecvDemo örnek programını değiştirirken, aşağıdaki program kodları eklenir.

```
int h=100; // Robot motorlarının dönme hızı

long bileri = 0xFF18E7;// Robotun ileri gitmesi için kullanılan U.K. kodu
long bgeri  = 0xFF4AB5;// Robotun geri gitmesi için kullanılan U.K. kodu
long bdur   = 0xFF38C7;// Robotun durması için kullanılan U.K. kodu
long bsol   = 0xFF10EF;// Robotun sola dönmesi için kullanılan U.K. kodu
long bsag   = 0xFF5AA5;// Robotun sağa dönmesi için kullanılan U.K. kodu
```

Eğitsel robotun motor pinleri tanımlanırken “`void setup()`” fonksiyonu içine aşağıdaki kodlar eklenir.

```
pinMode(5, OUTPUT);// sağ motor ileri pini
pinMode(6, OUTPUT);// sağ motor geri pini
pinMode(9, OUTPUT);// sol motor ileri pini
pinMode(10, OUTPUT);// sol motor geri pini
```

Eğitsel robota bağlı IR alıcı ve kod çözücünün uzaktan kumandadan aldığı kodları robotun hareket fonksiyonlarıyla eşleştirebilmesi için “`void loop()`” fonksiyonu içine aşağıdaki kodların eklenmelidir. Burada “`results.value`” değişkeni, uzaktan kumandadan anlık olarak aldığı verileri saklayan bir değişkendir.

```
if (results.value == bileri ) ileri();
if (results.value == bgeri ) geri();
if (results.value == bsag ) sag();
if (results.value == bsol ) sol();
if (results.value == bdur ) dur();
```

En son “`void loop()`” fonksiyonunun blok bitirme “}” parantezinin sonundan itibaren robotu yönlendiren `void ileri()`, `void geri()`, `void sag()`, `void sol()`, `void dur()` fonksiyonlarının eklenmelidir. Programın son hâli aşağıdaki gibi olacaktır.

```
#include <IRremote.h>
int RECV_PIN = 11;

int h=100; // Robot motorlarının dönme hızı

long bileri = 0xFF18E7;// Robotun ileri gitmesi için kullanılan U.K. kodu
long bgeri  = 0xFF4AB5;// Robotun geri gitmesi için kullanılan U.K. kodu
long bdur   = 0xFF38C7;// Robotun durması için kullanılan U.K. kodu
long bsol   = 0xFF10EF;// Robotun sola dönmesi için kullanılan U.K. kodu
long bsag   = 0xFF5AA5;// Robotun sağa dönmesi için kullanılan U.K. kodu
IRrecv irrecv(RECV_PIN);

decode_results results;
```



```

void setup()
{
  pinMode(5, OUTPUT); // sağ motor ileri pini
  pinMode(6, OUTPUT); // sağ motor geri pini
  pinMode(9, OUTPUT); // sol motor ileri pini
  pinMode(10, OUTPUT); // sol motor geri pini

  Serial.begin(9600);
  irrecv.enableIRIn(); // Alıcıyı başlat.
}

void loop() {
  if (irrecv.decode(&results)) {
    Serial.println(results.value, HEX);

    if (results.value == bileri ) ileri();
    if (results.value == bgeri ) geri();
    if (results.value == bsag ) sag();
    if (results.value == bsol ) sol();
    if (results.value == bdur ) dur();

    irrecv.resume(); // Yeni verileri al.
  }
}

void ileri(){
  analogWrite(5,h);
  analogWrite(6,0);
  analogWrite(9,h);
  analogWrite(10,0);
}

void geri(){
  analogWrite(5,0);
  analogWrite(6,h);
  analogWrite(9,0);
  analogWrite(10,h);
}

void sol(){
  analogWrite(5,h);
  analogWrite(6,0);
  analogWrite(9,0);
  analogWrite(10,h);
}

void sag(){
  analogWrite(5,0);
  analogWrite(6,h);
  analogWrite(9,h);
  analogWrite(10,0);
}

void dur(){
  analogWrite(5,0);
  analogWrite(6,0);
  analogWrite(9,0);
  analogWrite(10,0);
}

```



#### Sıra Sizde

Eğitsel robot gövdesine IR alıcı devresini bağlayıp diğer robot bağlantılarının düzgün olduğundan emin olduktan sonra mikrodenetleyici IDE programını açarak hız değişkenine 100 değerini atayınız. Robotun uzaktan kumanda kontrolüne ait program kodlarını yazınız. Tüm işlemler bittikten sonra programı mikrodenetleyici karta yükleyiniz. Robotun pillerini düzgün bir şekilde takınız ve açma kapama anahtarını açıp, robotun çalışmasını gözlemleyiniz.

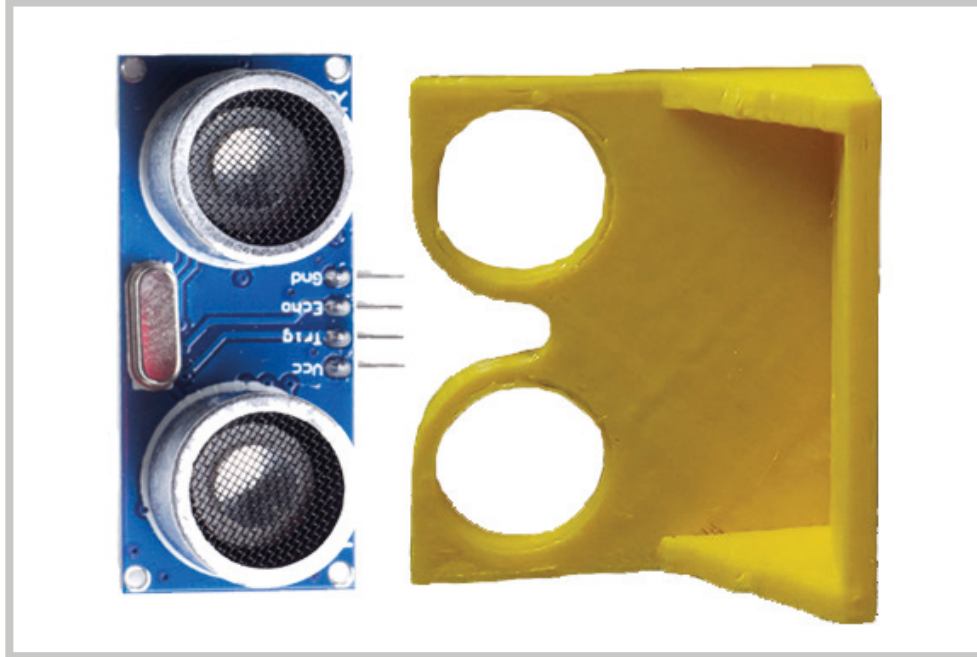


#### Değerlendirme

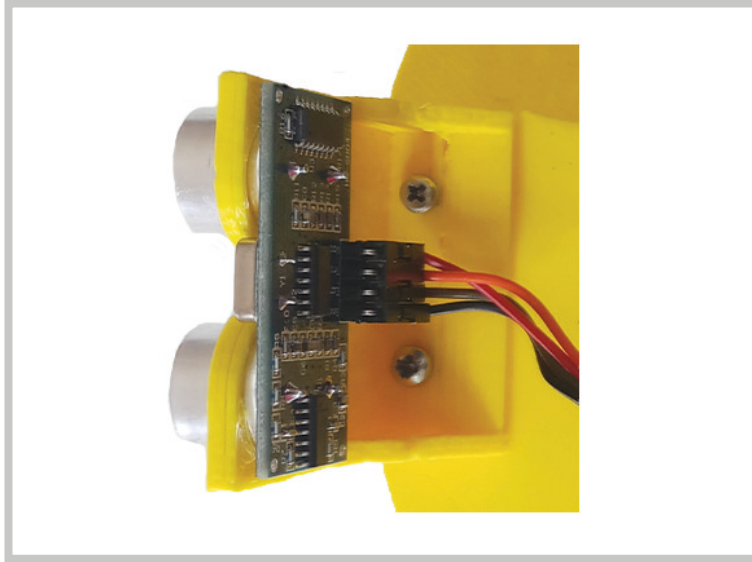
Çalışmalarınız öğrenme biriminin sonunda yer alan Kontrol Listesi kullanılarak değerlendirilecektir. Çalışmalarınızı yaparken değerlendirme ölçütlerini dikkate alınız.

### 3.7. EĞİTSEL ROBOT İLE ENGELDEN KAÇMA

Engelden kaçan robotlarda mesafe algılama sensörleri kullanılır. Mesafe algılama sensörlerinin birçok çeşidi vardır. Eğitsel robotta, HC-SR04 olarak da bilinen ultrasonik mesafe sensörü kullanılacaktır. Montaj aparatı kullanılarak eğitsel robotun ön tarafına Görsel 3.32’de görüldüğü gibi vidalarla ultrasonik mesafe sensörünün montajı yapılır ve program yazılacak hâle gelir. Ultrasonik mesafe sensörü ve montaj aparatı Görsel 3.31’de görülmektedir.

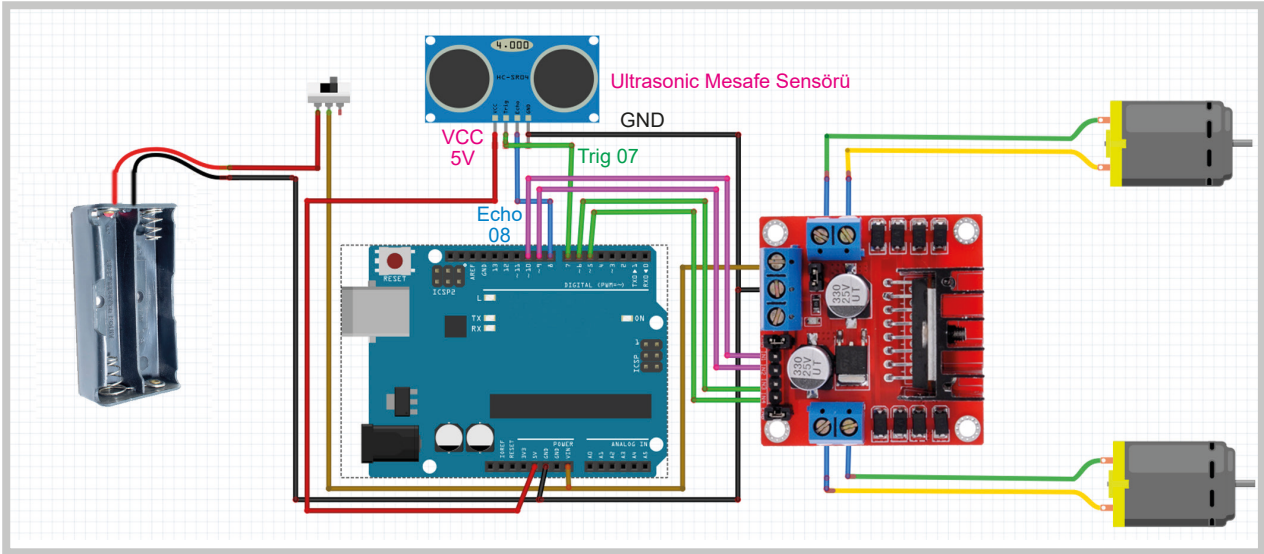


Görsel 3.31: Ultrasonik mesafe algılama sensörü ve montaj aparatı



Görsel 3.32: Ultrasonik mesafe algılama sensörünün eğitsel robota monte edilmesi

Ultrasonik mesafe sensörü 2 adet hoparlörden oluşur. Bunlardan birincisi, karşı tarafa 40 kHz lik 8 adet ses dalgası gönderir. Diğer hoparlör ise bu ses dalgasının bir cisme çarpıp geri dönmesini bekler. Gönderilen ses sinyali bir cisme çarpıp geri döndüğünde ise sesin gecikme süresi hesaplanır. Bu şekilde sensör ile cisim arasındaki mesafe ölçülür.



Görsel 3.33: Ultrasonik mesafe algılama sensörünün bağlantı şeması

Ultrasonik mesafe sensöründe, Görsel 3.33'te görüldüğü gibi 4 adet bağlantı pini bulunur. Bu pinlerden Vcc pini mikrodenetleyici kartın 5V pinine, GND pini ise mikrodenetleyici kartın GND pinine bağlanır. Geriye kalan Trig pini (ses sinyali gönderen pin) mikrodenetleyici kartın D7 pinine, Echo pini (ses sinyalini alan pin) ise mikrodenetleyici kartın D8 pinine bağlanır. Robotun tüm bağlantıları yapıldıktan sonra programı yazarken öncelikle "trig" ve "echo" isiminde 2 değişken belirlenip, pin numaralarının atanması gerekir. Sonraki aşamada "`void setup()`" fonksiyonu içinde `pinMode(trig, OUTPUT);` komutuyla mikrodenetleyici kartın "trig" portu çıkış, `pinMode(echo, INPUT);` komutuyla "echo" portu giriş yapılır.

### 3. Öğrenme Birimi: Robot Tabanlı Proje Geliştirme

Mesafe sensörüyle cisim arasındaki mesafe hesaplanırken, "void loop()" fonksiyonu içinde öncelikle `digitalWrite(trig,1);` komutuyla karşı tarafa bir ses dalgası gönderilir. 1 milisaniye beklenip `digitalWrite(trig,0);` komutuyla bu ses dalgası kesilir. Gönderilen sesi almak için `pulseIn(echo,1);` komutu kullanılarak "sure" isimindeki değişkene aktarılır. Mesafe hesaplaması yapılırken "mesafe" isiminde bir değişken oluşturularak, `int mesafe=(sure/2)/28.97;` komutuyla "sure" değişkeninin yarısı alınıp 28.97 sabit sayısıyla çarpılır. Burada sesin gidip gelme mesafesinden dolayı "sure" değişkeninin yarısı alınır. Komutta kullanılan sabit sayı ise ortam sıcaklığına göre hesaplanır. Mesafe belirlendikten sonra "`if (mesafe < 30)`" komutuyla robot ile engel arasındaki mesafe 30 cm'den küçükse robotu geri al ve döndür hareketini, "`else`" komutuyla da aradaki mesafe 30 cm'den büyükse ileri gitme hareketini yapacak kodların yazılması gerekir. Engelden kaçan eğitsel robotun program kodları aşağıdaki gibidir.

```
int trig = 7; // Mesafe sensörü trig pini için D7 portunu belirle
int echo = 8; // Mesafe sensörü trig pini için D8 portunu belirle
int hiz=100; // Motor dönüş hızını belirle

void setup(){
  pinMode(5, OUTPUT); // sağ motor ileri pini
  pinMode(6, OUTPUT); // sağ motor geri pini
  pinMode(9, OUTPUT); // sol motor ileri pini
  pinMode(10, OUTPUT); // sol motor geri pini
  pinMode(trig, OUTPUT); // Mesafe sensörü trig pini çıkış için
  pinMode(echo, INPUT); // Mesafe sensörü echo pini giriş için
}
void loop(){
  digitalWrite(trig, 1);
  delay (1);
  digitalWrite(trig, 0);

  int sure = pulseIn(echo, 1);
  int mesafe = (sure/2) / 28.97;

  if (mesafe < 30) // mesafe 30cm'den küçük ise robotu geri al ve döndür.
  {
    analogWrite(5, 0);
    analogWrite(6, hiz);
    analogWrite(9, 0);
    analogWrite(10, hiz);
    delay(150);

    analogWrite(5, 0);
    analogWrite(6, hiz);
    analogWrite(9, hiz);
    analogWrite(10, 0);
    delay(250);
  }
  else // mesafe 30cm'den büyük ise düz git.
  {
    analogWrite(5, hiz);
    analogWrite(6, 0);
    analogWrite(9, hiz);
    analogWrite(10, 0);
  }
}
```



#### Sıra Sizde

Eğitsel robotun gövdesine mesafe sensörünü bağlayıp diğer robot bağlantılarının düzgün olduğundan emin olduktan sonra mikrodenetleyici IDE programını açarak hız değişkenine 100 değerini atayınız. Engelden kaçan robota ait program kodlarını yazınız. Tüm işlemler bittikten sonra programı mikrodenetleyici karta yükleyiniz. Robotun pillerini düzgün bir şekilde takınız ve açma kapama anahtarını açıp, robotun çalışmasını gözlemleyiniz.



#### Değerlendirme

Çalışmalarınız öğrenme biriminin sonunda yer alan Kontrol Listesi kullanılarak değerlendirilecektir. Çalışmalarınızı yaparken değerlendirme ölçütlerini dikkate alınız.

**Normalde hareket etmeyen engeli görünce geri giden robot (yakalanmayan robot) kodları şunlardır:**

```
int trig = 7; // Mesafe sensörü trig pini için D7 portunu belirle
int echo = 8; // Mesafe sensörü echo pini için D8 portunu belirle
int hiz=150; // Motor dönüş hızını belirle

void setup(){
  pinMode(5, OUTPUT); // sağ motor ileri pini
  pinMode(6, OUTPUT); // sağ motor geri pini
  pinMode(9, OUTPUT); // sol motor ileri pini
  pinMode(10, OUTPUT); // sol motor geri pini
  pinMode(trig, OUTPUT); // Mesafe sensörü trig pini çıkış için
  pinMode(echo, INPUT); // Mesafe sensörü echo pini giriş için
}

void loop(){
  digitalWrite(trig, 1);
  delay(1);
  digitalWrite(trig, 0);
  int sure = pulseIn(echo, 1);
  int mesafe = (sure/2) / 28.97;

  if (mesafe < 30 ) // mesafe 30cm'den küçük ise robotu geri gel.
  {
    analogWrite(5, 0);
    analogWrite(6, hiz);
    analogWrite(9, 0);
    analogWrite(10, hiz);
  }
  else // mesafe 30cm'den büyük ise dur.
  {
    analogWrite(5, 0);
    analogWrite(6, 0);
    analogWrite(9, 0);
    analogWrite(10, 0);
  }
}
```



#### Sıra Sizde

Eğitsel robotun gövdesine mesafe sensörünü bağlayıp diğer robot bağlantılarının düzgün olduğundan emin olduktan sonra mikrodenetleyici IDE programını açarak hız değişkenine 150 değerini atayınız. Yakalanmayan robota ait program kodlarını yazınız. Tüm işlemler bittikten sonra programı mikrodenetleyici karta yükleyiniz. Robotun pillerini düzgün bir şekilde takınız ve açma kapama anahtarını açıp, robotun çalışmasını gözlemleyiniz.

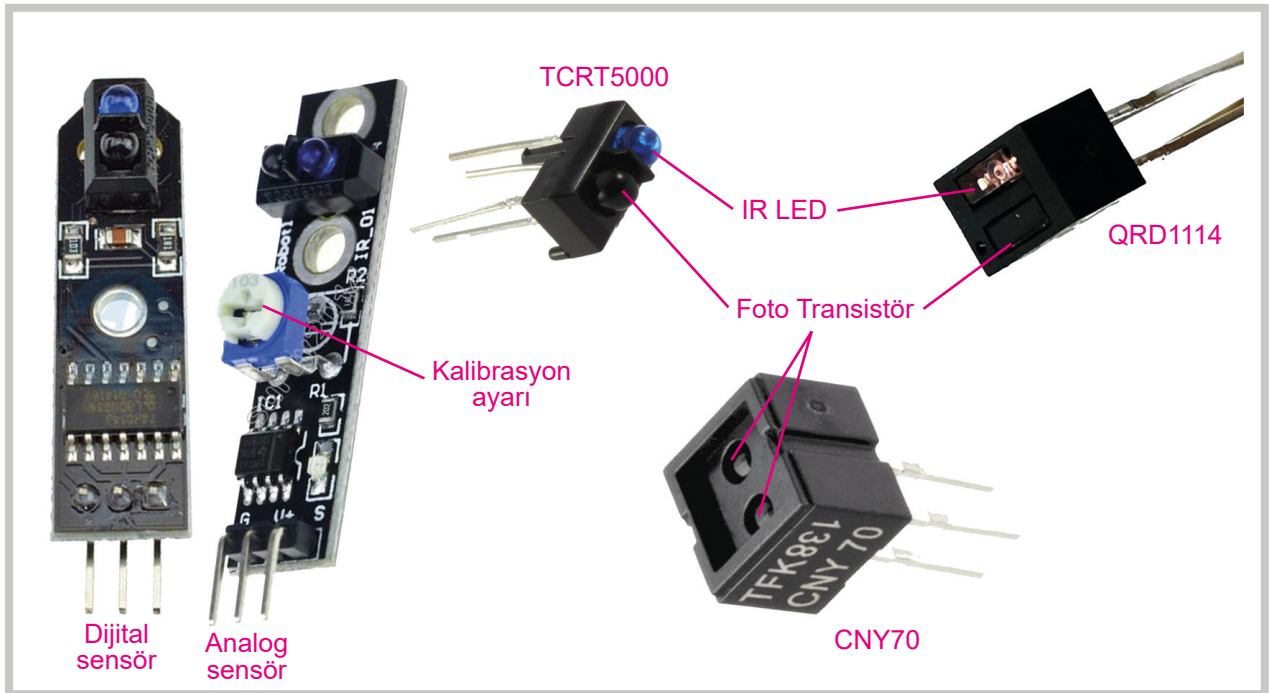


#### Değerlendirme

Çalışmalarınız öğrenme biriminin sonunda yer alan Kontrol Listesi kullanılarak değerlendirilecektir. Çalışmalarınızı yaparken değerlendirme ölçütlerini dikkate alınız.

### 3.8. EĞİTSEL ROBOT İLE ÇİZGİ İZLEME

Bu bölümde eğitsel robotun çizgi izleme uygulamasını nasıl yapacağına yer verilecektir. Çizgi izleme sensörleri yapısal olarak kızılötesi LED ile foto transistörden oluşur. Kızılötesi LED'den çıkan sinyaller karşı cisme çarpar. Cisim açık renk ise ışığı yansıtır. Bu şekilde sensör bir değer üretir. Eğer sinyaller yansımayacak kadar cisme uzak bir noktadaysa veya cisim siyah renkliyse (siyah renk ışığı yansıtmayacağından) 0 değerini üretir. Çizgi izleme sensörleri çıkış durumlarına göre; dijital çıkış verenler (0 - 1) ve analog çıkış verenler (0-1023) şeklinde iki çeşittir. Yapısal olarak bakıldığında ise kullanılan sensörün çeşidine ve özelliğine göre farklı şekillerde üretilir. Ayrıca çizgi izleme sensörleri tekli üretildiği gibi ihtiyaca göre üçlü, beşli, sekizli gibi farklı sayılarda paket hâlinde de üretilebilmektedir. Çizgi izleme sensör çeşitlerine Görsel 3.34'te yer verilmiştir.



Görsel 3.34: Çizgi izleme sensör çeşitleri

### 3. Öğrenme Birimi: Robot Tabanlı Proje Geliştirme

Görsel 3.34'te verilen çizgi izleme sensör çeşitlerinde farklı firmaların ürettiği (TCRT5000, CNY70, QRD1114 vb.) IR LED+Foto Transistör ikilisi görülmektedir. Bu ikiliden oluşan devre elemanları kullanılarak dijital veya analog çizgi izleme sensör devreleri üretilmektedir. Dijital sensörlerde sinyal kuvvetlendirmek için 7414 entegresi, analog sensörlerde LM358 entegresi kullanılır. Bazı durumlarda bu entegrelerin farklı versiyonları da kullanılabilir.

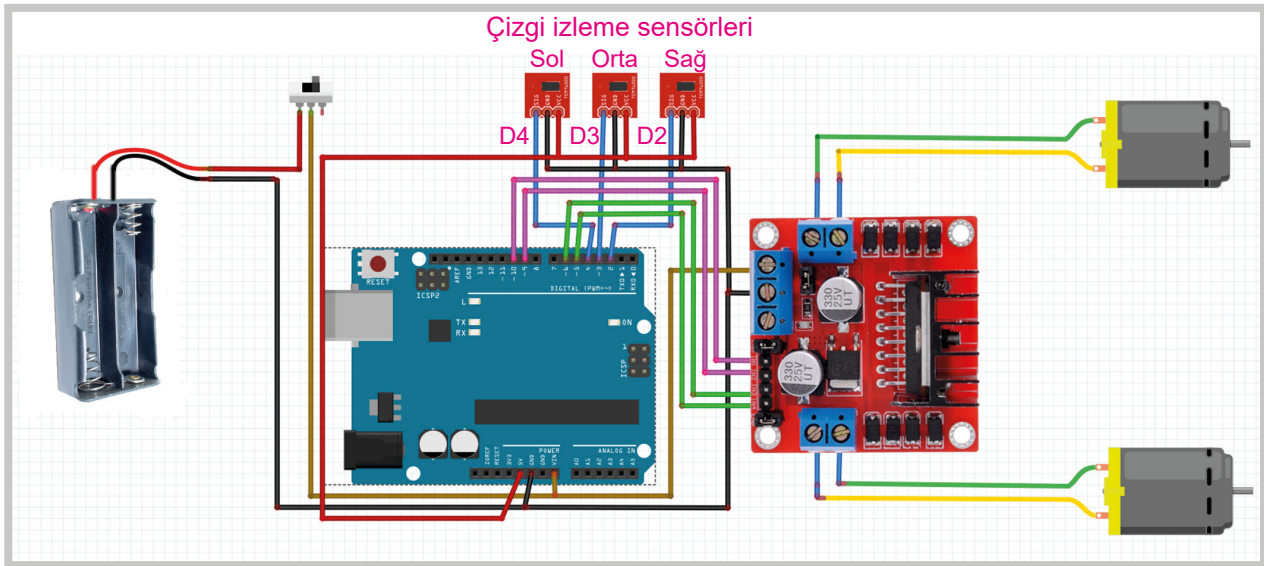
Eğitsel robotla beyaz zemin üzerinde siyah çizgiyi izleme işlemini gerçekleştirebilmek için 3 adet çizgi izleme sensörüne ihtiyaç vardır. Eğitsel robotta, dijital çıkış veren çizgi izleme sensörü kullanılmıştır. Montaj aparatı kullanılarak eğitsel robotun ön tarafına Görsel 3.35.a'da görüldüğü gibi vidalar ile çizgi izleme sensörlerinin montajı yapılır. Çizgi izleme sensörleri ve montaj aparatı Görsel 3.35.b'de görülmektedir. Çizgi izleme sensörü aparatı altta, mesafe sensörü aparatı üstte, robot gövdesi ortada kalacak şekilde vida delikleri üst üste getirilip, aynı vida kullanılarak birbirlerine tutturulacaktır.



a) Çizgi izleme sensörleri ve montaj aparatı

b) Çizgi izleme sensörlerinin robot üzerine montajının yapılması

**Görsel 3.35: Çizgi izleyen robot bağlantı aşaması**



**Görsel 3.36: Çizgi izleme sensörlerinin bağlantı şeması**

### 3. Öğrenme Birimi: Robot Tabanlı Proje Geliştirme

Çizgi izleme sensörlerinin Görsel 3.36'daki gibi elektriksel bağlantıları yapılırken, "Vcc" uçları mikrodenetleyici kartı üzerindeki "5V" pinine, "GND" uçları mikrodenetleyici kartı üzerindeki "GND" pinine bağlanır. Çizgi izleme sensörlerinin bağlantısı yapılırken, sağ sensörün "Out" ucu mikrodenetleyici kartı üzerindeki "D2" pinine, orta sensörün "Out" ucu mikrodenetleyici kartı üzerindeki "D3" pinine ve sol sensörün "Out" ucu mikrodenetleyici kartı üzerindeki "D4" pinine bağlanır. Tüm bağlantılar yapıldıktan sonra çizgi izleme sensörlerinin, beyaz zemin üzerinde bulunan siyah çizgi üzerindeki konumlarına göre motorlarının nasıl çalışacağı belirlenir. Motorların, çizgi üzerindeki sensörlerin konumlarına göre çalışması Tablo 3.5'te verilmiştir.

Çizgi izleme sensörlerinin bağlantıları yapıldıktan sonra, çizgi izleme sensörlerinin çalıştığını görmek için basit bir program yazılması ve çizgi izleyen sensörlerin gönderdiği verilerin, seri port ekranında incelenmesi gerekir. Öncelikle Görsel 3.37'deki program yazılıp mikrodenetleyici karta yüklenmelidir. Sonra beyaz bir A4 kâğıdı üzerine siyah bir elektrik bandı ile çizgi çizilir. Mikrodenetleyici IDE programından seri port ekranı açılır. Eğitsel robotta bulunan sensörler sırasıyla çizgi üzerinde gezdirilir. Seri port ekranındaki veriler incelendiğinde siyah çizginin üzerindeki sensör değeri 0, diğer sensör değerleri 1 olacaktır. Bunun sebebi siyah rengin ışığı yansıtmasıdır. Çizgi izleme sensörlerinden gelen verilerin okunmasını sağlayan program kodları aşağıdaki gibidir.

```
void setup()
{
  pinMode(2, INPUT); // sağ çizgi sensör giriş pini
  pinMode(3, INPUT); // orta çizgi sensör giriş pini
  pinMode(4, INPUT); // sol çizgi sensör giriş pini
  Serial.begin(9600); // Seri port veri akışını başlat
}
void loop() {
  int sR = digitalRead(2); //sağ çizgi sensör verisinin okunması
  int sC = digitalRead(3); //orta çizgi sensör verisinin okunması
  int sL = digitalRead(4); //sol çizgi sensör verisinin okunması
  Serial.print("Sol Sensor="); Serial.print(sL);
  Serial.print(", Orta Sensor="); Serial.print(sC);
  Serial.print(", Sag Sensor="); Serial.println(sR);
  delay(1);
}
```

```
Sol Sensor=0, Orta Sensor=1, Sag Sensor=1
Sol Sensor=1, Orta Sensor=0, Sag Sensor=1
Sol Sensor=1, Orta Sensor=0, Sag Sensor=1
Sol Sensor=1, Orta Sensor=0, Sag Sensor=1
Sol Sensor=1, Orta Sensor=1, Sag Sensor=0
Sol Sensor=1, Orta Sensor=1, Sag Sensor=0
Sol Sensor=0, Orta Sensor=1, Sag Sensor=1
Sol Sensor=0, Orta Sensor=1, Sag Sensor=1
Sol Sensor=0, Orta Sensor=1, Sag Sensor=1
Sol Sensor=1, Orta Sensor=0, Sag Sensor=1
Sol Sensor=1, Orta Sensor=0, Sag Sensor=1
Sol Sensor=1, Orta Sensor=1, Sag Sensor=0
Sol Sens
```

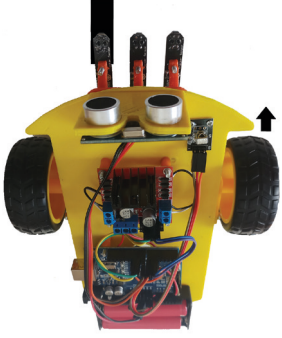
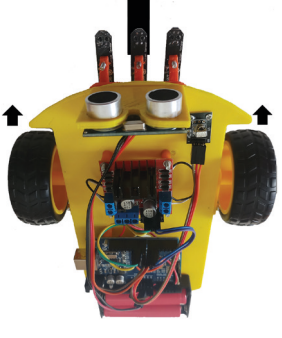
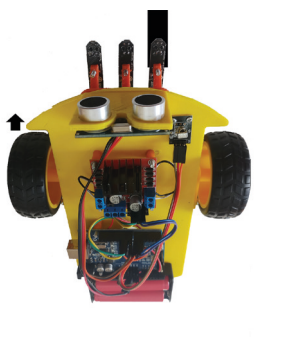
Görsel 3.37: Çizgi izleme sensörlerinden gelen verilerin okunması



### 3. Öğrenme Birimi: Robot Tabanlı Proje Geliştirme

Çizginin sağ, sol veya orta sensör tarafından görülmesi durumuna göre motorların hareketi şu şekilde sağlanır:

Tablo 3.5: Çizginin Durumuna Göre Motorların Hareket Durumları

Sol sensör çizgi üzerinde Sol=0, Orta=1, Sağ=1	Orta sensör çizgi üzerinde Sol=1, Orta=0, Sağ=1	Sağ sensör çizgi üzerinde Sol=1, Orta=1, Sağ=0
		
D9 = 0, D10=0, D5 = 1, D6=0	D9 = 1, D10=0, D5 = 1, D6=0	D9 = 0, D10=0, D5 = 1, D6=0
Robot sağ teker sabit, sağa doğru döner.	Robot ileri yönde hareket eder.	Robot sol teker sabit, sola doğru döner.



#### Sıra Sizde

Eğitsel robotun gövdesine çizgi izleme sensörlerini bağlayıp diğer robot bağlantılarının düzgün olduğundan emin olduktan sonra mikrodenetleyici IDE programını açarak çizgi izleme sensörlerinden gelen verilerin okunmasını sağlayan program kodlarını yazınız. Tüm işlemler bittikten sonra programı mikrodenetleyici karta yükleyiniz. Robotunuza pillerinizi düzgün bir şekilde takınız ve açma kapama anahtarınızı açıp çizgi izleyen sensörlerin gönderdiği verileri seri port ekranından gözlemleyiniz.



#### Değerlendirme

Çalışmalarınız öğrenme biriminin sonunda yer alan Kontrol Listesi kullanılarak değerlendirilecektir. Çalışmalarınızı yaparken değerlendirme ölçütlerini dikkate alınız.

Çizgi izleyen robot için tüm sensörlerin düzgün çalışması çok önemlidir. Görsel 3.43'teki program aracılığıyla sensörlerin çalıştığından emin olunduktan sonra çizgi izleyen robot kodlarına geçilebilir. Çizgi izleyen robotta, çizginin sağda, solda veya ortada olmasına göre motorlara ileri yönde enerji verilmelidir. Tablo 3.5 incelendiğinde, orta sensör çizgiyi gördüğünde (orta sensör değeri 0, diğer sensör değerleri 1 olduğunda) motorlar ileri yönde hareket edecektir. Sol sensör çizgiyi gördüğünde (sol sensör değeri 0, diğer sensör değerleri 1 olduğunda) yalnızca sağ motor ileri yönde hareket edecektir. Bu şekilde robot, orta sensör çizgiyi görene kadar sola doğru yönelecektir. Son durumda sağ sensör çizgiyi gördüğünde (sağ sensör değeri 0, diğer sensör değerleri 1 olduğunda) yalnızca sol motor ileri yönde hareket edecektir. Bu şekilde de robot orta sensör çizgiyi görene kadar sağa doğru yönelecektir.

Üç sensörlü çizgi izleyen robot kodu şu şekildedir:

```
int h=100;
void setup() {
  pinMode(5, OUTPUT); // sağ motor ileri pini
  pinMode(6, OUTPUT); // sağ motor geri pini
  pinMode(9, OUTPUT); // sol motor ileri pini
  pinMode(10, OUTPUT); // sol motor geri pini

  pinMode(2, INPUT); // Sağ Cizgi sensor
  pinMode(3, INPUT); // Orta Cizgi sensor
  pinMode(4, INPUT); // Sol Cizgi sensor
}
void loop() {
  int sR = digitalRead(2);
  int sC = digitalRead(3);
  int sL = digitalRead(4);

  if ((sL==1) && (sC==0) && (sR==1)) ileri();
  if ((sL==0) && (sC==1) && (sR==1)) sol();
  if ((sL==1) && (sC==1) && (sR==0)) sag();
}

void dur() {
  analogWrite(5, 0);
  analogWrite(6, 0);
  analogWrite(9, 0);
  analogWrite(10, 0);
}
void ileri() {
  analogWrite(5, h);
  analogWrite(6, 0);
  analogWrite(9, h);
  analogWrite(10, 0);
}
void sol() {
  analogWrite(5, h);
  analogWrite(6, 0);
  analogWrite(9, 0);
  analogWrite(10, 0);
}
void sag() {
  analogWrite(5, 0);
  analogWrite(6, 0);
  analogWrite(9, h);
  analogWrite(10, 0);
}
```



#### Sıra Sizde

Eğitsel robotun gövdesine çizgi izleme sensörlerini bağlayıp diğer robot bağlantılarının düzgün olduğundan emin olduktan sonra mikrodenetleyici IDE programını açıp 3 sensörlü çizgi izleyen robot programını yazınız. Tüm işlemler bittikten sonra programı mikrodenetleyici karta yükleyiniz. Robotun pillerini düzgün bir şekilde takınız ve açma kapama anahtarını açıp, robotun çalışmasını gözlemleyiniz.



#### Değerlendirme

Çalışmalarınız öğrenme biriminin sonunda yer alan Kontrol Listesi kullanılarak değerlendirilecektir. Çalışmalarınızı yaparken değerlendirme ölçütlerini dikkate alınız.

**Engeli gördüğünde duran, engel olmadığında çizgi izleyen robot kodları şunlardır:**

```
int trig = 7; // Mesafe sensörü trig pini için D7 portunu belirle
int echo = 8; // Mesafe sensörü echo pini için D8 portunu belirle
int h=100; // Motor dönüş hızını belirle
void setup(){
    pinMode(5, OUTPUT); // sağ motor ileri pini
    pinMode(6, OUTPUT); // sağ motor geri pini
    pinMode(9, OUTPUT); // sol motor ileri pini
    pinMode(10, OUTPUT); // sol motor geri pini

    pinMode(2, INPUT); // Sağ Cizgi sensor
    pinMode(3, INPUT); // Orta Cizgi sensor
    pinMode(4, INPUT); // Sol Cizgi sensor

    pinMode(trig, OUTPUT);
    pinMode(echo, INPUT);
}
void loop(){
    digitalWrite(trig, 1);
    delay(1);
    digitalWrite(trig, 0);
    int sure = pulseIn(echo, 1);
    int mesafe = (sure/2) / 28.97;
    delay(1);
    int sR = digitalRead(2);
    int sC = digitalRead(3);
    int sL = digitalRead(4);

    if (mesafe < 30 ) // mesafe 30 cm'den küçük ise robotu durdur.
    {
        dur();
    }
    else // mesafe 30 cm'den büyük ise çizgiyi takip et.
    {
```

### 3. Öğrenme Birimi: Robot Tabanlı Proje Geliştirme

```
if ((sL==1) && (sC==0) && (sR==1)) ileri();
if ((sL==0) && (sC==1) && (sR==1)) sol();
if ((sL==1) && (sC==1) && (sR==0)) sag();
}

}
void dur(){
analogWrite(5, 0);
analogWrite(6, 0);
analogWrite(9, 0);
analogWrite(10, 0);
}
void ileri() {
analogWrite(5, h);
analogWrite(6, 0);
analogWrite(9, h);
analogWrite(10, 0);
}
void sol() {
analogWrite(5, h);
analogWrite(6, 0);
analogWrite(9, 0);
analogWrite(10, 0);
}
void sag() {
analogWrite(5, 0);
analogWrite(6, 0);
analogWrite(9, h);
analogWrite(10, 0);
}
```



#### Sıra Sizde

Eğitsel robotun gövdesine çizgi izleme sensörlerini bağlayıp diğer robot bağlantılarının düzgün olduğundan emin olduktan sonra mikrodenetleyici IDE programını açıp engeli gördüğünde duran, engel olmadığında çizgi izleyen robot programını yazınız. Tüm işlemler bittikten sonra programı mikrodenetleyici karta yükleyiniz. Robotun pillerini düzgün bir şekilde takınız ve açma kapama anahtarını açıp, robotun çalışmasını gözlemleyiniz.

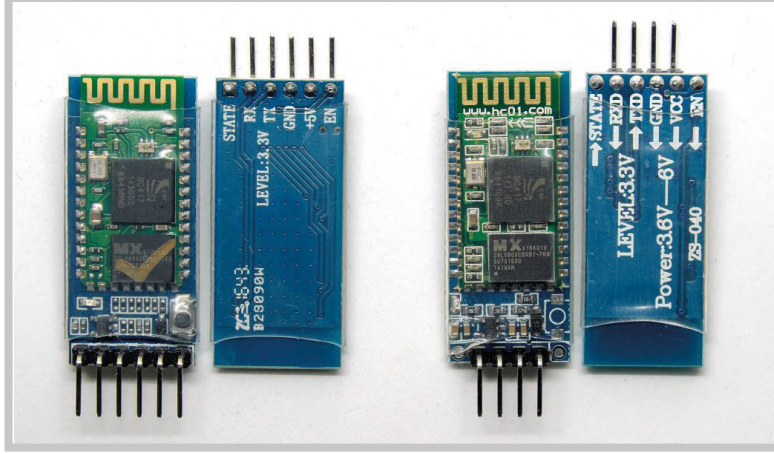


#### Değerlendirme

Çalışmalarınız öğrenme biriminin sonunda yer alan Kontrol Listesi kullanılarak değerlendirilecektir. Çalışmalarınızı yaparken değerlendirme ölçütlerini dikkate alınız.

## 3.9. EĞİTSEL ROBOTUN BLUETOOTH KONTROLÜ

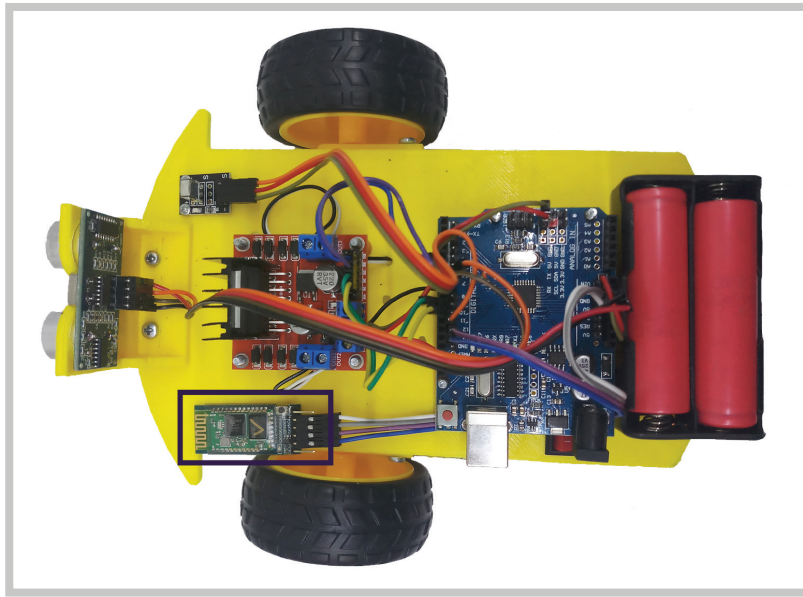
Bu bölümde eğitsel robotun bluetooth ile kontrol uygulamasına yer verilecektir. Bluetooth, başlangıçta cep telefonları ve mobil cihazlar için geliştirilmiş kablosuz haberleşme teknolojisidir. Günümüzde sadece mobil cihazlarda değil evlerde, otomobillerde, iş yerlerinde yaygın bir kullanım alanına sahiptir. Görsel 3.38'deki bluetooth modülü, herhangi bir mikrodenetleyici kart devresine bağlandığında bluetooth bağlantısı olan cep telefonu, tablet, bilgisayar gibi farklı ortamlardan mikrodenetleyici karta veri göndermesine olanak tanınır.



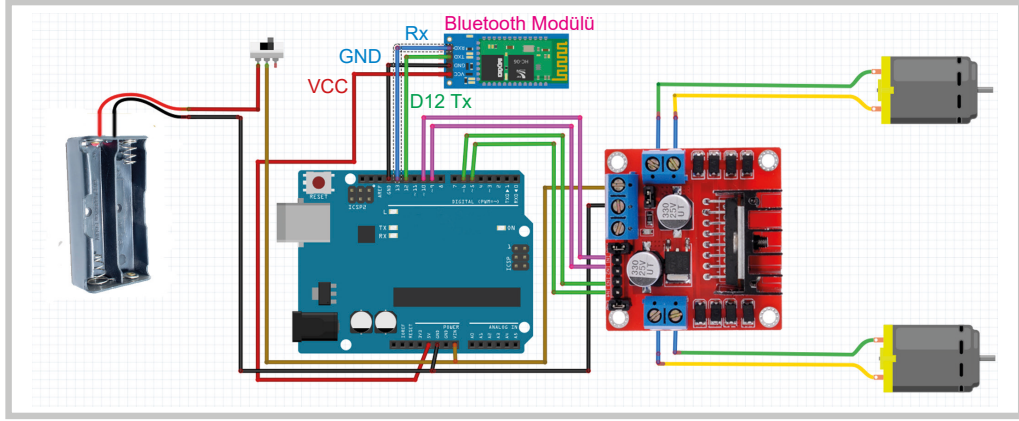
Görsel 3.38: Bluetooth modülleri

Eğitsel robotta bluetooth modülünün montajı, Görsel 3.39'daki gibi robot gövdesi üzerinde uygun bir bölüme (vida ile tutturma olanağı olmadığı için) sıcak silikon ile yapıştırılarak yapılabilir. Bluetooth modülünün kablo bağlantısını yapmak için Görsel 3.40'tan yararlanılabilir. Bluetooth modüllerinde her ne kadar 6 adet bağlantı ucu bulunsa da bunlardan 4 tanesini kullanmak yeterlidir.

Bluetooth modülünün Görsel 3.40'taki gibi elektriksel bağlantılarını yaparken "5V" pinini mikrodenetleyici kart üzerindeki "5V" pinine, "GND" pinini mikrodenetleyici kart üzerindeki "GND" pinine, Tx pinini mikrodenetleyici kart üzerindeki "D12" pinine, Rx pinini mikrodenetleyici kart üzerindeki "D13" pinine dişi-dişi jumper kablolarıyla bağlamak gerekir.



Görsel 3.39: Bluetooth modülünün eğitsel robot üzerine montajının yapılması



Görsel 3.40: Bluetooth modülünün bağlantı şeması

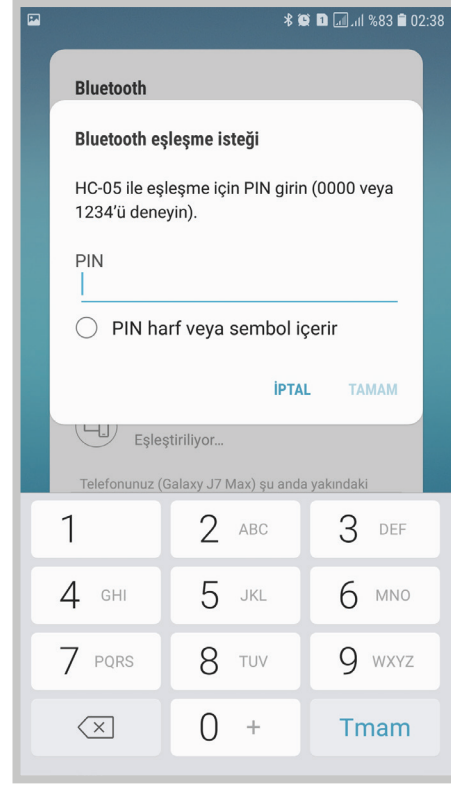
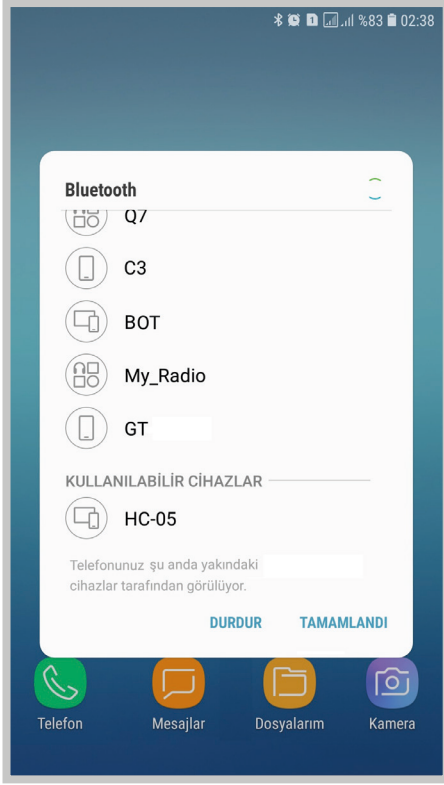
Bluetooth modülünün elektriksel bağlantısı yapıldıktan sonra bluetooth bağlantısı olan bir cihazdan (cep telefonu, tablet vb.) gönderilen veriyi alacak şekilde, basit bir program yazılmalıdır. Bluetooth modülünü mikrodenetleyici kartının D12 ve D13 pinlerini Tx, Rx olarak kullanabilmek için `<SoftwareSerial.h>` kütüphanesinden yararlanılacaktır. Programda, bluetooth modülünün farklı cihazdan (cep telefonu, tablet vb.) gönderdiği veriler, mikrodenetleyici IDE programının seri port ekranında gösterilecektir.

**Bluetooth modülünden aldığı verileri seri port ekranında gösteren program kodları şunlardır:**

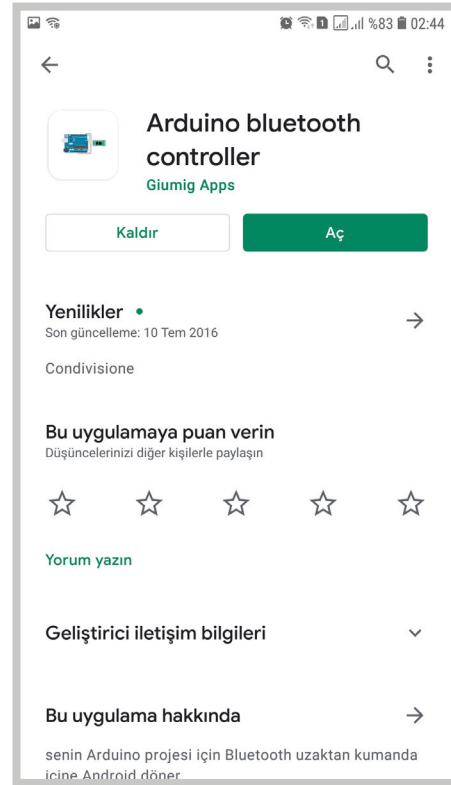
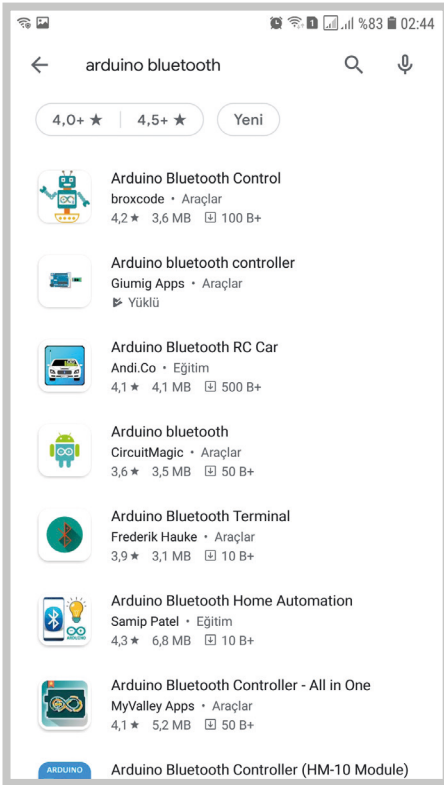
```
#include<SoftwareSerial.h>
SoftwareSerial bluetooth1(12,13); //tx=12,Rx=13
char veri;
void setup() {
  Serial.begin(9600);
  bluetooth1.begin(9600);
}
void loop() {
  if(bluetooth1.available())
  {
    veri=bluetooth1.read();
    Serial.println(veri);
  }
}
```

Bluetooth modülünden aldığı verileri seri port ekranında gösteren program kodları, mikrodenetleyici IDE programında yazılıp mikrodenetleyici kartına yüklenmelidir. Yükleme tamamlandıktan sonra eğitsel robot üzerindeki bluetooth modülü ile veri gönderecek cihazı (cep telefonu, tablet vb.) eşleştirmek gerekir. Öncelikle cihazın bluetooth bağlantısı açılıp Görsel 3.41'deki gibi robot üzerine takılı "HC-05" adındaki bluetooth modülünü taratmak gerekir. Bluetooth modülünün ismi ilk kullanımda HC-04, HC-05, HC-06, SPP-CA vb. olabilir. Bluetooth modülünün ismi tablet veya cep telefonunda kullanılabilir cihazlar listesinde yer aldığı anda üzerine dokunulması ve eşleştirme pininin girilmesi gerekir. Bluetooth modülünün eşleştirme pini ilk kullanımda 0000 veya 1234 olabilir. Eşleştirme pini kabul edildikten ve başarılı bir şekilde bluetooth modülü cihazla eşleştirildikten sonra tekrardan bu işlemleri yapmaya gerek yoktur. Cihazın bluetooth bağlantısı her açıldığında robot üzerinde bulunan bluetooth modülünü otomatik olarak görüp eşleşmeyi yapacaktır.

### 3. Öğrenme Birimi: Robot Tabanlı Proje Geliştirme



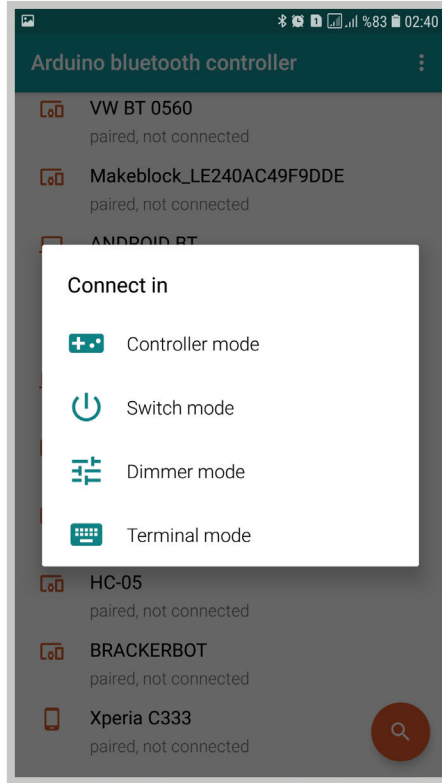
Görsel 3.41: Bluetooth modülünün tablet ve cep telefonu ile eşleştirilmesi



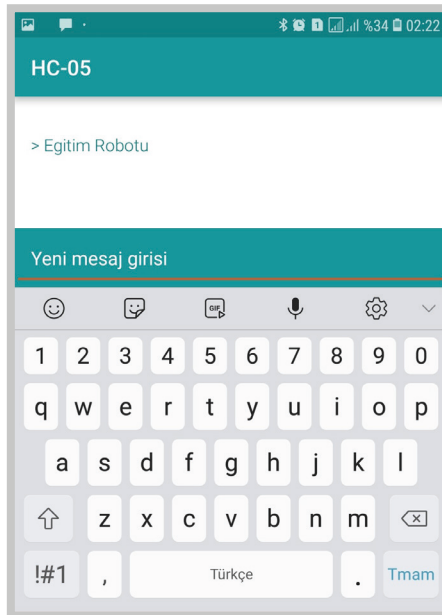
Görsel 3.42: Bluetooth kontrol uygulamasının yüklenmesi

### 3. Öğrenme Birimi: Robot Tabanlı Proje Geliştirme

Tablet veya cep telefonları gibi cihazlardan bluetooth modülüne veri gönderebilmek için bir uygulama programı yüklemek gerekir. Android cihazlar için play store uygulaması açılıp arama ekranından “Arduino Bluetooth” diye aratılır. Listede çıkan Görsel 3.41’deki uygulamalardan ücretsiz olan “Arduino Bluetooth Controller” uygulaması seçilir. Listeden istenilen uygulama seçildikten sonra cihaza kurulum gerçekleştirilir. Kurulumdan sonra program açılır ve bluetooth modülünün ismi seçilir. Görsel 3.43’teki ekrandan “Terminal Mode” seçeneği seçilir.



Görsel 3.43: Bluetooth kontrol uygulamasının açılması



Görsel 3.44: Bluetooth kontrol uygulamasında terminal mode seçeneği

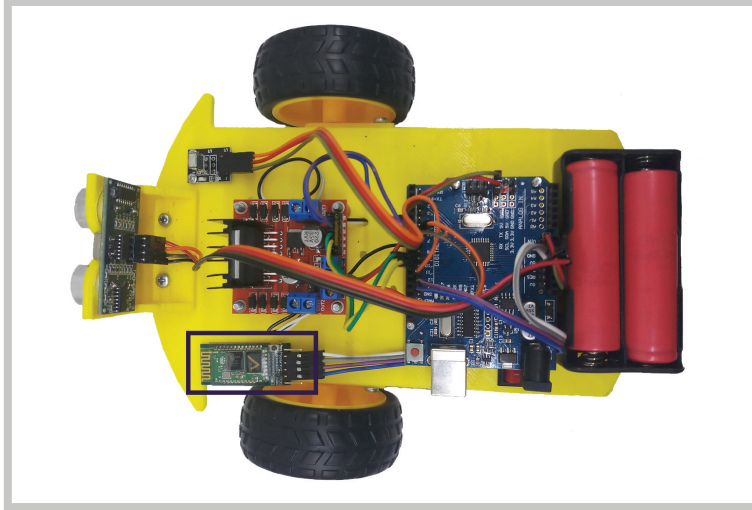


Terminal mode ekranından Görsel 3.44'teki gibi bluetooth modülüne gönderilmek istenen mesaj girilip tamam butonuna basıldığında, Görsel 3.45'teki gibi seri port ekranında gönderilen mesaj görüntülenir. Bluetooth modülünün cihazdan aldığı verilerin seri portta görüntülenmesini sağlayan program kodları aşağıdaki gibidir.

```
#include<SoftwareSerial.h>
SoftwareSerial bluetooth1(12,13); //tx=12,Rx=13
char veri;

void setup(){
  Serial.begin(9600);
  bluetooth1.begin(9600);
}

void loop(){
  if(bluetooth1.available()){
    {
      veri=bluetooth1.read();
      Serial.println(veri);
    }
  }
}
```



Görsel 3.45: Bluetooth modülünün eğitsel robot üzerine montajının yapılması



#### Sıra Sizde

Eğitsel robotun gövdesine bluetooth modülünü bağlayınız. Diğer robot bağlantılarının düzgün olduğundan emin olduktan sonra mikrodenetleyici IDE programını açınız. Bluetooth modülünden aldığı verileri seri port ekranında gösteren programı yazınız. Tüm işlemler bittikten sonra programı mikrodenetleyici karta yükleyiniz. Veri göndermek için kullanacağınız cihaza bluetooth kontrol uygulamasını yükleyiniz. Görsel 3.44'teki gibi **Terminal mode** ekranında bir mesaj gönderiniz.



#### Değerlendirme

Çalışmalarınız öğrenme biriminin sonunda yer alan Kontrol Listesi kullanılarak değerlendirilecektir. Çalışmalarınızı yaparken değerlendirme ölçütlerini dikkate alınız.

### 3. Öğrenme Birimi: Robot Tabanlı Proje Geliştirme

Bluetooth modülü ile cihaz arasında mesaj alışverişi sorunsuz bir şekilde gerçekleştiyse, eğitsel robotun bluetooth üzerinden kontrol edilmesini sağlayacak program kodlarına geçilebilir. Programı yazarken öncelikle "bluetooth1" isminde bir **SoftwareSerial** nesnesi oluşturulur. Sonraki aşamada **bluetooth1.read()** ; komutuyla bluetooth modülünden gelen veriler "veri" ismindeki ve **char** tipindeki değişkene aktarılır. En son aşamada ise gelen veriye göre **if(veri=='1')** şart komutuyla robotun hareket fonksiyonları çalıştırılır.

Eğitsel robotun bluetooth üzerinden kontrol edilmesini sağlayan program kodları şunlardır:

```
#include<SoftwareSerial.h>
SoftwareSerial bluetooth1(12,13); //tx=12,Rx=13

int h = 150;
char veri;

void setup() {

bluetooth1.begin(9600);
  pinMode(5, OUTPUT); // sağ motor ileri pini
  pinMode(6, OUTPUT); // sağ motor geri pini
  pinMode(9, OUTPUT); // sol motor ileri pini
  pinMode(10, OUTPUT); // sol motor geri pini

}

void loop() {
if(bluetooth1.available())
{
  veri= bluetooth1.read();

  if(veri=='1') ileri();
  if(veri=='2') geri();
  if(veri=='3') sol();
  if(veri=='4') sag();
  if(veri=='5') dur();

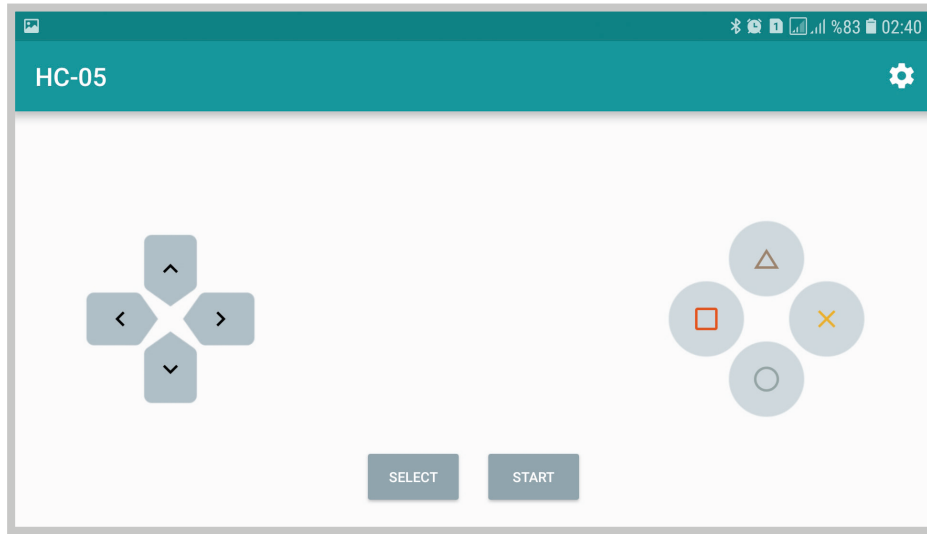
}

}

void ileri() {
analogWrite(5,h);
analogWrite(6,0);
analogWrite(9,h);
analogWrite(10,0);
}

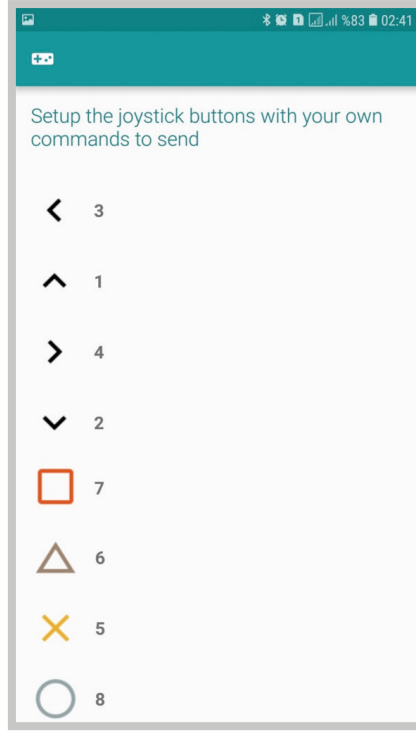
void geri() {
analogWrite(5,0);
analogWrite(6,h);
```

```
analogWrite(9,0);
analogWrite(10,h);
}
void sol(){
analogWrite(5,h);
analogWrite(6,0);
analogWrite(9,0);
analogWrite(10,h);
}
void sag(){
analogWrite(5,0);
analogWrite(6,h);
analogWrite(9,h);
analogWrite(10,0);
}
void dur(){
analogWrite(5,0);
analogWrite(6,0);
analogWrite(9,0);
analogWrite(10,0);
}
```

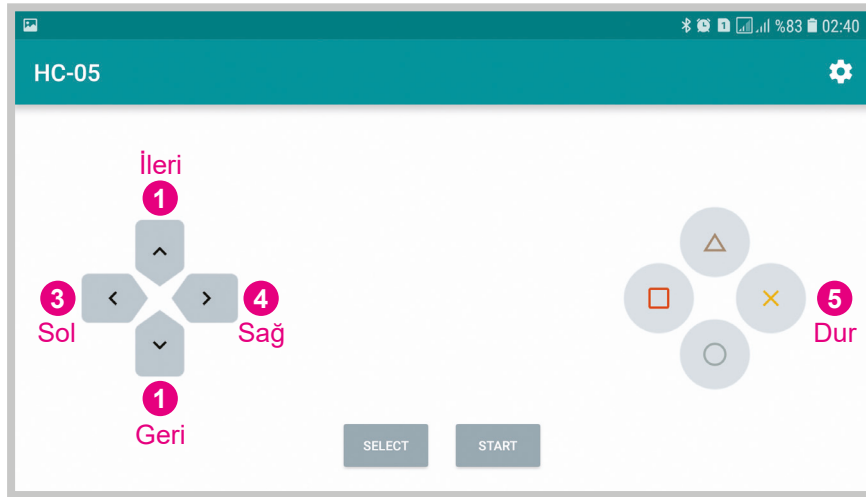


Görsel 3.46: Bluetooth kontrol uygulaması Controller Mode seçeneği

Eğitsel robotu bluetooth üzerinden kontrol edebilmek için öncelikle, mikrodenetleyici IDE programında ilgili programın kodlarının yazılıp, mikrodenetleyici karta yüklenmesi gerekir. Tablet veya cep telefonu gibi cihazların Görsel 3.46'daki gibi bluetooth kontrol uygulaması çalıştırılıp, **Controller Mode** seçeneği ile Görsel 3.47'deki pencerenin açılması gerekir. Bluetooth kontrol uygulamasındaki **Controller Mode** seçeneğinde bazı ayarlamalar yapılmalıdır. Görsel 3.47'deki program penceresinin sağ üst köşesindeki dişli çark simgesine tıklanıp butonlara basıldığında, gönderilecek rakamları belirlemek için Görsel 3.47'deki veri girişleri yapılmalıdır. Veri giriş işlemi yapıldıktan sonra tekrar Görsel 3.47'deki ekrana geri dönülerek eğitsel robot, bluetooth üzerinden kontrol edilmeye başlanabilir. Robotun kontrol edilmesinde Görsel 3.48'deki gibi sadece 5 buton kullanılmış olup robota ek fonksiyonlar eklemek için diğer butonlar da kullanılabilir.



Görsel 3.47: Bluetooth kontrol uygulaması Controller Mode ayarları



Görsel 3.48: Bluetooth kontrol uygulaması Controller Mode ile robotu yönlendirme



#### Sıra Sizde

Eğitsel robotun gövdesine bluetooth modülünü bağlayınız. Diğer robot bağlantılarının düzgün olduğundan emin olduktan sonra mikrodenetleyici IDE programını açınız. Eğitsel robotun bluetooth üzerinden kontrol edilmesini sağlayan programı yazınız. Tüm işlemler bittikten sonra programı mikrodenetleyici karta yükleyiniz. Görsel 3.48'deki gibi bluetooth kontrol uygulamasının Controller Mode seçeneğiyle robotunuzu kontrol ediniz.

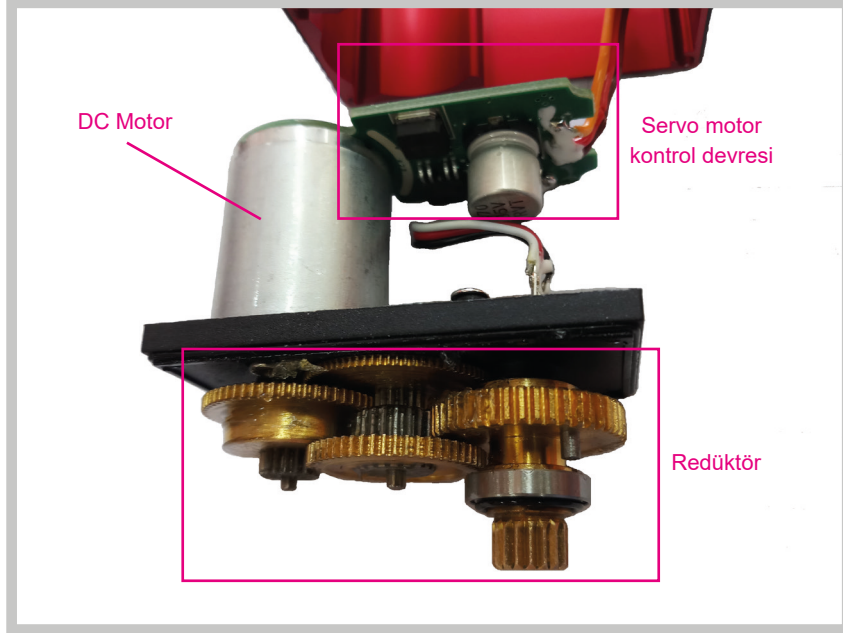


#### Değerlendirme

Çalışmalarınız öğrenme biriminin sonunda yer alan Kontrol Listesi kullanılarak değerlendirilecektir. Çalışmalarınızı yaparken değerlendirme ölçütlerini dikkate alınız.

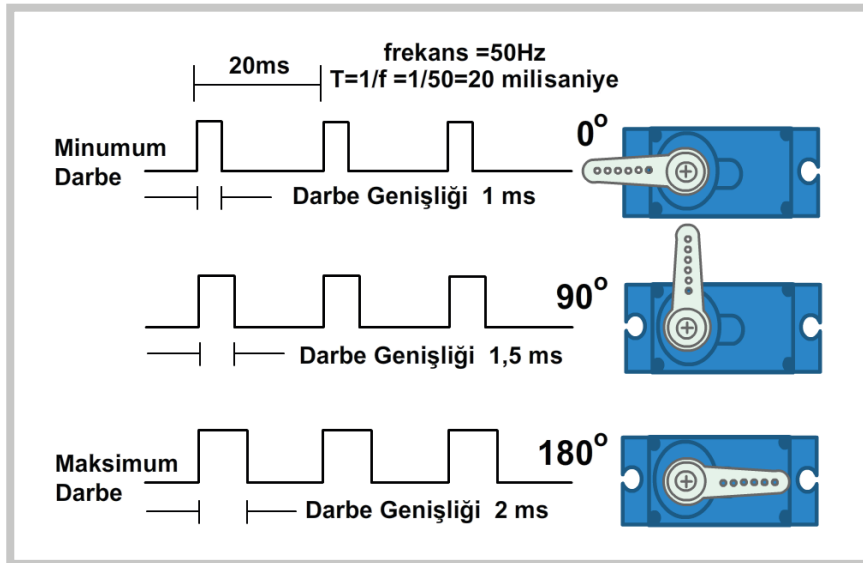
### 3.10. SERVO MOTOR İLE ROBOT UYGULAMALARI

Özellikle yürüme yeteneğine sahip (insansı robotlar, örümcek robot, dans eden robot vb.) robotlarda ve bazı robot kollarında servo motorlar kullanılır. Servo motorlar DC motorların gelişmiş versiyonlarıdır. Servo motorlar Görsel 3.49'da görüldüğü gibi enkoder, servo motor kontrol devresi, redüktör ve DC motordan oluşan gelişmiş motorlardır.



Görsel 3.49: Servo motor iç yapısı

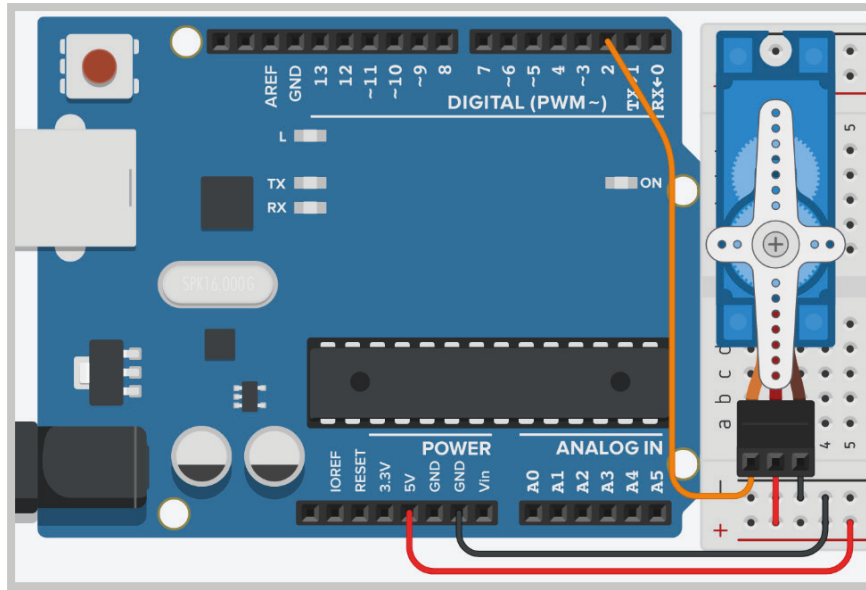
Servo motorları gönderilen PWM sinyalleri sayesinde, belirli açılarda döndürme imkânı vardır. Görsel 3.50'de görüldüğü gibi 20 milisaniyelik bir periyotta gönderilen pozitif darbe genişliği 1 milisaniye olduğunda motor mili  $0^\circ$  dönmektedir. Pozitif darbe genişliği 1.5 milisaniye olduğunda motor mili  $90^\circ$ , pozitif darbe genişliği 2 milisaniye olduğunda ise motor mili  $180^\circ$  dönmektedir. Ayrıca servo motorların dönüş kabiliyeti sınırlandırılarak  $90^\circ$ ,  $180^\circ$ ,  $270^\circ$  ve  $360^\circ$  dönebilecek şekilde üretilmektedir.



Görsel 3.50: Servo motoru belirli açılarda döndüren PWM sinyali

### 3.10.1. Açık Vererek Servo Motor Kontrolü

Servo motorlarda Görsel 3.51'de görüldüğü gibi 3 adet giriş pini bulunur. Bu pinlerden kırmızı renkli kablo ucunun bağlı olduğu pin, servo motorun (+) besleme ucu olup mikrodenetleyici kartın 5V pinine bağlanır. Servo motorun kahverengi kablo ucunun bağlı olduğu pin, servo motorun (-) besleme ucu olup mikrodenetleyici kartın GND pinine bağlanır. Servo motorun turuncu kablo ucunun bağlı olduğu pin ise PWM sinyal ucu olup, mikrodenetleyici kartın giriş çıkış portlarından birine bağlanır. Görsel 3.51'deki devrede servo motorun PWM sinyal ucu mikrodenetleyici kartın D2 portuna bağlanmıştır. Servo motorları çalıştıracak PWM sinyalleri `Servo.h` kütüphanesi aracılığıyla üretilir. `Servo servo1;` komutuyla servo nesnesi tanımlanır. `void setup()` fonksiyonu içine yazılan `servo1.attach(2);` komutuyla servo motorun sinyal ucunun bağlı olduğu port numarası belirlenir. `void loop()` fonksiyonu içine yazılan `servo1.write(0);` komutuyla servo motorun belirtilen açıya dönmesi sağlanır.



Görsel 3.51: Servo motorların mikrodenetleyici karta bağlanması

Açık vererek servo motor kontrolünü sağlayan örnek program kodları aşağıda görülmektedir.

```
#include <Servo.h>

Servo servo1; // servo1 adında servo nesnesi oluşturulur.

void setup() {
  servo1.attach(2); // Servo motoru 2. porta bağlar.
}

void loop() {

  servo1.write(0); // Servo motoru 0 derece döndür.
  delay(1000); // 1 saniye bekle.
  servo1.write(30); // Servo motoru 30 derece döndür.
  delay(1000); // 1 saniye bekle.
  servo1.write(60); // Servo motoru 60 derece döndür.
  delay(1000); // 1 saniye bekle.
  servo1.write(90); // Servo motoru 90 derece döndür.
```

```
delay(1000); // 1 saniye bekle.
servo1.write(150); // Servo motoru 150 derece döndür.
delay(1000); // 1 saniye bekle.
servo1.write(180); // Servo motoru 180 derece döndür.
delay(1000); // 1 saniye bekle.
}
```



#### Sıra Sizde

Görsel 3.51'deki devreyi breadboard üzerine kurunuz. Açık vererek servo motor kontrolünü sağlayan örnek program kodlarını mikrodenetleyici IDE programında yazınız. Tüm işlemler bittikten sonra programı mikrodenetleyici karta yükleyiniz. Servo motor üzerindeki değişimleri gözlemleyiniz.



#### Değerlendirme

Çalışmalarınız öğrenme biriminin sonunda yer alan Kontrol Listesi kullanılarak değerlendirilecektir. Çalışmalarınızı yaparken değerlendirme ölçütlerini dikkate alınız.

Servo motoru 0° ile 180° arasında döngü kullanarak dönmesini sağlayan örnek program kodları, aşağıda görülmektedir.

```
#include <Servo.h>

Servo servo1; // servo1 adında servo nesnesi oluşturulur.

int pos = 0; // Servo motorun açısını tutan değişken

void setup() {
  servo1.attach(2); // Servo motoru 2. porta bağlar.
}

void loop() {
  for (pos = 0; pos <= 180; pos += 1) {
    // 0 dereceden 180 dereceye 1 derece aralıkla döndüren for döngüsü
    servo1.write(pos);
    delay(15);
  }
  for (pos = 180; pos >= 0; pos -= 1) {
    // 180 dereceden 0 dereceye 1 derece aralıkla döndüren for döngüsü
    servo1.write(pos);
    delay(15);
  }
}
```



#### Sıra Sizde

Görsel 3.51'deki devreyi breadboard üzerine kurunuz. Servo motoru 0° ile 180° arasında döngü kullanarak dönmesini sağlayan örnek program kodlarını mikrodenetleyici IDE programında yazınız. Tüm işlemler bittikten sonra programı mikrodenetleyici karta yükleyiniz. Servo motor üzerindeki değişimleri gözlemleyiniz.

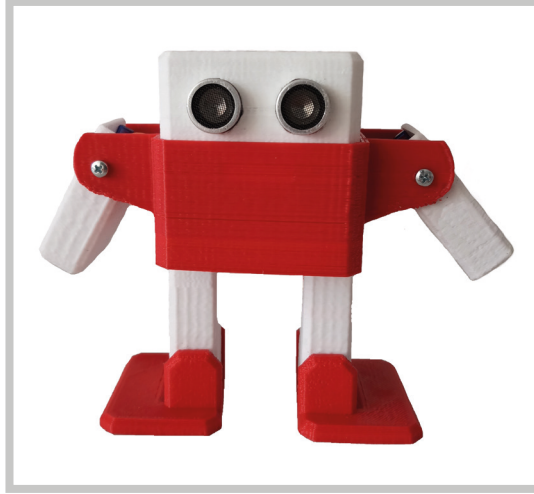


<http://kitap.eba.gov.tr/>  
KodSor.php?KOD=20809

#### Değerlendirme

Çalışmalarınız öğrenme biriminin sonunda yer alan Kontrol Listesi kullanılarak değerlendirilecektir. Çalışmalarınızı yaparken değerlendirme ölçütlerini dikkate alınız.

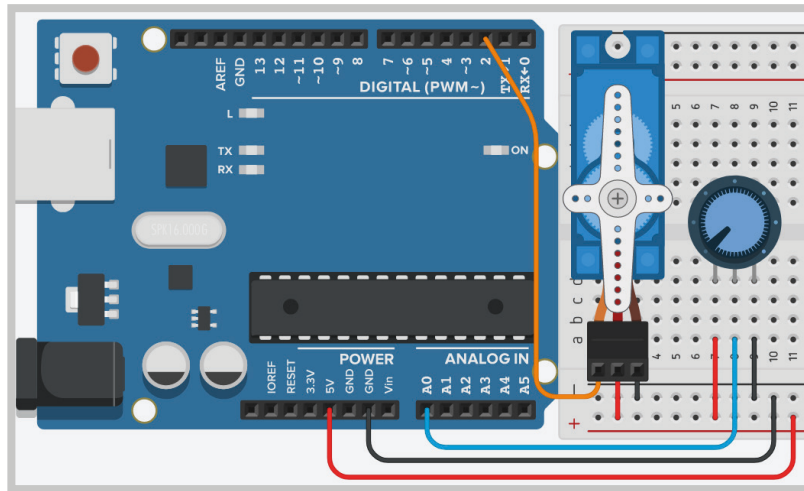
Servo motorun açı vererek kontrol edilmesiyle ilgili yürüme ve dans etme yeteneğine sahip Otto robot projesi bulunmaktadır. Görsel 3.52'de Otto robota ait 3 boyutlu yazıcıda yazdırılmış bir robot gövdesi görülmektedir. Otto robot projesine ait 3 boyutlu yazdırma parçaları ve program kodları, internet arama motorlarında araştırılarak ilgili sitelerin adreslerinden indirilebilir.



Görsel 3.52: Servo motorlarla yapılmış yürüyen ve dans eden robot projesi

#### 3.10.2. Potansiyometre ile Servo Motor Kontrolü

Servo motorlar açı verilerek kontrol edildiği gibi bir potansiyometre aracılığıyla da kontrol edilebilmektedir. Görsel 3.53'te servo motorun potansiyometre ile kontrol edilmesiyle ilgili devre görülmektedir. Devrede potansiyometrenin sağ ve sol uçları mikrodenetleyici kartın 5V ve GND pinlerine bağlanmaktadır. Potansiyometrenin orta ucu ise mikrodenetleyici kartın A0 analog girişine bağlanır. Devredeki servo motor bağlantısı Görsel 3.53'deki gibidir.



Görsel 3.53: Servo motorun potansiyometre ile kontrol edilmesi



Servo motorları çalıştıracak PWM sinyalleri `Servo.h` kütüphanesi aracılığıyla üretilir. `Servo servo1;` komutuyla `servo1` isminde bir servo nesnesi tanımlanır. `void setup()` fonksiyonu içine yazılan `servo1.attach(2);` komutuyla servo motorun sinyal ucunun bağlı olduğu port numarası belirlenir. `void loop()` fonksiyonu içine yazılan `pot1 = analogRead(A0);` komutuyla A0 analog giriş portundan okunan analog değer “pot1” değişkenine aktarılır. `aci1 = map(pot1,0,1023,0,180);` komutuyla “pot1” değişkeni içindeki 0 ile 1023 arasındaki veri 0 ile 180 arasında yeniden örneklenerek “aci1” değişkenine aktarılır. `servo1.write(aci1);` komutuyla “aci1” değişkeni içindeki değerle servo motorun belirli açıda dönmesi sağlanır.

Servo motorun potansiyometre ile kontrol edilmesine ait program kodları aşağıda görülmektedir.

```
#include <Servo.h>
Servo servo1; // Servo motoru kontrol etmek için servo1 nesnesi oluştur.

void setup() {
    servo1.attach(2); // Servo motoru 2. porta bağlar.
}

void loop() {
    int pot1 = analogRead(A0); // potansiyometre değerini okur.

    int aci1 = map(pot1, 0, 1023, 0, 180);
    // pot değerini 0 ile 180 arasında yeniden örnekler.

    servo1.write(aci1); // Açı değerini Servo motora gönderir.
    delay(15); // Servo motorların istenilen açığa gelmesi için bekler.
}
```



#### Sıra Sizde

Görsel 3.53'teki devreyi breadboard üzerine kurunuz. Servo motorun potansiyometre ile kontrol edilmesine ait program kodlarını mikrodenetleyici IDE programında yazınız. Tüm işlemler bittikten sonra programı mikrodenetleyici karta yükleyiniz. Potansiyometreyi çevirerek servo motor üzerindeki değişimleri gözlemleyiniz.



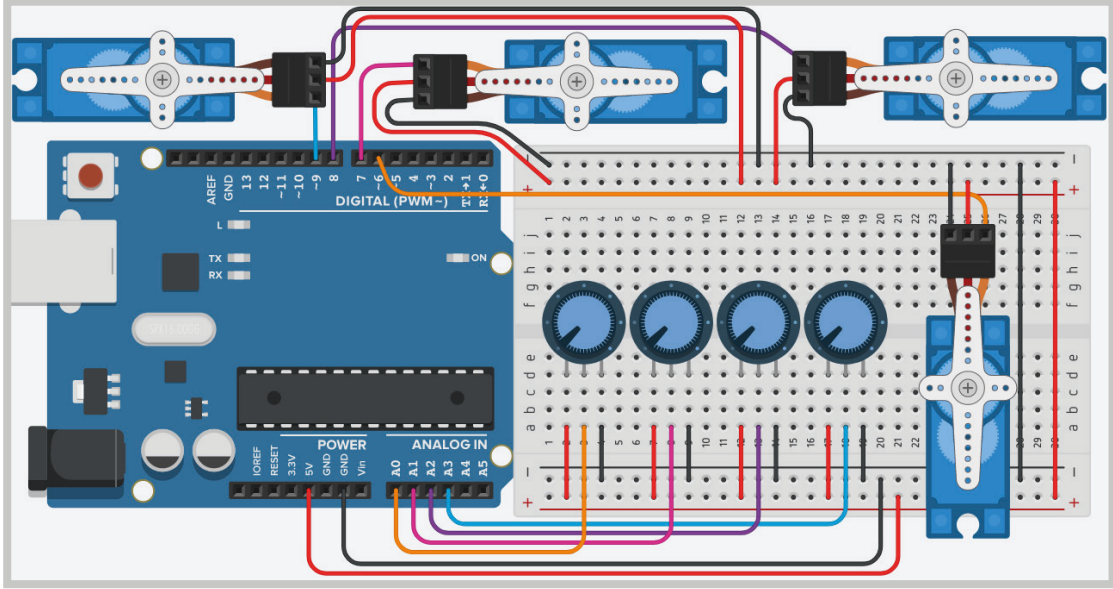
#### Değerlendirme

Çalışmalarınız öğrenme biriminin sonunda yer alan Kontrol Listesi kullanılarak değerlendirilecektir. Çalışmalarınızı yaparken değerlendirme ölçütlerini dikkate alınız.

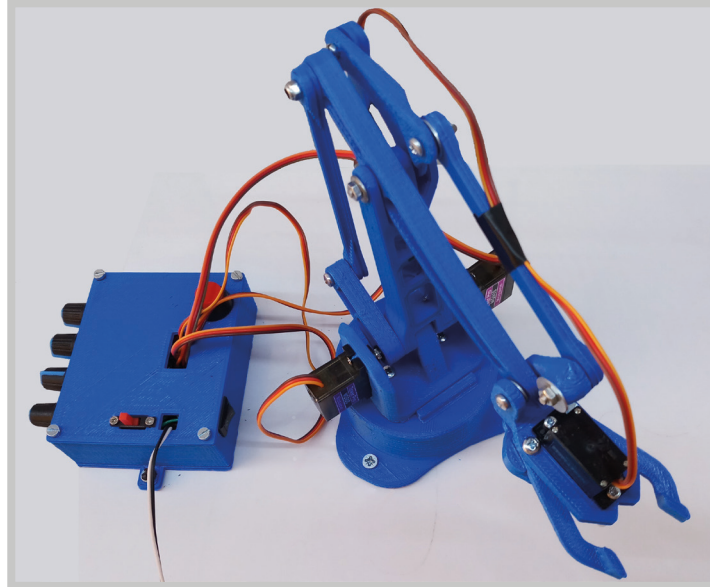
### 3. Öğrenme Birimi: Robot Tabanlı Proje Geliştirme

Servo motorun potansiyometre ile kontrol edilmesiyle ilgili robot kolu projeleri bulunmaktadır. Görsel 3.55'te 4 adet servo motor kullanılarak gerçekleştirilmiş bir robot kolu görülmektedir. Robot kolu projesine ait 3 boyutlu yazdırma parçaları internet arama motorlarında araştırılarak, ilgili sitelerin adreslerinden indirilebilir. Robot kolunun devresi Görsel 3.54'te verilmiştir.

Dört adet servo motor kullanılarak gerçekleştirilmiş robot kolu projesinin program kodları aşağıda görülmektedir.



Görsel 3.54: Servo motorun 4 potansiyometre ile kontrol edilmesi



Görsel 3.55: Servo motorlarla yapılmış robot kolu projesi

```
#include <Servo.h>
Servo servo1; // Servo motoru kontrol etmek için servo1 nesnesi oluştur.
Servo servo2; // Servo motoru kontrol etmek için servo2 nesnesi oluştur.
Servo servo3; // Servo motoru kontrol etmek için servo3 nesnesi oluştur.
Servo servo4; // Servo motoru kontrol etmek için servo4 nesnesi oluştur.

void setup() {
  servo1.attach(6); // Servo motoru 6. porta bağlar.
  servo2.attach(7); // Servo motoru 7. porta bağlar.
  servo3.attach(8); // Servo motoru 8. porta bağlar.
  servo4.attach(9); // Servo motoru 9. porta bağlar.
}

void loop() {
  int pot1 = analogRead(A0); // 1. potansiyometre değerini okur.
  int pot2 = analogRead(A1); // 2. potansiyometre değerini okur.
  int pot3 = analogRead(A2); // 3. potansiyometre değerini okur.
  int pot4 = analogRead(A3); // 4. potansiyometre değerini okur.

  int aci1 = map(pot1, 0, 1023, 0, 180);
  // 1. pot değerini 0 ile 180 arasında yeniden örnekler
  int aci2 = map(pot2, 0, 1023, 0, 180);
  // 1. pot değerini 0 ile 180 arasında yeniden örnekler
  int aci3 = map(pot3, 0, 1023, 0, 180);
  // 1. pot değerini 0 ile 180 arasında yeniden örnekler
  int aci4 = map(pot4, 0, 1023, 0, 180);
  // 1. pot değerini 0 ile 180 arasında yeniden örnekler

  servo1.write(aci1); // 1. Açık değerini 1. Servo motora gönderir.
  servo2.write(aci2); // 2. Açık değerini 2. Servo motora gönderir.
  servo3.write(aci3); // 3. Açık değerini 3. Servo motora gönderir.
  servo4.write(aci4); // 4. Açık değerini 4. Servo motora gönderir.
  delay(15); // Servo motorlarının istenilen açığa gelmesi için bekler.
}
```



#### Sıra Sizde

Görsel 3.54'teki devreyi breadboard üzerine kurunuz. Servo motorun potansiyometre ile kontrol edilmesine ait program kodlarını mikrodenetleyici IDE programında yazınız. Tüm işlemler bittikten sonra programı mikrodenetleyici karta yükleyiniz. Potansiyometreleri çevirerek servo motorlar üzerindeki değişimleri gözlemleyiniz.



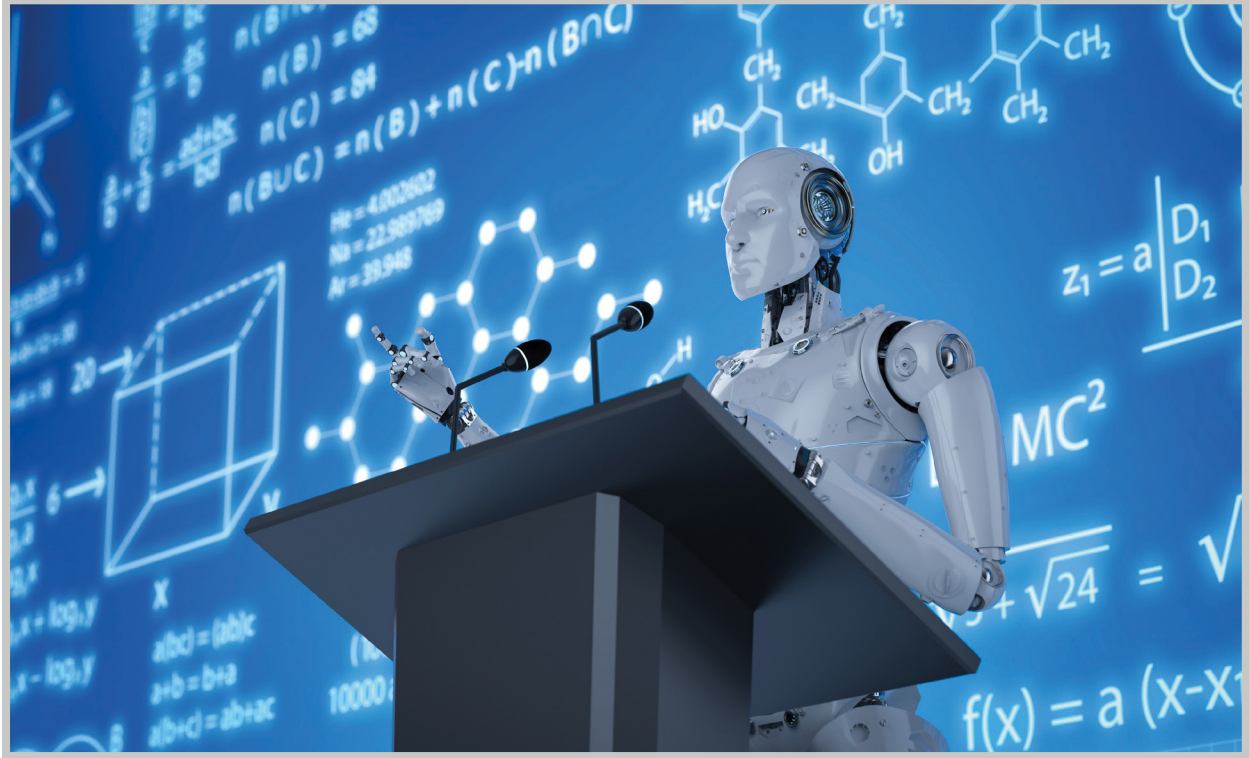
#### Değerlendirme

Çalışmalarınız öğrenme biriminin sonunda yer alan Kontrol Listesi kullanılarak değerlendirilecektir. Çalışmalarınızı yaparken değerlendirme ölçütlerini dikkate alınız.

## 1. Örnek Proje Çalışması

Herhangi bir toplumsal sorunu belirleyiniz. Bu sorunu ortadan kaldırmak için çalışma prensiplerini kendinizin belirlediği bir robotu hem mekanik hem kod olarak tasarlayınız.

### Robotik ve Yapay Zekânın Geleceği ve Etik

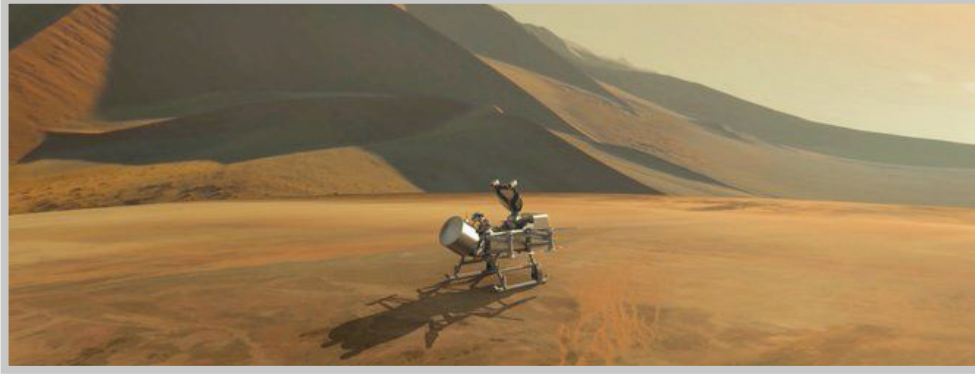


Günümüzde robotik ve yapay zekâ teknolojileri oldukça hızlı bir ilerleme göstererek, geçmişte tahmin edilemeyecek bir noktaya gelmiştir. Bu alanda yaşanan gelişmelerin ileride geleceği nokta, 2016 yılında Vincent C. Müller ve Nick Bostrom tarafından yapılan bir araştırmaya da konu olmuştur. Araştırma sonucuna göre; ilgili bilim çevreleri tarafından %50 ihtimalle, ileri teknoloji içeren (insana oldukça benzer yapıda teknolojilerin üretimi) makinelerin üretiminin 2040-2050 yıllarında olacağı belirtilirken, bu olasılık 2075 yılına gelindiğinde %90'a yükselecektir (Müller ve Bostrom, 2016).

Bu araştırmadan yaklaşık yüzyıl önce, bilim kurgu alanında da adı anılan ünlü yazarlardan biri olan Jules Gabriel Verne'nin kaleme aldığı (1828-1905) "Denizler Altında 20.000 Fersah", "Dünyanın Merkezine Yolculuk" gibi romanları, o yıllardan gelecekteki teknolojik gelişmelere ışık tutmuştur. Özellikle robotik ve yapay zekâ teknolojilerinde yaşanan hızlı değişim ve gelişmelerin olası etkileri de bilim çevreleri ve toplumlar tarafından sorgulanmaya başlanmıştır. Bu alanda çalışmalar yürüten bilim çevreleri, robotlar ve ilgili teknolojilerin ileri düzeyde oluşunun, etik açıdan dikkate alınması gereken sonuçlarına işaret etmiştir.

Robotik ve yapay zekâ teknolojileri insanlığı ileriye taşıyacak ve toplumsal alanın pek çok kısmında kolaylıklar sağlayacaktır. Ancak burada dikkat edilmesi gereken önemli noktalardan biri, toplumların gelişimine ışık tutan bu alanın, insani ve etik bir boyutunun da olduğu konusunun gözden kaçırılmaması gerektiğidir. Dolayısıyla her bilimsel çalışmada olduğu gibi, robot tasarımı ve geliştirilmesine ilişkin tüm teknolojilerin üretiminde insana, insanlığın ahlaki ve etik değerlerine zarar verecek hiçbir unsurun dâhil edilmemesi ve bu çalışmaların da etik ilkelere riayet edilerek yürütülmesi gerektiğidir.

## 2. Örnek Proje Çalışması



Ay'ın yüzeyinde belirli bir uzay misyonunu yerine getirmek için bir robot tasarlayacağınızı düşününüz. Tasarlayacağınız robot, Ay'ın yüzeyinde kısa bir tur atacaktır. Buna göre aşağıdaki sorular çerçevesinde araştırma yaparak, robotunuzu tasarlayınız.

- Ay'ın yüzeyi hakkında bir araştırma yapınız. Araştırma sonuçlarınızı not ediniz.
- Buna göre robotunuzun araştırma yapacağı bir yapay yüzey (Ay yüzeyi) oluşturunuz. Bu yüzeyde robotunuza hangi komutları (engelden kaçma, çizgi izleme, uzaktan kumanda ile yönlendirme vb.) vermeyi planlıyorsunuz? Buna göre bir plan oluşturunuz.
- Robotunuzun fiziksel donanımını hazırlayınız ve ilgili programları kullanarak mikrodenetleyici karta yükleme yapınız. Robotunuzu planladığınız misyon kapsamında hareket ettiriniz.

## 3. Örnek Proje Çalışması



Tarımda ekili tarlalardaki ürünler kuşlar, böcekler vb. tarafından zarar verilebilmektedir. Siz de ekili bir tarladaki ürünleri korumak amacıyla tarlada hareket eden ve belirli aralıklarla kuş sesleri çıkaran bir robot tasarlayınız. Aşağıdaki sorular çerçevesinde araştırma yaparak robotunuzu tasarlayınız.

- Tarım alanlarındaki ürünlere kuşların verebileceği zarardan korumak amacıyla neler yapılmaktadır, araştırınız.
- Robotunuzun hareket edeceği küçük bir alan düşününüz. Bu alanın topraklı ve engelli bir alan olabileceğini göz önünde bulundurunuz. Çeşitli malzemeler kullanarak yapay bir tarla alanı oluşturunuz.
- Tasarlayacağınız robotun, yapay tarla içinde hangi yönlere ve ne kadar uzağa hareket edeceğini hesaplayınız. Bu robotu ekinlere yaklaşan kuşları algılayıp dışarıya kuş sesi verecek bir şekilde tasarlayınız.
- Fiziksel donanımını hazırladığınız robotunuza, mikrodenetleyici IDE programında yazdığınız kodları, mikrodenetleyici geliştirme kartına yükleyip robotunuza belirlediğiniz komutları veriniz.

## ÖLÇME VE DEĞERLENDİRME

- Aşağıdaki sensörlerden hangisi engelden kaçan robot yapımında kullanılır?**
  - Mesafe sensörü
  - Sıcaklık sensörü
  - Renk sensörü
  - Işık sensörü
  - Bluetooth modülü
- Aşağıdakilerden hangisi bluetooth modülünün pinlerinden biridir?**
  - Echo
  - Trig
  - Rx
  - Out
  - Data
- Aşağıdaki mikrodenetleyici portlarından hangisinde hız kontrolü yapılmaz?**
  - D4
  - D5
  - D6
  - D9
  - D10
- Aşağıdakilerden hangisi motor sürücü entegrelerinden biridir?**
  - L7408
  - L7432
  - L7404
  - L298
  - L555
- Ultrasonik mesafe sensöründe süre bilgisini algılayan komut aşağıdakilerden hangisidir?**
  - Trig
  - Data
  - PulseIn
  - Echo
  - Mesafe
- Motorun döndürme kuvveti aşağıdakilerden hangisidir?**
  - Akım
  - Tork
  - Kinetik
  - Moment
  - Volt
- Çizgi izleme sensörlerinde IR LED'in gönderdiği ışığı algılayan aşağıdakilerden hangisidir?**
  - Foto LED
  - Foto Direnç
  - Foto Kondansatör
  - Foto Pil
  - Foto Transistör
- Uzaktan kumanda verilerini algılamada kullanılan kütüphane aşağıdakilerden hangisidir?**
  - IRcontrol
  - IRremote
  - RemoteControl
  - IRsence
  - IRrecv

9. Aşağıdakilerden hangisi DC motorlarda hız azaltma dişlilerinin genel adıdır?

- A) Endüktans
- B) Enkoder
- C) Stator
- D) Rotor
- E) Redüktör

10. Aşağıdakilerden hangisi DC motorda, motorun orta kısmındaki hareketli parçanın genel adıdır?

- A) Kömür
- B) Dişli grubu
- C) Rotor
- D) Fırça
- E) Stator

#### Meraklısına

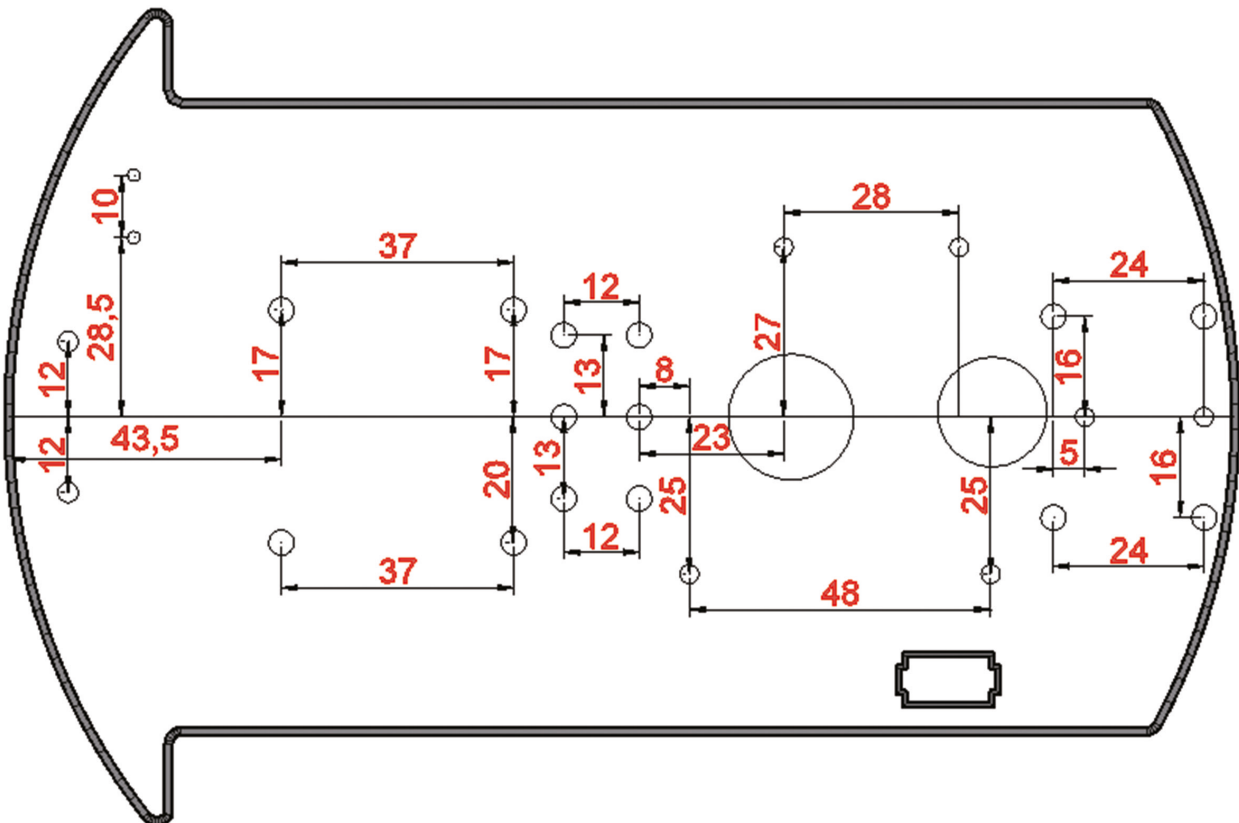
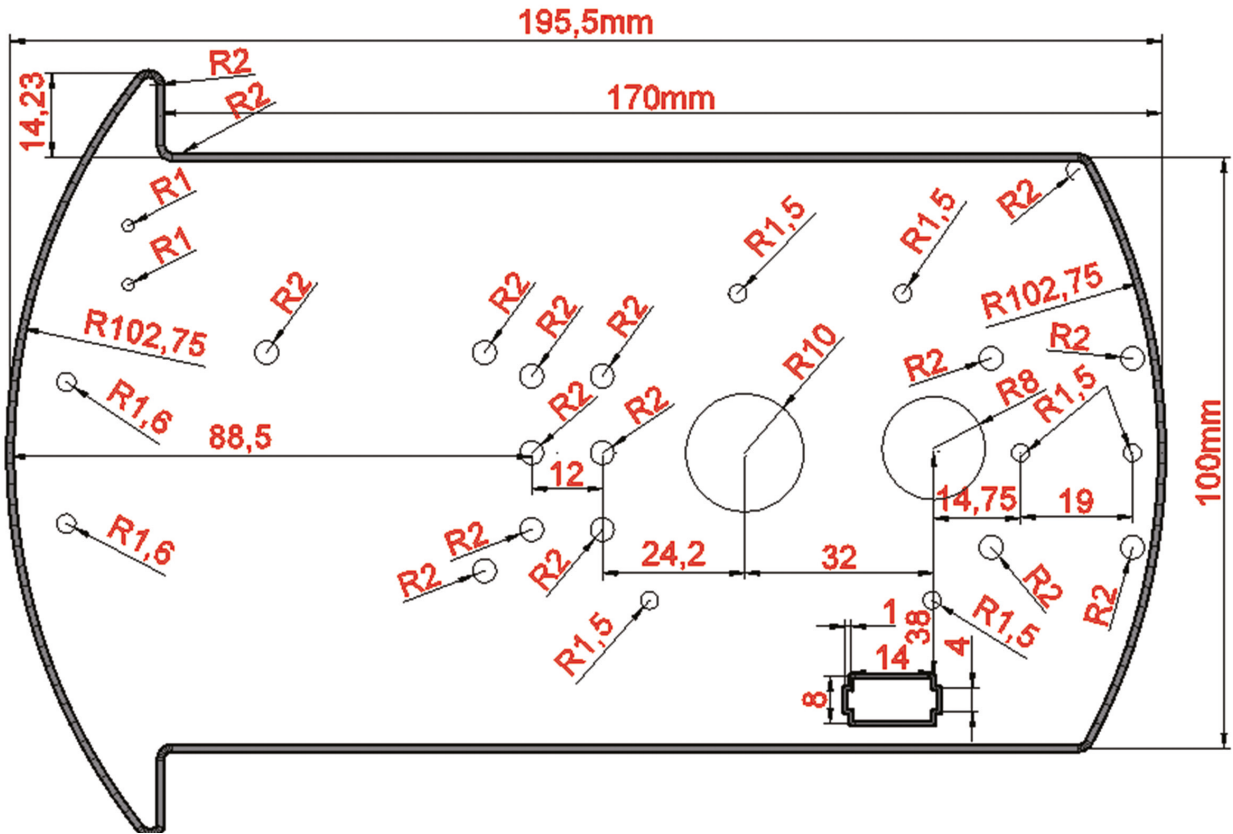
Her yıl Milli Eğitim Bakanlığı tarafından düzenlenen, Uluslararası Robot yarışmasını biliyor muydunuz? Mini sumo, temel ve ileri seviye çizgi izleme, hızlı çizgi izleyen, yumurta toplama, endüstriyel robotik kol gibi kategorilerde, her yıl farklı illerde düzenlenmektedir. Detaylı bilgilere [robot.meb.gov.tr](http://robot.meb.gov.tr) adresinden ulaşabilirsiniz.

## SIRA SİZDE KONTROL LİSTESİ

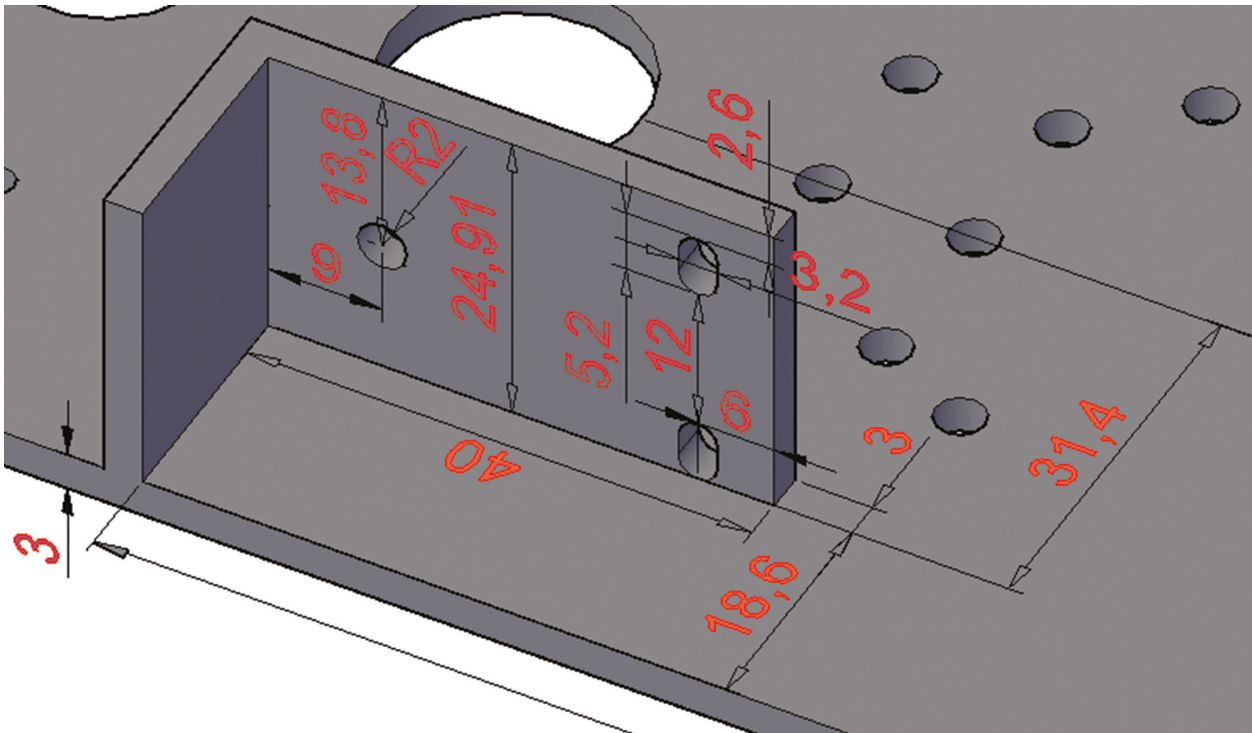
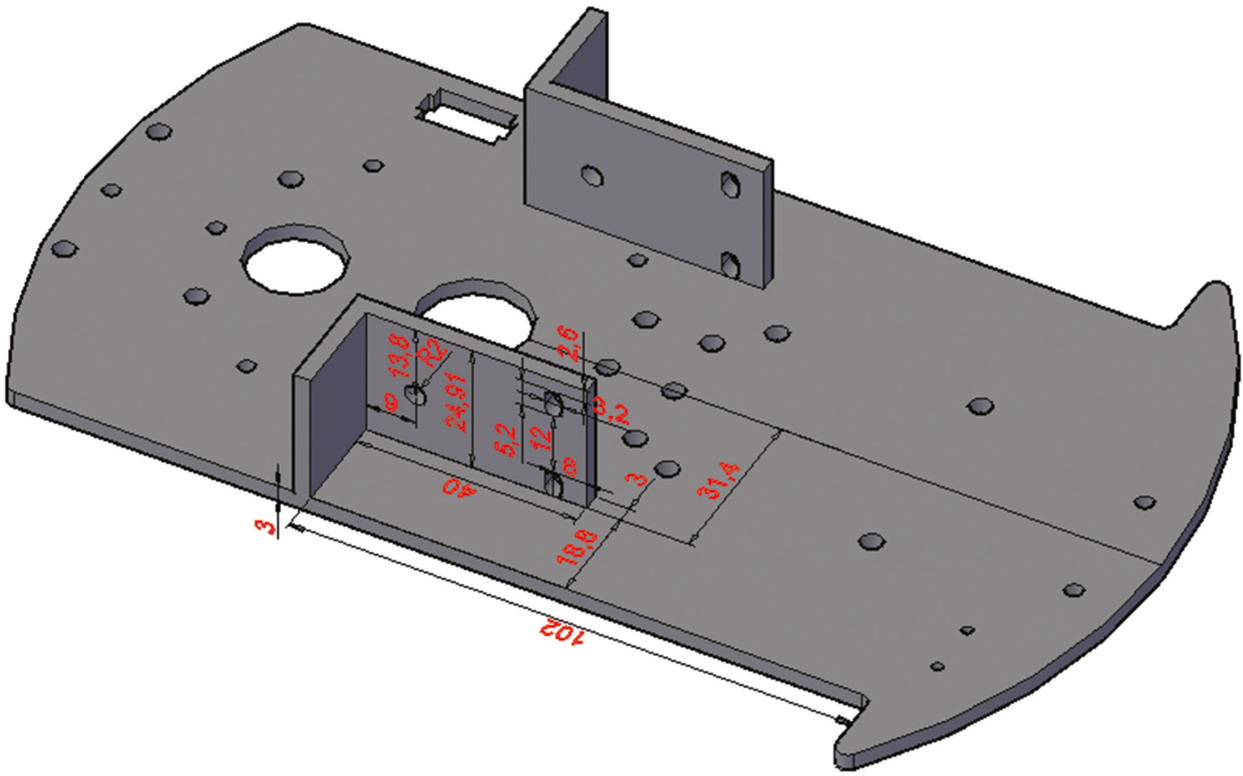
### Kontrol Listesi

Kriterler	Evet	Hayır
Uygun devre elemanları ve sensörlerin seçimini yapar.		
Devre elemanlarını breadboarda yerleştirir.		
Program kodlarını, simülatör programında uygun bir şekilde yazar.		
Simülasyonu başlatarak yazılan programın çalışmasını gözlemler.		
Devrede kullanılan sensörler ve diğer devre elemanlarının çalışmasını gözlemler.		
Çalışmada iş sağlığı ve güvenliği kurallarına dikkat eder.		
Zamanı verimli kullanır.		

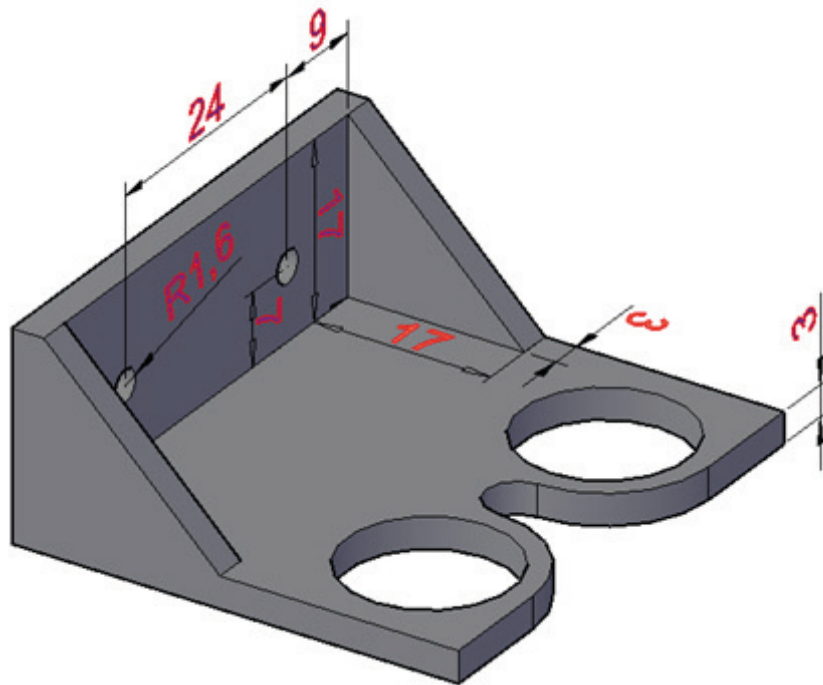
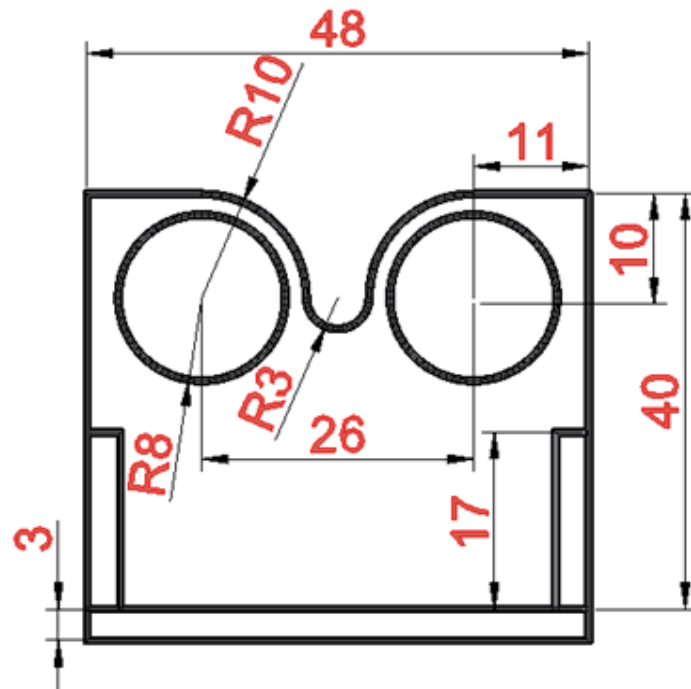
EK – 1: Robot Gövdesinin Ölçüleri



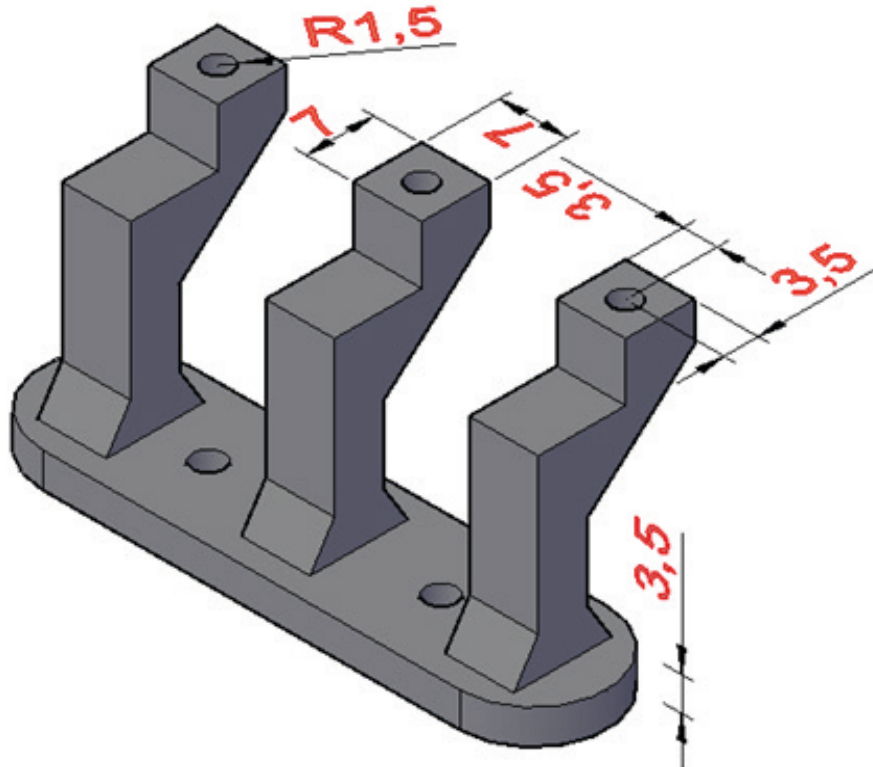
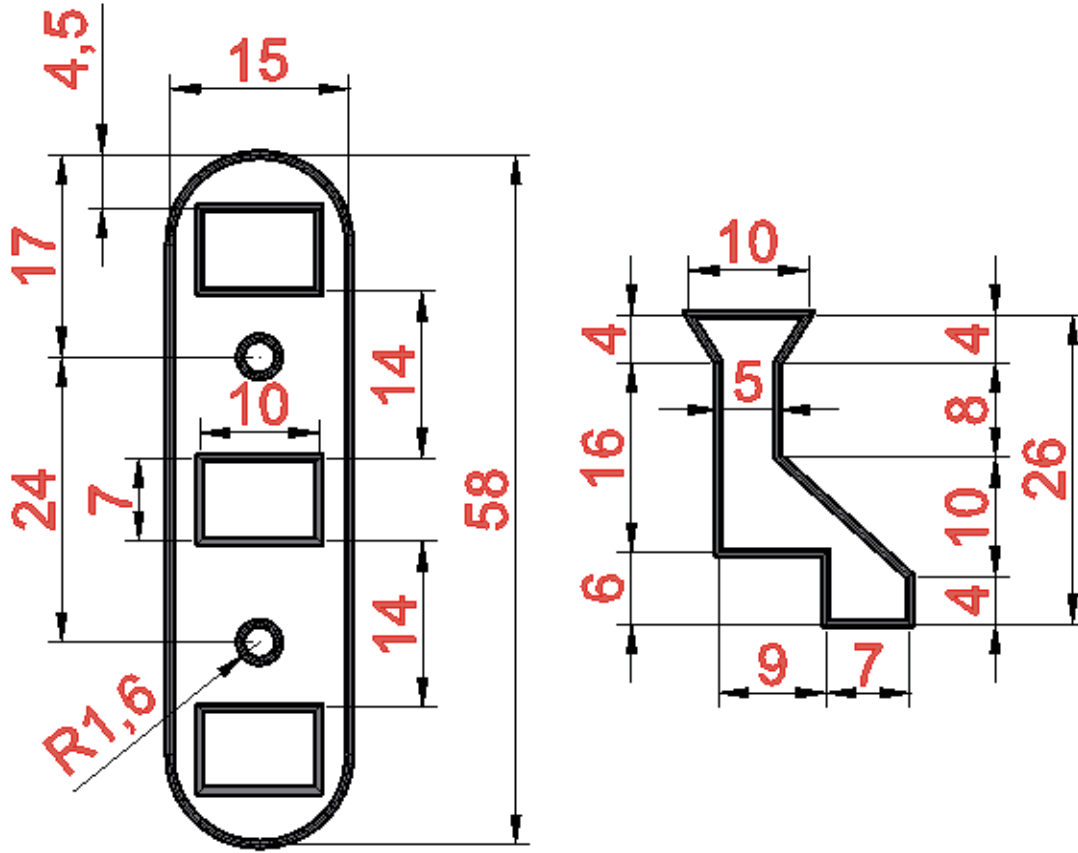




EK – 2: Mesafe Sensörü Aparatının Ölçüleri



## EK – 3: Çizgi İzleme Sensörü Aparatının Ölçüleri



## ÖLÇME VE DEĞERLENDİRME SORULARI CEVAP ANAHTARLARI

### 1. ÖĞRENME BİRİMİ

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
E	A	E	C	C	B	E	B	D	D	A	E	C	A	C	D

### 2. ÖĞRENME BİRİMİ

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
D	A	C	D	C	D	C	D	A	C	A	D	C	E	C	A	D	E	A	E

### 3. ÖĞRENME BİRİMİ

1	2	3	4	5	6	7	8	9	10
A	C	A	D	C	B	E	B	E	C

## GÖRSEL KAYNAKÇASI



Görsel Kaynakçasına ulaşmak için kodu tarayınız.

<http://kitap.eba.gov.tr/karekod/Kaynak.php?KOD=1539>









NOTLAR:

A series of horizontal dotted lines for taking notes.

NOTLAR:

A series of horizontal dotted lines for taking notes.