



# Robotbilim içindeki EdMaceralarınız<sup>1</sup>

## 10 Ders Planı

Etkinlik Kağıtları ve Çalışma Yaprakları



<sup>1</sup> Türkçe çevirisi ve uyarlaması STEM Makers Academy (www.stemmakers.academy) tarafından yapılmıştır.  
[www.meetedison.com](http://www.meetedison.com)

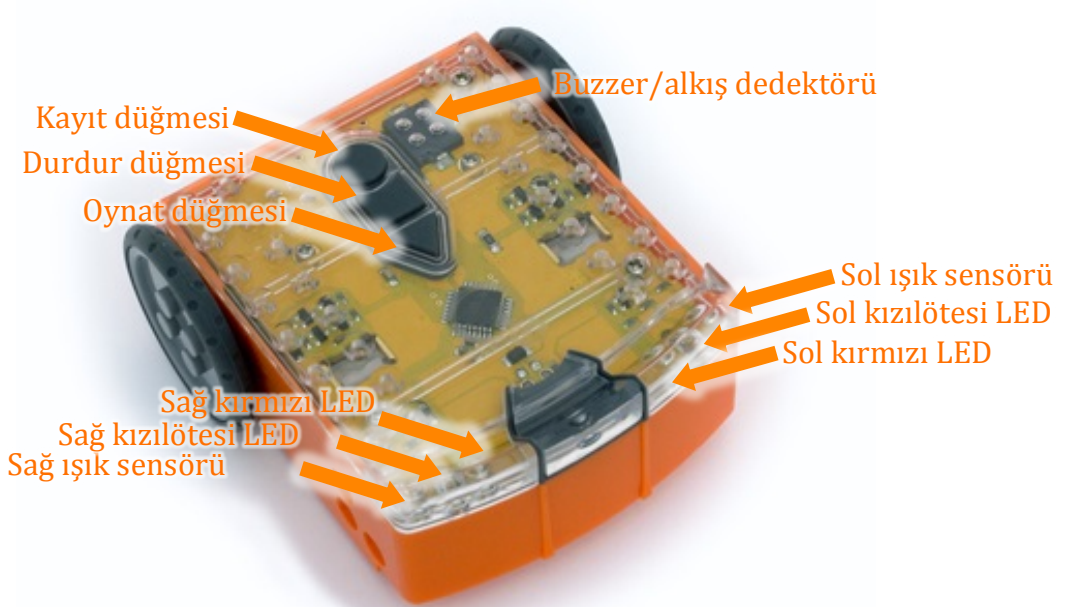


İsim: \_\_\_\_\_

## Ders 1 Çalışma Yaprağı 1.1 – Edison’u Tanıma

Edison LEGO ürünleriyle uyumlu küçük bir programlanabilir robottur.

Edison dünya ile etkileşim kurmak için sensörler ve motorlar kullanır. Aşağıdaki resimde Edison’un sensörleri, düğmelerini ve bağlantılarını görebilirsiniz.



Edison’un sensörleri ve düğmelerini tanıma

Oynat düğmesi – Programı başlat  
Durdur düğmesi – Programı durdurmak için bas  
Kayıt düğmesi– 1 basış = programı indir, 3 basış = barkodu oku



Edison’un güç düğmesi ve çizgi takip sensörü

Edison’un çizgi takip sensörü kırmızı bir LED lamba ve bir ışık sensöründen oluşur.

Çizgi takip sensörü daha önceden yüklenmiş özel barkodlarda okur.

EdComm kablo Edison’a karşıdan program yüklemek için kullanılır. Bilgisayarınıza veya tabletinize kulaklık soketiyle bağlanır.



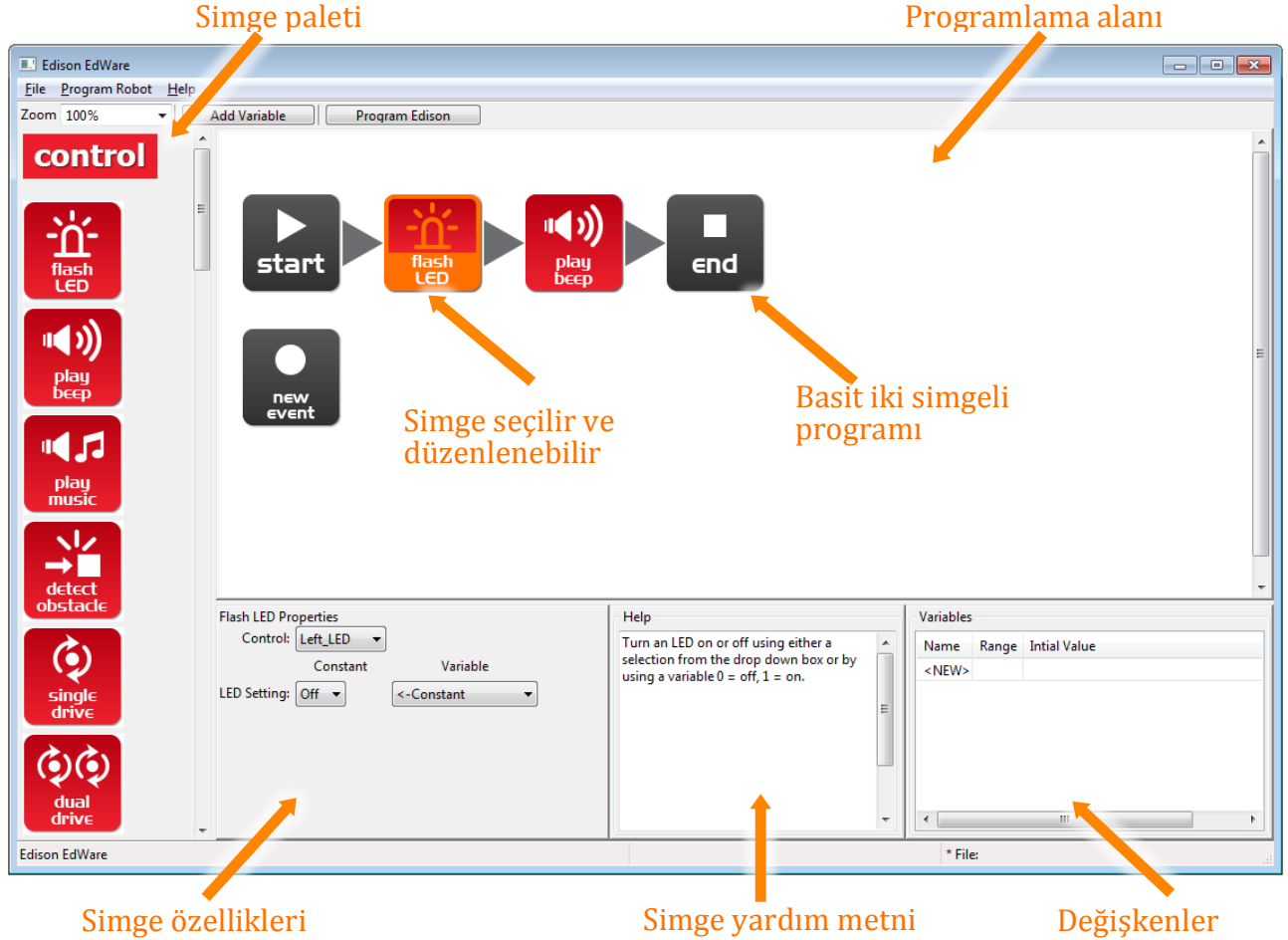
EdComm programlama kablosu



İsim: \_\_\_\_\_

## Ders 1 Çalışma yaprağı 1.3 – EdWare’la Tanışın

Bu EdWare ve Edison robotu programlamak için kullanılan yazılımdır.



Programlamaya başlamak için sol taraftaki paletten simgeler tutulur ve programlama alanının üzerine bırakılır. Simgeleri 'başla' ve 'bitir' simgeleri arasına yerleştiriniz.

Bir simge seçiniz ve simge özellikleri kutusunda Edison'un simgeye nasıl cevaplar vereceğini kontrol etmek için ayarlarını düzenleyiniz. Programlama yaparken bir rehber olarak yardım metnini kullanınız. Bir simge hakkında bilmeniz gereken her şey burada bulunabilir. Değişkenler bölümü Edison'un belleğinin küçük parçalarını oluşturabileceğiniz ve görebileceğiniz yerdir.

**Dört büyük simge gurubunun isimleri nelerdir?**

---

---

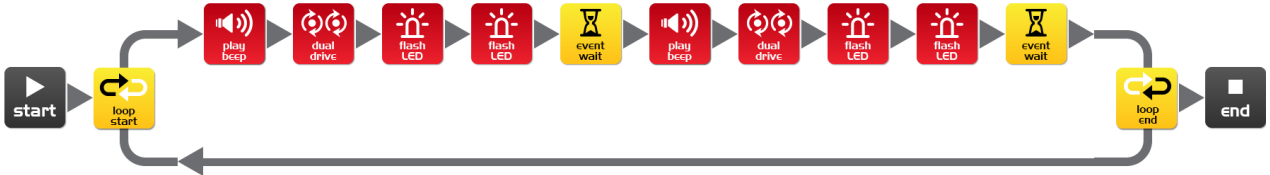
---

---

İsim: \_\_\_\_\_

## Ders 1 Çalışma Yaprağı 1.4 – Bir test programı indirme

Test programınız açınız. Dosya adı: **Test Program.edw** (Windows location: C:\Program Files\EdWare\My Programs)



Test program

Öğrencilerin bir programın nasıl görüldüğünü açıklayın. Edison her seferinde bir simgeyi göz önüne alır ve simgenin söylediğini yapar. Oklar, Edison'un simgeleri okuduğu yönü gösterir. Bu programda bir döngü vardır, bu yüzden program sonsuza kadar veya piller bitene kadar devam eder.

Test programını indirmek için cihazınıza EdComm kablosunu kulaklık soketine bağlayınız ve **sesi sonuna kadar açınız**. Edison'a, EdComm kablonun diğer ucunu gösterildiği gibi takınız.



Test Programını İndirmek için Bu Adımları Takip Ediniz:

1. Edison'un kayıt düğmesine bir kere basınız
2. EdWare'de 'Edison Programla' butonuna ve daha sonra 'İndirmeyi Başlat'a basınız
3. Programı başlatmak için Edison'un oynat düğmesine basınız

**Oynat düğmesine bastığınızda robot ne yaptı?**

---

---

**Bu eylemleri programdaki simgelerle ilişkilendirebilir misiniz? Lütfen düşüncelerinizi açıklayınız.**

---

---

**Program bilgisayardan robota nasıl ulaştı?**

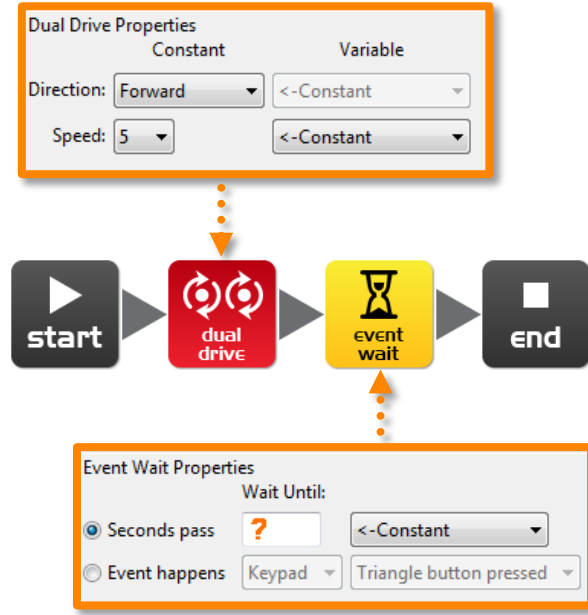
---

---

İsim: \_\_\_\_\_

## Ders 2 Çalışma yaprağı 2.1 – Robotu ileriye sürüş

Edison robotu ileri hareket ettirmek için aşağıdaki programı yazınız. Başlangıç ve bitiş noktası olarak masa üzerinde aktivite kağıdı veya renkli bir şerit kullanınız.



Robotun başlangıç çizgisinin arkasından başlamasını ve bitiş çizgisinden önce durmasını sağlamak için 'Geçen saniye' (değiştirilebilir?) Değerini saniye cinsinden ayarlayınız.

Minimum süre 0.01 saniye.

Maksimum süre 327.67 saniye.

Bitiş çizgisinden hemen önce robotunuzun ileri hareket edip durana kadar farklı miktarlarda süre deneyiniz.

Robotunuzun başlangıçtan bitişe gidene kadar geçen doğru zaman miktarı nedir?

\_\_\_\_\_

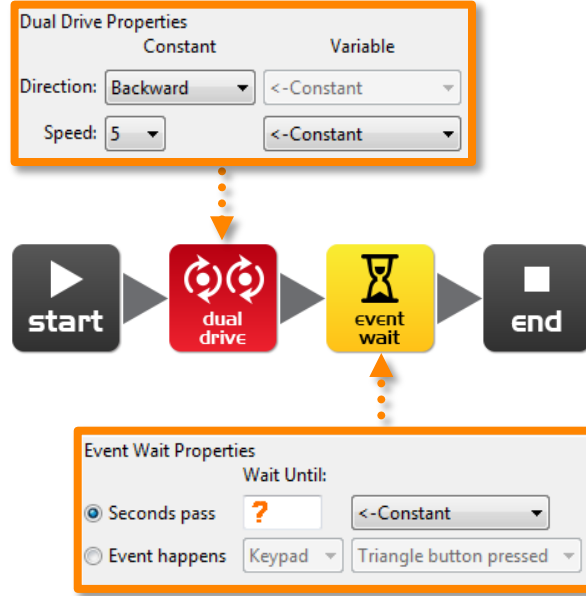
Robotun yaptığı işlemi ve nedenini anlatınız

\_\_\_\_\_  
\_\_\_\_\_  
\_\_\_\_\_

İsim: \_\_\_\_\_

## Ders 2 Çalışma yaprağı 2.2 – Robotu geriye sürüş

Edison robotu geri hareket ettirmek için aşağıdaki programı yazınız. Başlangıç ve bitiş noktası olarak masa üzerinde aktivite kağıdı veya renkli bir şerit kullanınız.



Robotun başlangıç çizgisinin arkasından başlamasını ve bitiş çizgisinden önce durmasını sağlamak için 'Geçen saniye' (değiştirilebilir?) Değerini saniye cinsinden ayarlayın.

Minimum süre 0.01 saniye.

Maksimum süre 327.67 saniye.

Bitiş çizgisinden hemen önce robotunuzun ileri hareket edip durana kadar farklı miktarlarda süre deneyiniz

Robotunuzun başlangıçtan bitişe gidene kadar geçen doğru zaman miktarı nedir?

\_\_\_\_\_

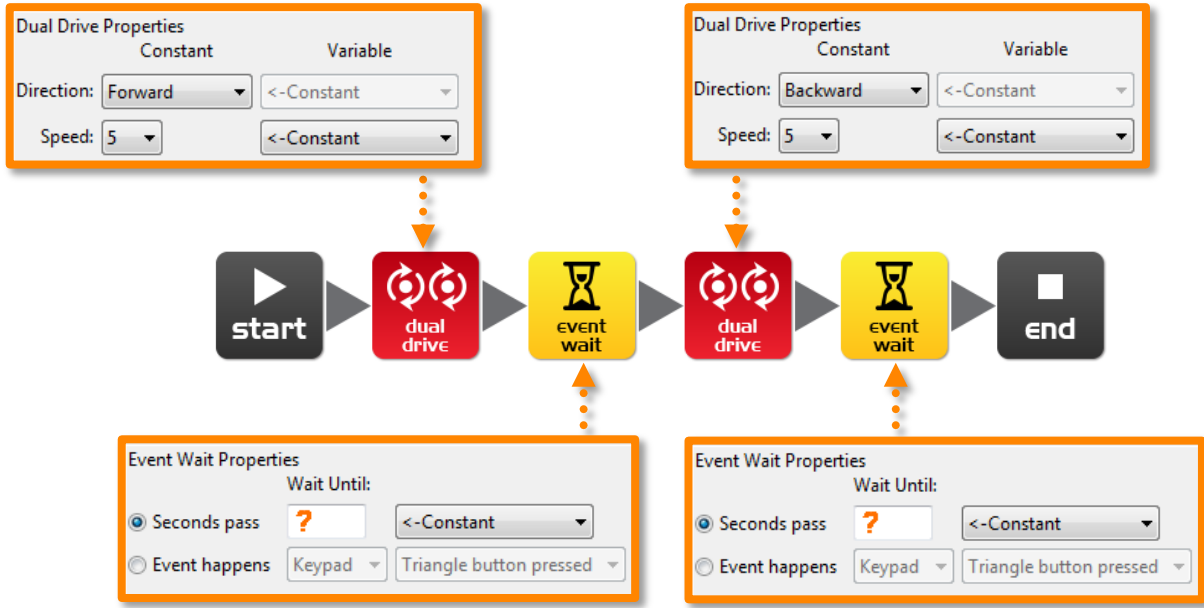
Robotun yaptığı işlevi ve nedenini anlatınız

\_\_\_\_\_  
\_\_\_\_\_  
\_\_\_\_\_

Name: \_\_\_\_\_

## Ders 2 Çalışma yaprağı 2.3 – İleriye ve daha sonra geriye

Edison robotu ileri ve daha sonra geri hareket ettirmek için aşağıdaki programı yazınız. Başlangıç ve bitiş noktası olarak masa üzerinde aktivite kağıdı veya renkli bir şerit kullanınız.



Robotun başlangıç çizgisinin arkasından başlayıp bitiş çizgisinden önce durmasını, ardından geriye doğru sürmeyi ve başlangıç satırından önce durmasını sağlamak için 'Geçen saniye' (değiştirilebilir?) değerini saniye cinsinden ayarlayın.

Minimum süre 0.01 saniye.

Maksimum süre 327.67 saniye.

Bitiş çizgisinden hemen önce robotunuzun ileri hareket edip durana kadar farklı miktarlarda süre deneyiniz.

Robotunuzun başlangıçtan bitişe gidene kadar geçen doğru zaman miktarı ne kadardır?

İleri \_\_\_\_\_ Geri \_\_\_\_\_

Robotun yaptığı şeyi ve nedenini anlatınız

---

---

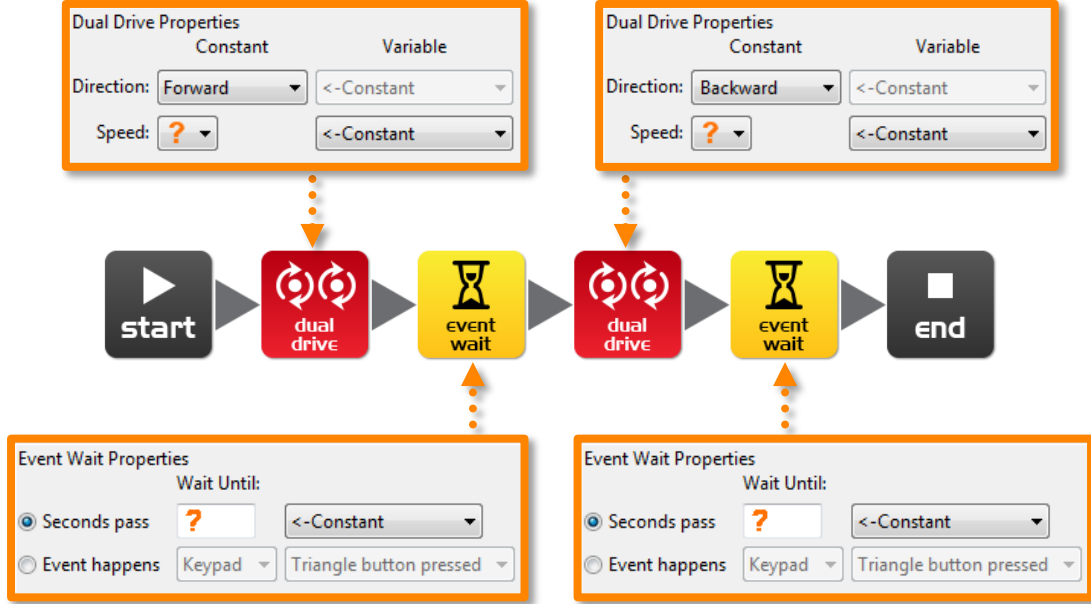
---



İsim: \_\_\_\_\_

## Ders 2 Çalışma yarağı 2.4 – Hız oyunu

Edison robotu ileri daha sonra geri hareket ettirmek için aşağıdaki programı yazınız. Başlangıç ve bitiş noktası olarak masa üzerinde aktivite kağıdı veya renkli bir şerit kullanınız.



Bu sefer süre ve hız değişir!

Programınıza bip sesi çal, flash LED veya hatta müzik çalmak gibi yeni simgeler ekleyebilirsiniz.

1. Başlama çizgisinin gerisinde başlayınız ve bitiş çizgisinden önce bitiriniz daha sonra geri doğru hareket ettiriniz ve mümkün olduğunca hızlı bir şekilde başlama çizgisinden önce durdurunuz.

**En hızlı ayarlarınız nedir?**

İleri sürüş hızı \_\_\_\_\_ İleri sürüş süresi \_\_\_\_\_

Geri sürüş hızı \_\_\_\_\_ Geri sürüş süresi \_\_\_\_\_

2. Başlama çizgisinin gerisinde başlayınız ve bitiş çizgisinden önce bitiriniz daha sonra geri doğru hareket ettiriniz ve mümkün olduğunca yavaş bir şekilde başlama çizgisinden önce durdurunuz

**En yavaş ayarlarınız nedir?**

İleri sürüş hızı \_\_\_\_\_ İleri sürüş süresi \_\_\_\_\_

Geri sürüş hızı \_\_\_\_\_ Geri sürüş süresi \_\_\_\_\_

İsim: \_\_\_\_\_

## Ders 2 Çalışma kağıdı 2.1



**BİTİŞ ÇİZGİSİ**

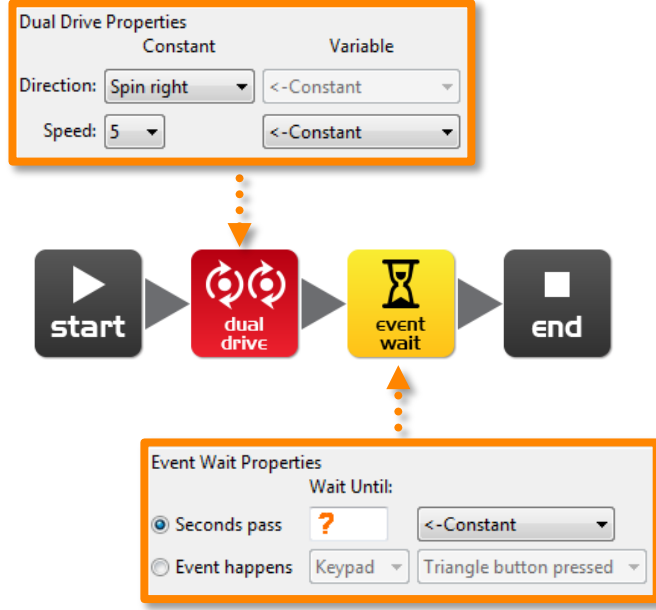


**BAŞLAMA ÇİZGİSİ**

İsim: \_\_\_\_\_

## Ders 3 Çalışma yaprağı 3.1 – 90 derece sağa dönüş

Edison robotu 90 derece sağa döndürmek için aşağıdaki programı yazınız. Başlangıç ve bitiş noktası olarak masa üzerinde aktivite kağıdı 3.1 veya renkli bir şerit kullanınız.



Robotun tam 90° sağa döndürülmesi için 'Geçen saniye' (değiştirilebilir?) Saniye cinsinden bir süreye ayarlayın.

Minimum süre 0.01 saniye.

Maksimum süre 327.67 saniye.

Başlangıç noktasından bitiş noktasına kadar robotunuz sağa dönene kadar farklı miktarlarda süre deneyiniz

Robotunuzun 90 derece dönmesi için geçen doğru zaman miktarı nedir?

\_\_\_\_\_

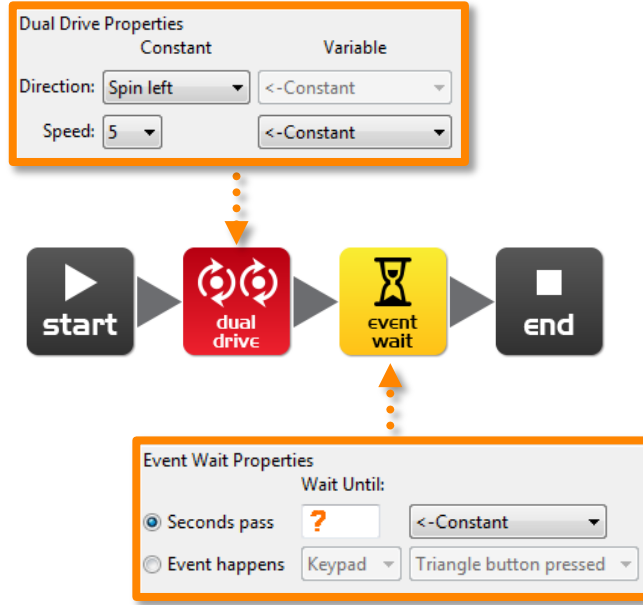
Robotun yaptığı şeyi ve nedenini anlatınız

\_\_\_\_\_  
\_\_\_\_\_  
\_\_\_\_\_

İsim: \_\_\_\_\_

## Ders 3 Çalışma yaprağı 3.2 – 180 derece sola dönüş

Edison robotu 180 derece sola döndürmek için aşağıdaki programı yazınız. Başlangıç ve bitiş noktası olarak masa üzerinde aktivite kağıdı 3.1 veya renkli bir şerit kullanınız



Robotun tam olarak 180° sola döndürülmesi için 'Geçen saniye' (değiştirilebilir?) Saniye cinsinden bir süreye ayarlayın.

Minimum süre 0.01 saniye.

Maksimum süre 327.67 saniye.

Başlangıç noktasından bitiş noktasına kadar robotunuz sola dönene kadar farklı miktarlarda süre deneyiniz

Robotunuzun 180 derece dönmesi için geçen doğru zaman miktarı nedir?

\_\_\_\_\_

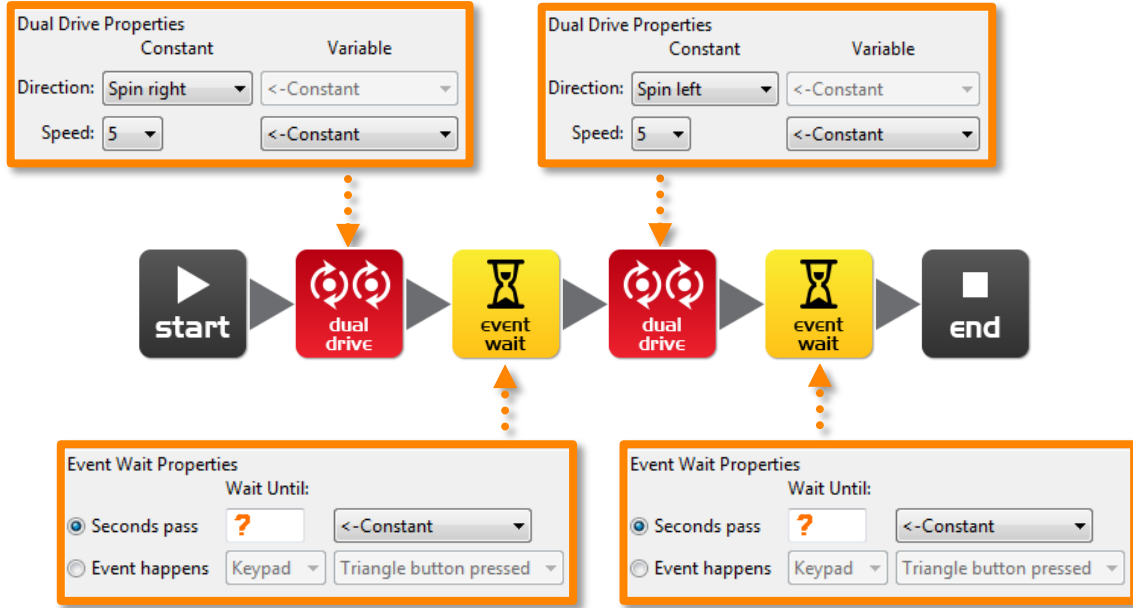
Robotun yaptığı şeyi ve nedenini anlatınız

\_\_\_\_\_  
\_\_\_\_\_  
\_\_\_\_\_

İsim: \_\_\_\_\_

## Ders 3 Çalışma yaprağı 3.3 – Sağa dönüş sonra sola dönüş

Edison robotu 90 derece sağa daha sonra 180 derece sola döndürmek için aşağıdaki programı yazınız. Başlangıç ve bitiş noktası olarak masa üzerinde aktivite kağıdı 3.1 veya renkli bir şerit kullanınız



Robotun tam olarak 90° sağa dönmesini sağlayın ve daha sonra 270° (90° + 180°) derece sola getirmek için 'Geçen saniye' (değiştirilebilir?) saniye cinsinden bir süreye ayarlayın.

Minimum süre 0.01 saniye.

Maksimum süre 327.67 saniye.

Başlangıç noktasından bitiş noktasına kadar robotunuz 90 derece sağa ve daha sonra bitiş noktasının altına 270 derece dönene kadar farklı miktarlarda süre deneyiniz.

Robotunuzun 90 derece daha sonra 270 derece dönmesi için geçen doğru zaman miktarı nedir?

90°: \_\_\_\_\_, 270°: \_\_\_\_\_

Robotun yaptığı şeyi ve nedenini anlatınız

---

---

---

İsim: \_\_\_\_\_

## Ders 3 Çalışma yaprağı 3.4 – Küçük labirent

Aktivite kağıdı 3.2'deki küçük labirent boyunca Edison robotunuzu sürmek için edindiğiniz bilgileri robotu programlamak için kullanınız.

Robot başlama çizgisinin gerisinden başlamalı ve bitiş çizgisinden sonra durmalıdır ve kazanmak için çizgiler üzerinde hareket etmemelidir.

Robotu ileri doğru hareket ettirmek ve gerekli dönüşleri yapmak için çoklu programlama simgelerini birleştirmeniz gerekecektir.

İpucu: İleri, sağa dön, ileri, sola dön, ileri

Robotun yaptığı şeyi ve nedenini anlatınız

---

---

---

Labirentte robotu hareket ettirmek için ne gibi zorluklarla karşılaştınız?

---

---

### İleri rekabet

Labirenti en hızlı kim bitirebilir?

Bir kronometre kullanmaya gerek yoktur. Sadece tüm bekleme durum simgelerine süre ekleyiniz.

Robot başlama çizgisinin gerisinden başlamalı ve bitiş çizgisinden sonra durmalıdır ve kazanmak için çizgiler üzerinde hareket etmemelidir.

Robotunuzun labirenti bitirme süresi nedir?

---

İsim: \_\_\_\_\_

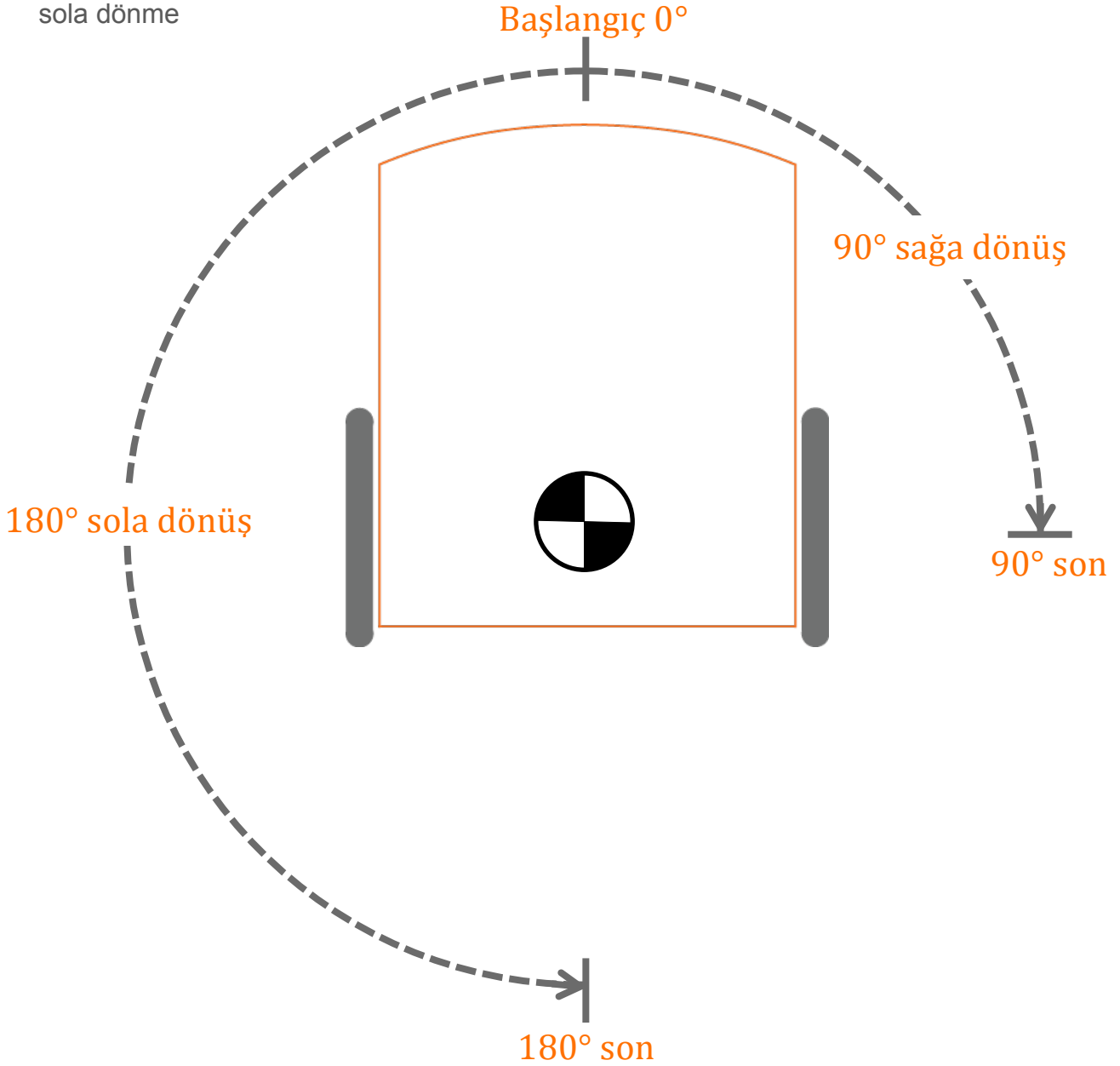
## Ders 3 Aktivite kağıdı 3.1 – Dönüş

Robotu şekilde gösterildiği gibi yerleştirin ve daha sonra dönüş programınızı oynatın. Robotunuz aşağıdakileri yapmalıdır.

Program 1 – 0° başlangıçtan 90° sona sağa dönme

Program 2 – 0° başlangıçtan 180° sona sola dönme

Program 3 – 0° başlangıçtan 90° sona sağa dönme daha sonra 180° sona 270° sola dönme



[www.meetedison.com](http://www.meetedison.com)





İsim: \_\_\_\_\_

## Ders 4 Çalışma yaprağı 4.1 – Görev

Kendiniz için bir robot sürüş görevi seçiniz ve Edison robotunuzu programlayınız.

Bazı örnekler aşağıdadır fakat kendi görevinizi belirleyebilirsiniz.

- Kupa veya kalemlik gibi bir nesnenin etrafında hareket ettiriniz
- Masanın köşesinden çıkmaksızın kenarlarında hareket ettiriniz
- Geniş bir kağıt parçası üzerine bir labirent oluşturunuz
- Bloklar kullanarak bir labirent oluşturunuz

Yaratıcı olmak için LED yak, bip sesi çal, müzik çalma gibi diğer programlama simgelerini kullanabileceğinizi unutmayınız.

Robotunuzun sürüş görevi nedir?

---

---

---

Bu programı yazarken güçlük çekilen noktalar nelerdi?

---

---

---

Programınızın içerdiği ek programlama simgeleri nelerdi ve bu simgeler ne yaptılar?

---

---

---

[www.meetedison.com](http://www.meetedison.com)

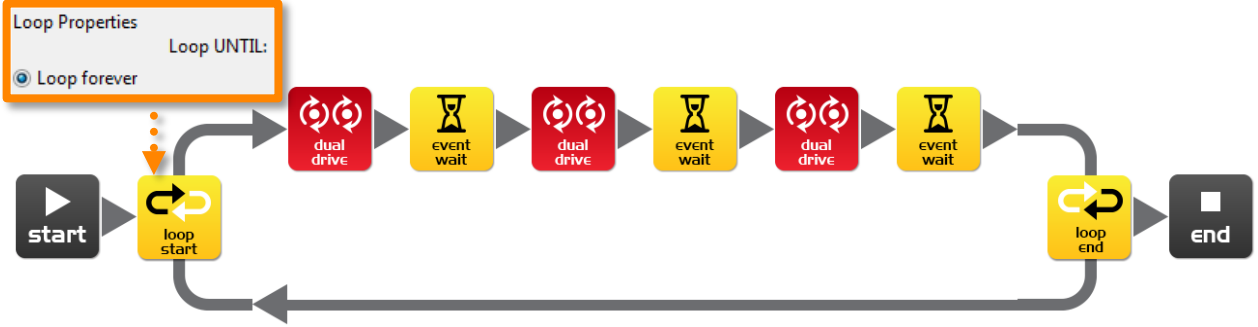
İsim: \_\_\_\_\_

## Ders 4 Çalışma yaprağı 4.2 – Meksika dalgası

Bu, herkesin robotunun aynı programı biraz farklı sürelerde yürüttüğü eğlenceli bir sınıf etkinliğidir. Sonuç bir Meksika dalgası veya koreografik bir dansa benzerdir.

### Döngüler

Hareketleri tekrar etmek için programınızda bir döngü kullanınız.



### Yapmanız gerekenler:

Robot hareketlerinin uygun bir dizisi ile kısa bir program yazınız. Programınızda ışık ve ses kullanmayı unutmayınız.

Hareket dizinizi yazıp kontrol ettiğinizde öğretmenin görüşünü alınız ve sınıfın en iyisini seçiniz.

Eğer sizinki seçilirse, sınıf ile program dosyanızı paylaşınız. Eğer başka birilerinininde programı seçilirse program dosyanızı açtığınızda başlangıca eylem bekle simgesi ekleyiniz. Öğretmenin söylediği süreyi ayarlayınız.

**Eylem bekle simgesi süresi:** \_\_\_\_\_ **Robot numarası:** \_\_\_\_\_ (robotları sıraya koymak için)

Başlangıçta en kısa, sonda en uzun gecikme süresine göre bir çizgi üzerine tüm robotları yerleştiriniz. Sonra herkes aynı anda oynat düğmesine basar.

Şimdi geri çekilin ve robot performanslarını izleyin.

**Robotunuzun hareketlerini betimleyiniz.**

---

---

---

İsim: \_\_\_\_\_

## Ders 5 Tasarım özeti çalışma yaprağı 5.1 – Programım

Gelecekte robotlar yardımcılarımız olacaktır. Uzun süredir robot elektrikli süpürelere sahibiz dolayısıyla bu gelecek çok uzakta değildir. Robotunuzun yapabileceği yararlı işleri düşünün ve bu görevi yapması için bir program yazınız.

Bazı örnekler aşağıda yer almaktadır;

- Müziğe uyarak dans etmek – Bir eğlence robotu
- Elektrikli süpürge – evde çok kullanışlı
- Güvenlik robotu– kalem kutunuzu güvende tutunuz

### 1. Tartışınız

Programlayabileceğiniz yararlı bir robot uygulamasına karar vermek için grup arkadaşınız veya diğer öğrenciler ile bunu tartışınız.

### 2. Açıklayınız

Programınızı yazmaya başlamadan önce robotunuzun çözeceği problemi ve bu problemi nasıl çözeceğini aşağıda açıklayınız.

Problem: \_\_\_\_\_

\_\_\_\_\_

Robotum bu problemi şunu yaparak çözecek: \_\_\_\_\_

\_\_\_\_\_

### 3. Programınızı yazınız

Simgeleri yerleştirmeye başlamadan önce programınızı planlayınız. Kullanmak için öğrendiğiniz simgeleri kullanınız fakat diğer simgeleri de deneyebilirsiniz.

### 4. Başarısızlık mı?

Herşey ilk seferinde düşündüğünüz şekilde çalışmaz. Hiç problem değil, tekrar deneyin fakat ilkinde yanlış giden şeyi belirleyiniz. **Programınız ilk seferinde neden çalışmadı?** Ünlü Thomas Edison'un ampulü keşfetmeden önce 10,000 kez başarısız olduğunu hatırlayınız. Bu yüzden **denemeye devam ediniz!**

\_\_\_\_\_

\_\_\_\_\_

\_\_\_\_\_

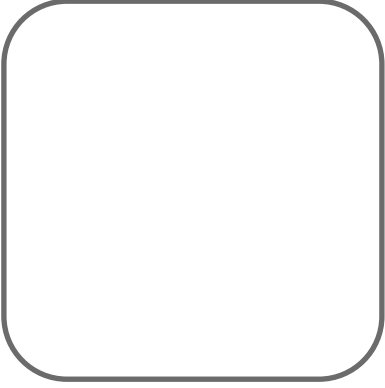
\_\_\_\_\_

İsim: \_\_\_\_\_

## Ders 5 Tasarım özeti çalışma yaprağı 5.2 - Programım

### 5. Programınızdaki programlama simgelerinden bazılarını açıklayınız

Aşağıya programlama simgelerini çizin ve renklendiriniz. Daha sonra onların programınızda ne yaptığını açıklayınız.



Bu simgenin adı nedir? \_\_\_\_\_

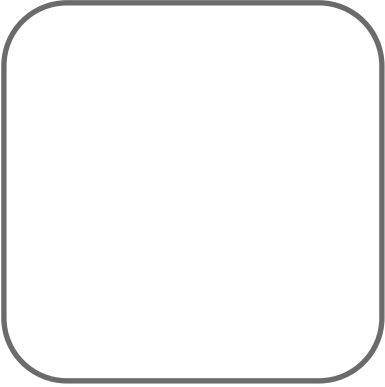
Bu simge ne yapar? \_\_\_\_\_

---

---

---

---



Bu simgenin adı nedir? \_\_\_\_\_

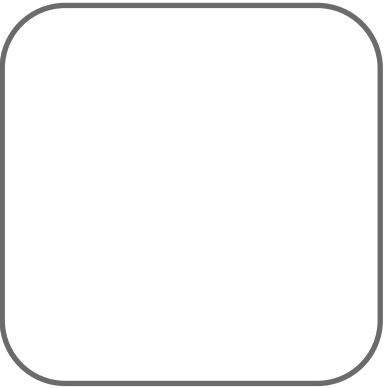
Bu simge ne yapar? \_\_\_\_\_

---

---

---

---



Bu simgenin adı nedir? \_\_\_\_\_

Bu simge ne yapar? \_\_\_\_\_

---

---

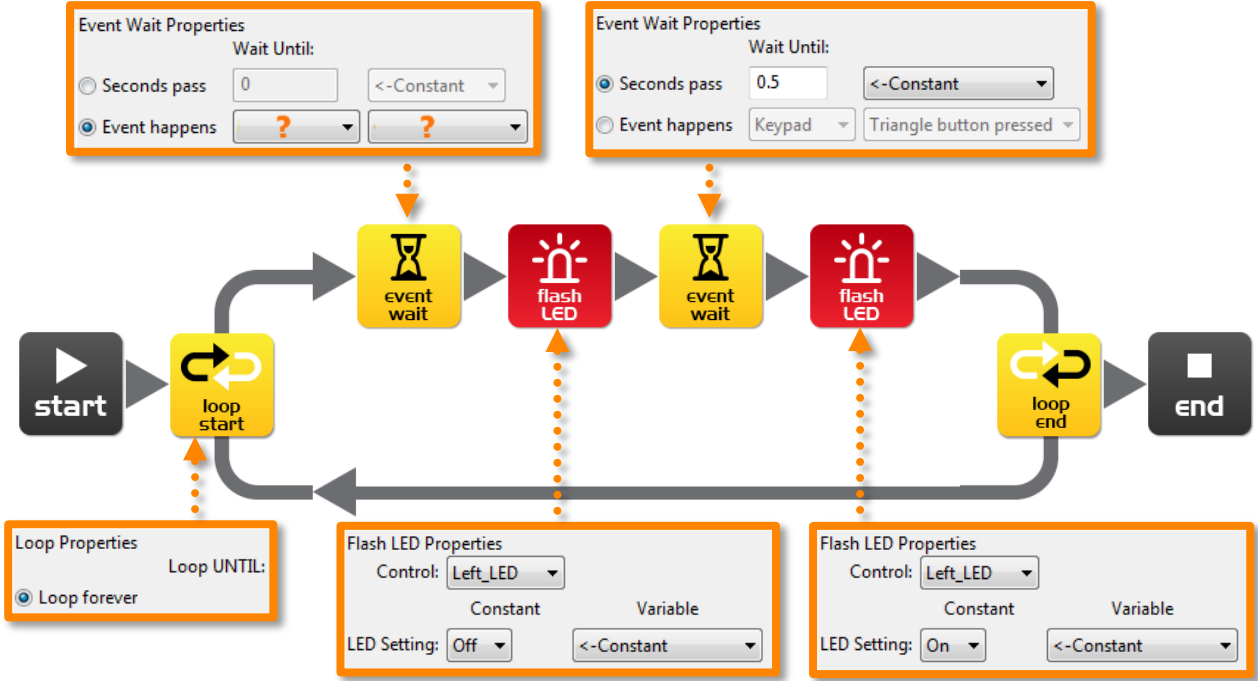
---

---

İsim: \_\_\_\_\_

## Ders 6 Çalışma yaprağı 6.1 – Alkışa yanıt olarak LED yakma

Alkış gibi yüksek bir sese yanıt olarak Edison robotun sol LED'ini yakıp söndürmek için aşağıdaki programı yazınız.



Bu programda ilk eylem bekle simgesi süre için beklemek amacıyla kullanılmıyor. Aslında program belirli bir eylem meydana gelene kadar bu noktadan devam etmeyecektir.

İlk eylem bekleme simgesinin özellikler kutusunda 'Eylemler' butonunu seçiniz ve daha sonra alkışa yanıt olarak gerekli olduğunu düşündüğünüz eylem tipini seçiniz.

Robot alkışı ne kadar mesafeden algılayabilir?

\_\_\_\_\_

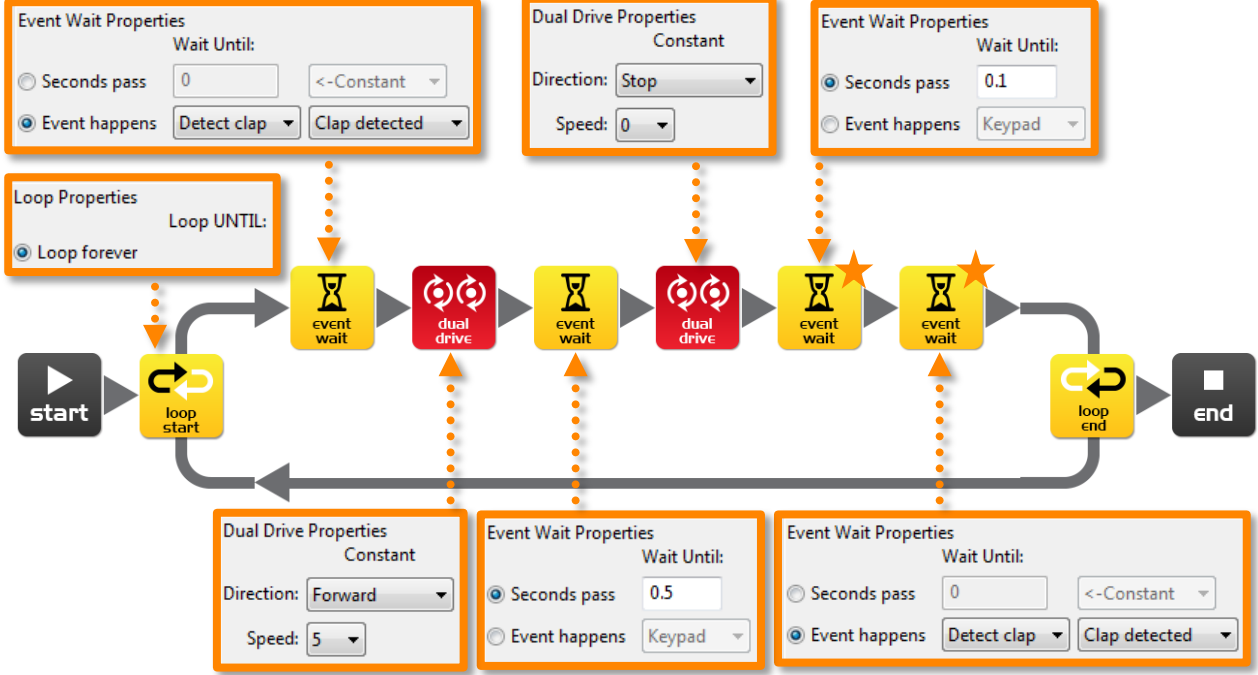
Bu programdaki döngü simgelerinin amacı nedir ve döngü simgeleri oralarda olmasaydı ne olurdu?

\_\_\_\_\_  
\_\_\_\_\_  
\_\_\_\_\_

İsim: \_\_\_\_\_

## Ders 6 Çalışma yaprağı 6.2 – Alkışa yanıt olarak sürüş

Alkışa yanıt olarak Edison robotu ileri hareket ettirmek için aşağıdaki programı yazınız.



Edison robotunun ses sensörü, sadece alkışlara karşı hassas değildir fakat herhangi bir yüksek sese de tepki verebilir. Bu döndüğü zamanda ses çıkartan robotun motorları, dişliler ve tekerleklerini içerir. Ses sensörünün robotun hareket etmesinden etkilenmesini önlemek için bu programda kullanılan iki ek simge vardır ve yukarıda turuncu yıldızlarla belirtilir. İlk eylem bekle simgesi 0.1 saniyeye ayarlıdır ve robotun motorunun durması için süre verir. İkinci eylem bekleme simgesi alkış beklemek için ayarlıdır ve devam etmek için programa izin verir çünkü motorların sesini tespit edecektir.

Eğer alkış sensörünü kullanıyorsanız motorları durdurduktan sonra bu iki simgeyi kullanmanız gerekecektir.

Farklı hareketler yapmak için ilk çift motorlu sürüş simgesinde farklı yönlerde ve eylem bekle simgesi ile deney yapınız.

**Başka hangi yön ve sürelerde denediniz?**

Deney 1: yön \_\_\_\_\_, süre \_\_\_\_\_

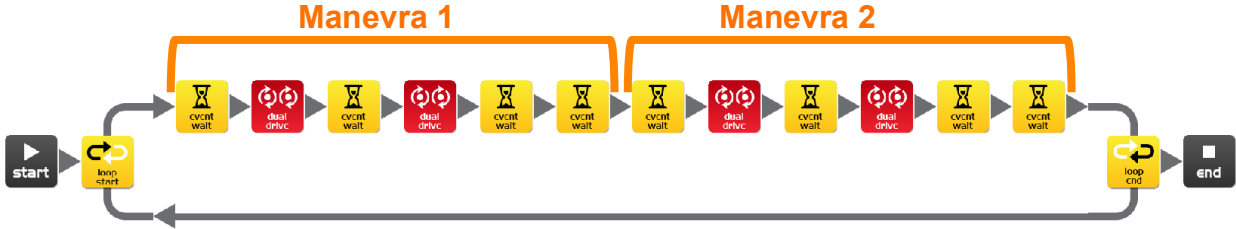
Deney 2: yön \_\_\_\_\_, süre \_\_\_\_\_

Deney 3: yön \_\_\_\_\_, süre \_\_\_\_\_

İsim: \_\_\_\_\_

## Ders 6 Çalışma yaprağı 6.3 – Alkışa yanıt olarak dans

Önceki iki uygulamada öğrendiğiniz şeyleri uygulayarak robotun alkışlamalara tepki vereceği bir dans rutini oluşturunuz.



En az iki dans manevrasına ihtiyacınız olacak fakat istediğiniz kadar ekleyebilirsiniz. Yukarıdaki program iki basit hareket manevrasına sahiptir. Bu iki manevrayı tekrarlar çünkü bir döngü içindedir.

Her bir alkışa iki manevra ekleyebilir ve deneyebilirsiniz.

### Alternatif görev

Eğer bir dans programı istemezseniz o zaman robotun alkışa yanıt olarak hareket ettiği ve daha sonra ikinci alkışa yanıt olarak geri döndüğü kısa bir engel dersi oluşturabilirsiniz.

Programınızda kaç tane manevra var?

\_\_\_\_\_

Robotunuzun dansını veya sürüş manevralarını açıklayınız.

\_\_\_\_\_  
\_\_\_\_\_  
\_\_\_\_\_

### Eğlenceli sınıf aktivitesi

Sınıfın en iyi dans rutinini seçiniz ve her öğrenci robotuna bunu robotuna programlasın. Daha sonra tüm robotları beraber yerleştiriniz ve robotların dansını görmek için herkes aynı anda alkışlasın.

İsim: \_\_\_\_\_

## Ders 7 Çalışma yaprağı 7.1 – Kızılötesi ile engel algılama

Edison robot kızılötesi bakış ile donatılmıştır. Kızılötesi ışık insan gözüyle görünmezdir bu yüzden bu ışığı göremezsiniz fakat bu Edison'a karanlıkta görmesine izin verir.

Edison'un engelleri algılaması için, kızılötesi ışık robotun sol ve sağ tarafından dışarı doğru yayılır. Eğer kızıl ötesi ışık duvar gibi bir engelden yansırsa o zaman yansıyan ışık Edison'un kızılötesi dedektörü tarafından tespit edilir. Kızılötesi dedektörü robotun ön kısmının ortasında yer alır.

Aşağıdaki gösterimde Edison'un solunda bir engel vardır bu yüzden sadece sol yayıcıdan kızılötesi ışık yansıtılır. Alınan sinyalden Edison solda bir engelin olduğunu fakat sağda olmadığını belirleyebilir.

Yayılan kızılötesi ışık kırmızı gösterilmiştir ve yansıyan kızıl ötesi ışık mavi renkte gösterilmiştir.



Bu engel için yayılan ve yansıyan kızılötesi ışıkları çiziniz.

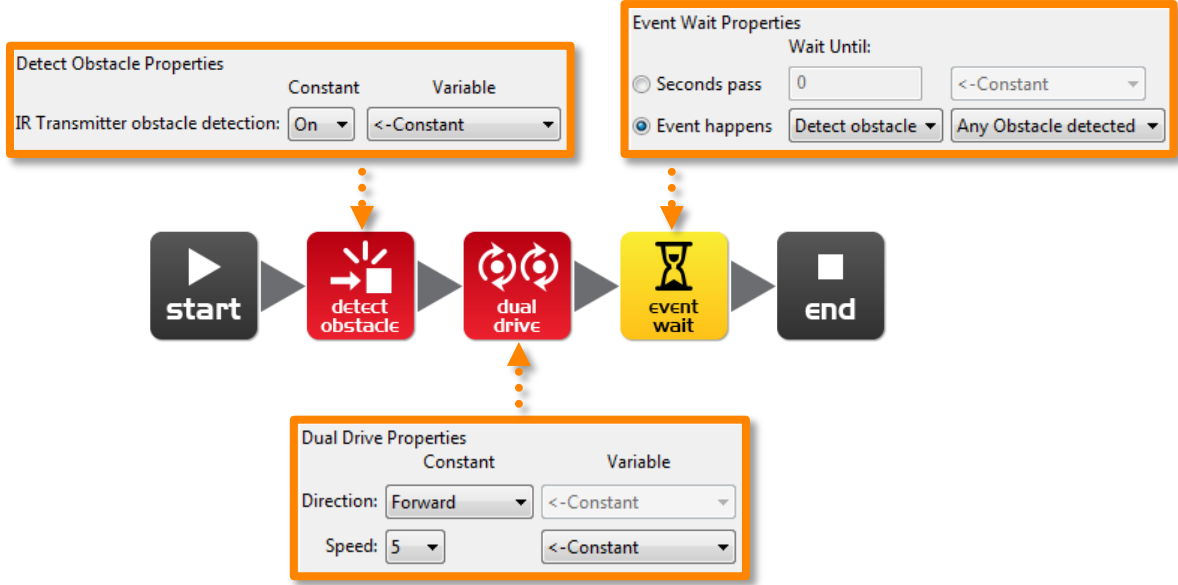




İsim: \_\_\_\_\_

## Ders 7 Çalışma yaprağı 7.2 – Engel algılama ve durma

Edison robotu bir engelle karşılaşınca kadar hareket ettirmek için aşağıdaki programı yazınız.



Kırmızı (kontrol grubu) engeli algıla simgesi engel algılama sistemini açmak ve kızıl ötesi ışık yayıcıları başlatmak için gereklidir.

Çift motorlu sürüş hızı robotun bir engelle çarpmadan önce onu tespit etmesine izin vermek için 5'e ayarlanır. Eğer hız çok fazlaysa o zaman robot engellere çarpacaktır.

Ne kadar uzaklıkta robot engelleri tespit edebildi?

\_\_\_\_\_

Bu tip görünmez algılamayı daha önce gördünüz mü? Nerede gördünüz?

\_\_\_\_\_  
\_\_\_\_\_

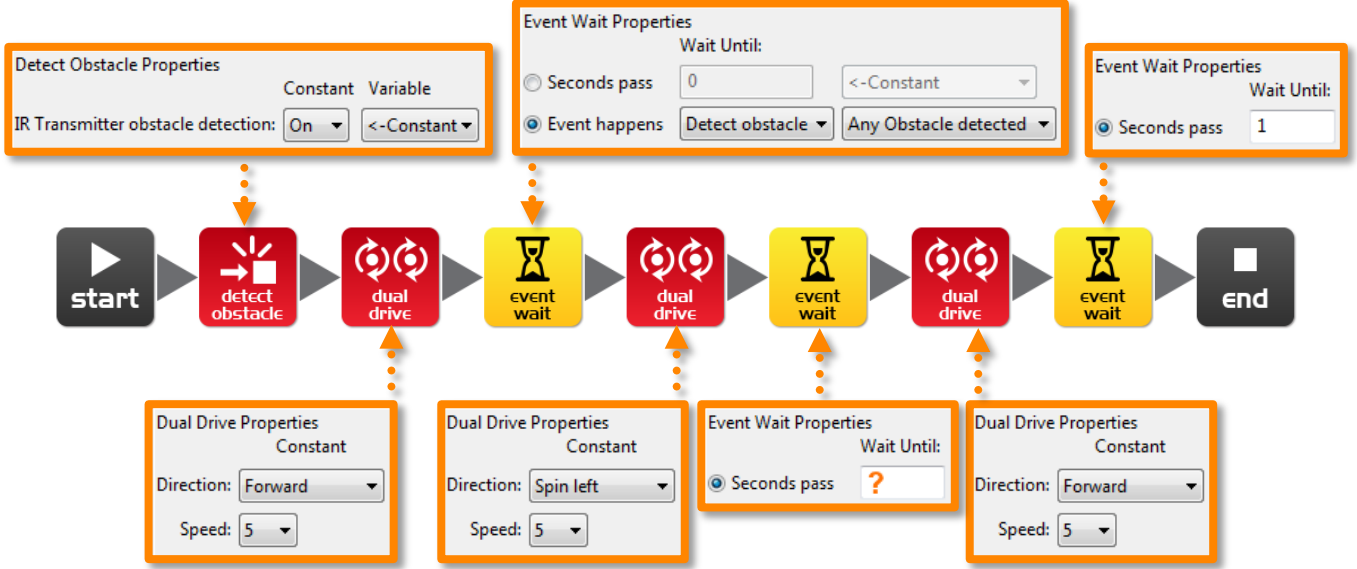
Bu tür algılama teknolojisinin başka nerelerde kullanılabileceğini düşünüyorsunuz?

\_\_\_\_\_  
\_\_\_\_\_

İsim: \_\_\_\_\_

## Ders 7 Çalışma yaprağı 7.3 – Engel algılama ve kaçınma

Edison robotu bir engelle karşılaşınca kadar hareket ettirmek daha sonra 180° döndürmek ve engelden 1 saniye uzaklaştırmak için aşağıdaki programı yazınız.



3. derste (çalışma yaprağı 3.2). robotu 180 derece döndermek için gereken sürenin doğru miktarını buldunuz. Yukarıda turuncu soru işaretle gösterilen yerde bu süreyi tekrar kullanınız.

Robotunuzu 180 derece döndermek için gereken sürenin doğru miktarı nedir?

\_\_\_\_\_

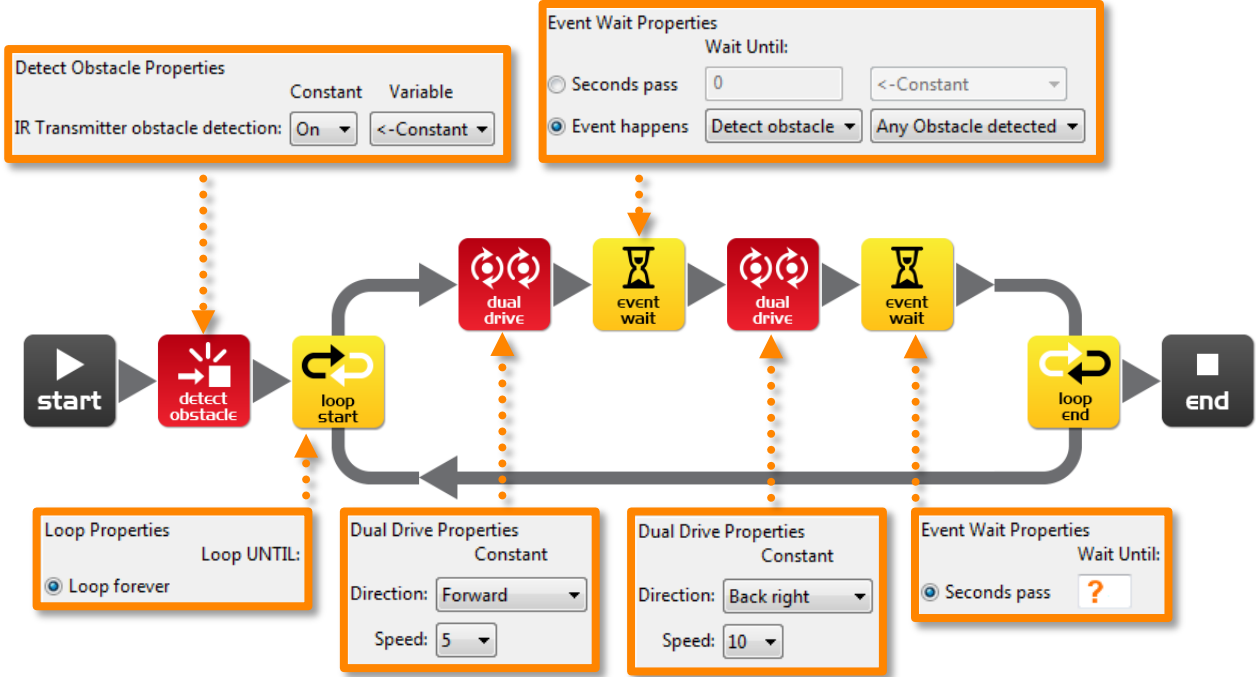
Bu programı eksik yapan şeyin ne olduğunu ve onu nasıl geliştirebileceğinizi düşününüz

\_\_\_\_\_  
\_\_\_\_\_  
\_\_\_\_\_  
\_\_\_\_\_  
\_\_\_\_\_

İsim: \_\_\_\_\_

## Ders 7 Çalışma yaprağı 7.4 – Engelden kaçınma

Edison robotun sürekli bir şekilde engellerden kaçınarak hareket etmesi için aşağıdaki programı yazınız.



İkinci bekleme durum simgesinde farklı miktarlarda süre ile deney yapmayı deneyiniz. Bu süre ayarı, robotun ne kadar süre döneceğini belirler ve yukarıda turuncu soru işaretiyle (?) gösterilir.

Robotun dönmesi için en iyi süre ayarının ne olduğunu düşünüyorsun?

\_\_\_\_\_

Bu süre ayarı niçin en iyisidir? Robot onu daha iyi yapmak için farklı olarak ne yapar?

\_\_\_\_\_  
\_\_\_\_\_  
\_\_\_\_\_



İsim: \_\_\_\_\_

## Ders 8 Çalışma yaprağı 8.1 – Çizgi takip sensörü

Edison robot çizgi takip sensörü ile donatılmıştır. Sensör iki temel elektronik bileşenden oluşur. Bunlar;

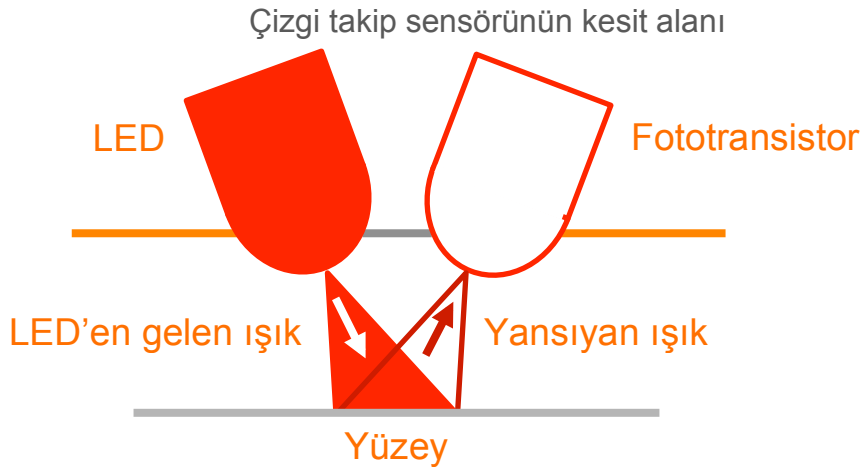
- 1) Kırmızı ışık yayıcı diyot (LED)
- 2) Fototransistör (ışık sensörü)

LED robotun hareket ettiği yüzey üzerine ışık saçar. Eğer Edison üzerindeki yuvarlak butona iki kez basarsanız LED hazır olacaktır. Robotu kaldırarak robotun yüzey üzerine ürettiği ışığın noktası çevresine bakınız. Siyah veya beyaz bir yüzeye yerleştirildiğinde ışık noktasının ne kadar parlak olduğunu karşılaştırın.

**Siyah veya beyaz yüzey üzerine yerleştirildiğinde nokta daha parlak (daha fazla ışık yansıtır) mı ?**

\_\_\_\_\_

Fototransistör bileşen bir ışık sensörüdür ve yüzeyden yansıyan ışığın miktarını ölçer.



Yukarıdaki örnekte gördüğümüz gibi, beyaz yüzeyden siyah bir yüzeye kıyasla daha fazla ışık yansır. Bu yüzden fototransistör beyaz bir yüzeyde siyah bir yüzeye kıyasla daha yüksek bir ışık okur. Bu, robotun sürüş yaptığı yüzeyde davranması ve tepki vermesi için programlanmasını sağlar. Siyah bir yüzey yansıtıcı olmayan olarak kabul edilir ve beyaz bir yüzey yansıtıcı olarak kabul edilir.

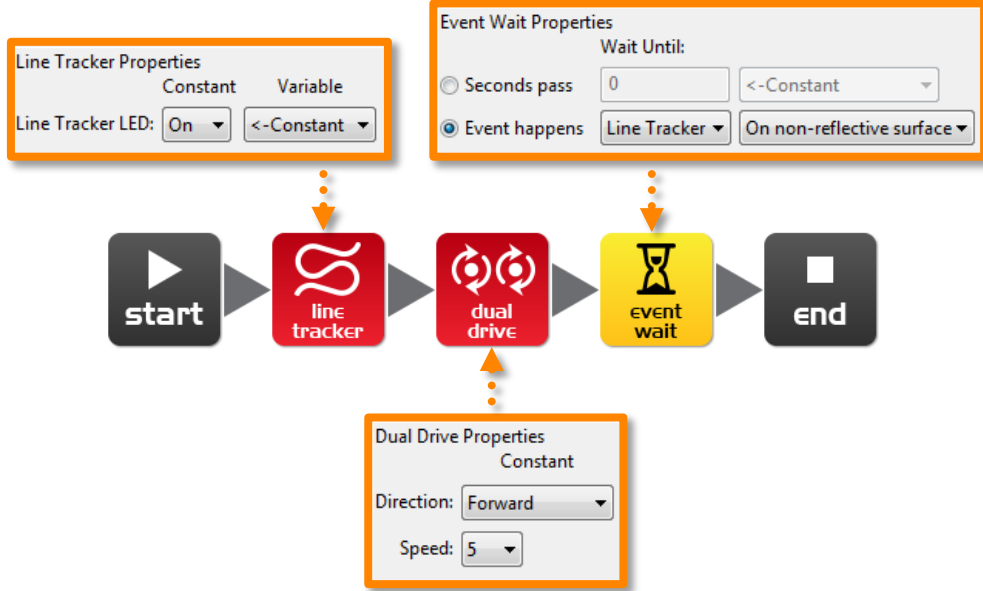
**Çizgi takipçisinin aşağıdaki yüzey renklerine, yansıtıcı veya yansıtıcı olmamasına nasıl tepki vereceğini düşünüyorsunuz? (İpucu: Işık kırmızı)**

**Kırmızı yüzey** \_\_\_\_\_, **Yeşil yüzey** \_\_\_\_\_, **Mavi yüzey** \_\_\_\_\_

İsim: \_\_\_\_\_

## Ders 8 Çalışma yaprağı 8.2 – Siyah bir çizgiye kadar sürüş

Siyah bir çizgiyle (yansıtıcı olmayan) karşılaşıncaya kadar beyaz yüzey (yansıtıcı) üzerinde Edison robotu hareket ettirmek için aşağıdaki programı yazınız.



Bir programda çizgi takip sensörünü kullanmak için öncelikle sensörü açmalısınız. Bu kırmızı LED'i de aktif eder.

Aktivite kağıdı 8.1'deki siyah çizgiyi kullanınız veya beyaz bir kağıt üzerine siyah bir çizgi çizin. Beyaz bir masa üzerine siyah elektrik bandında kullanabilirsiniz. Robotu siyah çizgiye doğru hareket ettiriniz ve robot duracaktır.

Aktivite kağıdı 8.1'de üç renkli çizgi vardır. Bunlar kırmızı, mavi ve yeşil. Edison'u çizgilerin her birine doğru hareket ettirin ve robotun durup durmayacağına bakınız.

Edison'un çok iyi algılayamadığı bir renk var mı? Hangi renk? \_\_\_\_\_

Bunun neden olduğunu düşünüyorsunuz?

---

---

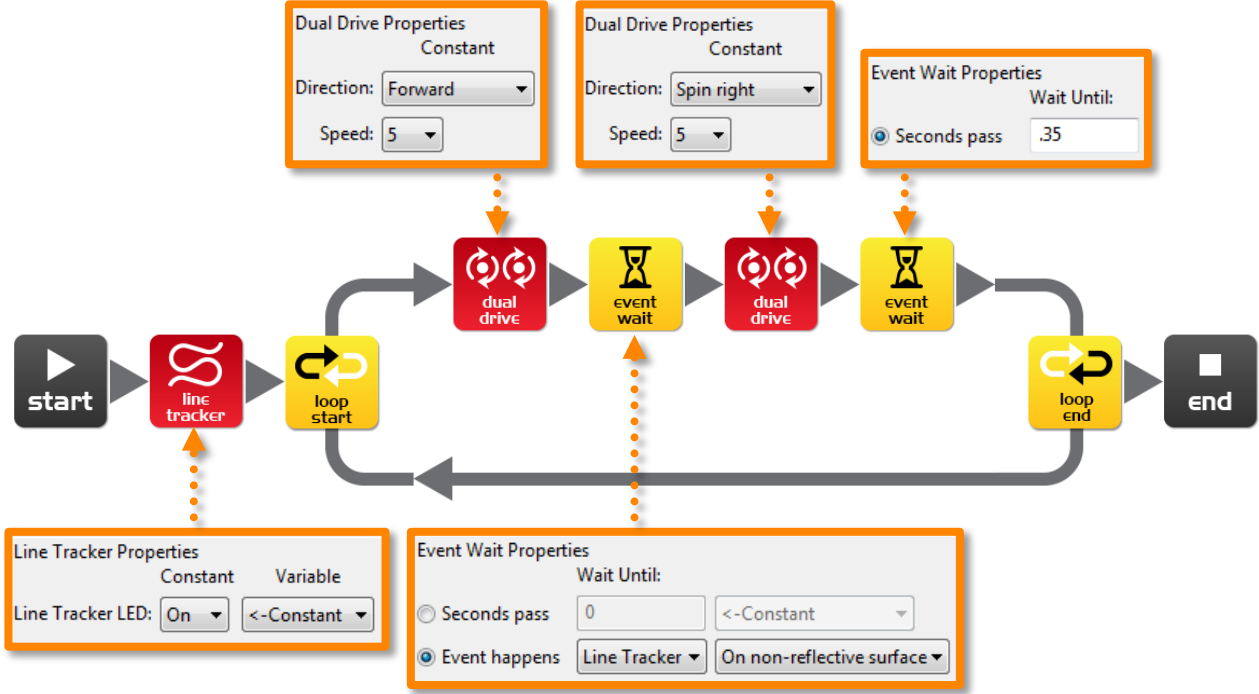
---

[www.meetedison.com](http://www.meetedison.com)

İsim: \_\_\_\_\_

## Ders 8 Çalışma yaprağı 8.3 – Bir sınır içerisinde sürüş

Bir sınır içerisinde Edison robotu hareket ettirmek için aşağıdaki programı yazınız.



Sınır olarak aktivite kağıdı 8.2'yi kullanınız veya büyük bir kağıt parçası ve kalın bir siyah kalem kullanarak kendiniz oluşturunuz.

Geniş bir sınır oluşturmak için beyaz bir masa üzerinde siyah elektrik bandı da kullanabilirsiniz.

Eğer geniş bir sınır oluşturursanız, ne olduğunu görmek için pek çok robot ekleyiniz.

Ayrıca, farklı hızlarla deneme yapınız.

Problemler var olmadan önce robot ne kadar hızlı gidebilir?

\_\_\_\_\_

Robor çok hızlı ilerlediğinde ne olur?

\_\_\_\_\_

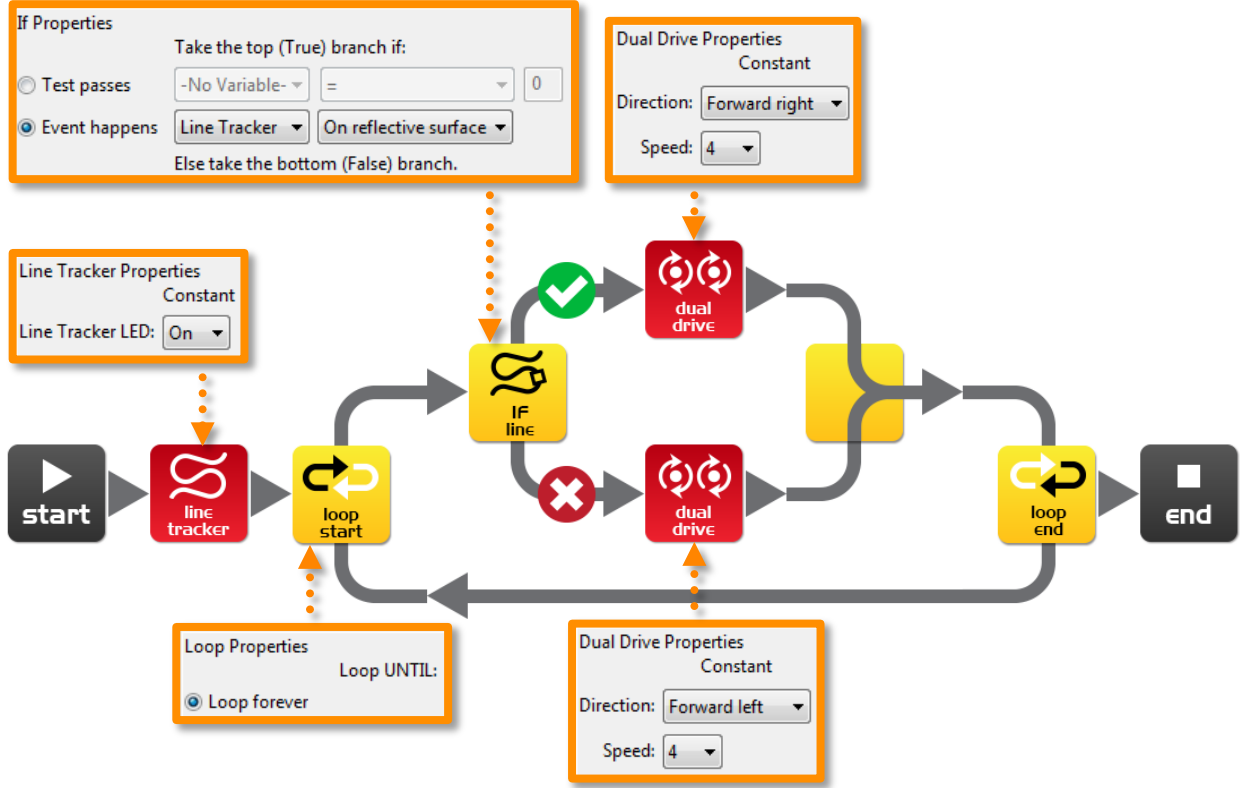
\_\_\_\_\_

\_\_\_\_\_

İsim: \_\_\_\_\_

## Ders 8 Çalışma yaprağı 8.4 – Bir çizgiyi takip etme

Edison robotun siyah bir çizgiyi takip etmesi için aşağıdaki programı yazınız.



Bu program robotun ne yapacağına karar vermesine izin vermek için Eğer simgesini kullanır. Eğer çizgi takip sistemi yansıtıcı bir yüzey üzerinde ise o zaman 4. hızda sağa ileri hareket eder. Eğer çizgi takip sistemi yansıtıcı bir yüzey üzerinde değilse o zaman 4. hızda sola ileri hareket eder. Robot beyaz üzerinde olduğunda robot beyazdan uzaklaşır. Robot siyah üzerinde olduğunda robot siyahtan uzaklaşır. Bundan dolayı robot çizginin kenarında ilerler.

Robotu çalışma yaprağı 8.2'deki hattın içerisine yerleştiriniz ve çizgiyi takip etmesini izleyin.

Robot hangi yönde hareket ediyor (saat yönünde veya saat yönünün tersi)? \_\_\_\_\_

Robotu çalışma yaprağı 8.2'deki hattın dışarısına yerleştiriniz ve çizgiyi takip etmesini izleyin.

Robot hangi yönde hareket ediyor (saat yönünde veya saat yönünün tersi)? \_\_\_\_\_

Neden robotun parkurda aynı yönde gitmediğini düşünüyorsunuz?

---

---

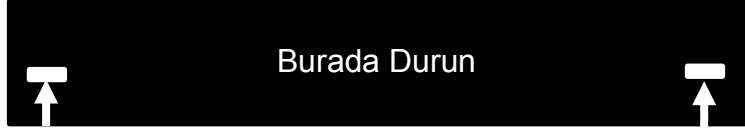
---



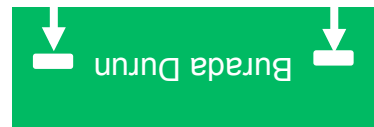
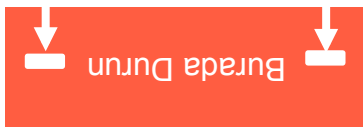
İsim: \_\_\_\_\_

## Ders 8 Aktivite kağıdı 8.1 – Çizgi takip sensörü

Çalışma yaprağı 8.2'deki programınızı kontrol etmek için bu aktivite kağıdını kullanınız



Aşağıdaki renkli çizgileri kullanarak çalışma yaprağı 8.1'deki cevaplarınızda doğrulayabilirsiniz. **Edison** hangi renkler için durur?

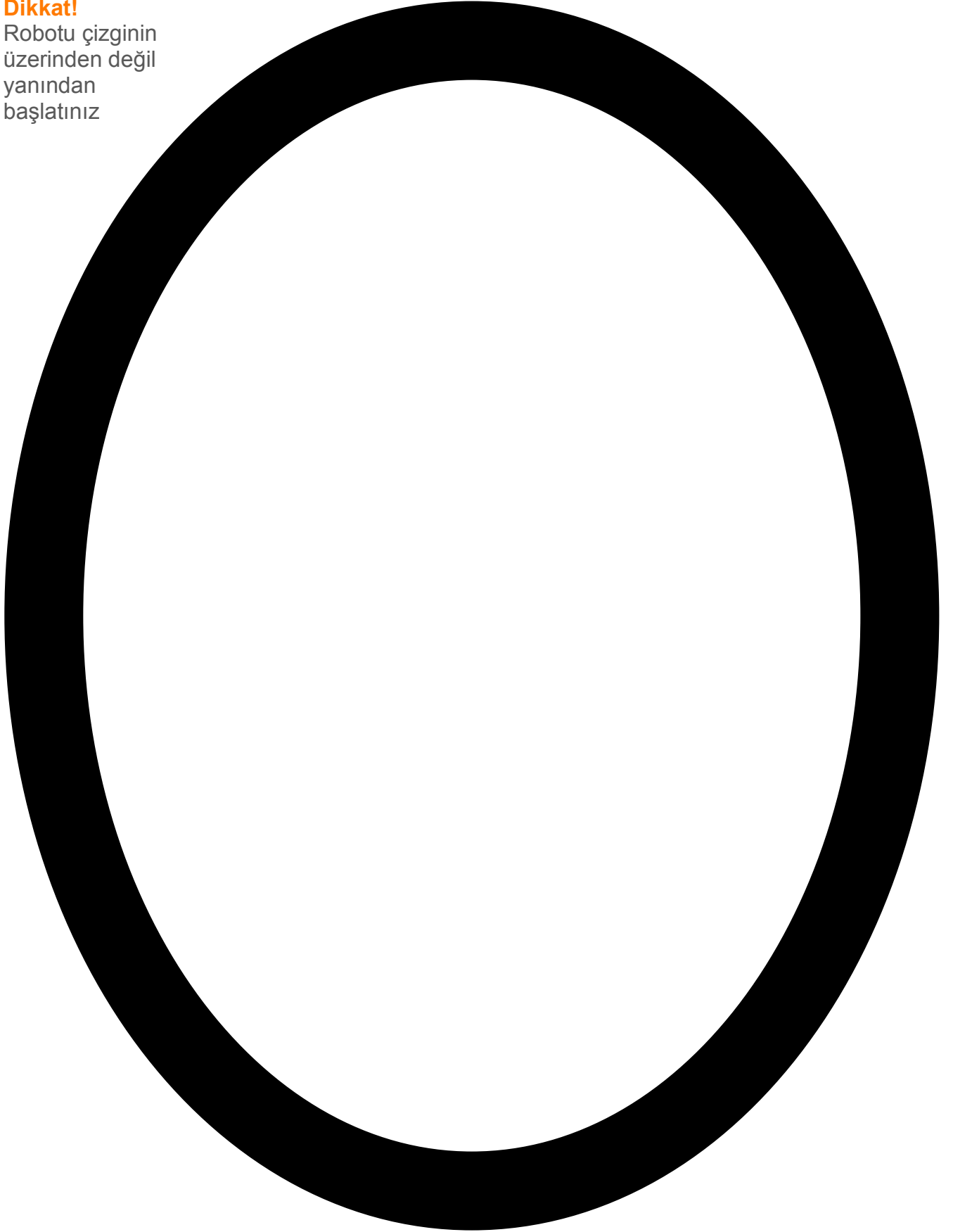


İsim: \_\_\_\_\_

## Ders 8 Aktivite kağıdı 8.2 – Çizgi takip sensörü

### Dikkat!

Robotu çizginin  
üzerinden değil  
yanından  
başlatınız



İsim: \_\_\_\_\_

## Ders 9 Çalışma yaprağı 9.1 – Değişkenleri anlama

Değişken verileri saklamak için bilgisayar hafızasının küçük bölümüdür. Değişkenleri bu kadar kullanışlı yapan şey, program çalışırken bu verilerin değişebileceğidir, bu nedenle adı değişkendir.

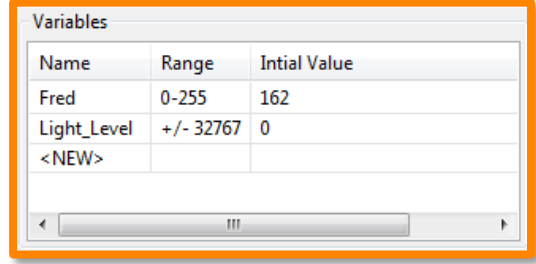
Değişkenler 10, 106, 1.482 vb gibi sayıları depolar ve matematiksel işlem yapmak için bilgisayar programına izin verir. Bu bilgisayarların çok iyi olduğu bir şeydir.

Edison bayt ve kelimeler olarak isimlendirilen iki tür değişkene sahiptir. Bayt değişkenler 0'dan 255'e kadar sayıları depolayabilir. Kelime değişkenler -32,767'den +32,767'ye kadar olan sayıları depolayabilir.

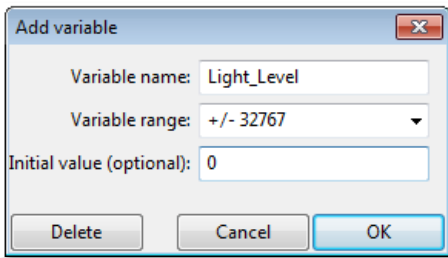
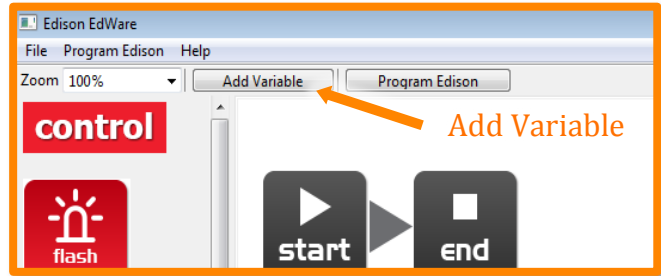
Değişkenleri kolay kullanmak için onlara isimler veririz. Bu değişkenlerde ne tür bilgilerin depolandığını hatırlamamız için bizlere yardım eder. EdWare'da değişkenlerinizi istediğiniz gibi adlandırabilirsiniz. Birini 'Fred' olarak isimlendirebilirsiniz, ancak Fred'in hangi tip bilginin saklandığını hatırlamak çok yararlı bir isim olmayabilir. Daha iyi bir isim 'Işık\_Seviyesi' olabilir. Bu tür bir ad, değişkenin ne için kullanıldığını ve orada ne tür veri bulunacağını hatırlamayı kolaylaştırır.

Artık değişkenleri bildiğimize göre, ışık alarm programımız için EdWare'de bir tane oluşturalım.

Sol köşedeki Değişken Ekle butonuna tıklayalım ve bir pop-up kutusu belirecektir.



Name	Range	Initial Value
Fred	0-255	162
Light_Level	+/- 32767	0
<NEW>		



Değişkeninizin adını Işık\_Seviyesi olarak yazınız, değişken aralığını +/-32767 olarak seçiniz ve başlangıç değerini sıfıra ayarlayınız. Şimdi Tamam'a tıklayınız ve sağ alt köşede değişkenler tablosuna değişkeniniz eklenecektir.

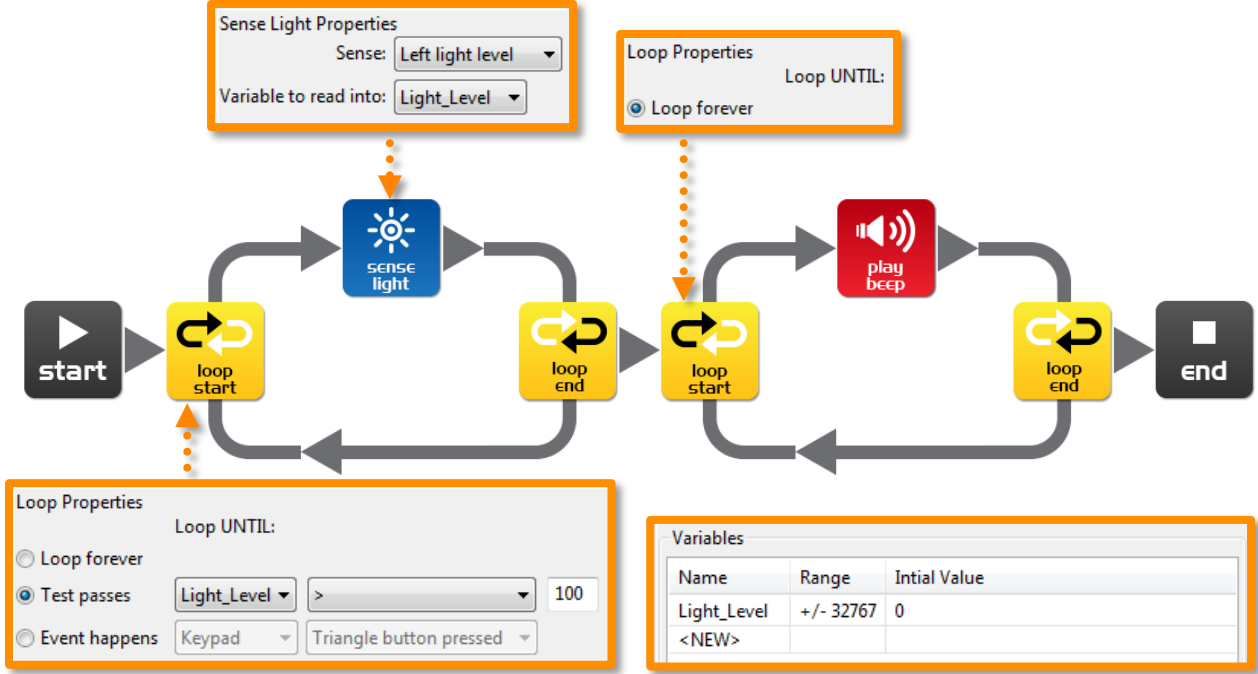
Aşağıdaki verileri depolamak için hangi tür değişkeni kullanırdınız (bayt veya kelimeler)?

12 \_\_\_\_\_, 192 \_\_\_\_\_, 801 \_\_\_\_\_, -42 \_\_\_\_\_, 27,901 \_\_\_\_\_

İsim: \_\_\_\_\_

## Ders 9 Çalışma kağıdı 9.2 – Işık alarmı

Odadaki ışıklar açıldığında Edison'un bir alarm sesi çıkarması için aşağıdaki programı yazınız.



Işık sensörü simgesi sol ışık sensöründen ışık seviyesini okur ve Işık\_Seviyesi olarak isimlendirilen değişkenin okumayı yerleştirir.

İlk döngü yapılacak şeyi belirlemek için matematik kullanır.

Işık\_Seviyesi olarak isimlendirilen değişkendeki değer 100'den daha büyük olana kadar döner.

Işık\_Seviyesindeki değer 100'den büyük olduğunda döngüden çıkılır ve alarm sesi üreten sonraki döngüye gider.

Edison robotu karanlığa koyunuz ve oyna butonuna basınız. Işıklar açıldığında robot alarm sesi çıkarır.

**Böyle bir alarmın yararlı olabileceği gerçek yaşam durumu düşünebilir misiniz?**

---

---

**Karanlık için bir alarm yapmak için programda yapılması gereken değişiklikler nelerdir?**

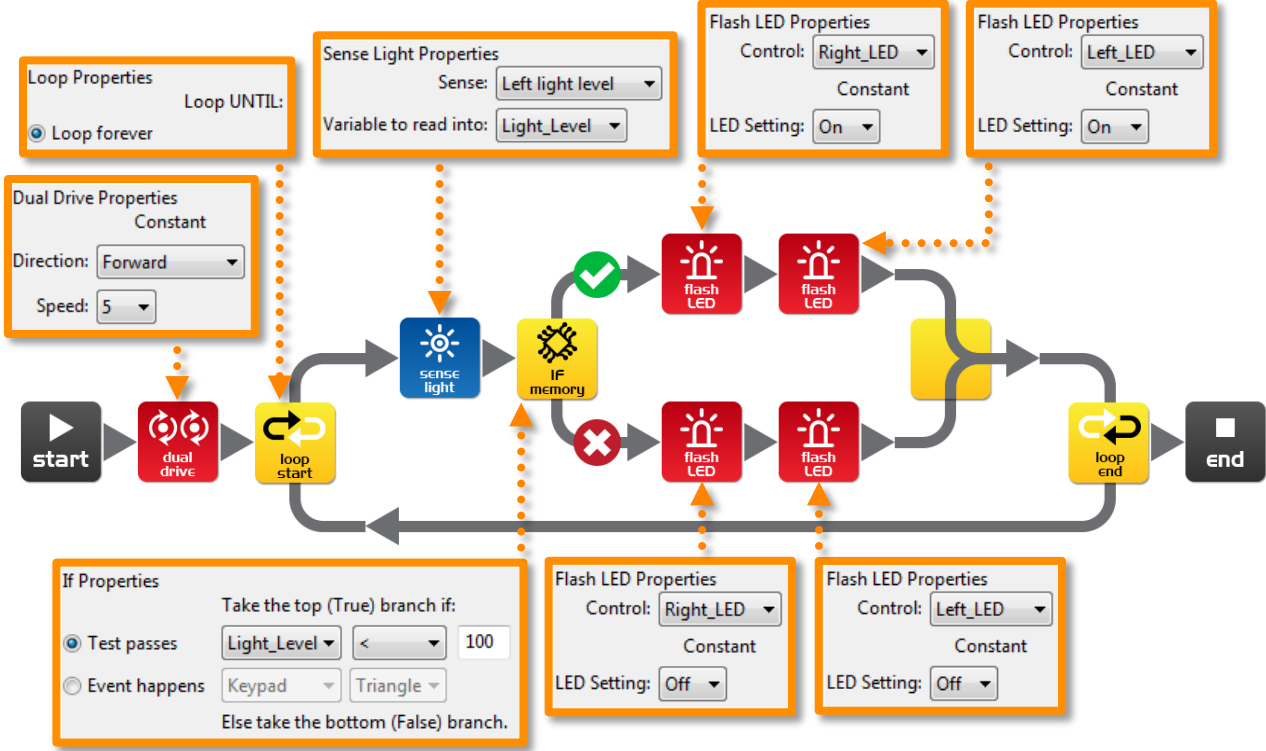
---

---

İsim: \_\_\_\_\_

## Ders 9 Çalışma yaprağı 9.3 – Otomatik ışıklar

Karanlık olduğunda Edison robotun iki LED ışığı yakması için aşağıdaki programı yazınız.



Bir tünel boyunca veya kanepenin altında robotu hareket ettiriniz ve ön ışıkların yanmasını izleyiniz.

Bu programda programın yolunu belirlemek için 'küçük'(<) sembolünü kullanıyoruz. Eğer Işık\_Seviyesi değişkeni 100'den 'küçük ise o zaman doğru yolu alır ve LED'leri açar.

Eğer sembolündeki değer (100) ile denemeler yapınız.

Değeri daha büyük yaptığınızda ne olur?

---

---

Değeri daha küçük yaptığınızda ne olur?

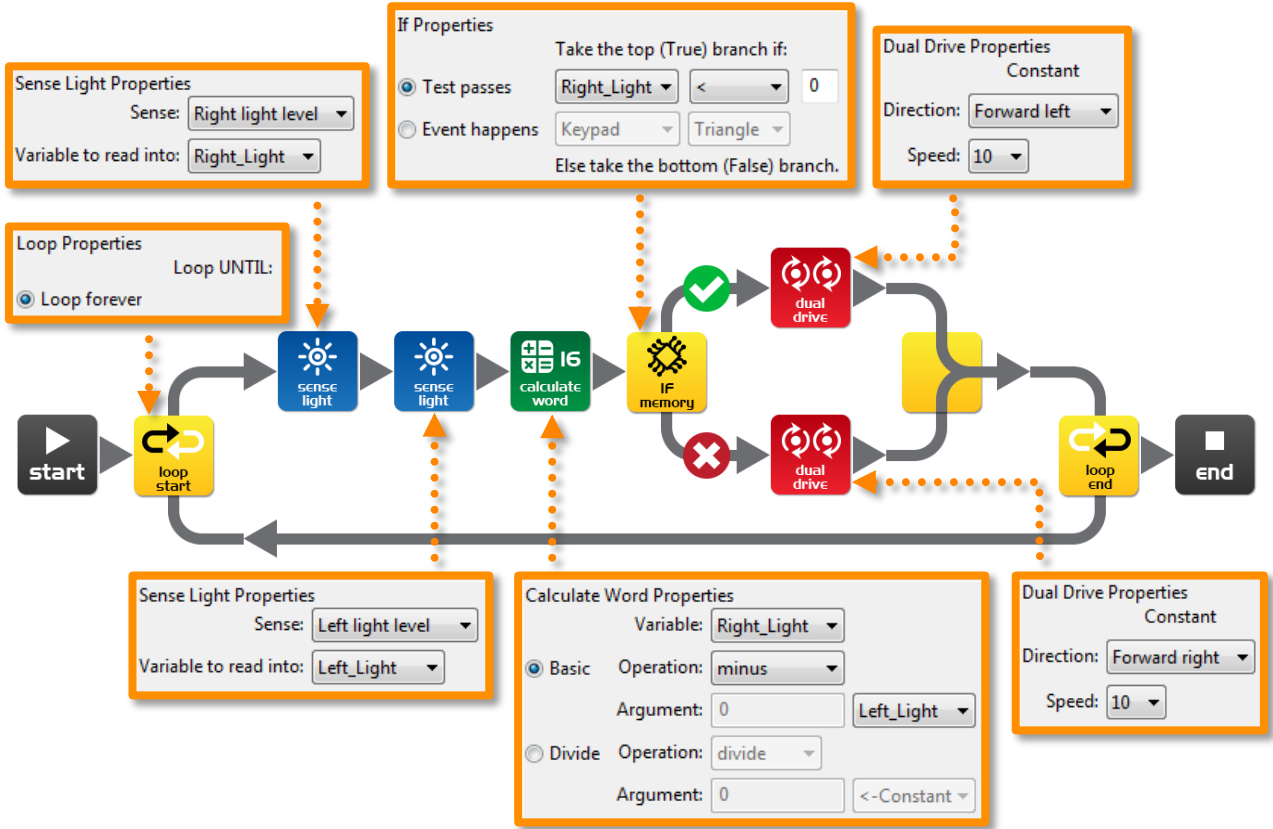
---

---

İsim: \_\_\_\_\_

## Ders 9 Çalışma yaprağı 9.4 – Işığın takibi

Edison robotun bir el feneri ışığını takip etmesi için aşağıdaki programı yazınız.



Edison'a bir el feneri tutunuz ve robot ışığa doğru hareket edecektir.

Bu program iki değişken arasında bir hesaplama yapar. Kelime hesaplama simgesinde 'Sağ\_Işık' değişkeni 'Sol\_Işık' değişkeninden çıkartılır. Sonuç (cevap) 'Sağ\_Işık' değişkenine geri yerleştirilir. Sonuçların neye benzeyeceğine bakalım:

	Sağ_Işık	Sol_Işık	Hesaplama	Sonuç
Sağdaki fener	200	100	$200 - 100 =$	-100
Soldaki fener	100	200	$100 - 200 =$	+100

Yukarıda görebileceğiniz gibi fener sağda olduğunda sonuç sıfırın altındadır (negatif sayı). Fener solda olduğunda sonuç sıfırın üzerindedir (pozitif sayı)

Eğer simgesi şunu sorar: Sonuç sıfırdan daha az mı?. Eğer doğruysa robot sola hareket eder (ışığa doğru), eğer yanlışsa robot sağa hareket eder (ışığa doğru).

Eğer 'küçük' sembolünü (<) 'büyük' (>) sembolüyle değiştirirseni ne olurdu?

İsim: \_\_\_\_\_

## Ders 10 Tasarım özeti çalışma yaprağı 10.1 – Programım

'İnsanların Başvurması Gerekmiyor' videosu gelecekte robotların nasıl kullanabileceğine yönelik anlayış sağlar. Robotunuzun yapabileceği uygun bir görev düşününüz ve daha sonra o görevi yapması için bir program yazınız.

Bazı örnekler aşağıdadır:

- Kurtarma robotu –Robot kayıp bir kişiyi (küçük bir figür veya bebek) bulmak için bir sınır içerisinde hareket eder. Robot kişinin yerini tespit ettiğinde alarm çalar.
- Sürücüsüz araba – Robot insanlar, diğer arabalar veya binalar (oyuncak) çarpmaksızın tasarlanmış bir yol da hareket eder.
- Savaş robotları – Savaşa insan askerlerden ziyade robotları gönderiniz. Bir sumo ringi ve iki veya daha fazla robotu birbirini araması için programlayınız ve daha sonra diğerini sumo ringinin dışına çıkartması için görevlendiriniz.

### 1. Tartışınız

Programlayabileceğiniz yararlı bir robot uygulamasına karar vermek için grup arkadaşınız veya diğer öğrenciler ile bunu tartışınız.

Ortaya çıkan fikirlerden bazıları nelerdi?

---

---

---

---

---

---

Fikirlerden birinin neden mümkün olmadığının sebepleri nelerdi?

---

---

---

---

---

---

[www.meetedison.com](http://www.meetedison.com)

İsim: \_\_\_\_\_

## 2. Açıklayınız

Programınızı yazmaya başlamadan önce robotunuzun çözeceği problemi ve bu problemi nasıl çözeceğini aşağıda açıklayınız.

Problem:

---

---

---

---

---

Çözüm:

---

---

---

---

---

## 3. Programınızı yazınız

Simgeleri yerleştirmeye başlamadan önce programınızı planlayınız. Kullanmak için öğrendiğiniz simgeleri kullanınız fakat diğer simgeleri de deneyebilirsiniz.

## 4. Başarısızlık mı?

Herşey ilk seferinde düşündüğünüz şekilde çalışmaz. Hiç problem değil, tekrar deneyin fakat ilkinde yanlış giden şeyi belirleyiniz. **Programınız ilk seferinde neden çalışmadı?** Ünlü Thomas Edison'un ampulü keşfetmeden önce 10,000 kez başarısız olduğunu hatırlayınız. Bu yüzden **denemeye devam ediniz!**

---

---

---

---

---

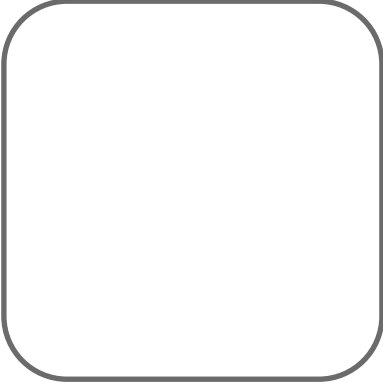


İsim: \_\_\_\_\_

## Ders 10 Tasarım özeti çalışma yaprağı 10.2 - Programım

### 5. Programınızdaki programlama simgelerinden bazılarını açıklayınız

Aşağıya programlama simgelerini çiziniz ve renklendiriniz. Daha sonra onların programınızda ne yaptığını açıklayınız.



Bu simgenin adı nedir? \_\_\_\_\_

Bu simge ne yapar? \_\_\_\_\_

---

---

---

---



Bu simgenin adı nedir? \_\_\_\_\_

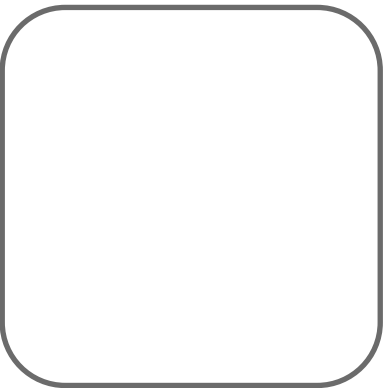
Bu simge ne yapar? \_\_\_\_\_

---

---

---

---



Bu simgenin adı nedir? \_\_\_\_\_

Bu simge ne yapar? \_\_\_\_\_

---

---

---

---

İsim: \_\_\_\_\_

## Öğrenci başarı çizelgesi

Program	Onay
2.1 Robotu ileri hareket ettirme	
2.2 Robotu geri hareket ettirme	
2.3 Robotu ileri ve geri hareket ettirme	
2.4 Sürat oyunu	
3.1 Sağa dönüş	
3.2 Sola dönüş	
3.3 Sağa ve daha sonra sola dönüş	
3.4 Küçük labirent	
4.1 Görev sürüşü	
4.2 Meksika dalgası robot stili	
5 Tasarım özeti 1– Programım	
6.1 El çırpıma yanıt olarak LED yakıp söndürme	
6.2 El çırpıma yanıt olarak hareket etme	
6.3 El çırpıma yanıt olarak dans etme	

Program	Onay
7.1 Bir engeli tespit etme ve durma	
7.2 Bir engeli tespit etme ve kaçınma	
7.3 Bir engeli tespit etme ve bir döngüden kaçınma	
7.4 Sağ ve sol engel tespiti	
8.2 Siyah çizgiye kadar hareket etme	
8.3 Bir sınır içereisinde hareket etme	
8.4 Çizgiyi takip etme	
9.2 Işık seviyesi alarmı	
9.3 Otomatik ışıklar	
9.4 Işığı takip etme	
10 Tasarım özeti 2 – Programım	
Kendi programım 1	
Kendi programım 2	
Kendi programım 3	