



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

## 10 Ders Planı



Bu ders planları RoboticsWPS ile birlikte üretilmiştir.

**ROBOTICS**  
*Robotic Workshops*  
coding for schools

[www.roboticswps.com.au](http://www.roboticswps.com.au)

<sup>1</sup> Türkçe çevirisi ve uyarlaması MeetEdison'un tek yetkili Türkiye distribütörü STEM Makers Academy ([www.stemmakers.academy](http://www.stemmakers.academy)) tarafından yapılmıştır.



## Ders Planları-İçindekiler

Robotik WPS'i Tanıma .....	4
Özet .....	5
Ders Planına Genel Bakış .....	6
Ders 1: Tanıyın ve Kurun.....	9
Ders 1 Çalışma Yaprağı 1.1 – Edison'u Tanıma .....	16
Ders 1 Çalışma Yaprağı 1.2 – Barkod Programlama .....	17
Ders 1 Çalışma yaprağı 1.3 – EdWare'la Tanışın .....	18
Ders 1 Çalışma Yaprağı 1.4 – Bir test programı indirme .....	19
DERS 2: ROBOT HAREKETİ – SÜRÜŞ .....	20
Ders 2 Çalışma yaprağı 2.1 – Robotu ileriye sürüş .....	21
Ders 2 Çalışma yaprağı 2.2 – Robotu geriye sürüş.....	22
Ders 2 Çalışma yaprağı 2.3 – İleriye ve daha sonra geriye.....	23
Ders 2 Çalışma yarağı 2.4 – Hız oyunu.....	24
Ders 2 Çalışma kağıdı 2.1 .....	25
DERS 3: ROBOT HAREKETİ – DÖNDÜRME .....	26
Ders 3 Çalışma yaprağı 3.1 – 90 derece sağa dönüş .....	27
Ders 3 Çalışma yaprağı 3.2 – 180 derece sola dönüş .....	28
Ders 3 Çalışma yaprağı 3.3 – Sağa dönüş sonra sola dönüş .....	29
Ders 3 Çalışma yaprağı 3.4 – Küçük labirent .....	30
Ders 3 Aktivite kağıdı 3.1 – Dönüş .....	31
Ders 3 Aktivite kağıdı 3.2 – Küçük labirent.....	32
DERS 4: LABİRENT MEYDAN OKUMASI VE MEKSİKA DALGASI.....	33
Ders 4 Çalışma yaprağı 4.1 – Görev .....	34
Ders 4 Çalışma yaprağı 4.2 – Meksika dalgası .....	35
DERS 5: TASARIM ÖZETİ 1 – PROGRAMIM .....	36
Ders 5 Tasarım özeti çalışma yaprağı 5.1 – Programım .....	37
Ders 5 Tasarım özeti çalışma yaprağı 5.2 – Programım .....	38
DERS 6: ALKIŞ ALGISI.....	39
Ders 6 Çalışma yaprağı 6.1 – Alkışa yanıt olarak LED yakma .....	40
Ders 6 Çalışma yaprağı 6.2 – Alkışa yanıt olarak sürüş.....	41
Ders 6 Çalışma yaprağı 6.3 – Alkışa yanıt olarak dans.....	42

DERS 7: ENGEL ALGILAMA .....	43
Engel algılayıcısını kalibre ediniz .....	44
Ders 7 Çalışma yaprağı 7.1 – Kızılötesi ile engel algılama .....	45
Ders 7 Çalışma yaprağı 7.2 – Engel algılama ve durma .....	46
Ders 7 Çalışma yaprağı 7.3 – Engel algılama ve kaçınma.....	47
Ders 7 Çalışma yaprağı 7.4 – Engelden kaçınma .....	48
Ders 7 Çalışma yaprağı 7.5 – Sağ ve sol engel algılaması .....	49
DERS 8: ÇİZGİ ALGILAMA VE TAKİP .....	50
Ders 8 Çalışma yaprağı 8.1 – Çizgi takip sensörü .....	51
Ders 8 Çalışma yaprağı 8.2 – Siyah bir çizgiye kadar sürüş .....	52
Ders 8 Çalışma yaprağı 8.3 – Bir sınır içerisinde sürüş .....	53
Ders 8 Çalışma yaprağı 8.4 – Bir çizgiyi takip etme .....	54
Ders 8 Aktivite kağıdı 8.1 – Çizgi takip sensörü .....	55
Ders 8 Aktivite kağıdı 8.2 – Çizgi takip sensörü .....	56
DERS 9: IŞIĞA TEPKİ.....	57
Ders 9 Çalışma yaprağı 9.1 – Değişkenleri anlama .....	58
Ders 9 Çalışma kağıdı 9.2 – Işık alarmı .....	59
Ders 9 Çalışma yaprağı 9.3 – Otomatik ışıklar .....	60
Ders 9 Çalışma yaprağı 9.4 – Işığı takibi.....	61
DERS 10: TASARIM ÖZETİ 2 – PROGRAMIM .....	62
Ders 10 Tasarım özeti çalışma yaprağı 10.1 – Programım .....	63
Ders 10 Tasarım özeti çalışma yaprağı 10.2 – Programım .....	65
Öğrenci başarı çizelgesi .....	66

## Robotik WPS'i Tanıma

Michele Miller 2009'da kurulan Robotics WPS'in kurucusu ve yöneticisidir. Michele, Avustralya, New South Wales and Queensland'da kayıtlı bir öğretmendir ve Sydney Teknoloji Üniversitesi (UTS) INSEARCH'de öğretim görevlisi olup 20 yılı aşkın tecrübeye sahiptir.

2010'da NSW'de öğretim yaparken Sydney'deki UTS INSEARCH'de robotik programı koordine etmeye davet edildi. O günden bu yana Robotics WPS ekibi bir dizi kodlama atölyesi geliştirdi ve şu anda Avustralya okullarına robotik ve programlama atölyeleri yapıyor. Her yıl 5,000'in üzerinde öğrenci Robotics WPS atölyelerine katılmaktadır.

Robotik WPS ayrıca, personel gelişimi günleri ve atölyeler sağlayarak okullara yardım etmektedir



Michele Miller ve istekli bir robotik öğrencisi

## Özet

Bu dersler çeşitli robotlar kullanılarak 5 yıldan fazla bir süredir Robotics WPS tarafından gerçekleştirilen atölyelerde tasarlandı ve test edildi. Dersler şimdi Edison robot ile test edilmekte ve çok başarılı olduğu ortaya konulmaktadır.

Bu ders planlarını ister oldukları gibi isterse de kendi özelleştirilmiş planlarınızı geliştirecek bir başlangıç noktası olarak kullanabilirsiniz. Bazı öğretmenlerin öğrencilerinin yeteneklerine göre küçük değişiklikler yapmaları gerekebilir. Dersler basit hareket davranışları gibi daha kolay programlama simgeleri ile başlar ve çizgi takibi ve sumo güreşi gibi sınıf favorileri ile sonlanır.


Her yaşta ki öğrenciler, bu ders planlarıyla ilgili programlama sürecinden, problem çözme ve işbirliğinden gerçekten hoşlanırlar.

Bazı öğrencilerin başlangıç kavramlarını kolayca kavrayacakları ve bir sonraki etkinlik için hazır olduklarını göreceksiniz. Bu ders planları, bağımsız öğrenmeyi ve bir sonraki derse geçiş yapmayı sağlar. Bu ders planları öğrencilerinizin dersten derse geçişi ile süreç ile ilgili farkındalığını geliştirecektir.

VO.2 Your EdVenture into Robotics 10 Lesson Plans

### Lesson 2 Worksheet 1 - Drive the robot forward

Write the following program to drive the Edison robot forward. Use the activity sheet or marked lines on a desk or as start and stop markers.



Set the 'Seconds pass' (replace?) to an amount of time in seconds to have the robot start from behind the start line and stop before the finish line.

Minimum amount of time is 0.01 seconds.  
Maximum amount of time is 327.67 seconds.

Try different amounts of time until your robot drives forward and stops just before the finish line.

What is the correct amount of time to get your robot from the start to the finish?  
\_\_\_\_\_

Describe what the robot is doing and why?  
\_\_\_\_\_  
\_\_\_\_\_  
\_\_\_\_\_

www.meetedison.com www.roboticswps.com.au page 20

## Çalışma Yaprakları

Her bir ders için birkaç tane çalışma yaprağı oluşturulmuştur. Çalışma yaprakları öğrencilerin takip etmesi gereken talimatları içerdiğinden öğrencilerin bağımsız bir şekilde çalışmalarına izin verir.

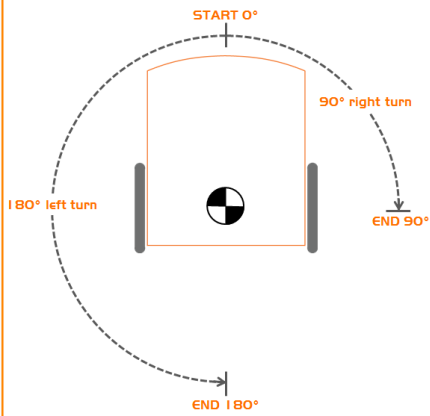
Çalışma yaprakları öğrencilerin öğrenmelerini göstermek ve güçlüklerine yanıt vermek amacıyla öğrenciler için sorular da içerir.

VO.5 Your EdVenture into Robotics 10 Lesson Plans

### Lesson 3 Activity sheet 3.1 - Turning

Place the robot down as per the outline then play your turning program. Your robot should:

Program 1 - Turn right from start 0° to end 90°  
Program 2 - Turn left from start 0° to end 180°  
Program 3 - Turn right from start 0° to end 90°, then turn left 270° to end 180°



www.meetedison.com www.roboticswps.com.au page 31

## Aktivite Kağıtları

Derslerin bazıları aktivite kağıtlarını içerir. Aktivite kağıtları, öğrencilerin programlarını çabuk bir şekilde kontrol etmeleri ve deneylerini çalıştırmaları için bir çalışma alanı sağlar.

## Ders Planına Genel Bakış

Birçok ders planı 90 dakikayı tamamlamak için tasarlanmıştır fakat bu öğrencinin yaşına ve yeteneğine göre değişecektir. Tüm ders planları Avustralya müfredatı ile bağlantılıdır. Lütfen [meetedison.com/](http://meetedison.com/) adresindeki referansların tam bir listesine bakın.

### Ders 1: Tanıyın ve kurun

**Teknoloji Becerileri** - Öğrenciler robota bir programı nasıl indirecekleri ve programlama ortamı hakkında bilgi sahibi olurlar.

1. Barkod programlarını kullanarak Edison'u kurun ve bilgi sahibi olun
2. Yazılımı açın ve simgelerin nasıl taşınacağı konusunda bilgi sahibi olun
3. Test programını indirin

### Ders 2: Robot hareketi - hareket ettirme

**Dizisel programlamaya giriş**– Öğrenciler robotun talimat simgelerine nasıl yanıtlar verdiğini ve zaman, hız ve uzaklık kavramlarını nasıl bir araya getirdiğini öğrenir

- Program 1 – Robotu ileri doğru hareket ettirme (Çalışma yaprağı 2.1)
- Program 2 – Robotu geri doğru hareket ettirme (Çalışma yaprağı 2.2)
- Program 3 – Robotu ileri ve geri hareket ettirme (Çalışma yaprağı 2.3)
- Program 4 – Hız oyunu (Çalışma yaprağı 2.4)

### Ders 3: Robot hareketi – döndürme

**Dizisel programlama ve temel geometri** – Öğrenciler robotun zaman ve geometriye nasıl cevap verdiğini ve robotun hareket etme kontrolünü nasıl sağlayacaklarını öğrenir.

- Program 1 – Sağa dönüş (Çalışma yaprağı 3.1)
- Program 2 – Sola dönüş (Çalışma yaprağı 3.2)
- Program 3 – Sağa ve daha sonra sola dönüş (Çalışma yaprağı 3.3)
- Program 4 – Küçük labirent (Çalışma yaprağı 3.4)

### Ders 4: Labirent meydan okuması ve Meksika dalgası

**Öğrenmeyi pekiştirmek** – Öğrenciler iki eğlenceli açık uçlu aktiviteyi yapmak için 1. dersten 3. derse bilgilerini kullanır.

- Program 1– Görev sürüşü (Çalışma yaprağı 4.1)
- Program 2 – Meksika dalgası robot stili (Çalışma yaprağı 4.2)

### Ders 5: Tasarım özeti 1 – Programım

**Yaratıcı düşünme ve problem çözme** – Öğrenciler kendi görevleriyle karşımıza çıkar ve robotun nasıl bir çözüm sağlayacağını kavramsallaştırmak için çözüm yolu bulur. Öğrenciler kendi konusunu seçebilir, programının amacını ifade edebilir ve gerçek dünyada nerede kullanılabileceğini açıklayabilir.

1. Robotun çözebileceği bir problem veya eğlenceli bir müzik hareketi belirleyiniz
2. Problemi veya robotun yapması gereken hareketlerini tanımlayınız
3. Programı yazınız ve onu test ediniz
4. Başarısızlık– İlk deneme hiçbir zaman başarılı değildir. Onu kanıtlayın ve denemeye devam edin!
5. Yapmak istediğiniz şeyi ve kullanılan programlama simgelerini tanımlayınız.

6. Demo – Öğrenciler sınıfa robotlarının programını gösterir

### Ders 6: Alkış algılama

**Girişler (sensörler)** – Öğrenciler robotun dış uyarıcıya (alkış) nasıl yanıt vereceğini öğrenir. Bu ders eğlenceli bir sınıf aktivitesini de içerir.

- Program 1 – Alkışa yanıt olarak LED yakıp söndürünüz (Çalışma yaprağı 6.1)
- Program 2 – Alkışa yanıt olarak hareket ettiriniz (Çalışma yaprağı 6.2)
- Program 3 – Alkışa yanıt olarak dans ettiriniz (Çalışma yaprağı 6.3)

### Ders 7: Engelleri tespit ediniz

**Engelleri tespit etme kavramı ve yapay zekaya giriş** – Öğrenciler robotun çevresindeki engellere yanıt olarak karar vermesini programlar.

- Kızıl ötesi engel algılamayı anlama (Çalışma yaprağı 7.1)
- Program 1 – Bir engeli tespit etme ve durma (Çalışma yaprağı 7.2)
- Program 2 – Bir engeli tespit etme ve kaçınma (Çalışma yaprağı 7.3)
- Program 3 – Döngüde bir engeli tespit etme ve kaçınma (Çalışma yaprağı 7.4)
- Program 4 – Sağ ve sol engel tespiti (Çalışma yaprağı 7.5)

### Ders 8: Çizgi algılama ve takip etme

**Endüstride kullanılan robot davranışlara benzer** – Öğrenciler, gelişmiş otomatik fabrikalarda ve ambarlarda kullanılan benzer, temel çizgi algılama ve kontrolü hakkında bilgi edinirler.

- Çizgi takip sensörünü anlama (Çalışma yaprağı 8.1)
- Program 1 – Siyah çizgiye kadar hareket ettirmek (Çalışma yaprağı 8.2)
- Program 2 – Bir sınır içerisinde hareket ettirmek (Çalışma yaprağı 8.3)
- Program 3 – Bir çizgiyi takip etmek (Çalışma yaprağı 8.4)
- Video – İnsanların başvurması gerekmiyor

### Ders 9: Işığa Tepki

**Çevresel Ölçüm ve Matematik Programlama** – Öğrenciler ışık seviyesi ölçümü, onları hafızada depolama ve robotun davranışlarını kontrol etmek için matematiksel işlemler gerçekleştirmeyi öğrenir.

- Değişkenleri anlama (Çalışma yaprağı 9.1)
- Program 1 – Işık seviyesi alarmı (Çalışma yaprağı 9.2)
- Program 2 – Otomatik ışıklar (Çalışma yaprağı 9.3)
- Program 3 – Işığı takip etme (Çalışma yaprağı 9.4)
- Robotikte hayvan davranışları

### Ders 10: Tasarım Özeti 2 – Programım

**Yaratıcı düşünme ve problem çözme** – Bu öğrencilerin kendi görevlerine ve robotun nasıl bir çözüm sağlayacağını kavramsallaştırmak için çözüm yolu bulduğu ikinci tasarım özeti.

1. Robotun çözebileceği bir problem veya eylem belirleyiniz
2. Problemi veya robotun yapması gereken hareketlerini tanımlayınız
3. Programı yazınız ve onu test ediniz

4. Başarısızlık– İlk deneme hiçbir zaman başarılı değildir. Onu kanıtlayın ve denemeye devam edin!
5. Yapmak istediğiniz şeyi ve kullanılan programlama simgelerini tanımlayınız.
6. Demo – Öğrenciler sınıfa robotlarının programını gösterir

### NAPLAN Sınavına Uygunluk (Avustralya ulusal öğrenci sınavı)

Öğrenciler bilim, teknoloji, mühendislik ve matematik (STEM) gibi alanlarda hızlı öğrenmeye yardımcı olan robotbilimini çekiçi buluyorlar. Fakat bu yüksek düzeydeki katılımdan yararlanılabilecek diğer eğitim unsurları da vardır.

Ders planlarının başından sonuna kadar öğrenciler programlarını ve robotlarının yapacağı eylemleri betimlemek için yazılı cevaplar sağlar. Tasarım özeti çalışma yapraklarında öğrenciler kendi programlarının planlarlar ve oluştururlar. Öğrenciler ayrıca programlarında kullanılan simgeler için betimsel davranışları da yazarlar.

Robotics WPS deneyimi pek çok öğrencinin görsel bir uyarandan bilgiyi alıp yazılı metne aktarmakta güçlük çektiğini göstermektedir.




Bu ders planları öğrencilerin görsel bir uyarandan (resim, grafik veya çizelge) yazılı cevaplar vermesine yardımcı olur. Bu NAPLAN sınavında kritik bir beceridir.

Bu öğrencilerin yeteneklerinin ve programlarından edindikleri bilgiyi yazılı metinlere nasıl transfer edeceği anlayışının gelişiminde büyük gelişmelerin görüldüğü Robotics WPS alanlarının birisidir.

Your EdVenture into Robotics 10 Lesson Plans

**Lesson 5 Design brief worksheet 5.2 – My program**

**5. Describe some of the programming icons in your program**  
Draw and colour in the programming icons below. Then describe what they do in your program.

	<p>What is the name of this icon? <u>dual drive</u></p> <p>What does this icon do? <u>In my program the dual drive icon drives my Robot forward at speed 5.</u></p>
	<p>What is the name of this icon? <u>event wait</u></p> <p>What does this icon do? <u>In my program the event wait icon keeps my robot driving forward for 1.5 seconds. When this time has finished the program continues.</u></p>
	<p>What is the name of this icon? <u>flash LED</u></p> <p>What does this icon do? <u>In my program the flash LED icon turns on the left LED to draw attention to my dancing robot.</u></p>

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

Tamamlanmış bir tasarım özeti örneği

## Ders 1: Tanıyın ve Kurun

**Teknoloji Becerileri** - Öğrenciler robota bir programı nasıl indirecekleri ve programlama ortamı hakkında bilgi sahibi olurlar

Edison'u programlamadan önce birkaç şeyin hazır olmasına ihtiyaç duyarız.

1. Edison'u hazırlama
2. Edison'u tanıma
3. Edison'un yazılımı EdWare'ı yüklemek
4. Bir test programı indirerek her şeyin çalıştığını test etmek

### Edison'u Hazırlama

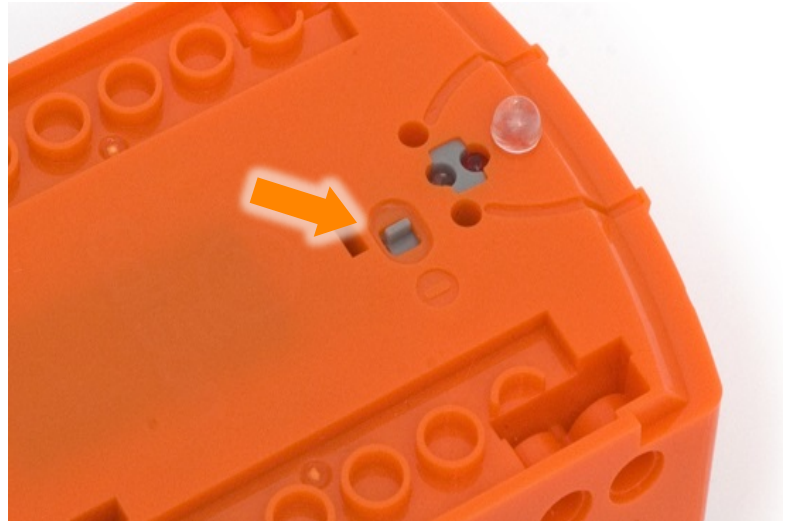


Pillerin doğru yönde olduğundan emin olunuz

Pil bölümünü açınız ve programlama kablosunu çıkartınız. Şimdi 4 tane 'AAA' pili yerleştiriniz. Pillerin doğru yönde olduğundan emin olmak için görüntüyü inceleyiniz ve pil kapağını kapatınız.

Güç anahtarını açık konuma getirerek Edison'u açınız. Edison'un kırmızı LED ışıkları şimdi yanıp sönmeye başlayacaktır.

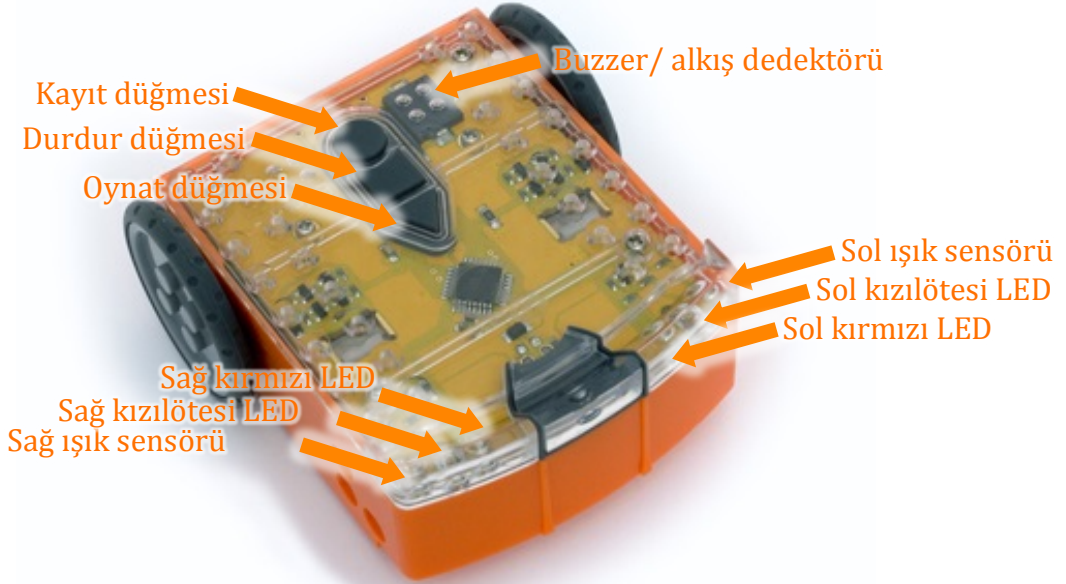
Edison çalışmaya hazır!



Düğmeyi sembol üzerine doğru itiniz

## Edison'u Tanıma (Çalışma Kağıdı 1.1)

Öğrencilerin Edison'u tanımaları ve tüm sensörlerin nerede olduğunu ve üzerinde yer alan üç düğmenin işlevini bilmeleri gerekecektir. Öğrencilerin Çalışma Yapağı 1'deki görüntüleri gözden geçirmelerini sağlayın. Öğrencilerin bu ders kitabındaki 10 ders planı süresince bunları tekrar gözden geçirmeleri faydalı olabilir.



Edison'un sensörleri ve düğmelerini tanıma

Oynat düğmesi – Programı başlat

Durdu düğmesi – Programı durdurmak için bas

Kayıt düğmesi – 1 basış = programı indir, 3 basış = barkodu oku

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

Çizgi takip sensörü daha önceden yüklenmiş programları aktive eden özel barkodları da okur.



Edison'un güç düğmesi ve çizgi takipsensörü



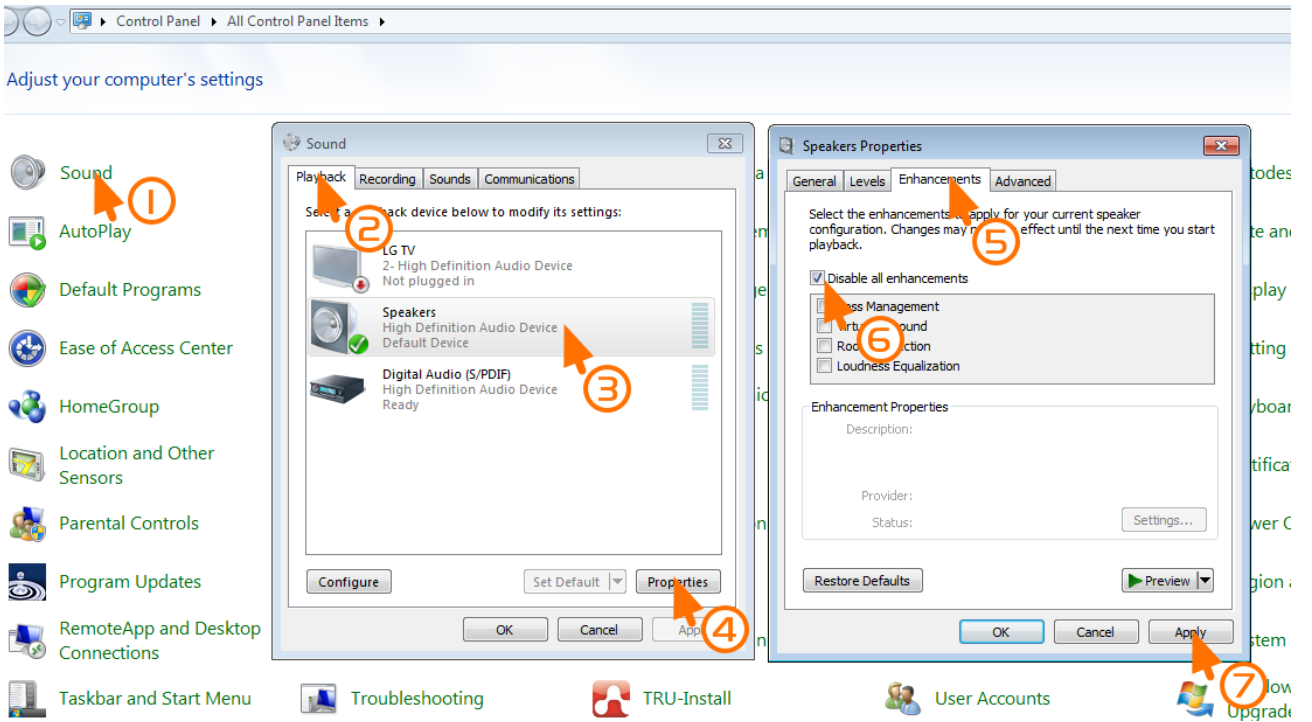
EdComm programlama kablosu

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

## EdWare'ı Yükleme

EdWare programlama yazılımı Windows, Mac, Linux, İOS, Android ve Rspberry Pi için ulaşılabilir. Yükleme dosyalarını ve işletim sisteminize EdWare'ı nasıl yükleyeceğinizin öğrenmek için [meetedison.com/downloads](http://meetedison.com/downloads) internet sitesini ziyaret ediniz.

Windows tabanlı bilgisayarlarda ses geliştirmeleri (ekolayzer) devre dışı bırakılmalıdır. Bunu yapmak için denetim masasına gidiniz ve aşağıdaki adımları takip ediniz.



Windows 7'deki tüm geliştirmeleri devre dışı bırakma

## Barkod Programlama (Çalışma yaprağı 1.2)

Öğrencileri öğrenecekleri şey ve robotik hakkında heyecanlandırmak için çalışma yaprağı 2'yi dağıtın. Çalışma yaprağı Edison'un tarayabileceği dört barkod içerir.

Öğrenciler, her bir barkodu tarar ve onların nasıl çalıştıklarını ve kullandıkları sensörleri betimler.

### Alkışla kontrollü sürüş

Edison düz bir yüzeye yerleştirilir ve oynat düğmesine (üçgen) basılır. Şimdi Edison'a yakın olacak şekilde alkışlayınız. Edison sağına dönecektir. Şimdi iki defa alkışlayınız ve Edison yaklaşık 30 cm ileri hareket edecektir. Parmağınızla Edison'a bir defa daha sonra iki defa hafifçe vurmaya da deneyiniz. Edison bir ses sensörüne sahiptir ve alkışa yanıt vermek için bu sensörü kullanır.

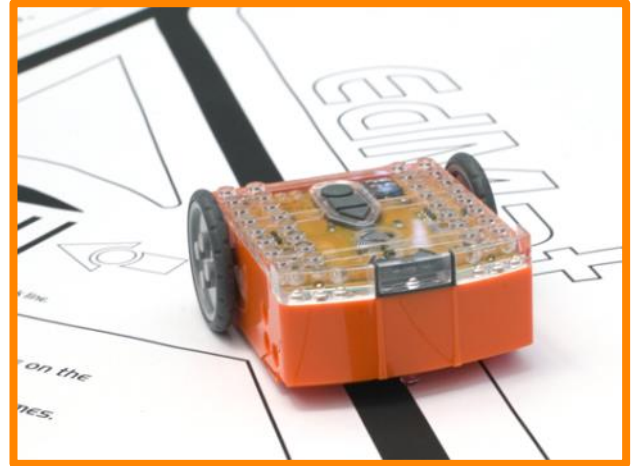
### Engellerden kaçınma

Oynat (üçgen) düğmesine basınız ve Edison'un bir nesneye yaklaşmasını ve daha sonra engele çarpmadan kaçınmasını izleyiniz. Edison yolundaki engelleri tespit etmek için kızıl ötesi ışınları kullanıyor.

### Çizgi takibi

Çizgi takip sensörü beyaz zemin üzerinde kalsın diye Edison'u çizginizin bir kenarına yerleştirin. Şimdi oynat (üçgen) düğmesine basınız ve Edison'un çizgiyi takip etmesini izleyiniz.

Edison'un çizgi takip sensörü yüzey üzerine ışık saçar ve daha sonra geri yansıyan ışığın miktarını ölçer. Beyaz, yüksek ışık okuması vererek çok ışık yansıtır, siyah ise düşük ışık okuması vererek az ışık yansıtır.



Edison'un çizgi takibi

Edison çizgiyi takip etmek için sürekli bir memnuniyetsizlik durumundadır. Çizgiden çıktığında, çizgiye gelmek için sağa döner fakat çizgide olduğunda çizgiden çıkmak için sola döner. Bu Edison'un çizgi kenarında ilerlemesine neden olur.

### Işık takibi

Bir el feneri ve güneş ışığı gibi gerçekten parlak bir ışığın uzağında düz bir yüzeye ihtiyacınız olacaktır. Oynat düğmesine basınız ve Edison'a el fenerini tutunuz. Edison parlak ışık kaynağını gördüğünde ona doğru ilerleyecektir. El fenerini hareket ettirerek Edisonu süreceğiniz yeri kontrol edebilirsiniz. *Bu davranış size bir şey hatırlatıyor mu?*

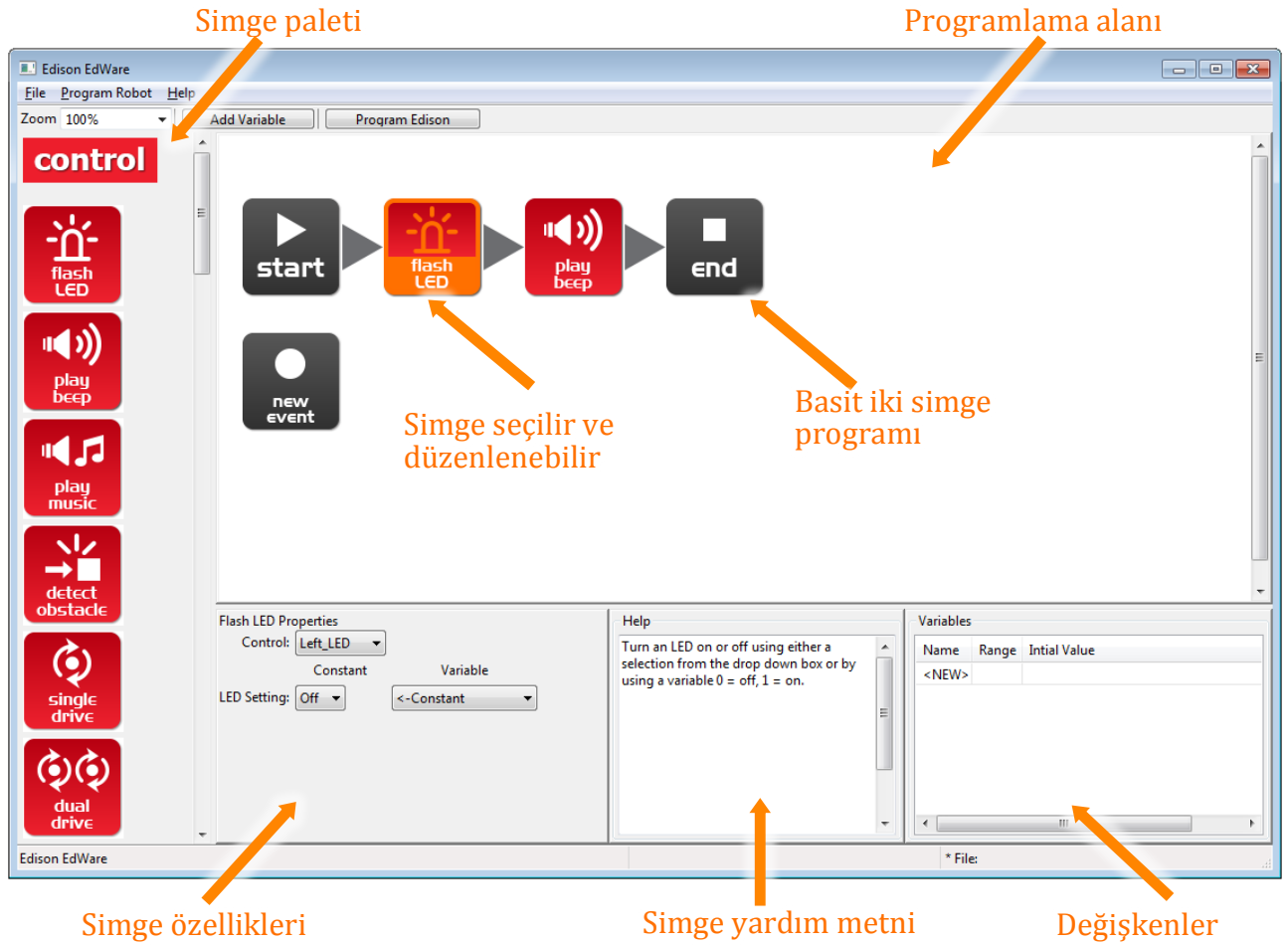
Bu Edison'un en ilgi çekici programlarından biridir. Çünkü Edison, gördüğümüz bazı uçan böceklerin davranışlarını taklit eder. Eminim parlak bir yaz gecesinde parlak bir ışığın etrafında dolanan güveler görmüşsünüzdür. Bu davranışlar ışığa yönelim olarak adlandırılır ve güneşe karşı büyüyen bitkilerde bulunur.

## EdWare'la Tanışın (Çalışma yaprağı 1.3)

Öğrenciler Edison ile yaptıkları etkileşimden heyecanlandıktan ve motive olduktan sonra EdWare'i kullanmaya başlayabilirler.

Aşinalık kazanmak için, öğrenciler yazılımı açar ve simgeleri soldaki simge paletinden hareket ettirir ve onları program alanına yerleştirir.

Burada EdWare'in Windows'ta görünüm şekli yer almaktadır. Diğer işletim sistemleri için EdWare çok benzer görünür.



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

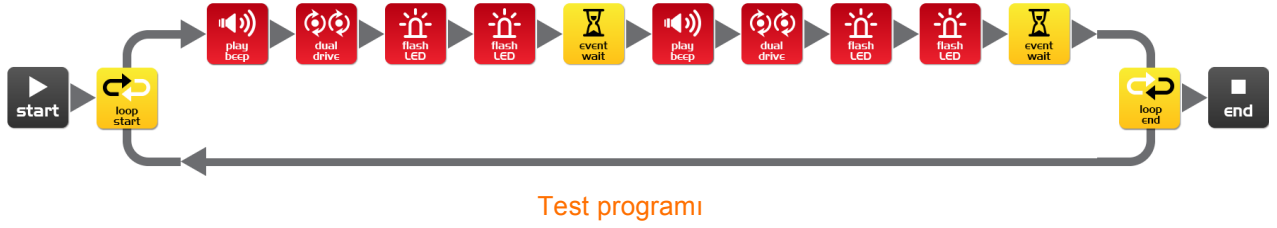
Değişkenler bölümü Edison'un belleğinin küçük parçalarını oluşturabileceğiniz ve görebileceğiniz yerdir. Daha fazlası Ders 9'da.

## Test programı (Çalışma yaprağı 1.4)

Öğrenciler şimdi test programını açabilir.

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

Test programı aşağıdaki gibi görünür.



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.

Öğrenciler şimdi robotlarına test programını indirebilir.

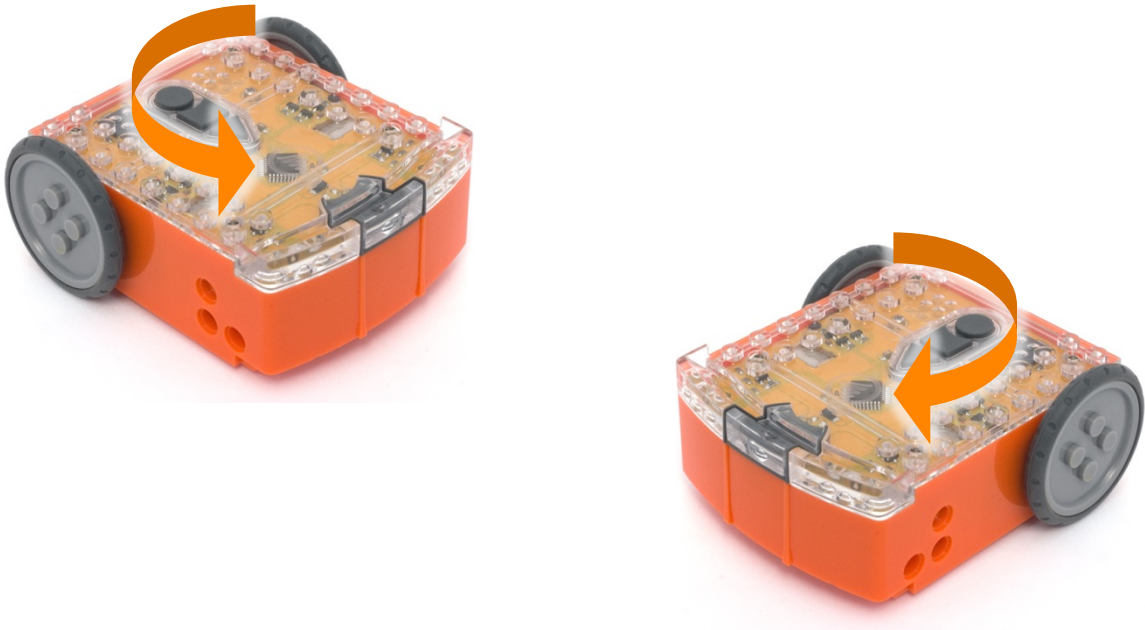
Cihazınıza EdComm kabloyu kulaklık soketine bağlayınız ve **sesi sonuna kadar** açınız. Edison'a EdComm kablonun diğer ucunu aşağıda gösterildiği gibi takınız.



Test Programını İndirmek için Bu Adımları Takip Ediniz:

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

Edison şimdi test programını çalıştıracak ve sağa ve sola dönecek, ışıkları yanıp sönecek ve bip sesi çıkartacaktır.



Öğrencilere, robotun oklarla gösterilen sırayla herbir simgeyi teker teker okuduğunu açıklayınız. Herbir simge robota yapılacak şeyi söyler.

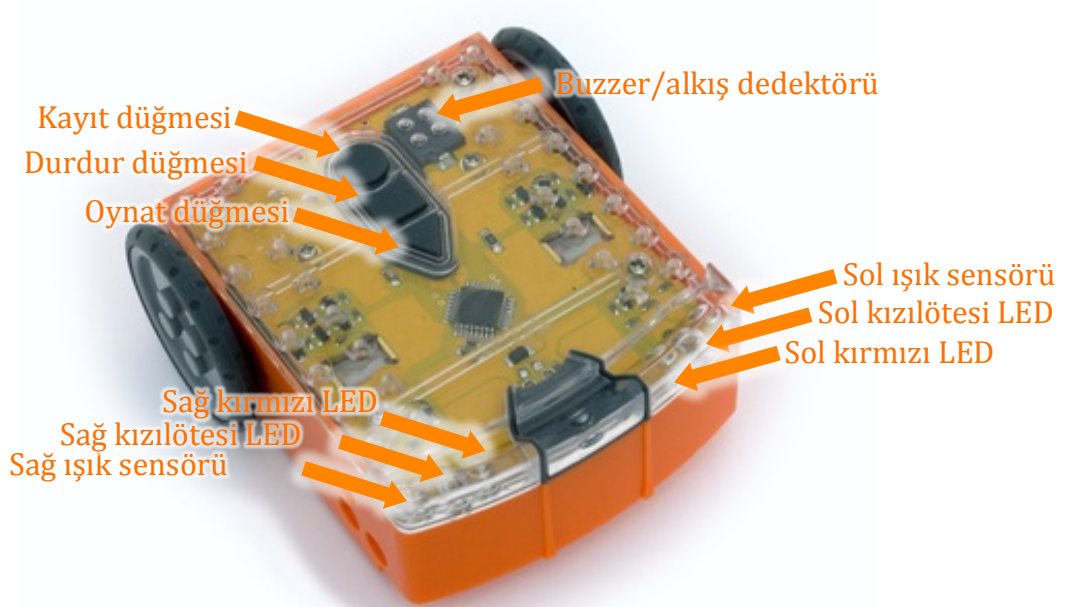
Ayrıca öğrencilere programın bilgisayar / tableten kabloyla Edison robotuna gittiğini açıklayın. Kablo kulaklık jakından gelen sesleri, robotun aldığı ve hafızasında depoladığı sinyale dönüştürür.

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

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

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

Çizgi takip sensörü daha önceden yüklenmiş özel barkodları okur.



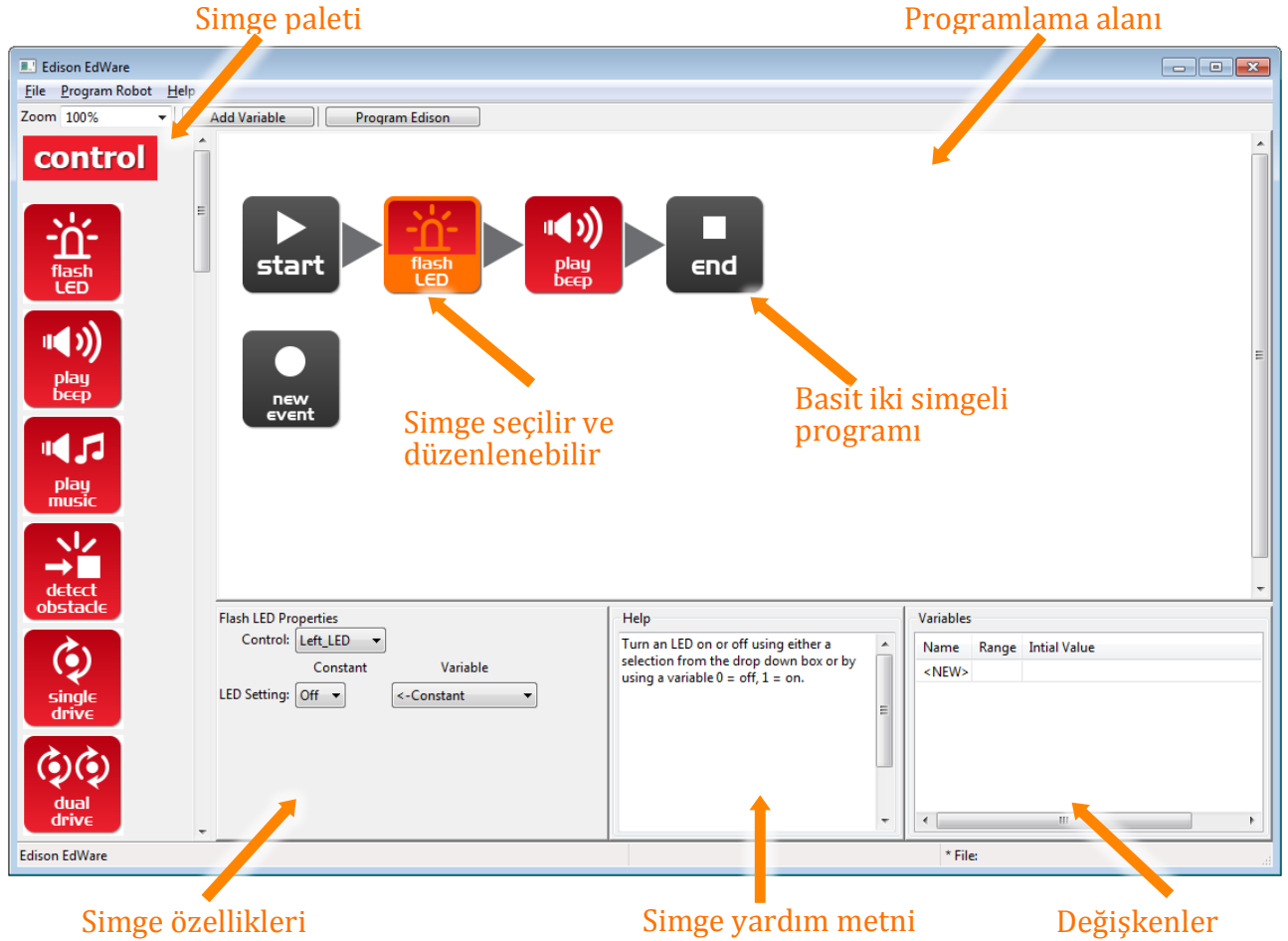
EdComm programlama kablosu



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?**

---

---

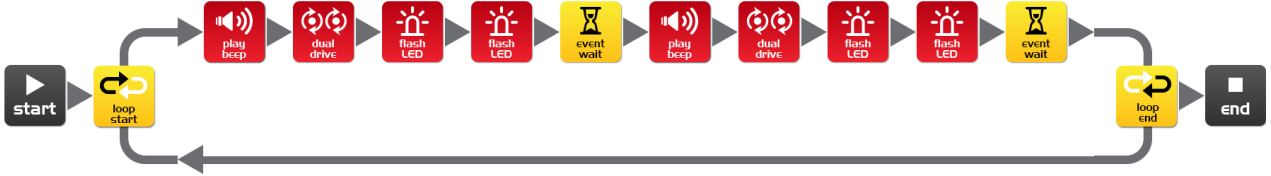
---

---

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?**

---

---

## DERS 2: ROBOT HAREKETİ – SÜRÜŞ

**Dizisel Programlamaya Giriş**– Öğrenciler dizideki talimat simgelerine robotun nasıl yanıt verdiğini ve zaman, hız ve uzaklık kavramlarını nasıl bir araya getirdiğini öğrenir.

Bu derste öğrenciler robotu belli bir mesafe ileri ve geri hareket ettirmek için temel kontrol sürüş simgelerini kullanacaktır.

### Program 1 – Robotu ileriye sürüş (Çalışma kağıdı 2.1)

Masa üzerinde çalışma kağıdını veya renkli bir şeridi kullanarak öğrencilerden robotun başlama işaretinden bitiş işaretine kadar belli bir mesafe ileri hareket ettiği bir program yazmalarını isteyiniz.

### Program 2 – Robotu geriye sürüş (Çalışma kağıdı 2.2)

Masa üzerinde çalışma kağıdını veya renkli bir şeridi kullanarak öğrencilerden robotun başlama işaretinden bitiş işaretine kadar belli bir mesafe geri hareket ettiği bir program yazmalarını isteyiniz.

### Program 3 – Robotu ileriye ve geriye sürüş (Çalışma yaprağı 2.3)

Masa üzerinde çalışma kağıdını veya renkli bir şeridi kullanarak öğrencilerden robotun başlama işaretinden bitiş işaretine kadar belli bir mesafe ileri ve geri hareket ettiği bir program yazmalarını isteyiniz.

### Program 4 – Hız oyunu (Çalışma yaprağı 2.4)

Öğrencilerden sürat, zaman ve uzaklık ile ilgili bir deney yapmalarını isteyiniz.

Öğrenciler kendi başlama ve bitiş noktalarını oluşturabilirler.

Robotun bitiş noktasına gidip geri gelebildiği en hızlı süre nedir?

Robotun bitiş noktasına gidip geri gelebildiği en yavaş süre nedir?

Yaratıcı öğrenciler, programlarına yanıp sönen LED ışıklar ve bipler ekleyebilir.

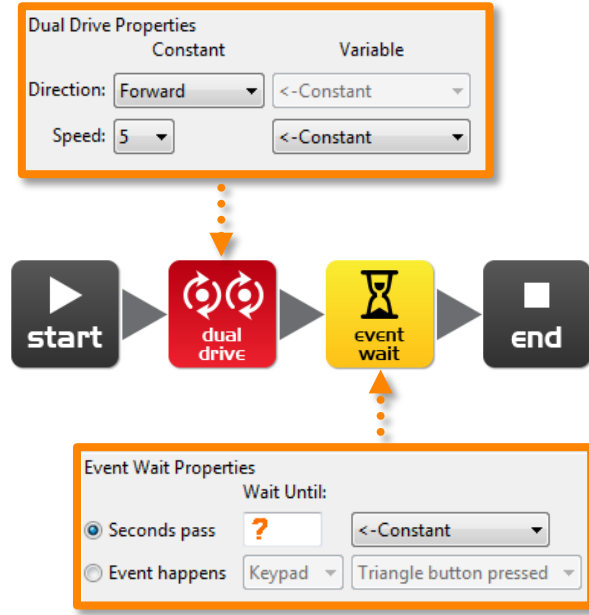
Herbir çalışma kağıdı öğrencilerden robotun ne yaptığını ve neden yaptığını betimlemelerini ister. Tipik bir cevap şöyledir:

*Robot ileri gider çünkü program ileri hareketi ifade eden bir sürüş simgesine sahiptir. Robot bitiş çizgisine hareket eder ve durur çünkü eylem bekle simgesindeki süre 0.35 saniyedir. Bu robotun o mesafeyi ne kadar sürede aldığıdır.*

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 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?

\_\_\_\_\_

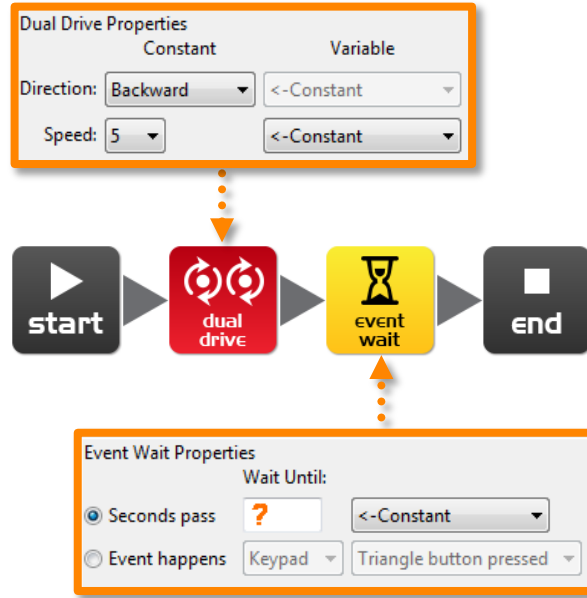
Robotun yaptığı işlemi ve nedenini anlatınız

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

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?

\_\_\_\_\_

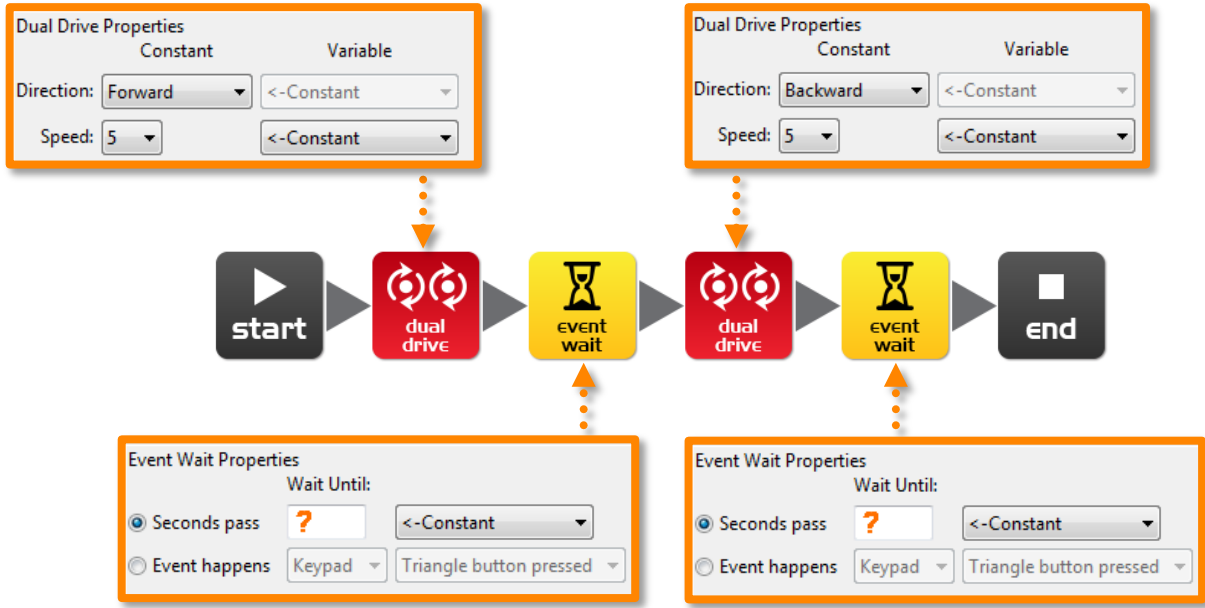
Robotun yaptığı işlevi ve nedenini anlatınız

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

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

---

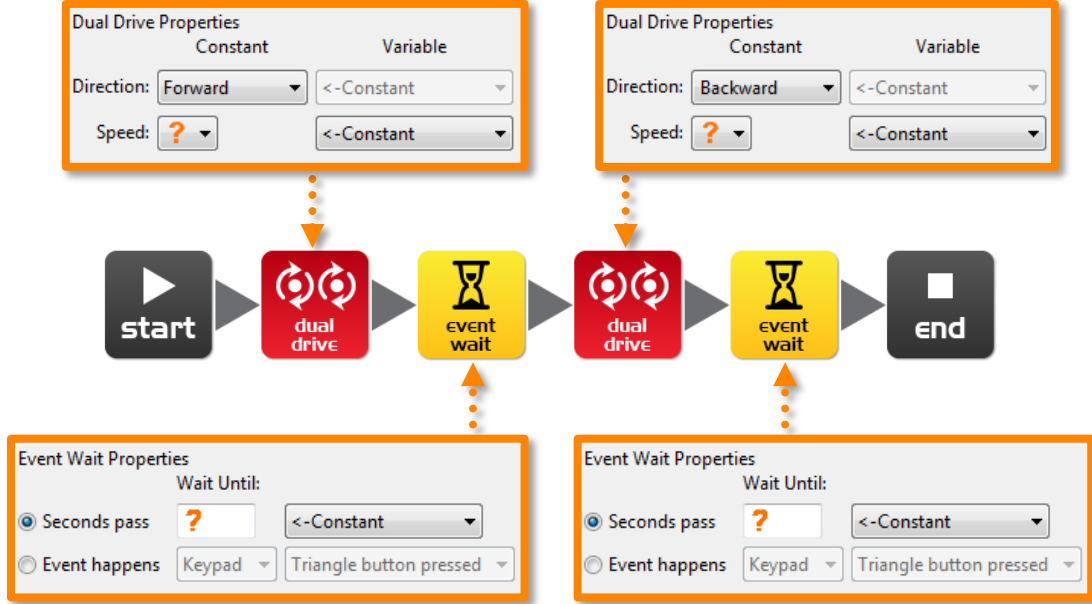
---

---

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**

## DERS 3: ROBOT HAREKETİ – DÖNDÜRME

Dizisel programlama ve temel geometri – Öğrenciler robotun zaman ve geometriye nasıl cevap verdiğini ve robotun hareket etme kontrolünü nasıl sağlayacaklarını öğrenir.

Bu derste öğrenciler robotu farklı açılarda ( $90^\circ$ ,  $180^\circ$  ve  $270^\circ$ ) döndürmek için basit kontrol simgelerini kullanacaklardır.

### Program 1 – Sağa döndürme (Çalışma yaprağı 3.1)

Masa üzerinde ya çalışma kağıdı 3.1 ya da renkli şerit kullanarak robotu 90 derece sağa çevirecek bir program yazmak için öğrencilere talimatlar verin.

### Program 2 – Sola döndürme (Çalışma yaprağı 3.2)

Masa üzerinde ya çalışma kağıdı 3.1 ya da renkli şerit kullanarak robotu 180 derece sola çevirecek bir program yazmak için öğrencilere talimatlar verin.

### Program 3 – Sağa daha sonra sola döndürme (Çalışma yaprağı 3.3)

Masa üzerinde ya çalışma kağıdı 3.1 ya da renkli şerit kullanarak robotu 90 derece sağa sonra 270 derece sola ( $90+180$ ) çevirecek bir program yazmak için öğrencilere talimatlar verin.

### Program 4 – Küçük labirent (Çalışma yaprağı 3.4)

Masa üzerinde ya çalışma kağıdı 3.1 ya da renkli şerit kullanarak bir küçük bir labirentte dolaşabilecek bir program yazmalarını söyleyin. Öğrenciler bu görevi tamamlamak için 2. ve 3. derslerden öğrendiklerini kullanacaklardır.

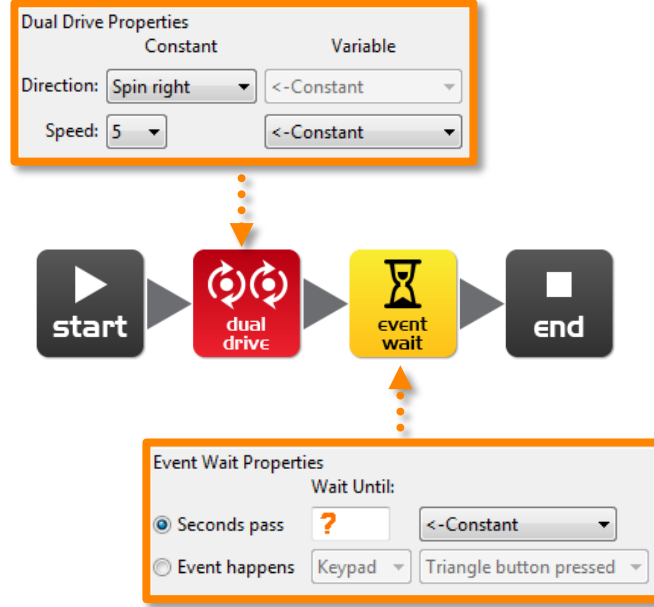
Herbir çalışma kağıdı öğrencilerden robotun ne yaptığını ve nicin yaptığını betimlemelerini ister. Tipik bir cevap şöyledir:

*Robot döner çünkü program sola dönüşü ifade eden bir sürüş simgesine sahiptir.  
Robot 90 derece döner çünkü eylem bekle simgesindeki süre 0.35 saniyedir. Bu robotun o mesafeyi ne kadar sürede aldığıdır.*

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?

\_\_\_\_\_

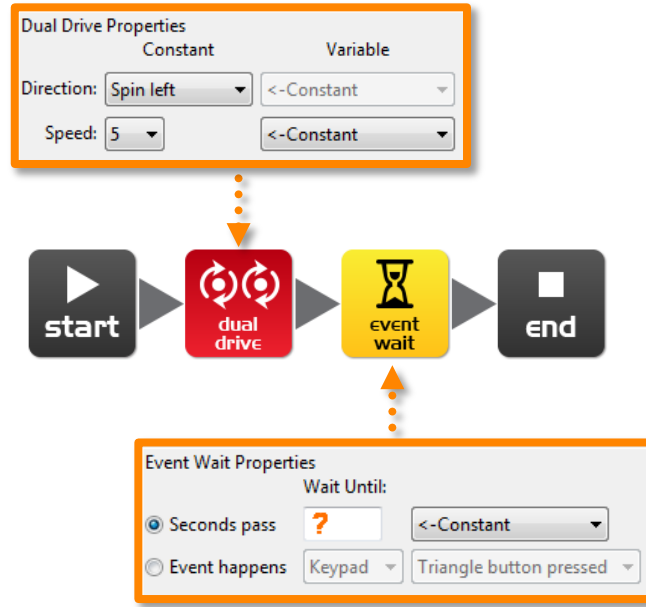
Robotun yaptığı şeyi ve nedenini anlatınız

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

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?

\_\_\_\_\_

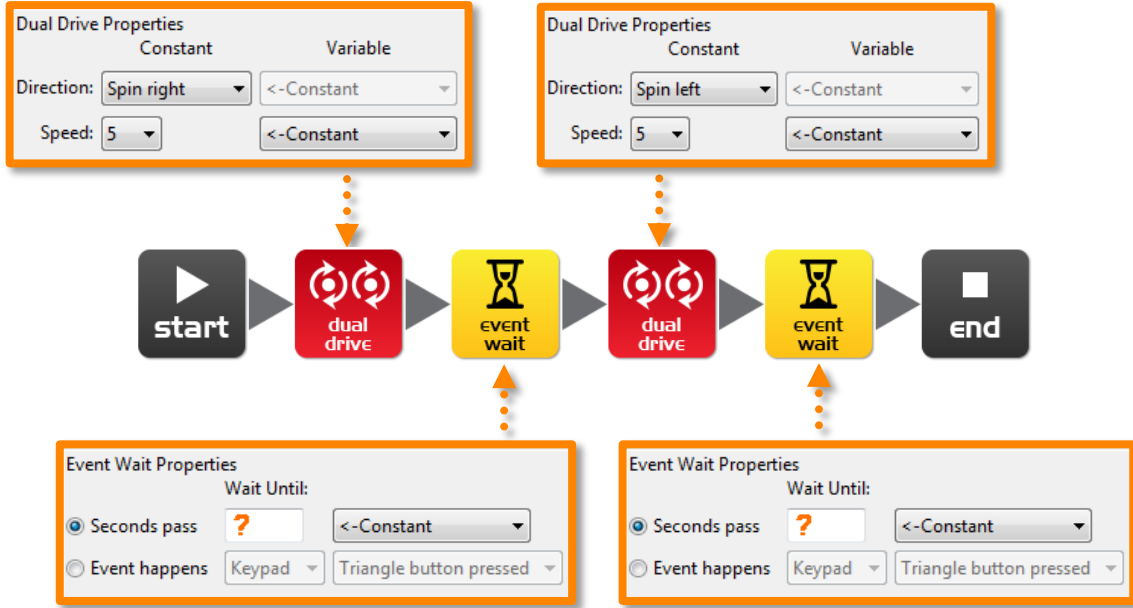
Robotun yaptığı şeyi ve nedenini anlatınız

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

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: \_\_\_\_\_

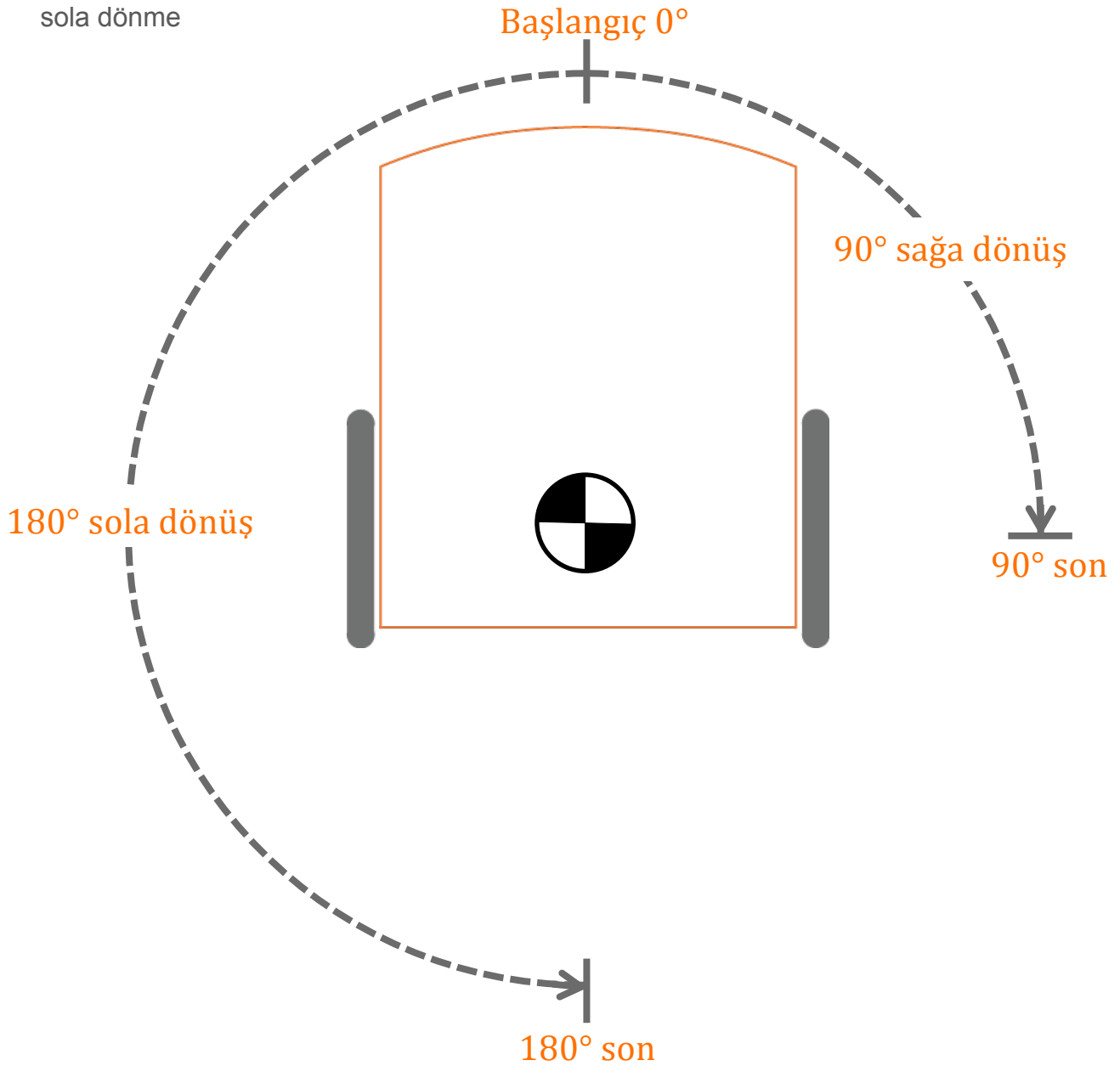
## Ders 3 Aktivite kağıdı 3.1 – Dönüş

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





## DERS 4: LABİRENT MEYDAN OKUMASI VE MEKSİKA DALGASI

**Öğrenmeyi pekiştirme** – Öğrenciler açık uçlu iki aktiviteyi gerçekleştirmek için 1. dersten 3. derse bilgilerini kullanırlar.

Bu derste öğrenciler iki programlama aktivitesi tamamlarlar.

### Program 1- Görev sürüşü

Öğrenciler robotları için bir görev sürüşü seçerler ve daha sonra bu görev sürüşü için bir program yazarlar. Bazı örnekler aşağıdadır.

- Kupa veya kalemlik gibi bir nesnenin etrafında hareket ettiriniz
- Masanın köşesinden çıkmaksızın kenarlarında hareket ettiriniz
- Robotun geçmesi için büyük bir kağıt parçası üzerine bir labirent oluşturunuz
- Robotun geçmesi için bloklar kullanarak bir labirent oluşturunuz

### Program 2 - Meksika dalgası robot stili

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

Öğrenciler robot hareketlerinin uygun bir dizisi ile kısa bir program yazarlar. Öğrenciler programlarında ışık ve ses kullanmaya da teşvik edilir.

Sınıf (veya öğretmen) en uygun hareketler ile programı seçer. Daha sonra bu program diğer tüm öğrenciler ile paylaşılır. Sonra her bir öğrenci programın başlangıcına bir eylem bekle simgesi ekler.

Bir Meksika dalgası etkisi oluşturmak için her bir öğrenci yeni bir eylem bekle simgesi içinde küçük miktarlarda farklı süreler programlar. Örneğin;

- Robot 1 – Eylem bekleme süresi 0 saniye
- Robot 2 – Eylem bekleme süresi 0.2 saniye
- Robot 3 – Eylem bekleme süresi 0.4 saniye
- Robot 4 – Eylem bekleme süresi 0.6 saniye
- Robot 5 – Eylem bekleme süresi 0.8 saniye
- Robot 6 – Eylem bekleme süresi 1.0 saniye
- Robot 7 – Eylem bekleme süresi 1.2 saniye
- Robot 8 – Eylem bekleme süresi 1.4 saniye
- Robot 9 – Eylem bekleme süresi 1.6 saniye
- Robot 10 – Eylem bekleme süresi 1.8 saniye

1'den 10'a kadar robotlar gecikme sürelerine göre bir çizgi üzerine yerleştirilir. Daha sonra herkes aynı anda basar ve performansı izlemek için bekler.

Not: Apple IOS işletim sistemi dosyaları içe aktarmaya izin vermez bu yüzden bu aktive eğer iPad kullanıyorsanız mümkün olmayabilir.

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?

---

---

---

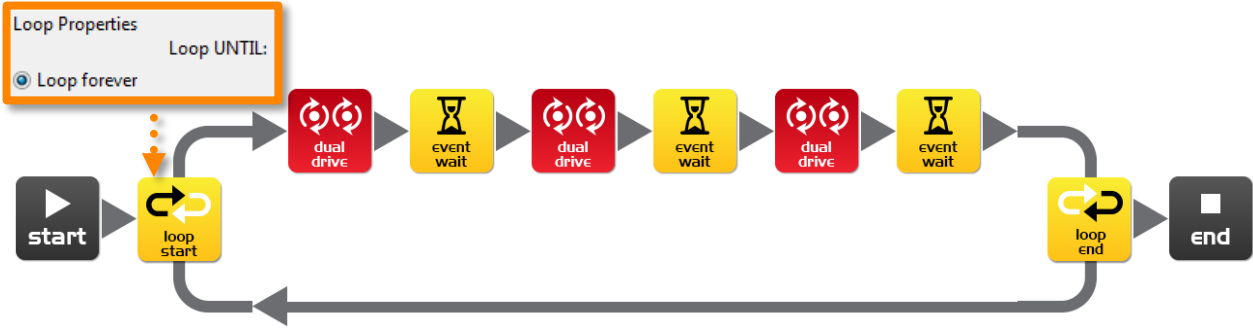
İsim: \_\_\_\_\_

## 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.**

---

---

---

## DERS 5: TASARIM ÖZETİ 1 – PROGRAMIM

**Yaratıcı düşünme ve problem çözme** – Öğrenciler kendi görevleriyle karşımıza çıkar ve robotun nasıl bir çözüm sağlayacağını kavramsallaştırmak için çözüm yolu bulur. Bu iki ders süresi alabilir. Alternatif olarak bu ev ödevi uygulaması olabilir. Öğrenciler kendi konusunu seçebilir, programının amacını ifade edebilir ve gerçek dünyada nerede kullanılabileceğini açıklayabilir.

### 1. Robotun çözebileceği bir problem veya eğlenceli bir müzik hareketi belirleyiniz

Gelecekte robotlar yardımcılarımız olacaktır. Uzun süredir robot elektrikli süpürelere sahibiz, dolayısıyla bu gelecek çok uzakta değildir. Bir öğrencinin robotu ne kadar yararlı veya eğlenceli görev yerine getirebilir?

Örnekler;

- Müziğe uyararak dans etmek – Robot öğrencilerin en sevdiği müziğe uyararak dans eder.
- Elektrikli süpürge – Öğrenciler geniş bir kâğıt üzerine bir oturma odası zemini çizebilir ve robotu tüm alanı süpürmek için programlayabilir.
- Güvenlik robotu– Öğrenciler bir güvelik alanı belirleyebilir (kağıt veya küçük bir nesne üzerine çizilmiş kare). Robotun işi alan etrafında sürüş yaparak güvenlik alanını korumaktır.

### 2. Problemi veya robotun yapması gereken hareketlerini tanımlayınız

Öğrenciler tasarım özeti çalışma kâğıdını kullanarak robot programlarının çözdüğü problemi ve robot programının problemi nasıl çözdüğünü tanımlar.

Örnek: **Problem:** *‘İnsanlar iyi dans edebilir fakat diğer dansçılarla senkronize olamazlar.’*  
**Robotum bu problemi şu şekilde çözecektir:** *‘Hızlı hareketler ve aynı dans hareketini diğer robotlarla beraber aynı anda yaparak.’* Bu insan dansını izlemekten çok daha eğlenceli olacaktır.

### 3. Programı yazınız ve test ediniz

Öğrenciler programlarını yazar ve kontrol eder.

### 4. Başarısızlık

Kaçınılmaz bir şekilde ilk deneme asla başarılı değildir. Bazı öğrenciler deneyişlerini sürdürmek için ekstra teşviğe ihtiyaç duyabilir. Başarısızlık programlamanın normal bir parçasıdır (veya herhangi bir mühendislik alanının). Ünlü Thomas Edison ampülü icat etmeden önce 10,000 kez başarısız oldu. Öğrenciler tasarım özeti çalışma yaprağına başarısızlıklarını açıklayabilir.

### 5. Yapmak istediğiniz şeyi ve kullanılan programlama simgelerini tanımlayınız

Öğrenciler tasarım özeti çalışma yaprağı 5.2’yi kullanarak programlarında kullanacakları bir takım simgeleri seçer ve bu simgelerin programda ne yaptıklarını açıklar. Öğrencilerin daha fazla programlama simgesi açıklaması için ek sayfalar yazdırın.

### 6. Demo

Öğrenciler robotlarının davranışlarını gösterir ve fikirleri, programları, karşılaştıkları problemleri ve onları nasıl çözüldüğü hakkında konuşur.

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 uyararak 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 ampülü keşfetmeden önce 10,000 kez başarısız olduğunu hatırlayınız. Bu yüzden **denemeye devam ediniz!**

---

---

---

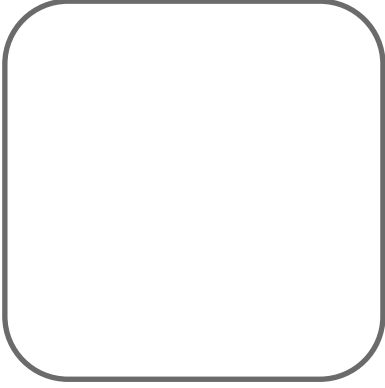
---

İsim: \_\_\_\_\_

## Ders 5 Tasarım özeti çalışma yaprağı 5.2 - Programım

### 5. Programınızdaki programlama simgelerinden bazılarını açıklayınız

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



Bu simgenin adı nedir? \_\_\_\_\_

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

---

---

---

---



Bu simgenin adı nedir? \_\_\_\_\_

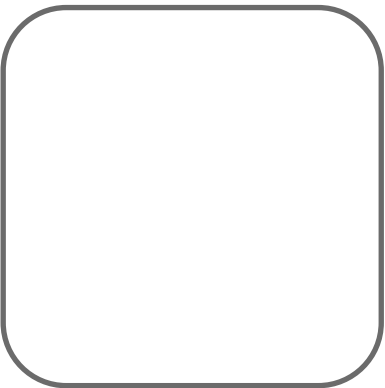
Bu simge ne yapar? \_\_\_\_\_

---

---

---

---



Bu simgenin adı nedir? \_\_\_\_\_

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

---

---

---

---

## DERS 6: ALKIŞ ALGISI

**Girişler (sensörler)** – Öğrenciler robotun dış uyarıcıya (alkış) nasıl yanıt vereceğini öğrenir.

Bu derste öğrenciler bir robotun dış uyarıcıları nasıl algıladığını ve robotun yanıtlarının nasıl programlanacağı ve robot davranışlarının nasıl oluşturulacağını öğreneceklerdir.

### Program 1 – Alkışa yanıt olarak LED yakma (Çalışma yaprağı 6.1)

Öğrenciler, programa devam etmeden önce bir olayın meydana gelmesini bekleyecek robot programlama konusunda bilgi sahibi olurlar. Bu durumda olay alkış gibi bir yüksek sestir.

### Program 2 – Alkışa yanıt olarak sürüş (Çalışma yaprağı 6.2)

Öğrenciler şimdi bir olaya (alkış) yanıt olarak robotu hareket ettirmek için daha önceki derslerde ki öğrenmelerini birleştirir.

### Program 3 – Alkışa yanıt olarak dans (Çalışma yaprağı 6.3)

Öğrenciler alkışa yanıt olarak kendi robot hareketlerini oluşturur. Öğrenciler iki seçeneğe sahiptir:

1. Çoklu dans manevrası ile bir robot dansı rutini oluşturunuz
2. Alkışa yanıt olarak bir engel boyunca robotu hareket ettiren bir program oluşturunuz.

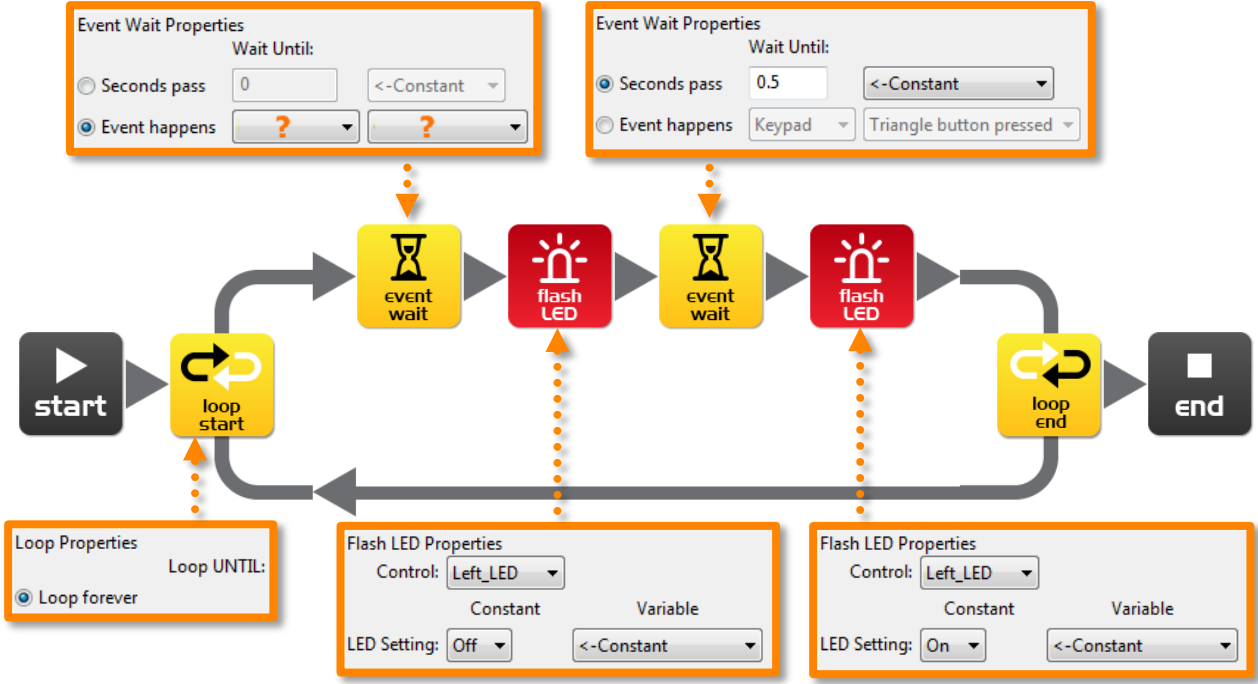
### Eglenceli Sınıf Aktivitesi

En iyi dans rutinini seçiniz ve diğer tüm öğrenciler bu rutin ile robotlarını planlayınız. Daha sonra tüm robotları beraberce yerleştiriniz ve öğrenciler tüm robotların birlikte dansını görmek için uygun zamanda alkışlayın. Alkışları 1, 2, 3, 4 ile hesaplamamız gerekecektir. Alkış 5'te başlar ve 6, 7, 8 için devam eder.

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?

\_\_\_\_\_

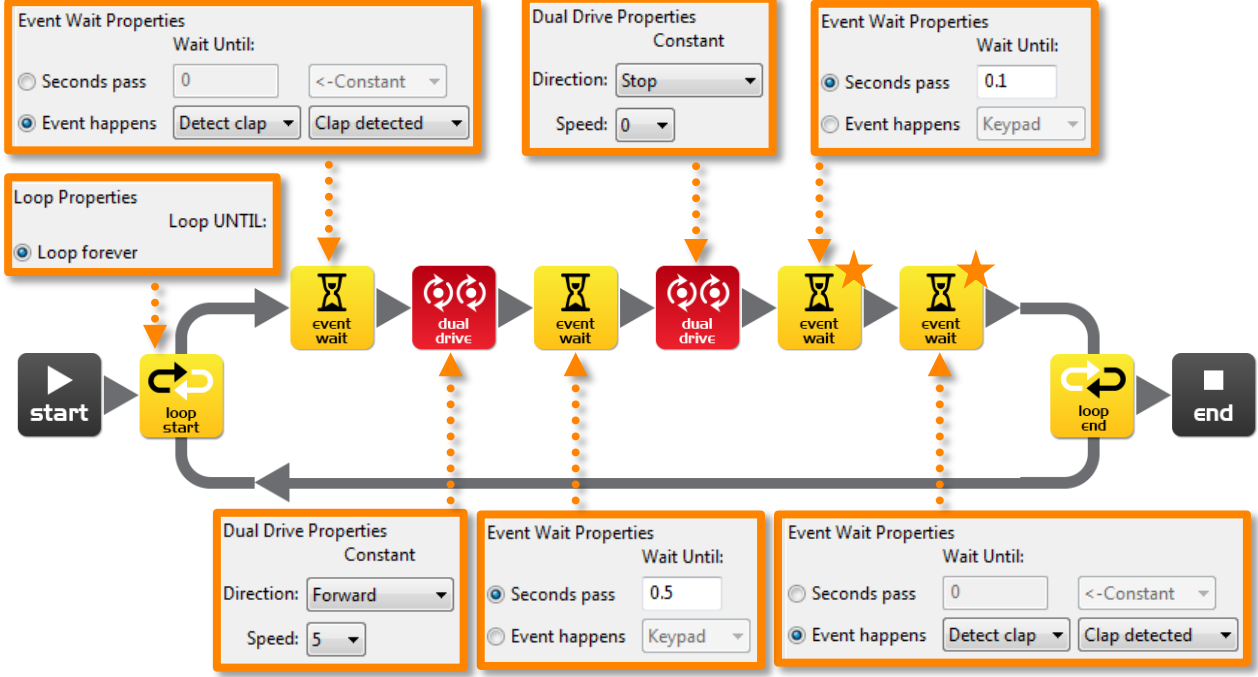
Bu programdaki döngü simgelerinin amacı nedir ve döngü simgeleri oralarda olmasaydı ne olurdu?

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

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 \_\_\_\_\_

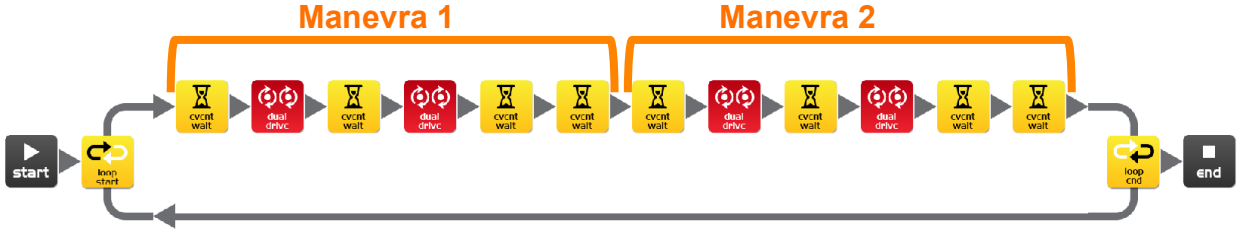
Deney 2: yön \_\_\_\_\_, süre \_\_\_\_\_

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.

## DERS 7: ENGEL ALGILAMA

Engelleri tespit etme kavramı ve yapay zekaya giriş – Öğrenciler robotun çevresindeki engellere yanıt olarak karar vermesi (yapay zeka) için robotu programlar.

### Kızıl ötesi ile engel algılamayı anlama (Çalışma yaprağı 7.1)

Öğrenciler, robotun kızılötesi engel tespit teknolojisine ve robotun bu yolla engelleri nasıl tespit edebildiğine aşina olurlar.

### Program 1 – Bir engeli algılama ve durma (Çalışma yaprağı 7.2)

Öğrenciler, yolunda tespit edilen bir engele kadar robotu hareket ettiren bir program yazar. Robot daha sonra bir çarpışmadan kaçınarak durur.

### Program 2 – Bir engeli algılama ve kaçınma (Çalışma yaprağı 7.3)

Öğrenciler, yolunda tespit edilen bir engele kadar robotu hareket ettiren bir program yazar. Robot daha sonra bir çarpışmadan kaçınarak uzaklaşır.

### Program 3 – Döngüde bir engeli algılama ve kaçınma (Çalışma yaprağı 7.4)

Öğrenciler, yolunda tespit edilen bir engele kadar robotu hareket ettiren sürekli bir döngü içeren program yazar. Robot daha sonra bir çarpışmadan kaçınarak uzaklaşır. Daha sonra robot döngüden dolayı kaçınmak için diğer engelleri arayarak hareket etme için devam eder.

### Program 4 – Sağ ve sol engel algılama (Çalışma yaprağı 7.5)

Öğrenciler sürekli bir döngü içeren ve sağda ve solda bir engelin olup olmadığına yanıt vermek için yeteneğe sahip olan bir program yazar. Öğrenciler bunu yapmak için Eğer simgesi kullanır.

Eğer simgesinin önemi insan rehberliği olmaksızın robota karar verme yetisi verir. Bir robotta bu meydana geldiğinde bu şu an otonom robot olarak isimlendirilmektedir çünkü robot yapay zekaya sahiptir. Bu etkileyici bir kavramdır. Robot şimdi canlı mı?

Bu ders sürücüsü olmayan ve inşalar, binalar ve diğer arabalarla çarpışmalardan kaçınmak için sensörler kullanan robot arabalar hakkında tartışmaya teşvik etmek için de kullanılabilir

Eğer engel tespiti çok hassassa veya yeterince hassas değilse öğrenciler aşağıdaki sayfada yer alan talimatları kullanarak kalibre edebilir.

İsim: \_\_\_\_\_

## Engel algılayıcısını kalibre ediniz

Edison'un engel algılayıcı sisteminin hassasiyetinin kontrol edebilirsiniz. Sistem daha hassas yapılırsa Edison daha uzaktaki engelleri tespit edebilir ve daha az hassas yapılırsa sadece çok yakındaki engelleri tespit edecektir.

### Barkod okuma

1. Edison'u sağ kenarda barkoda bakacak şekilde yerleştiriniz
2. Üç defa kayıt düğmesine basınız
3. Edison ileri hareket edecek ve barkodu tarayacaktır.



Barkod – Engel algılayıcısı kalibre et

### Maksimum hassasiyeti ayarla

İlk olarak yukarıdaki barkodu tarayın daha sonra oynat düğmesine basınız. Edison şimdi kalibrasyon modundadır. Edison'un önündeki herhangi bir engeli kaldırın.

1. Sol kırmızı LED yanıp söne kadar tekrarlayan bir şekilde oynat düğmesine basınız (bu hassasiyeti artırır).
2. LED'in yanıp sönmeye tamamlanana kadar tekrarlayan bir şekilde kayıt düğmesine basınız (bu hassasiyeti azaltır)
3. Sağ tarafı kalibre etmek için durdur düğmesine basınız
4. Sağ kırmızı LED yanıp söne kadar tekrarlayan bir şekilde oynat düğmesine basınız şimdi LED'in yanıp sönmeye tamamlanana kadar tekrarlayan bir şekilde kayıt düğmesine basınız.
5. Durdur düğmesine basınız ve kalibrasyon tamamdır.

### Özel Hassasiyet

Edison'un önüne bir engeli yerleştirerek ve 1'den 5'e adımları tekrarlayarak engellerin belirlendiği uzaklığı ayarlayabilirsiniz.

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.



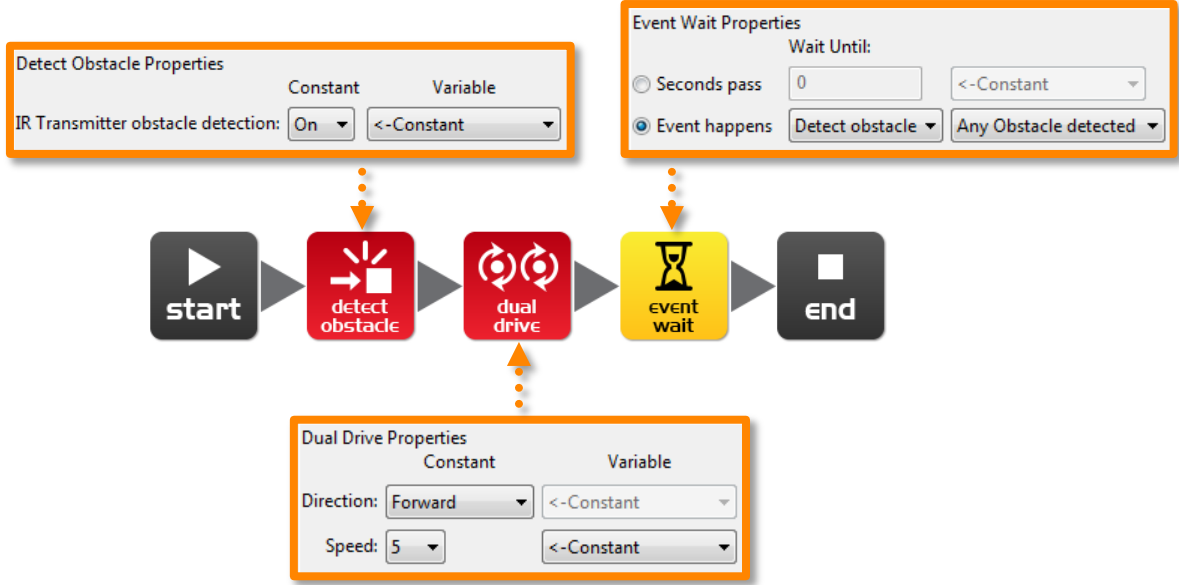
Bu engel için yayılan ve yansıyan kızılötesi ışıkları çiziniz.



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?

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

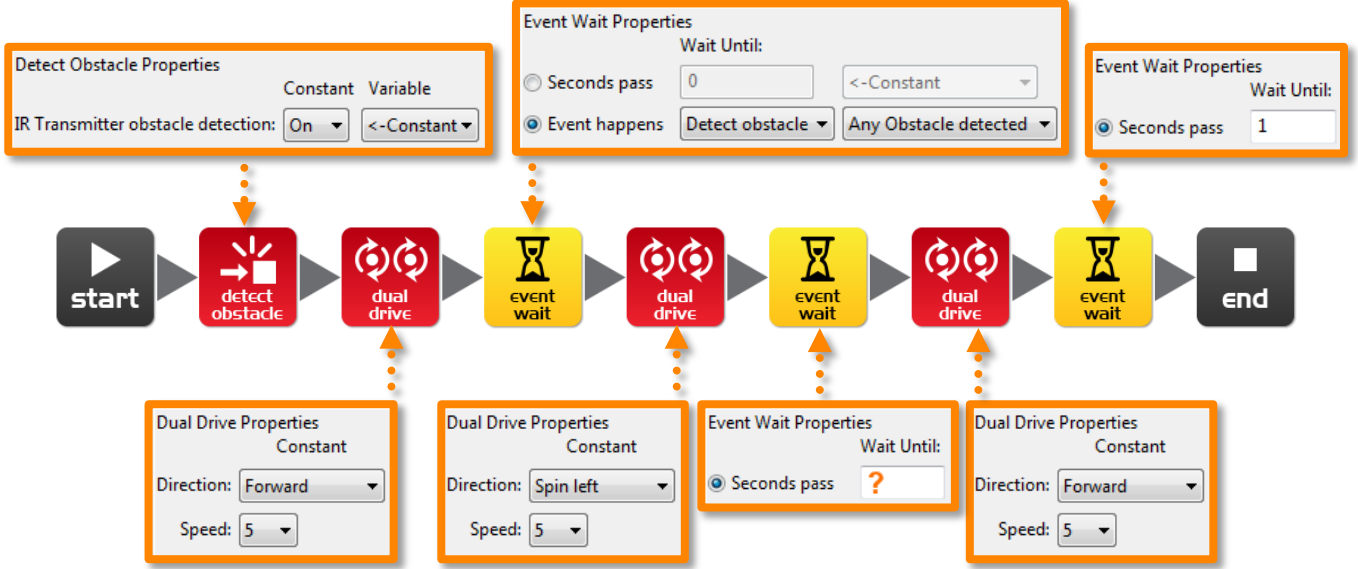
Bu tür algılama teknolojisinin başka nerelerde kullanılabileceğini düşünüyorsunuz?

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

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

---

---

---

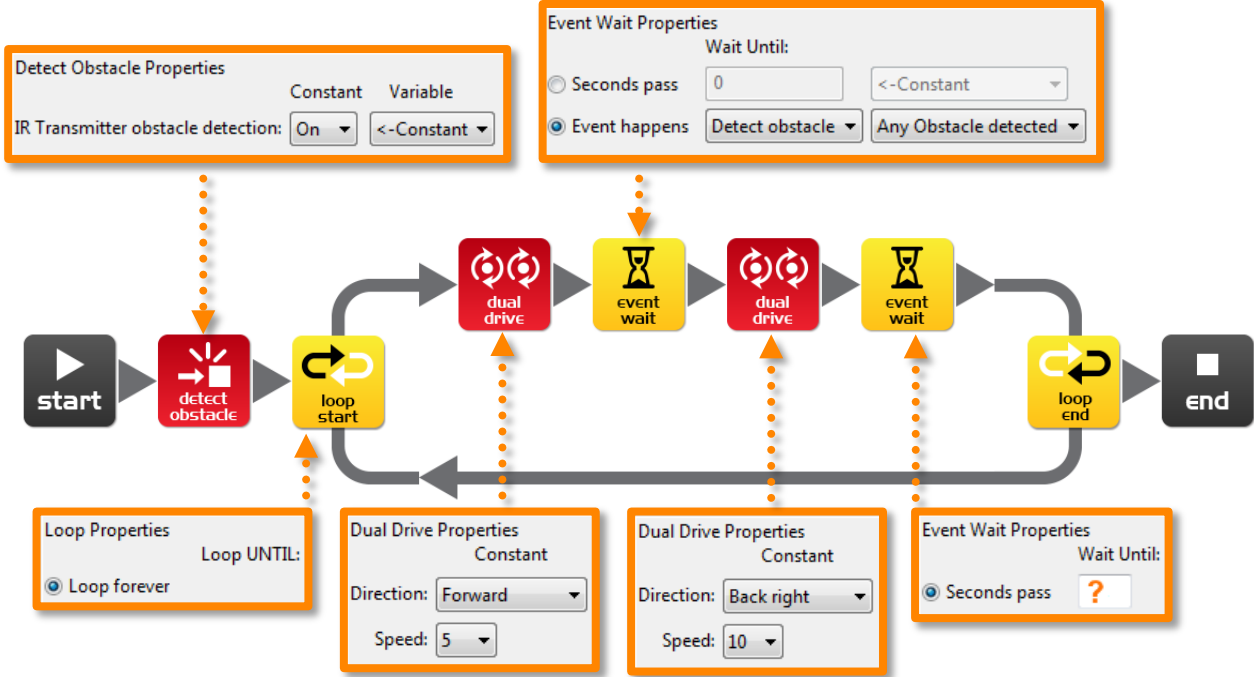
---

---

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 miktarda 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?

\_\_\_\_\_

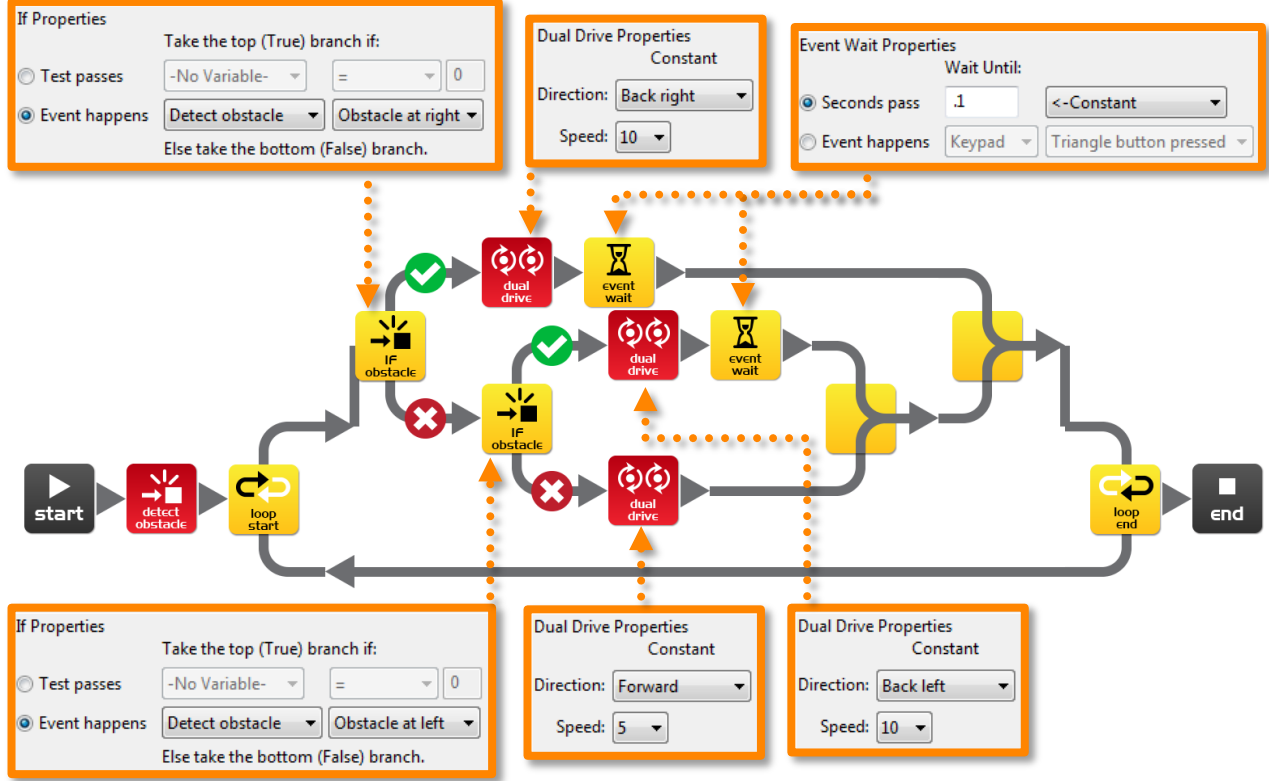
Bu süre ayarı niçin en iyisidir? Robot onu daha iyi yapmak için farklı olarak ne yapar?

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

İsim: \_\_\_\_\_

## Ders 7 Çalışma yaprağı 7.5 – Sağ ve sol engel algılaması

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



Yukarıdaki robotta Eğer simgelerini kullanıyoruz. Bunlar çok önemli simgelerdir çünkü insan rehberliği olmaksızın robota karar verme yetisi verirler. Bir robotta bu meydana geldiğinde o zaman robot otonom robot olarak isimlendirilmektedir çünkü robot yapay zekaya sahiptir.

Eğer simgesi bir şartın doğru olup olmadığını sorar. Eğer sonuç doğru ise program tik işaretli yoldan ilerler. Eğer sonuç yanlış ise program çarpı işaretli yoldan ilerler

Yukarıdaki programın bir engelin bulunduğu yere bağlı olarak alabileceği üç farklı yol vardır. **Bu üç yolun robotun ne yapmasına neden olacağını kendi sözcüklerinizle açıklayınız.**

Tespit edilen engel yok: \_\_\_\_\_

Sağda tespit edilen engel: \_\_\_\_\_

Solda tespit edilen engel: \_\_\_\_\_

**Robot kararlar verebildiği için canlı mı? Neden böyle düşünüyorsun?**

\_\_\_\_\_

## DERS 8: ÇİZGİ ALGILAMA VE TAKİP

Endüstri robotik davranışlara benzer – Öğrenciler, gelişmiş otomatik fabrikalarda ve ambarlarda kullanılan benzer, temel robot algılama ve kontrolü hakkında bilgi edinirler.

### Çizgi takip sensörünü anlama (Çalışma yaprağı 8.1)

Öğrenciler robotun çizgi takip sistemine ve sürüş yüzeyinin siyah veya beyaz olup olmamasına göre robotun nasıl tespit edebildiğine aşinadır.

### Program 1 – Siyah bir çizgiye kadar sürüş (Çalışma yaprağı 8.2)

Öğrenciler robotun siyah bir çizgiyle karşılaşınca kadar robotu hareket ettiren bir program yazar. Öğrenciler farklı renklerin belirlenmesi hakkında soruları da doğrulayabilir.

### Program 2 – Bir sınır içerisinde sürüş (Çalışma yaprağı 8.3)

Öğrenciler bir sınır içerisinde robotu hareket ettiren bir program yazar. Çizgi takip sistemi sınırı tespit eder ve robot etrafında döner. Öğrenciler çalışma yaprağı 2'yi kullanabilir veya geniş bir kağıt ve beyaz bir masanın üzerine siyah bir işaret veya siyah elektrik bandı kullanarak kendileri oluşturabilirler.

### Program 3 – Bir çizgiyi takip etme (Çalışma yaprağı 8.4)

Öğrenciler siyah bir çizgiyi takip eden program yazar. Öğrenciler çalışma yaprağı 2'yi kullanabilir veya geniş bir kağıt ve beyaz bir masanın üzerine siyah bir işaret veya siyah elektrik bandı kullanarak kendileri oluşturabilirler.

### Video – İnsanların başvurusu gerekmiyor

Çizgi takibi bugün pek çok otomatik fabrikalarda ve ambarlarda kullanılan çok temel bir robotik davranıştır fakat ya gelecekte! Geleceğin çalışma yeri, daha fazla robotla nasıl görünecek? Hatta var olacak mı? Aşağıdaki video robotların yaygınlaşmasının insanların geleceği için ne anlam ifade ettiği hakkında sınıf içinde tartışmayı tetikler.

<https://www.youtube.com/watch?v=7Pq-S557XQU> (15 dakika)



İsim: \_\_\_\_\_

## Ders 8 Çalışma yaprağı 8.1 – Çizgi takip sensörü

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

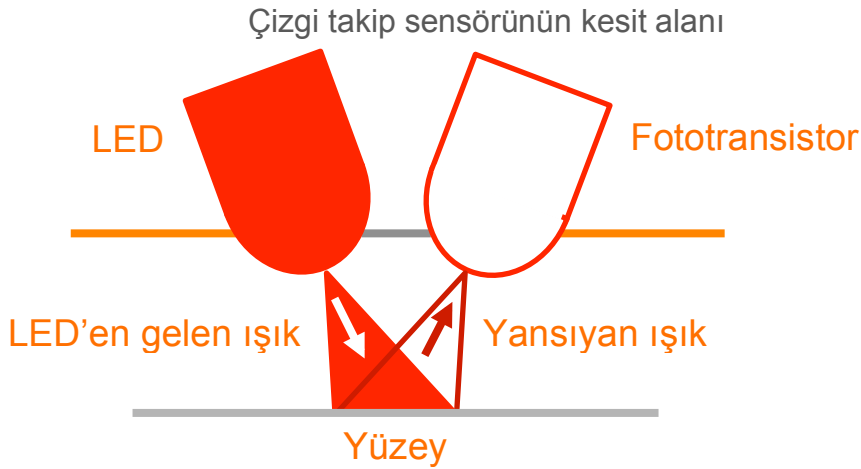
- 1) Kırmızı ışık yayıcı diyot (LED)
- 2) Fototransistör (ışık sensörü)

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 gördüğü 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.

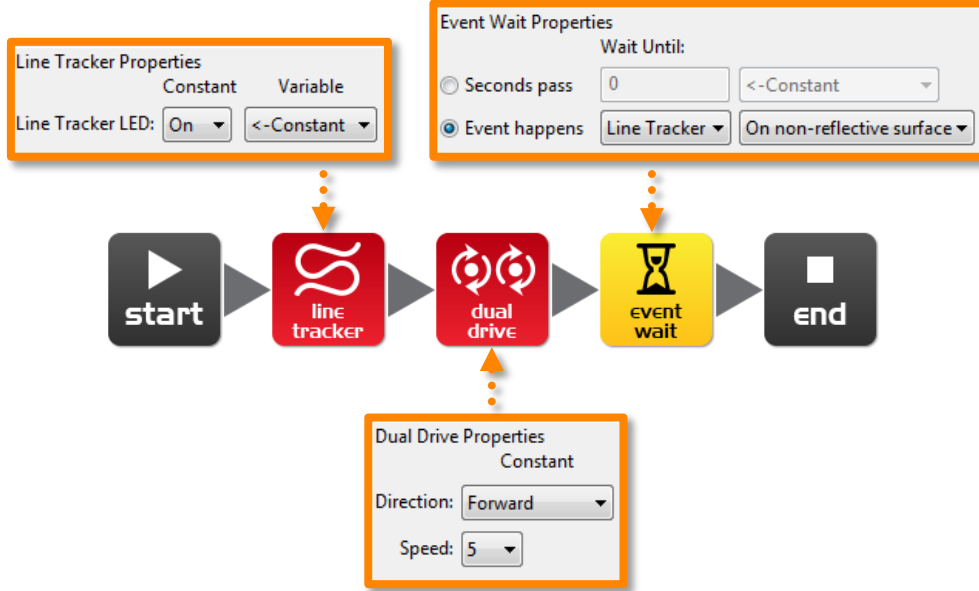
**Çizgi takipçisinin aşağıdaki yüzey renklerine, yansıtıcı veya yansıtıcı olmamasına nasıl tepki vereceğini düşünüyorsunuz? (İpucu: Işık kırmızı)**

**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 çiziniz. 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?

---

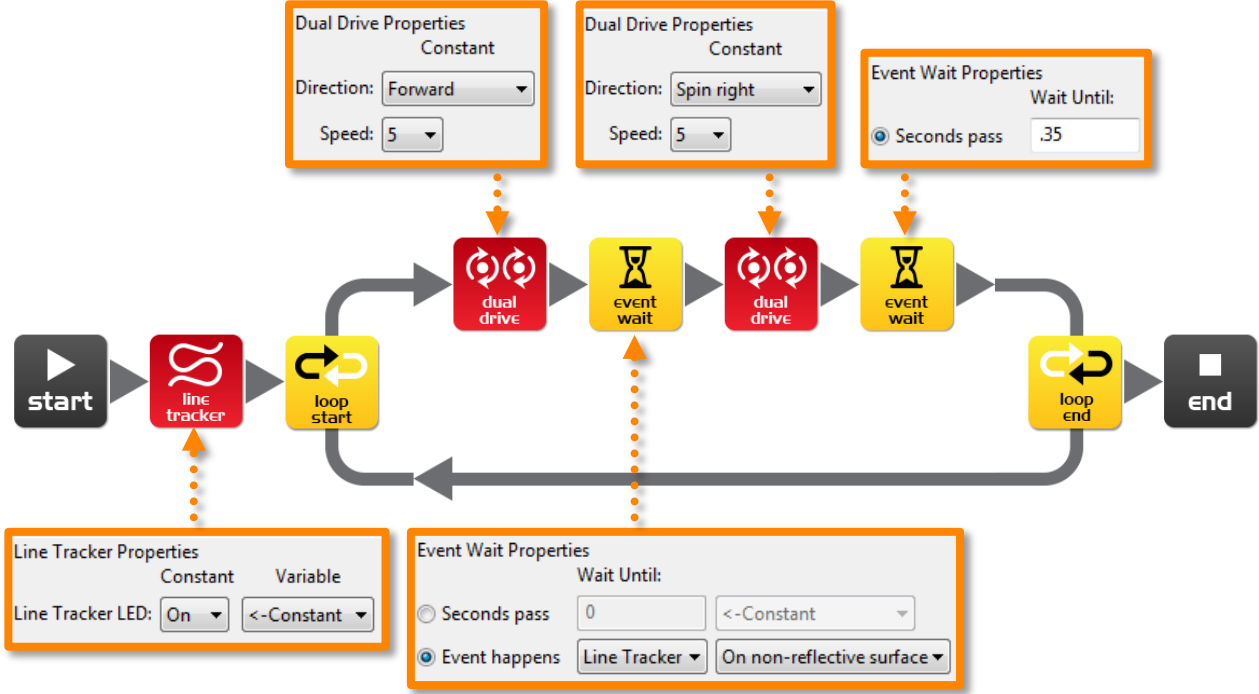
---

---

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?

\_\_\_\_\_

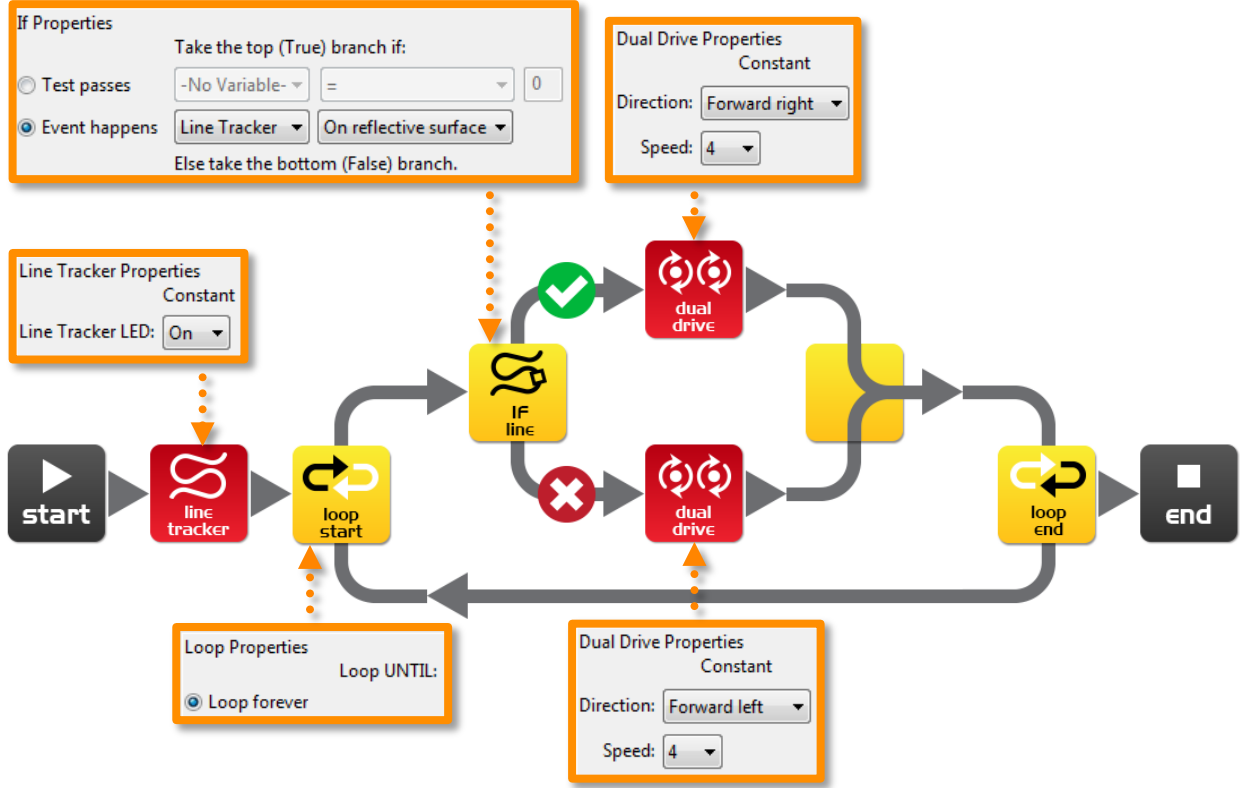
Robor çok hızlı ilerlediğinde ne olur?

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

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?

---

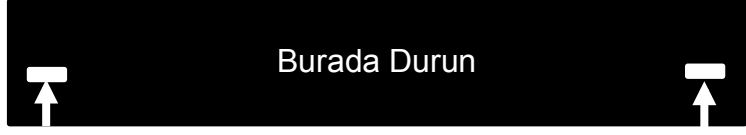
---

---

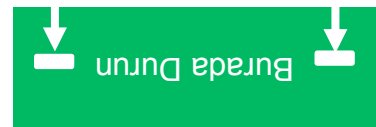
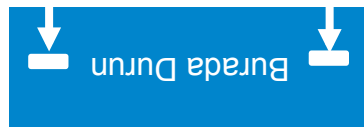
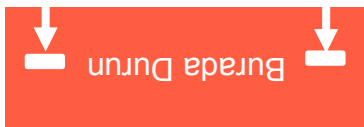
İsim: \_\_\_\_\_

## 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?

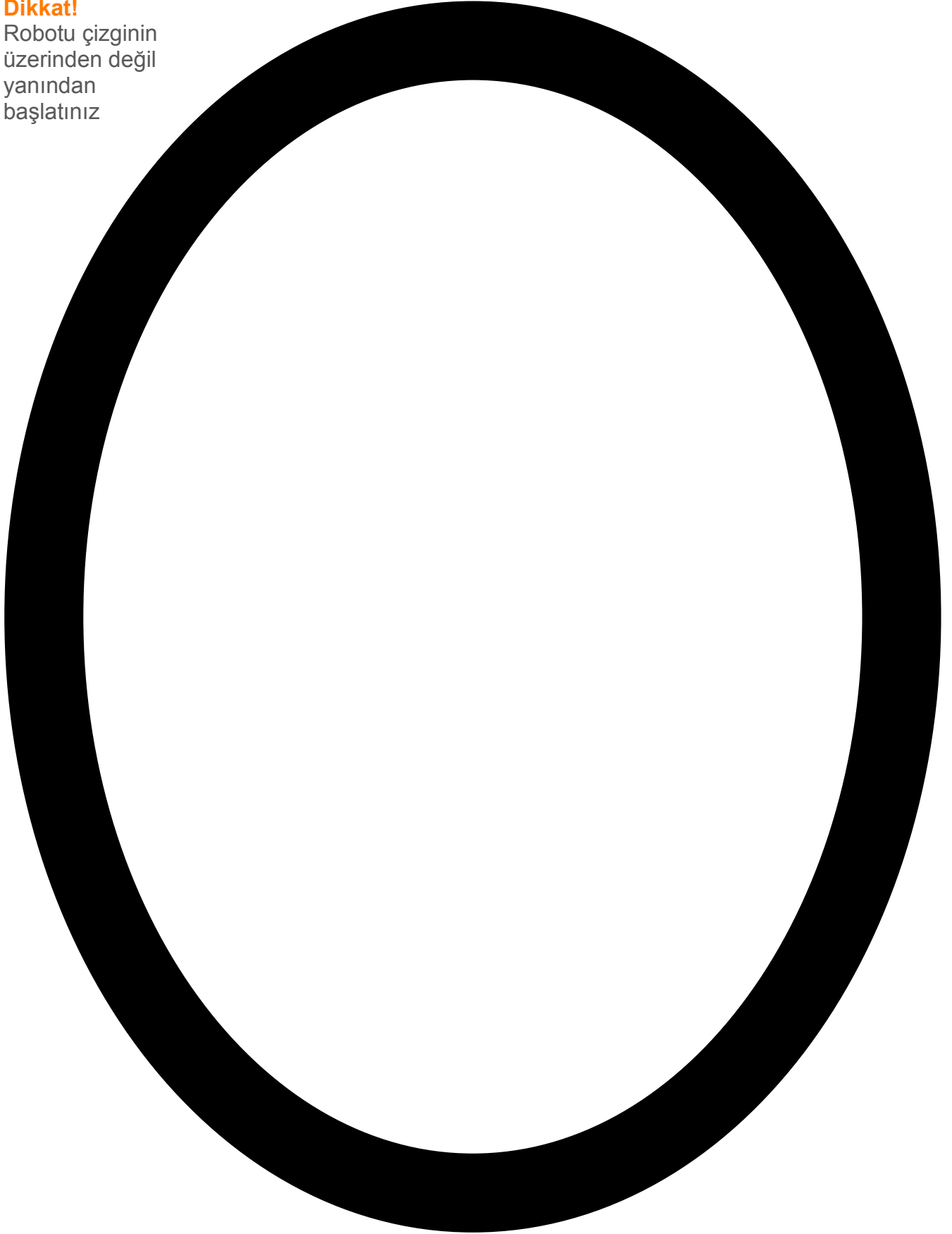


İsim: \_\_\_\_\_

## Ders 8 Aktivite kağıdı 8.2 – Çizgi takip sensörü

### Dikkat!

Robotu çizginin  
üzerinden değil  
yanından  
başlatınız



## DERS 9: IŞIĞA TEPKİ

**Çevresel ölçüm ve matematik programlama** – Öğrenciler robotların davranışlarını kontrol etmek için ışık seviyeleri ölçümü, hafızada saklama ve matematiksel işlem gerçekleştirme hakkında bilgi edinirler.

### Değişkenleri anlama (Çalışma yaprağı 9.1)

Öğrenciler programlamada bir değişkenin ne olduğu ve Edison robotta nasıl ayarlanabileceği hakkında bilgi edinirler. Öğrenciler aynı zamanda iki farklı boyutlu değişken (bayt ve kelimeler) hakkında bilgi edinir. (16 bit -32,767, +32,767)

### Program 1 – Işık seviyesi alarmı (Çalışma yaprağı 9.2)

Öğrenciler odadaki ışıkların açıldığı zaman bir alarm sesi çıkartan program yazarlar. Bu program bir 'büyük' (>) karşılaştırması kullanarak temel matematik ve değişkenlerin kullanımını tanıtır.

### Program 2 – Otomatik ışıklar (Çalışma yaprağı 9.3)

Öğrenciler ışık seviyelerini izleyerek robotu hareket ettirecek bir program yazarlar. Eğer robot karanlık bir alanda hareket ederse, ön ışıklar otomatik bir şekilde açılır. Bu program bir 'küçük' (<) karşılaştırması kullanır.

### Program 3 – Işığın takip etme (Çalışma yaprağı 9.4)

Öğrenciler el feneri gibi parlak bir ışığa karşı robotu hareket ettirecek bir program yazar. Bu program, çıkarma ve ardından 'küçük' (<) karşılaştırması kullanarak daha fazla bilişimsel matematiği tanıtmaktadır.

### Robotikte hayvan davranışları

Yukarıdaki ışık takip programı, sıcak bir gece sokak lambasına çekilen bir güveninkine çok benzer bir robot davranışı gösterir.

Bu hem böcek hem de robotların zekası hakkında iyi bir tartışma sağlayabilir.

**Niçin ışığa çekilen bir böcek canlıdır fakar robot değildir?**

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.

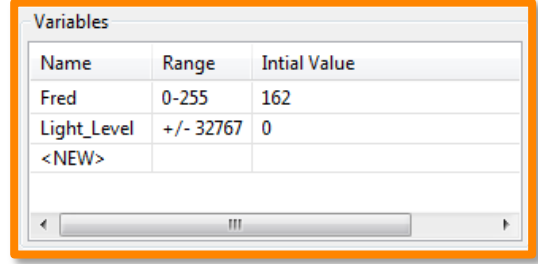
Değişkenler 10, 106, 1.482 vb gibi sayıları depolar ve matematiksel işlem yapmak için bilgisayar programına izin verir. Bu bilgisayarların çok iyi olduğu bir şeydir.

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.

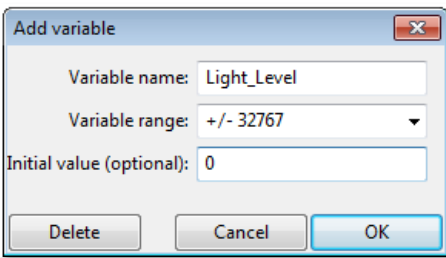
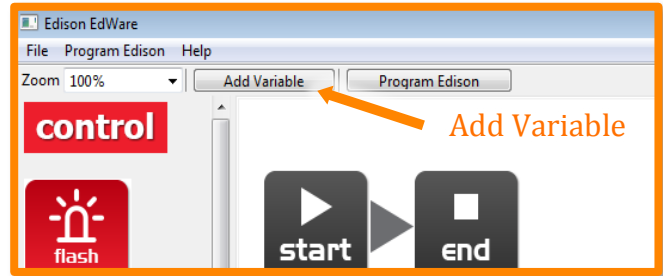
Değişkenleri kolay kullanmak için onlara isimler veririz. Bu değişkenlerde ne tür bilgilerin depolandığını hatırlamamız için bizlere yardım eder. EdWare'da değişkenlerinizi istediğiniz gibi adlandırabilirsiniz. Birini 'Fred' olarak isimlendirebilirsiniz, ancak Fred'in hangi tip bilginin saklandığını 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.

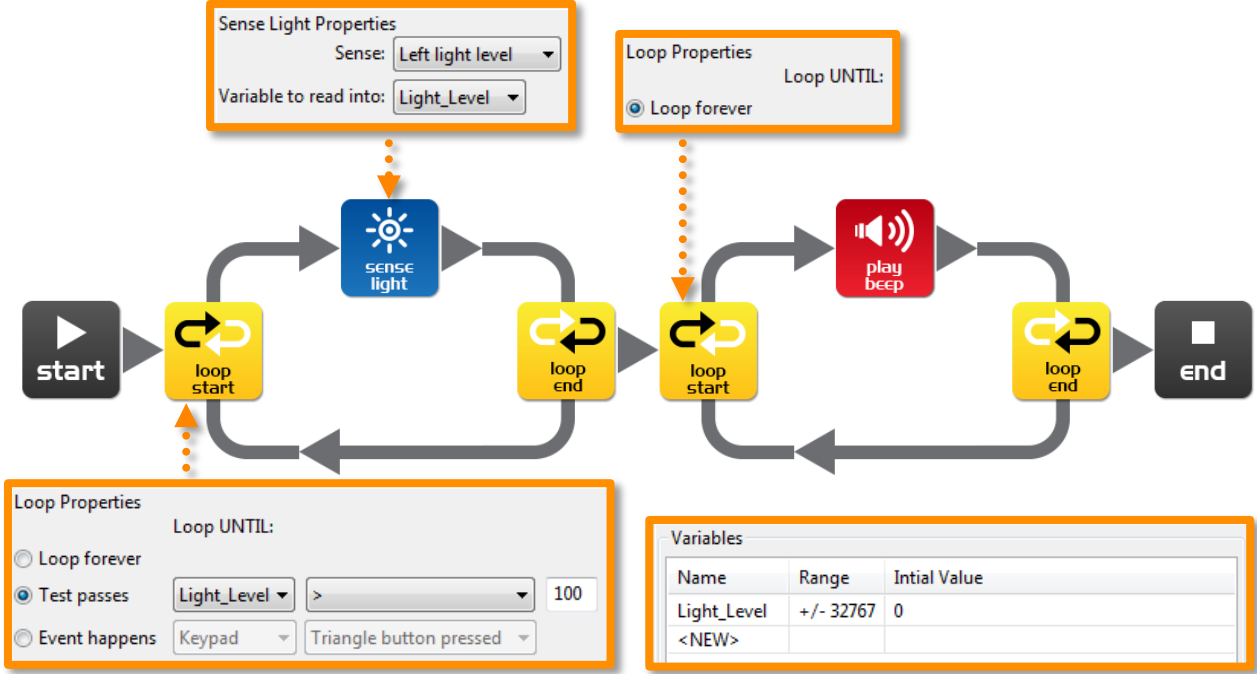
Aşağıdaki verileri depolamak için hangi tür değişkeni kullanırdınız (bayt veya kelimeler)?

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?**

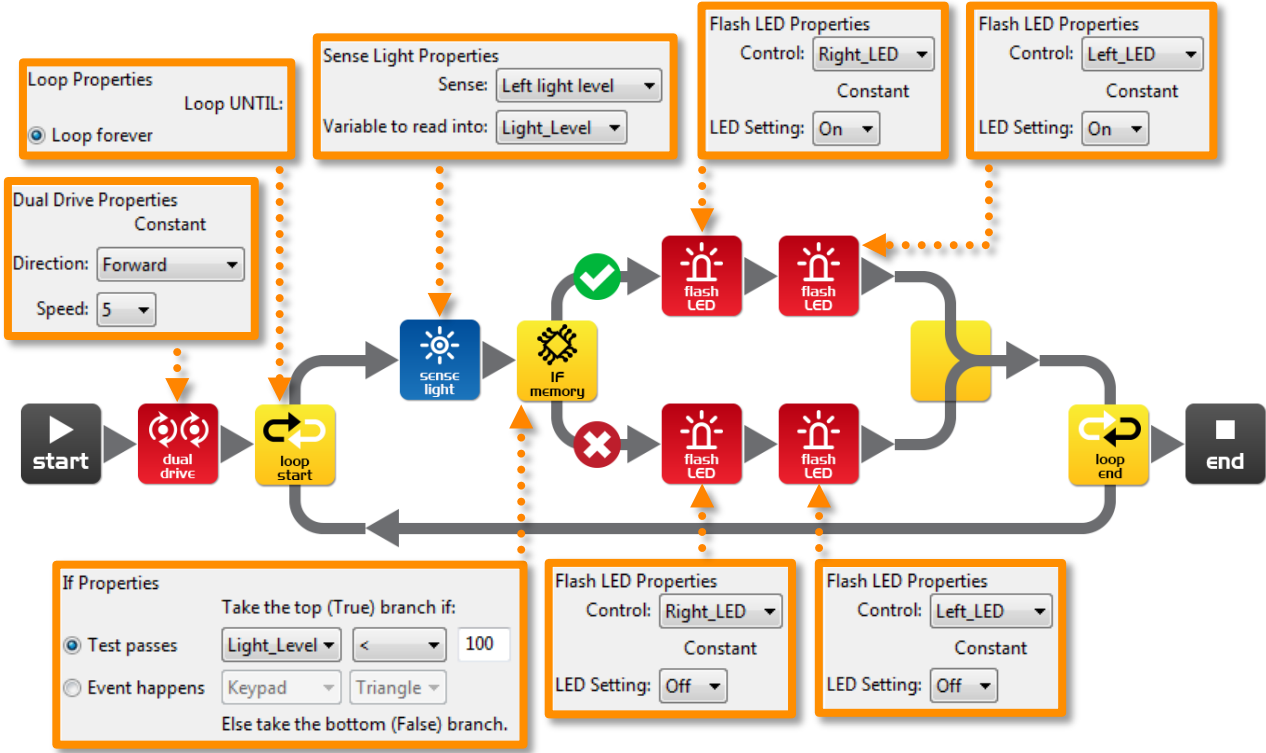
---

---

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?

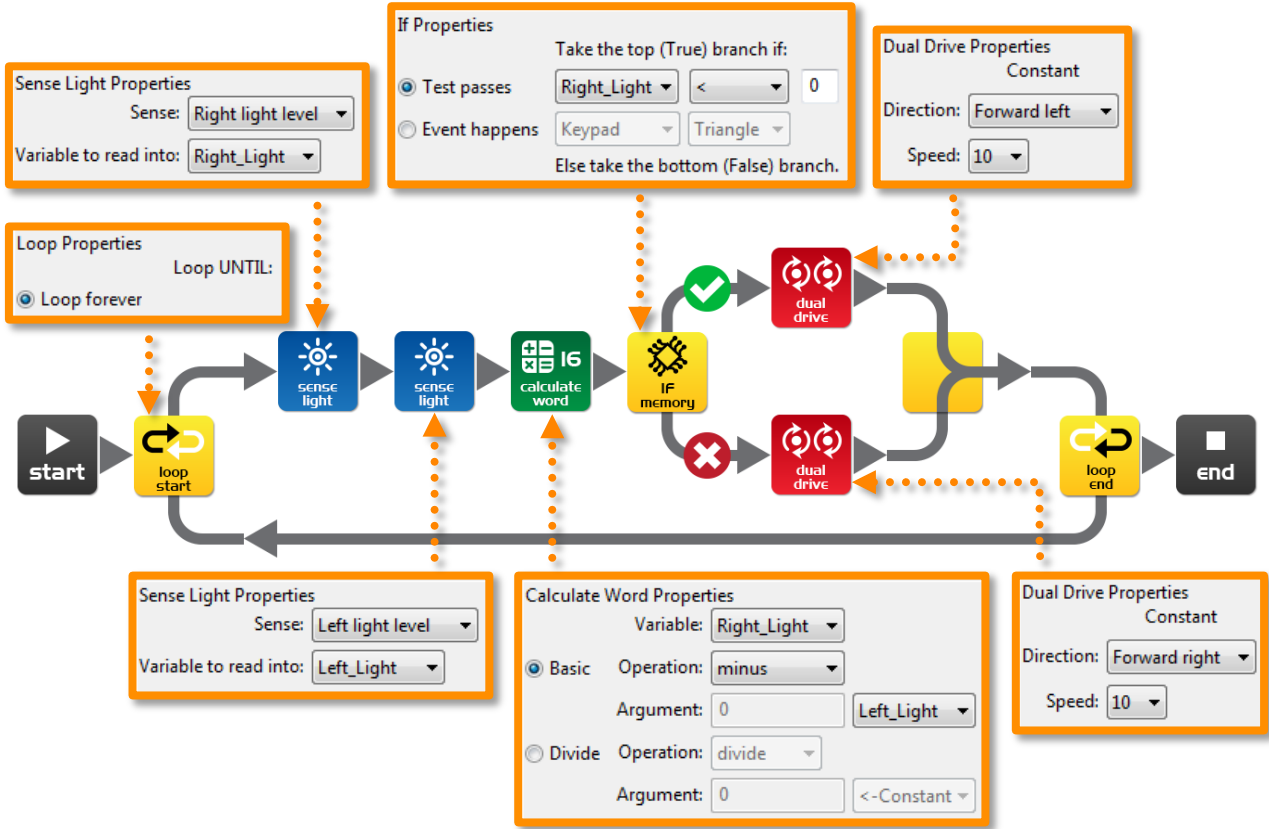
---

---

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?

## DERS 10: TASARIM ÖZETİ 2 – PROGRAMIM

**Yaratıcı düşünme ve problem çözme** – Bu ikinci tasarım özetinde öğrenciler kendi görevleriyle karşımıza çıkar ve robotun nasıl bir çözüm sağlayacağını kavramsallaştırmak için çözüm yolu bulur. Bu iki ders süresi alabilir. Alternatif olarak bu ev ödevi uygulaması olabilir. Öğrenciler kendi konusunu seçebilir, programının amacını ifade edebilir ve gerçek dünyada nerede kullanılabileceğini açıklayabilir. Bu tasarım özetinde öğrencilerden ilk tasarım özetinde yaptıklarından daha fazla bilgi vermeleri istenir. Öğrenciler şimdi daha fazla programlama deneyimine de sahip ve robot sensörlerini kullanabilirler.

### 1. Robotun çözebileceği bir problem veya yapabileceği bir eylem belirleyiniz

Öğrenciler “İnsanların Başvurması Gerekmiyor” videosunu izledi ve robotların gelecekte nasıl kullanılabileceğinin daha çok farkında olmalıdır. Bu öğrencilerden yeni fikirlerin gelmesini teşvik etmelidir. Aşağıda bazı örnekleri bulunmaktadır.

- 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. İki robot bir sumo güreşinde rekabet eder.

### 2. Problemi veya robotun yapması gereken hareketlerini tanımlayınız

Öğrenciler tasarım özeti çalışma kâğıdını kullanarak robot programlarının çözdüğü problemi ve robot programının problemi nasıl çözdüğünü tanımlar.

Örnek: **Problem:** ‘İnsan sürücülerin sürüş anında kolay bir şekilde kafaları karışır ve hata yapabilir. Bu hatalar ölümcül olabilir.’ **Çözüm:** ‘Bu program engel takibi kullanarak bir robotun yol boyunca nasıl hareket edebildiğini ve insanlar, diğer araçlar veya binalara çarpmadan kaçabildiğini bu yüzden çok güvenli olduğunu gösterir.’

### 3. Programı yazınız ve test ediniz

Öğrenciler programlarını yazar ve kontrol eder.

### 4. Başarısızlık

Kaçınılmaz bir şekilde ilk deneme asla başarılı değildir. Başarısızlık programlamanın normal bir parçasıdır (veya herhangi bir mühendislik alanının). Öğrenciler tasarım özeti çalışma yaprağına başarısızlıklarını açıklayabilir.

### 5. Yapmak istediğiniz şeyi ve kullanılan programlama simgelerini tanımlayınız

Öğrenciler tasarım özeti çalışma yaprağı 10.2’yi kullanarak programlarında kullanacakları bir takım simgeleri seçer ve bu simgelerin programda ne yaptıklarını açıklar. Öğrencilerin daha fazla programlama simgesi açıklaması için ek sayfalar yazdırın.

### 6. Demo

Öğrenciler robotlarının davranışlarını sergiler ve fikirleri, programları, karşılaştıkları sorunları ve bunları nasıl çözdüklerini konuşurlar.

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 olmadığını sebepleri nelerdi?

---

---

---

---

---

---

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 simgelerde 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!**

---

---

---

---

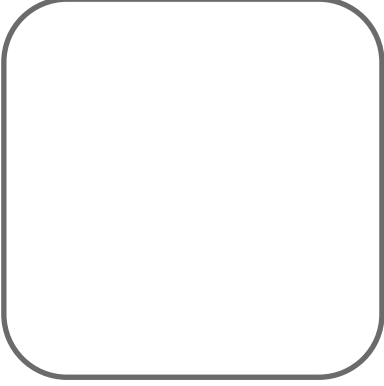
---

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? \_\_\_\_\_

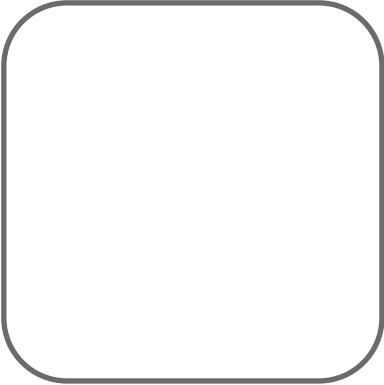
Bu simge ne yapar? \_\_\_\_\_

---

---

---

---



Bu simgenin adı nedir? \_\_\_\_\_

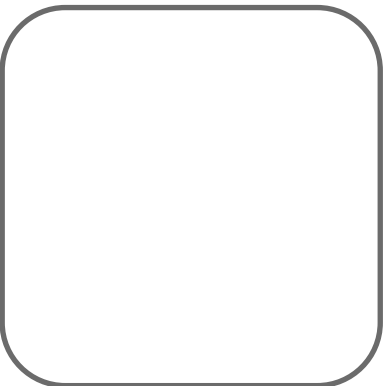
Bu simge ne yapar? \_\_\_\_\_

---

---

---

---



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 çırpma yanıt olarak LED yakıp söndürme	
6.2 El çırpma yanıt olarak hareket etme	
6.3 El çırpma 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 içeresinde 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	



## İletişim

PO Box 8052, Grange, SA 5022

+61 8 8166 7193 (Australia)

support@meetedison.com

www.meetedison.com



## İletişim

STEM Makers Bilişim Yazılım Eğitim Turizm Danışmanlık A.Ş.

Çankaya Üniversitesi Kuluçka Merkezi

Çukurambar Mah. Öğretmenler Caddesi No: 14

A Blok. 4. Kat, 421/B 06790 Çankaya

Ankara / TÜRKİYE

bilgi@stemmakers.academy

[www.stemmakers.academy](http://www.stemmakers.academy)