# 9. Hafta – Mesafe Sensörü

## Haftanın Amacı:

Bu haftanın amacı, öğrencilerin mesafe (distance) sensörünün kullanımı ile ilgili temel bilgi ve becerileri kazanmasıdır. Öğrenciler bu temel bilgi ve becerileri kullanarak görece gelişmiş programlar oluşturacaktır. Öğrenciler bu amaç doğrultusunda mesafe sensörünün ışıklarını düzenleme, önde bulunan aracı takip etme, duvar takibi yapma, etrafındaki nesneleri tespit edip ona doğru yönelme gibi problemlere çözüm üretecektir. Buna ek olarak öğrenciler Python’da fonksiyon kavramını öğrenecek ve bu kavramı problem çözümünde kullanacaktır.

## Haftanın Kazanımları:

* Öğrenciler mesafe sensörünün görevini ve nasıl çalıştığını anlatabilirler.
* Öğrenciler robotun mesafe sensöründen gelen değere göre hareket etmesini sağlayabilirler.
* Öğrenciler araç takip programı yazabilirler.
* Öğrenciler mesafe sensörünün ışıklarını programlayabilirler.
* Öğrenciler Python fonksiyonları oluşturup bu fonksiyonları programlarında kullanabilirler.
* Öğrenciler duvar takip programı yazabilirler.
* Öğrenciler nesne tespit ve ulaşma programı oluşturabilirler.

## Kullanılacak Malzemeler:

Robot seti, bilgisayar ve çalışma alanı

## GÖZLE VE UYGULA

|  |
| --- |
| ***Rehber Öğretmen İçeriği –*** ***Mesafe Sensörü Hakkında*** |
| *Mesafe sensörü isminden de anlaşılacağı üzere mesafe ölçmeye yarayan bir sensördür. 5 cm ile 2 metre arılığındaki (bu değerlerden 2 cm fazla veya eksik değerler de elde edilebilir) nesne ve yüzeyleri algılayarak bu cisim ve yüzeylerin mesafe sensörüne olan uzaklıklarını ölçebilir. Ölçüm yaparken karşısına doğru insan kulağı ile algılanamayan ses dalgaları gönderir ve bu ses dalgalarının karşıdaki nesneye çarpıp sensöre geri dönme süresini kullanarak cismin uzaklığını hesaplar. Algılanması istenen cisim veya yüzey mümkün mertebe mesafe sensörünün tam karşısında bulunmalıdır. Yani cismin veya yüzeyin mesafe sensörüne olan açısı sıfıra yakın olmalıdır. Bu açı arttıkça ölçüm değerlerindeki hata payı artacaktır ve ölçüm yapılamaz hale gelinecektir. Mesafe sensöründe kısa menzil ölçüm özelliği bulunmaktadır. Kısa menzil ölçüm özelliği modunda mesafe sensörü 5-30 cm arasındaki cisim ve yüzeyleri algılayıp ölçüm yapabilir. Kısa menzil modunda yapılan ölçümün doğruluğu artar. Mesafe sensörünün “gözlerinin” etrafında dört adet programlanabilir ışık bulunmaktadır. Bu ışıklar birbirinden bağımsız olarak programlanabilir.*  *Mesafe sensörünün kullanılabilmesi için öncelikle bir DistanceSensor nesnesi oluşturulmalıdır, başka bir değişle DistanceSensor sınıfından bir nesne oluşturulmalıdır. Oluşturulacak nesnenin ismi sensör ismi ile uyumlu olarak mesafe\_sensoru olarak seçilmiştir ve aşağıda bu nesnenin nasıl oluşturulacağı gösterilmiştir.*  *mesafe\_sensoru=DistanceSensor("F")*  *Oluşturulan nesneye başka bir isim de verilebilir. Burada mesafe sensörü F portuna bağlandığı için bu parametre girilmiştir. Başka bir porta bağlanan sensör için o portun ismi kullanılmalıdır. Mesafe sensörü nesnesi oluşturulduktan sonra sensörün ışıklarını açıp kapama, mesafe ölçme ve mesafe bazlı bekleme işlemleri gerçekleştirilebilir. Sensörün bütün ışıklarının yakılması için light\_up\_all metodu kullanılır. Bu metodun kullanımı aşağıda verilmiştir.*  *mesafe\_sensoru.light\_up\_all(parlaklık)*  *Parlaklık parametresi 0 ile 100 arasında tam sayı değerleri alabilir. 0 kapalı, 100 tam parlaklık anlamına gelir. 0-100 arasındaki değerler ise ara parlaklık miktarlarını göstermek için kullanılır. Eğer herhangi bir parlaklık değeri girilmezse varsayılan değer olarak 100 kullanılır.*  *Sensörün “gözleri” etrafındaki ışıkları ayrı ayrı yakmak için light\_up metodu kullanılır. Bu metodun kullanımı aşağıda verilmiştir.*  *light\_up(sağ üst, sol üst, sağ alt, sol alt)*  *Light\_up metodunun ilk parametresi sağ üst ışık kaynağının parlaklığını belirlemek için kullanılır. Benzer şekilde parlaklık miktarı 0-100 arasındaki tam sayılar aracılığı ile belirlenir. Bu metodun ikinci parametresi sol üst, üçüncü parametresi sağ alt ve son parametresi sol alt ışık kaynağının parlaklık miktarlarını belirlemek için kullanılır.* |

### Gözle: Sol Gözünü Kırpan Robot

Bu etkinlikte mesafe sensörünün sol gözünün ışıklarını açıp kapamak suretiyle kırpılması sağlanacaktır. Göz kırpma işlemi sürekli gerçekleştirilmelidir. Bu görev için aşağıdaki kod kullanılabilir.

mesafe\_sensoru=DistanceSensor("F")

while True:

    mesafe\_sensoru.light\_up(0,100,0,100)

    wait\_for\_seconds(0.5)

    mesafe\_sensoru.light\_up(0,0,0,0)

    wait\_for\_seconds(0.5)

### Uygula: Gözlerini Sırayla Kırpan Robot

Bu etkinlikte mesafe sensörünün gözleri sırayla kırpılacaktır. Önce sol göz kırpılacak ardından sağ göz kırpılacak ve bu işlem sürekli tekrarlanacaktır. Bu görev için aşağıdaki kod kullanılabilir.

mesafe\_sensoru=DistanceSensor("F")

while True:

    mesafe\_sensoru.light\_up(0,100,0,100)

    wait\_for\_seconds(0.5)

    mesafe\_sensoru.light\_up(0,0,0,0)

    wait\_for\_seconds(0.5)

    mesafe\_sensoru.light\_up(100,0,100,0)

    wait\_for\_seconds(0.5)

    mesafe\_sensoru.light\_up(0,0,0,0)

    wait\_for\_seconds(0.5)

|  |
| --- |
| ***Rehber Öğretmen İçeriği –*** ***Fonksiyonlar*** |
| Python’da tekrar tekrar kullanılacak kodlar için fonksiyonlar tanımlanır. Örneğin daha önce anlatılan wait\_for\_seconds fonksiyonu programın istenen süre için duraklatılmasını sağlar veya range fonksiyonu istenen aralıkta tam sayı üretmek için kullanılır. Bu fonksiyonlar Python’un yazarları tarafından oluşturulup Python kütüphanesine eklenmiştir. Programcılar bu fonksiyonları istedikleri zaman çağırıp kullanırlar. Programcılar da kendi fonksiyonlarını tanımlayabilir. Fonksiyon tanımlamak için def anahtar kelimesi kullanılır. Bundan sonra fonksiyonun ismi yazılır. Ardından parantez açıp kapatılarak parametre alınacaksa parametreler virgül ile ayrılarak parantezin içerisine yazılır. Eğer parametre almayacaksa parantezin içi boş bırakılır. Parantezden sonra iki nokta üst üste konulur. Buradan sonra fonksiyonun gövdesindeki komutlar yazılacaktır. Bir tab büyüklüğünde girinti yapılarak yazılan bütün komutlar, fonksiyon gövdesinde sayılacaktır ve fonksiyon çağrıldığında gövdedeki bütün komutlar çalıştırılacaktır. Aşağıda bir fonksiyonun nasıl tanımlanacağı gösterilmiştir. Eğer tanımlanan fonksiyon parametre almazsa parantezin içerisi boş bırakılmalıdır.  def fonksiyon\_ismi(parametre\_1, parametre\_,...,parametre\_n):      komut\_1      komut\_2      ...      komut\_n  Fonksiyon tanımlandıktan sonra çağrılarak kullanılabilir. Fonksiyonu çağırırken fonksiyon ismi ve var ise gerekli parametreler yazılır. Yukarıda yazılan fonksiyonun çağrılıp çalıştırılması için aşağıdaki komut kullanılabilir.  fonksiyon\_ismi(parametre\_1, parametre\_,...,parametre\_n):  Programda fonksiyon tanımı fonksiyon çağrısından önce olmalıdır. Fonksiyon çağrısından sonra tanımlanmış veya tanımı yapılmamış bir fonksiyonun kullanımı hataya sebep olacaktır. |

### Uygula: Fonksiyon Kullanarak Göz Kırpma

Gözlerini Sırayla Kırpan Robot etkinliğinde sağ ve sol gözler ayrı ayrı kırpılmaktadır. Burada göz kırpma işlemi için arka arkaya benzer komutlar yazılmıştır. Bu durum gereksiz kod yazılmasına sebep olmaktadır. Bunun yerine bu etkinlikte goz\_kirp fonksiyonu tanımlanarak kullanılacaktır. Aşağıda bu fonksiyonun tanımı verilmiştir. Rehber öğretmen gerekli kodları yazarak öğrencilere açıklar ve fonksiyonların faydalarından bahseder.

def goz\_kirp(goz):

    if goz=="sol":

        mesafe\_sensoru.light\_up(0,100,0,100)

    elif goz=="sag":

        mesafe\_sensoru.light\_up(100,0,100,0)

    wait\_for\_seconds(0.5)

    mesafe\_sensoru.light\_up(0,0,0,0)

    wait\_for\_seconds(0.5)

Hangi gözün kırpılması gerektiği goz\_kirp fonksiyonuna parametre olarak gönderilmelidir. Fonksiyon içerisinde koşul ifadesi aracılığı ile gelen parametreye göre sol veya sağ gözün kırpılması sağlanır. Bu fonksiyon farklı şekillerde de yazılabilir. Örneğin mesafe\_sensoru değeri bu fonksiyona parametre olarak gönderilebilir. Bu haliyle mesafe sensörünün ismi değiştirildiğinde program çalışmayabilir. Fonksiyon tanımlandıktan sonra artık program içerisinde çağrısı yapılarak kullanılabilir.

while True:

    goz\_kirp("sol")

    goz\_kirp("sag")

Yukarıdaki kod ile sürekli olmak kaydıyla, sırayla sol ve sağ gözlerinin kırpılması sağlanmaktadır. Görüldüğü üzere fonksiyon bir kere tanımlandıktan sonra onu kullanmak için sadece çağrılması yeterlidir. Böylece fonksiyonlar bize daha kısa ve daha kolay yönetilebilen programlar yazmada yardımcı olur.

*Uygula: Hub Düğmelerine Göre Göz Kırpan Robot*

Bu etkinlikte sol hub düğmesi tıklandığı süre boyunca sol gözünü, sağ hub düğmesi tıklandığı süre boyunca sağ gözünü ve herhangi bir düğme tıklanmadığında sırayla sol ve sağ gözlerini sürekli kırpan bir robot programı yazılmalıdır. Bu görev için aşağıdaki kod kullanılabilir.

while True:

    if hub.left\_button.is\_pressed():

        goz\_kirp("sol")

    elif hub.right\_button.is\_pressed():

        goz\_kirp("sag")

    else:

        goz\_kirp("sol")

        goz\_kirp("sag")

|  |
| --- |
| ***Rehber Öğretmen İçeriği –*** ***Mesafe Sönsörü İle Ölçüm*** |
| Mesafe sensörü ile cm, inç ve yüzde cinsinden ölçümler yapabilir. Yüzde cinsinden ölçüm yapıldığında elde edilen mesafe 200 cm’ye oranlanır. Örneğin 100 cm’de bir cisim algılandığında cisim mesafenin yüzde ellisinde olduğu için 50 değeri döndürülür. Santimetre, inç ve yüzde cinsinden ölçümler yapıldığında eğer 200 cm mesafe içerisinde bir cisim bulunmuyorsa yok anlamına gelen None değeri döndürülür. Aşağıda sırayla santimetre, inç ve yüzde ölçümleri için kullanılan metotlar verilmiştir.  get\_distance\_cm()  get\_distance\_inches()  get\_distance\_percentage()  Bu metotlar ile kısa menzil ölçüm de yapılabilir. Daha önce belirtildiği üzere kısa menzil ölçümlerde 30 cm’ye kadar ve daha kesin ölçümler yapılabilir. Kısa menzil ölçüm için bu metotlara True parametresi girilmelidir. Bu durum aşağıda örneklenmiştir.  get\_distance\_cm(True)  get\_distance\_inches(True)  get\_distance\_percentage(True) |

### Gözle: Belirli Bir Mesafeye Kadar Giden Robot

Bu etkinlikte robotun karşısında bulunan engele 10 cm yaklaşıp burada durmasını sağlayan bir program yazılacaktır. Rehber öğretmen kodu açıklayarak anlatır. Bu görev için aşağıdaki kod kullanılabilir. Etkinlikte kullanılacak robot ve engel Resim 9.1’de gösterilmiştir.

mesafe\_sensoru=DistanceSensor("F")

surus\_tabani=MotorPair("C","D")

surus\_tabani.set\_default\_speed(50)

while mesafe\_sensoru.get\_distance\_cm()>10:

    surus\_tabani.start()

surus\_tabani.stop()

LEGO, oyuncak, iç mekan, döşeli içeren bir resim

Açıklama otomatik olarak oluşturuldu

*Resim 9.1 Bir Mesafeye Kadar Giden Robot*

### Uygula: Fonksiyon Kullanarak Belirli Bir Mesafeye Kadar Giden Robot

Bu etkinlikte öğrencilerden cm\_ye\_kadar\_ilerle isimli bir fonksiyon yazmaları istenecektir. Bu fonksiyona gönderilen parametre robotun engele kaç cm kalana kadar ilerleyeceğini belirleyecektir. Örneğin cm\_kadar\_ilerle(20) komutu ile sürüş tabanı engele 20 cm kalana kadar ilerleyecektir. Rehber öğretmen için gerekli kod aşağıda verilmiştir.

mesafe\_sensoru=DistanceSensor("F")

surus\_tabani=MotorPair("C","D")

surus\_tabani.set\_default\_speed(50)

def cm\_kadar\_ilerle(mesafe):

    while mesafe\_sensoru.get\_distance\_cm()>mesafe:

        surus\_tabani.start()

    surus\_tabani.stop()

cm\_kadar\_ilerle(20)

|  |
| --- |
| ***Rehber Öğretmen İçeriği –*** ***Mesafe Sönsörü Olayları*** |
| Mesafe Sensöründe belirli bir mesafeden daha yakın olana kadar bekle ve belirli bir mesafeden daha uzak olana kadar bekle isimli iki olay bulunur. Belirli bir mesafeden yakın olana kadar bekle olayının kullanımı aşağıda verilmiştir. Burada mesafe istenen mesafedir. Birim ise santimetre için “cm”, inç için “in” ve yüzde için “%” olarak girilmelidir.  wait\_for\_distance\_closer\_than(mesafe,birim)  Belirli bir mesafeden daha uzak olana kadar bekle olayının kullanımı aşağıda gösterilmiştir. Aynı şekilde mesafe, beklenecek mesafedir ve birim de mesafenin birimidir.  wait\_for\_distance\_farther\_than(mesafe,birim) |

### Gözle: Olay Kullanarak Belirli Bir Mesafeye Kadar Giden Robot

Bu etkinlikte olay kullanılarak belirli bir mesafeye kadar ilerleyen robotun kodu yazılacaktır. Bu görev için wait\_for\_distance\_closer\_than olayı kullanılmalıdır. Bu görev için gerekli kod aşağıda verilmiştir. Rehber öğretmen kodu açıklayarak anlatır.

mesafe\_sensoru=DistanceSensor("F")

surus\_tabani=MotorPair("C","D")

surus\_tabani.set\_default\_speed(50)

surus\_tabani.start()

mesafe\_sensoru.wait\_for\_distance\_closer\_than(20,"cm")

surus\_tabani.stop()

### Uygula: Olay Kullanarak Belirli Bir Mesafeye Kadar Uzaklaşan Robot

Bu etkinlikte öğrencilerden bir engelden 30 cm’ye kadar uzaklaşan robot kodunu yazmaları istenmektedir. Bu kod için mesafe sensörünün ilgili olayı kullanılmalıdır. Rehber öğretmen için gerekli kod aşağıda verilmiştir.

mesafe\_sensoru=DistanceSensor("F")

surus\_tabani=MotorPair("C","D")

surus\_tabani.set\_default\_speed(-50)

surus\_tabani.start()

mesafe\_sensoru.wait\_for\_distance\_farther\_than(30,"cm")

surus\_tabani.stop()

### Gözle ve Uygula: Öndeki Aracı Takip Eden Robot

Bu etkinlikte amaç öndeki aracı 20 cm uzaklıktan takip eden robot programını yazmaktır. Öndeki araç ilerlerse aradaki mesafe 20 cm kalacak şekilde robot da ilerlemelidir, öndeki araç durduğunda robot da durmalıdır ve öndeki araç geri gelirse robot da 20 cm takip mesafesini koruyarak geriye doğru gitmelidir. Etkinlikte önde bulunan araç yerine kitap benzeri bir cisim kullanılabilir. Bu görev için aşağıdaki kod kullanılabilir.

mesafe\_sensoru=DistanceSensor("F")

surus\_tabani=MotorPair("C","D")

surus\_tabani.set\_default\_speed(50)

takip\_mesafesi=20

tepki\_gucu=5

while True:

    aradaki\_mesafe=mesafe\_sensoru.get\_distance\_cm()

    if aradaki\_mesafe!=None:

        tepki=(aradaki\_mesafe-takip\_mesafesi)\*tepki\_gucu

    surus\_tabani.start(0,tepki)

Rehber öğretmen programı öğrencilere anlatır ve onların da programı yazmalarını sağlar. Konuyu öğrencilere anlatırken düzenli olarak öndeki araç ile robotun arasındaki mesafenin (aradaki\_mesafe) ölçüldüğünü, bu mesafenin 20 cm (takip\_mesafesi) dışına çıktığında robotun aradaki mesafe ölçüsünde karşı tepki verdiğini, verilen tepkinin gücünün de tepki\_gucu değişkeniyle ayarlandığını vurgular. Farklı tepki güçleri için (örn, 3, 7 ve 10) programı çalıştırıp öğrencilere deneterek tepki gücünün pratik olarak ne anlama geldiğini öğrencilerin anlamasını sağlar.

### Uygula: Öndeki Aracı Takip Eden Gelişmiş Robot

Bu etkinlikte yine öndeki aracı 20 cm mesafe ile takip eden bir robot yapılacaktır. Fakat bu sefer robota birkaç özellik daha eklenecektir. Ek olarak

1. 0.5 saniyede bir konsol ekranında aradaki\_mesafe değişkeninin değerini yazdırmalıdır,
2. Aradaki mesafe 20’den büyük ise ışık matrisinde ileri ok, 20’den küçük ise geri ok ve aradaki mesafe 0 ise gülen yüz görünmelidir,
3. Aradaki mesafe 20’den büyük ise hoparlörden uygun bir bip sesi çalınmalıdır, 20’den küçük ise geriye gittiğini gösteren bir bip sesi çalınmalıdır ve 20 ise herhangi bir şey çalınmamalıdır,
4. Robot kuvvet sensörüne dokunulduğunda durmalıdır ve ses vermeyi bırakmalıdır.

Bu görev için aşağıdaki kod kullanılabilir. Rehber öğretmen öğrencilerin kodu kendilerinin yazmalarını istemelidir. Öğrencilerin yazdıkları kod verilen koddan farklılıklar gösterebilir.

import time

hub = PrimeHub()

kuvvet\_sensoru=ForceSensor("A")

mesafe\_sensoru=DistanceSensor("F")

surus\_tabani=MotorPair("C","D")

surus\_tabani.set\_default\_speed(50)

takip\_mesafesi=20

tepki\_gucu=5

birinci=time.ticks\_ms()

while not kuvvet\_sensoru.is\_pressed():

    aradaki\_mesafe=mesafe\_sensoru.get\_distance\_cm()

    ikinci= time.ticks\_ms()

    if ikinci-birinci>0.5:

        birinci=time.ticks\_ms()

        print(aradaki\_mesafe)

    if aradaki\_mesafe!=None:

        tepki=(aradaki\_mesafe-takip\_mesafesi)\*tepki\_gucu

        if aradaki\_mesafe>20:

            hub.light\_matrix.show\_image("ARROW\_N")

            hub.speaker.start\_beep(70)

        elif aradaki\_mesafe<20:

            hub.light\_matrix.show\_image("ARROW\_S")

            hub.speaker.start\_beep(100)

        else:

            hub.light\_matrix.show\_image("HAPPY")

            hub.speaker.stop()

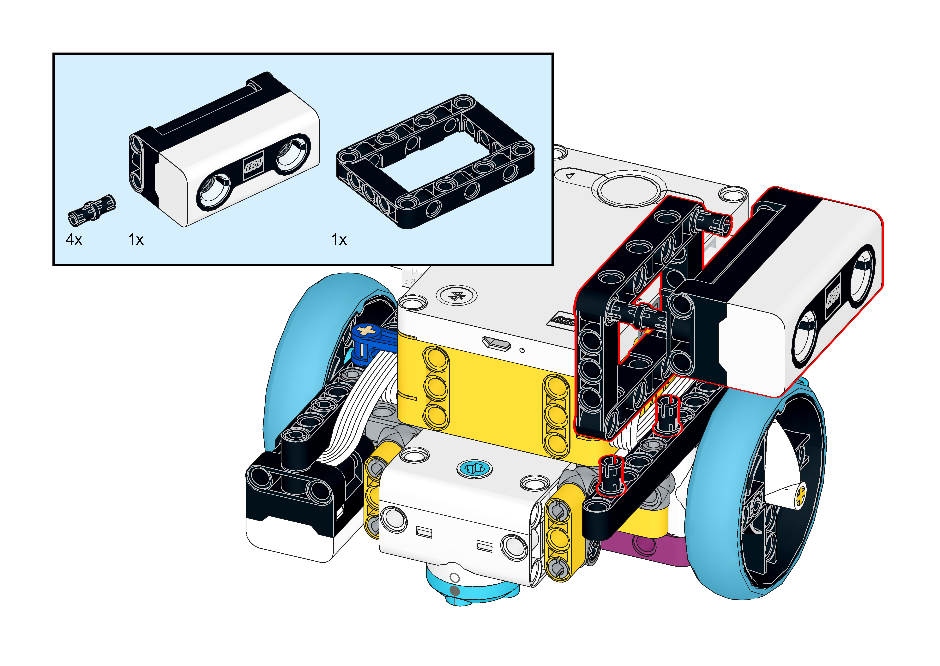
    surus\_tabani.start(0,tepki)

surus\_tabani.stop()

hub.speaker.stop()

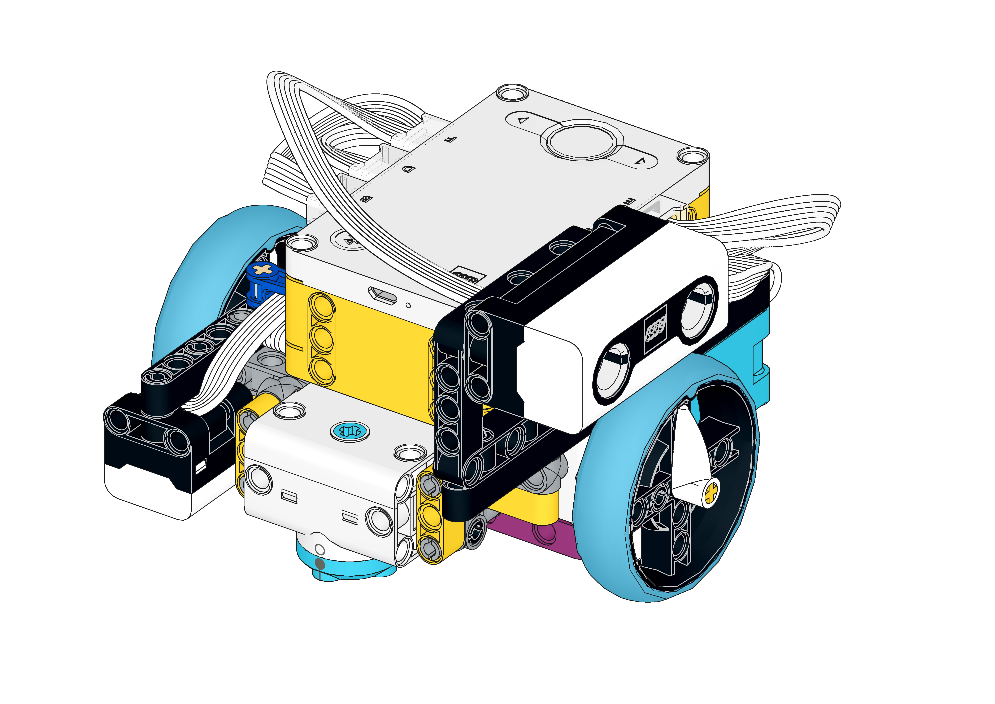
### Gözle: Düz Duvar Takibi Yapan Robot

Bu etkinlikte robotun duvar takibi yapması istenmektedir. Bu görev için robotun aşağıdaki şekilde gösterildiği gibi düzenlenmesi gerekmektedir.



*Resim 9.2 Duvar Takibi Yapan Robot*

Gerekli inşa işlemlerinin ardından “*mesafe sensörü sol tarafa bakan robot*” aşağıdaki şekilde gösterildiği gibi elde edilmiş olunur. Robot üzerinde kuvvet sensörü bağlı bulunabilir. Bu şekilde kuvvet sensörü gösterilmemiştir. Robotun üzerinde kuvvet sensörü varsa çıkarılmasına gerek yoktur.



*Resim 9.3 Mesafe Sensörü Sol Tarafa Bakan Robot*

Robot sol tarafında duran düz duvarı takip edeceği için mesafe sensörü sol tarafa takılmıştır. Robot duvara paralel bir şekilde bırakılacaktır ve bırakılma uzaklığı duvardan 15 cm’yi geçmemelidir. Robot duvar ile uzaklığı 10 cm olacak şekilde duvarı takip ederek ileriye doğru hareket etmelidir. Bu görev için aşağıdaki kod kullanılabilir. Rehber öğretmen kodu yazarak öğrencilere açıklar.

mesafe\_sensoru=DistanceSensor("F")

surus\_tabani=MotorPair("C","D")

istenen\_mesafe=10

duzeltme\_gucu=6

while True:

    gercek\_mesafe=mesafe\_sensoru.get\_distance\_cm()

    if gercek\_mesafe!=None:

        sapma=gercek\_mesafe-istenen\_mesafe

    tepki=-sapma\*duzeltme\_gucu

    surus\_tabani.start(tepki,15)

Mesafe sensörü bazen gerçek mesafeden bağımsız olarak beklenmedik değerler üretebilmektedir. Bu durumun sıklığı pürüzlü ve eğik yüzeylerde artmaktadır. Bu yüzden etkinlikte kullanılan duvar düz ve pürüzsüz olmalıdır.

### Uygula: Duvar Takibi Yapan Gelişmiş Robot

Bu etkinlikte robot yine duvar takibi yapmak üzere duvara paralel bir şekilde bırakılacaktır. Fakat bir öncekinden farklı olarak robot duvardan 150 cm uzaklığa kadar konulabilir. Bunun yanında artık bir tane düz duvar yerine aşağıdaki şekilde gösterildiği üzere basamak şeklinde ayrık iki veya daha fazla duvar bulunacaktır. Duvarların uzunluğu yaklaşık olarak 1 m olabilir. Çok kısa veya çok uzun olmamalıdır.



*Resim 9.4 Duvar Takibi Yapan Gelişmiş Robot*

Duvarların arasındaki mesafe 15 cm ile 100 cm arasında olabilir. İstenirse basamak şeklinde olmak kaydıyla üçüncü bir duvar eklenebilir. Robot ile ilk duvar arasındaki mesafe 150 cm’ye kadar çıkabilir. Fakat robot duvara paralel olacak şekilde bırakılmalıdır. Robot başlangıçta ve duvar geçişlerinde duvara olan mesafesini uygun bir konuma getirmelidir ve ardından duvarı takip etmeye devam etmelidir. Rehber öğretmen için aşağıdaki kod verilmiştir.

mesafe\_sensoru=DistanceSensor("F")

surus\_tabani=MotorPair("C","D")

istenen\_mesafe=10

duzeltme\_gucu=6

while True:

    gercek\_mesafe=mesafe\_sensoru.get\_distance\_cm()

    if gercek\_mesafe!=None:

        sapma=gercek\_mesafe-istenen\_mesafe

    if sapma>=15 and sapma<=150:

        surus\_tabani.stop()

        surus\_tabani.move(10,"cm",0,50)

        surus\_tabani.move\_tank(8,"cm",-50,50)

        surus\_tabani.move(sapma-4,"cm",0,20)

        surus\_tabani.move\_tank(8,"cm",50,-50)

    else:

        tepki=-sapma\*duzeltme\_gucu

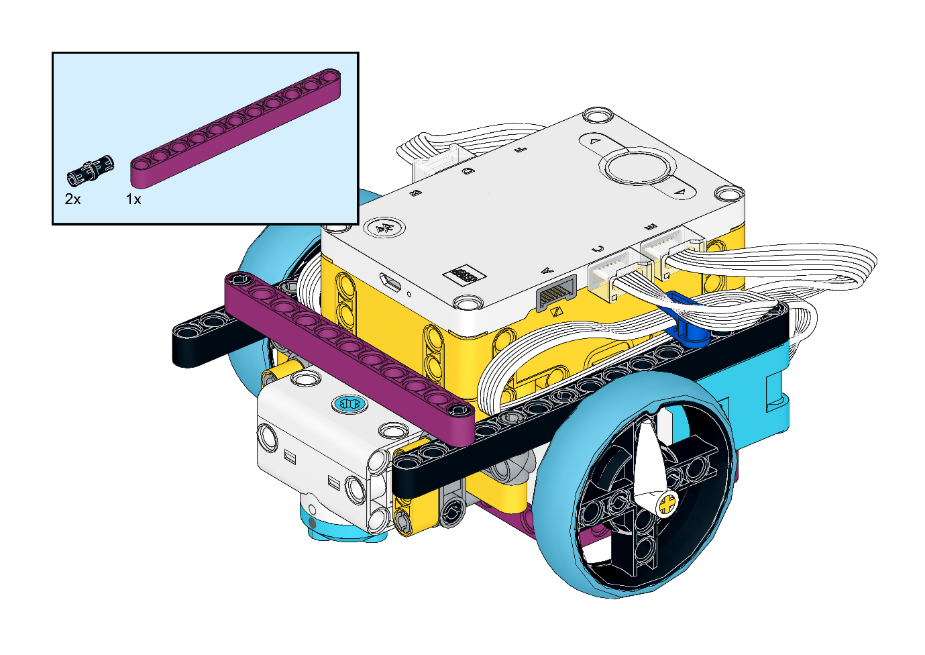
        surus\_tabani.start(tepki,15)

Öğrencilere bu etkinliği yapmaları için gerekli süre verilmelidir. Bu süre içerisinde rehber öğretmen onlara sadece ipuçları vererek öğrencileri yönlendirmelidir. Verilen süre bittiğinde öğrenciler başarılı bir şekilde kodu yazamamışlarsa verilen kodu onlara açıklayarak anlatır.

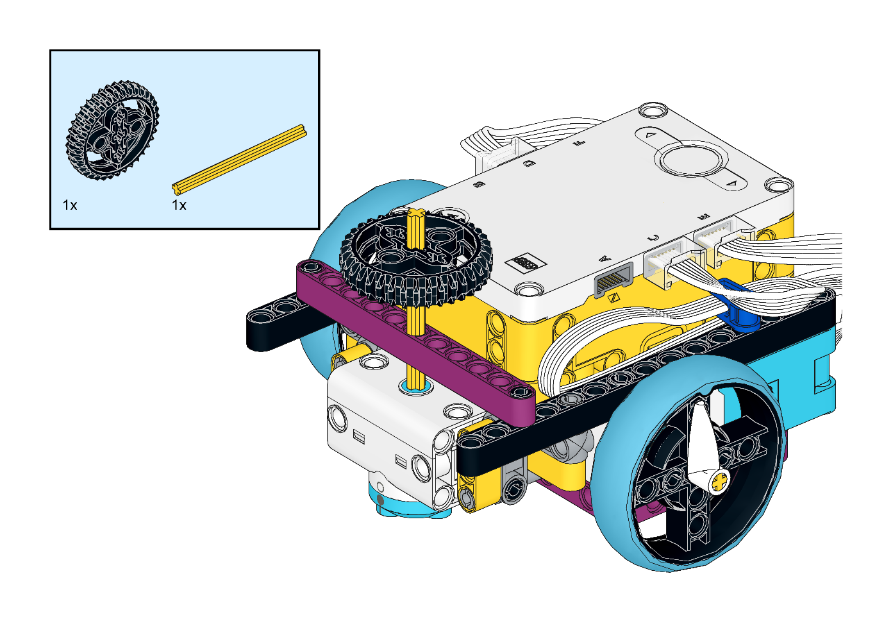
## TASARLA

Hedef Tespit Robotu

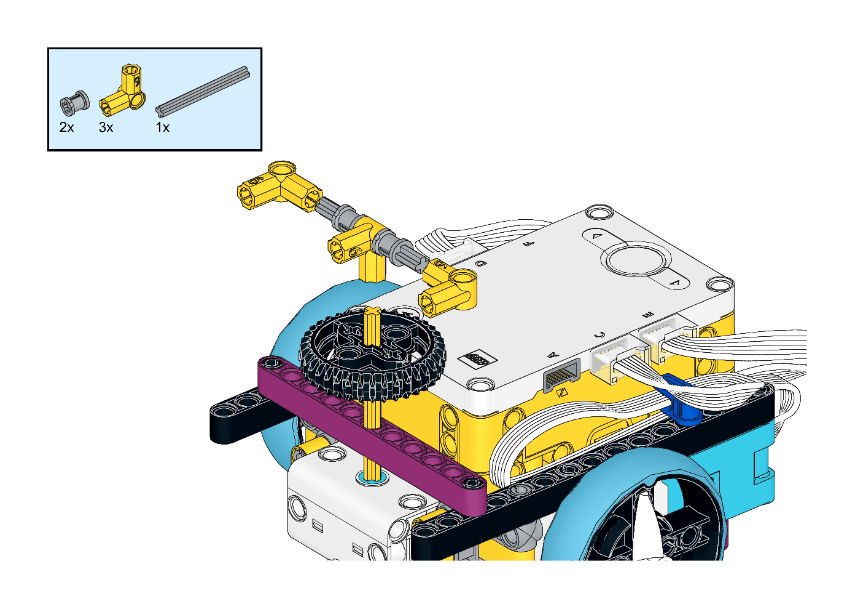
Bu etkinlikte öğrencilerden önünde bir yerde bulunan hedefi tespit edip, onun yanına giden robotu programlamaları beklenmektedir. Bu görev için uzaklık sensörü ve hareket sensörü birlikte kullanılmalıdır. Uzaklık sensörü 180 derece etrafını tarayıp hedef bulunup bulunmadığını kontrol edecektir. Hedefi bulduğunda, bulduğu yönde duracaktır. Hareket sensörü ise robotu bu yöne doğru hareket etmesi için kullanılacaktır. Bu görev için robot tasarımının değiştirilmesi gerekmektedir. Aşağıda robotun inşa adımları verilmiştir. Robot inşa adımları öğrenciler ile paylaşılmalıdır.



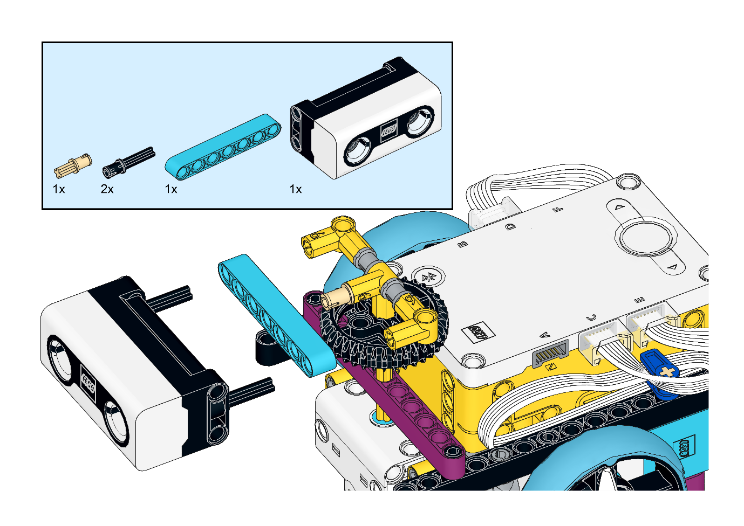
*Resim 9.5 Hedef Tespit Robotu Birinci Adım*



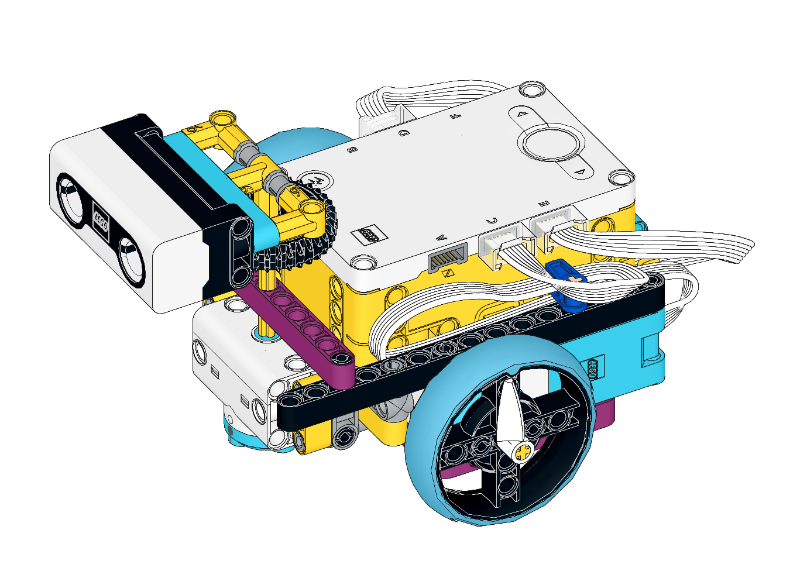
*Resim 9.6 Hedef Tespit Robotu İkinci Adım*



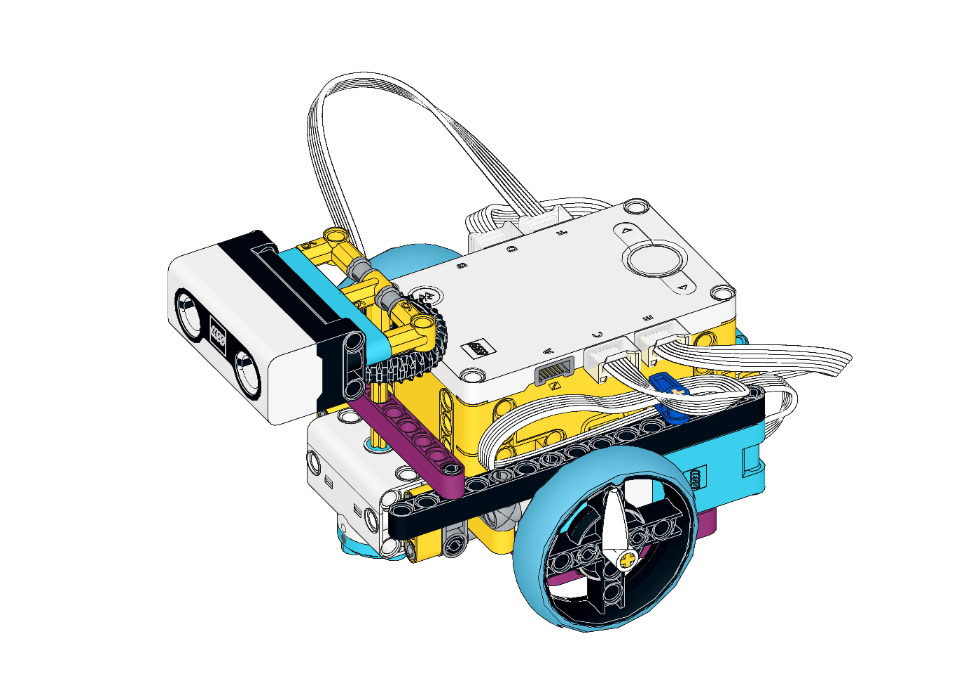
*Resim 9.7 Hedef Tespit Robotu Üçüncü Adım*



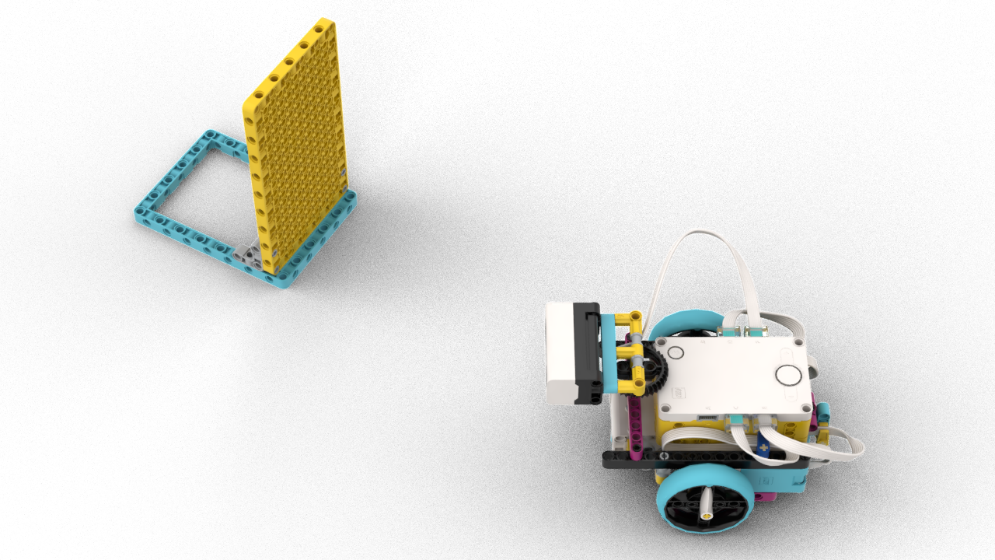
*Resim 9.8 Hedef Tespit Robotu Dördüncü Adım*



Resim 9.9 Hedef Tespit Robotu Beşinci Adım



*Resim 9.10 Hedef Tespit Robotu Altıncı Adım*



*Resim 9.11 Hedef Tespit Robotu Görev Alanı*

Öğrenciler robot üzerinde gerekli değişikliği yapıp hedefi hazırladıktan sonra rehber öğretmen robotun çalışmasını biraz daha detaylandırır. Mesafe sensörü 10’ar derecelik açılar ile dönerek önündeki alanı 180 derece tarar. Taradığı alan içerisindeki 50 cm mesafede bir hedef varsa robot mesafe sensörü aracılığı ile birkaç adeğer okur ve bu değerlerin ortalamasını alarak hedef tespiti yapar. Robot hedefe yönelerek ona doğru hareket eder ve ona 10 cm kala durur. 180 derece tamamlandığında bir hedef bulamazsa 10 cm ileriye gider ve yeniden mesafe sensörü ile tarama işlemine başlar ve bu işlemleri hedefi bulana kadar tekrar eder.

*Tanımlama*: Öğrenciler kodu yazmaya başlamadan önce problemi kâğıt kalem kullanarak tanımlamalıdır. Öğrenciler doğrudan kod yazmaya geçmek isteyebilir. Rehber öğretmen onları tanımlama adımına yönlendirmelidir. Öğrencilerin problemi çözmeden önce onun tasarımı hakkında düşünmeleri önemlidir. Aşağıda örnek tanımlama adımları verilmiştir. Bu adımlar öğrenciler ile paylaşılmamalıdır. Öğrenciler kendi adımlarını oluşturmalıdır.

1. Hedef robotun önünde bulunacağı için 180 derecelik tarama yeterlidir.
2. 180 derece belirli adımlara bölünerek, adım adım taranmalıdır.
3. Hedefin tespiti mesafe sensöründen gelen bir değer üzerinden yapılmamalıdır. Yer tespiti için mesafe sensöründen gelen birkaç değerin ortalaması alınmalıdır.
4. Tarama esnasında bir hedef tespit edilirse robot hedefe doğru yönelip hareket etmeli ve hedefe 10 cm kadar yaklaşmalıdır.
5. Hedef tespit edilemezse robot 10 cm ilerleyip tespit alanında ilerlemelidir.
6. Robot hedefi tespit edene kadar ii-iv adımları tekrarlanmalıdır.

*Fikir Üretme:* Öğrenciler bu adımda çözüme dair daha somut çözüm önerileri sunmalıdır. Algoritmayı detaylandırma, hangi programlama yapılarını kullanacağına, algoritmanın gerçekleştirilmesi için ne gibi programlama tekniklerinin kullanacağına karar verip tanımlama bu adımda gerçekleştirilir. Aşağıda örnek adımlar verilmiştir. Bu adımlar rehber öğretmen içi örnek niteliğindedir. Öğrenciler kendi adımlarını oluşturmalıdır.

1. Mesafe sensörü adımlarla 180 derece tarama yapacaktır. Adımlar 3, 5 veya 10 derece olarak belirlenebilir. Adım derecesine kod uygulandıktan sonra hedefe yaklaşma hata payı ve geçen süre göz önünde bulundurularak karar verilebilir.
2. Robotun hedefe yönelme açısı mesafe sensörünün hedefi tespit ettiği açı olmalıdır. Bu açı mesafe sensörünün bağlı olduğu motordan edinilebilir. Bu işlemin benzeri daha önce *“Motor Açısına Göre Hareket Eden Robot (Hafta 8)”* etkinliğinde gerçekleştirilmiştir. Orada öğrenilen deneyimler burada kullanılabilir.
3. Büyük motorun başlangıç açısı 90 derece olarak ayarlanabilir. Başlangıcın ardından her bir taramada büyük motor -90 ile 90 derece arasında tarama işlemi yapmalıdır.
4. Tespit işlemi, ilk tespit anında bitirilmek yerine üst üste birkaç tane tespit yapılarak bunların ortalaması hedef açı olarak alınabilir.
5. Hedefe 10 cm yaklaşmak için mesafe sensörü kullanılacaktır. Bunun için mesafe sensörünün olayları kullanılabilir.

## ÜRET

Hedef Tespit Robotu

Öğrenciler tasarlama adımının ardından, ürettikleri fikirleri uygulamaya geçirmek için robot başına geçerler ve gerekli kodları yazmaya başlarlar. Rehber öğretmen bu adımda öğrencilerin sorduğu sorulara doğrudan yanıtlar vermemelidir. Problemi çözüp çözmeme sorumluluğu öğrencilere aittir. Rehber öğretmen bu adımda sadece yönlendirme yapmalıdır. Kesinlikle sorunun çözümünü öğrenciler ile paylaşmamalıdır. Rehber öğretmen için örnek bir çözüm aşağıda paylaşılmıştır.

hub = PrimeHub()

hub.light\_matrix.show\_image("TARGET")

mesafe\_sensoru=DistanceSensor("F")

surus\_tabani=MotorPair("C","D")

buyuk\_motor=Motor("E")

surus\_tabani.set\_default\_speed(20)

buyuk\_motor.set\_default\_speed(30)

buyuk\_motor.run\_to\_position(0)

buyuk\_motor.run\_to\_position(90)

buyuk\_motor.set\_degrees\_counted(90)

bulunmadi=True

hedef\_acilari=[]

hedef\_acisi=0

while bulunmadi:

    if buyuk\_motor.get\_degrees\_counted()>-90:

        buyuk\_motor.run\_for\_degrees(-10)

        wait\_for\_seconds(0.2)

        mesafe=mesafe\_sensoru.get\_distance\_cm()

        if mesafe!=None and mesafe<50:

            sayac=1

            buyuk\_motor.set\_default\_speed(20)

            while mesafe!=None and mesafe<50 and sayac<=5:

                hedef\_acilari.append(-1\*buyuk\_motor.get\_degrees\_counted())

                buyuk\_motor.run\_for\_degrees(-5)

                wait\_for\_seconds(0.2)

                mesafe=mesafe\_sensoru.get\_distance\_cm()

                hub.speaker.beep(90)

                sayac+=1

            bulunmadi=False

    else:

        buyuk\_motor.run\_to\_position(0)

        buyuk\_motor.run\_to\_position(90)

        surus\_tabani.move(10,"cm")

for aci in hedef\_acilari:

    hedef\_acisi+=aci

hedef\_acisi=hedef\_acisi/len(hedef\_acilari)

buyuk\_motor.run\_to\_position(0)

hub.motion\_sensor.reset\_yaw\_angle()

if hedef\_acisi>0:

    while hub.motion\_sensor.get\_yaw\_angle()<hedef\_acisi:

        surus\_tabani.start\_tank(20,-20)

    surus\_tabani.stop()

else:

    while hub.motion\_sensor.get\_yaw\_angle()>hedef\_acisi:

        surus\_tabani.start\_tank(-20,20)

    surus\_tabani.stop()

surus\_tabani.start()

mesafe\_sensoru.wait\_for\_distance\_closer\_than(10)

surus\_tabani.stop()

for i in range(8):

    hub.speaker.beep(100)

    hub.light\_matrix.show\_image("YES")

    wait\_for\_seconds(0.2)

    hub.light\_matrix.off()

    wait\_for\_seconds(0.2)

## DEĞERLENDİR

Günün sonunda rehber öğretmen öğrencilere aşağıdaki soruları yönelterek öğrenilenlerin üzerine sınıfça düşünmesini sağlar.

* Fonksiyon nedir?
* Fonksiyon ne zaman kullanılır?
* Fonksiyonları kullanmanın avantajları nelerdir?
* Hedef Tespit Robotu etkinliğinde fonksiyonlar kullanılabilir mi? Yanıtınızı açıklayınız.
* Bugün gerçekleştirdiğiniz programlama etkinliklerinde en çok zorlandığınız şeyler (örn. konu, kavram, işbirlikli çalışma ve yöntem) neler oldu? Bunları aşmak için neler yaptınız?
* Mesafe sensörü ile bir proje yapsaydınız nasıl bir proje yapardınız?
* Bir sonraki derse daha verimli bir öğrenme deneyimi elde etmek için neleri yapıp neleri yapmamanız gerekir?

## İLAVE ETKİNLİKLER

### Kaç Duvar Olduğunu Sayan Robot

Bu etkinlikte aşağıdaki resimde gösterildiği gibi duvarlar aralıklarla yan yana dizilecektir. Duvarlar arasında 10-15 cm mesafe bulunacaktır. Duvar için kitap ve mukavva benzeri bir cisim kullanılabilir. Resimde sadece üç tane duvar gösterilmiştir. Duvar sayısı öğrenciler için belirsiz olacaktır. Öğretmen istediği kadar duvar koyabilir. Öğrenciler robotu duvardan istedikleri bir uzaklığa duvara paralel olacak şekilde koyabilirler.



*Resim 9.12 Duvar Sayan Robot*

Öğrenciler bir kod yazarak öğretmenin kaç adet duvar koyduğunu buldurmalıdır. Duvar sayısı bulunduktan sonra robot durmalı, bir adet bip sesi çalmalı ve ışık matrisinde kaç adet duvar bulunduğunu yazmalıdır. Bu etkinlik yapılırken öğrenciler doğrudan kod yazmaya başlamamalı, öncelikle yukarıda bir örneği verilen tasarlama adımlarını (tanımlama ve fikir üretme) gerçekleştirmelidir.

### LEGO Kutusu Etrafında Dönen Robot Yarışması

Bu etkinlikte öğrencilerden Spike Prime kutusu etrafında dönen robot kodunu yazmaları istenir. Öğrenciler kutunun kenar uzunluklarını tahmini veya gerçek ölçüm olarak kullanamazlar. Robot bir duvar takip algoritması ile solundaki duvarı takip ederken boşluk ile karşılaştığında dönmelidir. Duvar takip etme ve dönme algoritmaları sürekli tekrar edildiğinde robot kutunun etrafında dönmüş olacaktır. Rehber öğretmen öğrencilere gerekli açıklamaları yapar, onlara makul bir süre verir. Süre sonunda bütün gruplardan görevi kutu etrafında robotlarını kullanarak gerçekleştirmelerini ister.

Kazanan grubu öğretmen ve grup puanları belirleyecektir. Her grup diğer gruplara puan verecektir. Gruplar kendilerini puanlayamaz. Öğretmen de her gruba bir puan verecektir. Verilen puanlar toplandığında en yüksek puanı alan grup yarışmayı kazanır. Grupların puanlarında eşitlik olması durumunda kazananı öğretmen belirleyecektir.

|  |
| --- |
| **PROJE HAZIRLIYORUM- TANIMLAMA ve EMPATİ AŞAMASI** |
| Öğrenciler çevrelerinden veya günlük hayatlarından uygun bir problem belirleyip bu probleme kendi tasarladıkları bir robot ile çözüm üreteceklerdir. Geçen hafta proje grupları oluşturulmuş ve proje konusunun araştırılması istenmiştir.  Bu hafta gruplardan belirledikleri robot projesi için empati çalışması yapmaları ve problemlerini tanımlamaları beklenir. Açıklamalarının aşağıdaki bilgileri içermesi gerektiği belirtilir. Empati çalışması sonucunu paylaşmaya başlamadan önce gruplara hazırlanma, gözden geçirme ve tekrar etme için süre verilir.   * Problem cümlesi: Robotun ne yapması isteniyor? * Robotun kullanılması planlanan ortamdaki ihtiyaçlar nelerdir? * Robotun kullanılması planlanan ortamdaki beklentiler nelerdir?   Açıklanan projelerle ilgili olarak diğer öğrencilerin ve rehber öğretmenlerin fikirleri alınır.  **Rehber Öğretmene Not:** Öğrenci tarafından belirlenen robot görevlerinin gerçekleştirilebilir olması önemlidir. Atölyelerde bulunan imkânlarla bu görevin başarılabilir olması gerekir. Bu safhada gerçekleştirilemeyecek bir görevin seçilmesi veya öğrencilerin bu tür bir projeye yönlendirilmesi, proje sürecinde vakit kaybetmelerine ve motivasyonlarının olumsuz etkilenmesine neden olabilir. Bu nedenle rehber öğretmenin proje önerilerini çok iyi incelemesi/irdelemesi ve proje önerilerinin gerçekleştirilebilir/yapılabilir olduğundan emin olması gerekir.  **Sonraki Haftaya Hazırlık:** Öğrencilerden, sonraki hafta için belirledikleri projeler için "tanımlama" adımlarını gerçekleştirmeleri istenir. Tanımlama adımında, "empati" adımında toplanan bilgiler bir bütün hâline getirilir. Süreçte ortaya çıkabilecek problemleri çözebilmek için nelerin yapılması gerektiği ile ilgili fikirler ortaya konur. Çözümün başarılı olabilmesi için ihtiyaçların iyi tanımlanmış olması ve bu ihtiyaçların giderilmesi için uygun çözüm önerilerinin sunulması önemlidir. |