# 8. Hafta – Hareket (Gyro) Sensörü

## Haftanın Amacı:

Bu haftanın amacı öğrencilerin hareket (gyro) sensörü hakkındaki temel bilgi ve becerileri kazanmasını sağlamaktır. Hareket sensörünün üç ekseli dönme hareketi kullanılarak çeşitli etkinlikler yapılacaktır. Hareket sensörünün yön özelliği kullanılarak farklı programlar yazılacaktır. Hub hareketleri temel alınarak robot kodları yazılacaktır. Üç eksenli açı, yön ve hareket özellikleri Python programlama dili ile birlikte problem çözümünde kullanılacaktır. Öğrencilerden bu özellikleri kullanarak orta-ileri seviye algoritmalar geliştirmeleri ve kod yazmaları beklenmektedir.

## Haftanın Kazanımları:

* Öğrenciler hareket sensörünü ve sensörün nasıl çalıştığını bilirler.
* Öğrenciler hareket sensörünün üç eksenli dönme açısını Python komutlarını kullanarak tespit eder ve program içerisinde kullanabilir.
* Öğrenciler sapma açısını kullanarak robotun çeşitli dönme hareketleri yapmasını sağlarlar.
* Öğrenciler Hub’ın yön bilgisini kullanarak orta-ileri seviyede programlar yazabilirler.
* Öğrenciler Hub’ın sallanama, tıklatılma ve düşme durumlarını Python programlama dilini kullanarak saptar ve bu bilgiyi problem çözümünde kullanabilir.

## Kullanılacak Malzemeler:

Robot seti, bilgisayar ve çalışma alanı

## GÖZLE VE UYGULA

|  |
| --- |
| ***Rehber Öğretmen İçeriği –*** ***Hareket (Gyro) Sensörü Hakkında*** |
| *Gyro sensörü olarak da isimlendirilen hareket sensörü üç eksenli açısal hız ölçümü (jiroskop) ve üç eksenli ivme ölçümü olmak üzere toplamda altı eksenli ölçüm yapabilen bir sensördür. Ayrıca hareket sensörü Hub’ın sallandığı, düştüğü veya Hub’a dokunulduğu bilgisini Python komutları aracılığı ile programcıya sağlayabilmektedir. Bu tip sensörler günümüzde akıllı telefonlar, navigasyon sistemleri, oyun kumandaları ve robotlar gibi birçok teknolojik üründe kullanılmaktadırlar. Spike Prime robot setinde (EV3 robot setinin aksine) hareket sensörü Hub ile bütünleşiktir. Ayrıca EV3 robot setinde yaşanan kayma problemi (robotun dönme hareketi durduğu halde, sensörün açı değerinin artmaya devam etmesi) Spike Prime setinde giderilmiştir.*  *Hareket sensörü elektronik programlama bağlamında farklı bir anlama gelmektedir. Elektronik programlamada hareket sensörü, merdiven otomatlarında olduğu gibi, etrafta canlı veya cansız bir nesnenin hareket edip etmediğini kontrol etmek için kullanılır. Prime robot setinde ise hareket sensörü üç eksenli jiroskop ve üç eksenli ivme ölçer barındırmaktadır. Rehber öğretmen öğrencilere bu terminolojiden bahseder ve öğrencilere Elektronik Programlama ve Nesnelerin İnterneti dersinde kullanılacak olan hareket sensörünün buradakinden farklı olduğunu vurgular.*  *Spike Prime robot setinde Hub’ın üç eksenli açısal hareketi ve bu eksenlerde yaptığı açı değişim miktarı ile ilgili ölçümler yapılabilir. Üç eksende yapılan açısal hareket için sapma (yaw), yuvarlanma (roll) ve yunuslama (pitch) isimleri kullanılmıştır. Sapma (yaw) ismi sürüş tabanının tekerleri yerde iken sağa veya sola doğru yaptığı hareket için kullanılır bu hareket aşağıdaki resimde gösterilmiştir.*    *Resim 8.1 Sapma Açısı*  *Yuvarlanma bir merdanenin yuvarlanmasına benzetilebilir. Yuvarlanma sürüş tabanının sağ veya sol yanına yatmasını ifade etmek için kullanılır. Bu hareket için aşağıdaki resim örnek olarak verilebilir.*    *Resim 8.2 Yuvarlanma Açısı*  *Son hareket şekli ise yunuslama olarak adlandırılmıştır. Yunusların su yüzüne inip çıkarken yaptıkları harekete benzemektedir. Bu hareket şeklinde sürüş tabanının üstü veya altı kaldırılmaktadır. Yunuslama hareketi için aşağıdaki şekil örnek olarak gösterilebilir.*    *Resim 8.3 Yunuslama Açısı*  *Bu üç hareket şekli aslında üç boyutlu düzlemin her birindeki açısal hareketleri tarif etmek için kullanılmaktadır. Bunlar x, y ve z eksenlerindeki dönme hareketi olarak da adlandırılabilir. Öğrenciler x, y, z eksenlerini tam olarak gözünde canlandıramayabilir. Rehber öğretmen konuyu anlattıktan sonra* [*https://www.youtube.com/watch?v=pQ24NtnaLl8*](https://www.youtube.com/watch?v=pQ24NtnaLl8) *bağlantısındaki videoyu kullanarak öğrencilerin üç eksenli hareketi somutlaştırmasına yardımcı olabilir.* |

### Gözle: Hub’ın Açısal Hareketlerini Öğreniyorum

Bu etkinlikte amaç Hub’ın açısal hareketlerini öğretmeye yöneliktir. Sapma (yaw) hareketinin açısının alınması için *get\_yaw\_angle metodu* kullanılır. Açı değeri derece cinsindendir. Bu metodun kullanımı aşağıda gösterilmiştir.

hub = Primehub()

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

Yuvarlanma (roll) hareketi sonucunda oluşan açı değerinin ölçülmesi için *get\_roll\_angle* metodu kullanılır. Elde edilen açı değeri derece cinsindendir. Bu metodun yazım kuralı aşağıda gösterilmiştir.

hub.motion\_sensor.get\_roll\_angle()

Yunuslama (pitch) hareketi sonucunda oluşan açı değerlerini ölçmek içim *get\_pitch\_angle* metodu kullanılır. Bu metot derece cinsinden değer döndürür. Metodun kullaanımı aşağıda verilmiştir.

hub.motion\_sensor.get\_pitch\_angle()

Rehber öğretmen öğrencilere sapma, yuvarlanma ve yunuslama hareketlerini göstererek anlatır ve gerektiğinde video ile öğrencilerin somut olarak bu hareketleri gözlemlemelerini sağlar. Bu hareketler sonucunda oluşan açıların değerlerinin alınması için kullanılması gereken komutları anlatır. Rehber öğretmen aşağıdaki kodu tahtaya yansıtır ve öğrencilerin bu kodu yazıp çalıştırmalarını ister. Öğrencilerin Hub’ı ilgili eksenlerde döndürerek onların bu kod ile sapma, yuvarlanma ve yunuslama açılarındaki değişimi gözlemlemelerini sağlar.

while True:

    sapma=hub.motion\_sensor.get\_yaw\_angle()

    yuvarlanma=hub.motion\_sensor.get\_roll\_angle()

    yunuslama=hub.motion\_sensor.get\_pitch\_angle()

    print("Sapma Açısı: ", sapma)

    print("Yuvarlanma Açısı:", yuvarlanma)

    print("Yunuslama Açısı:", yunuslama)

    print("-------------------------------")

    wait\_for\_seconds(1)

Bu üç farklı hareket şeklinden sadece sapma hareketinde açı değerinin sıfırlanması sağlanabilir. Sürüş tabanı ilk çalıştırıldığında sapma açısı 0 derece olarak belirlenir. Daha sonraki hareketlerinde burayı başlangıç noktası olarak kabul eder ve sapma açısını ona göre belirler. Bazı durumlarda sapma açısının sıfırlanması gerekebilir. Bunun için *reset\_yaw\_angle* metodu kullanılır. Rehber öğretmen aşağıdaki kodu öğrencilere açıklayarak anlatır. *Reset\_yaw\_angle* metodu sayesinde son verilen açı miktarının bir anda değişiklik gösterdiğini vurgular.

hub = PrimeHub()

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

while hub.motion\_sensor.get\_yaw\_angle()<=160:

    surus\_tabani.start\_tank(20,0)

    print(hub.motion\_sensor.get\_yaw\_angle())

surus\_tabani.stop()

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

print(hub.motion\_sensor.get\_yaw\_angle())

### Gözle ve Uygula: Sürüş Tabanını 90 Derece Döndürüyorum

Bu etkinlikte sürüş tabanının 30 cm düz gittikten sonra 90 derece sağa dönmesini gerekmektedir. Fakat dönme işlemi için hareket sensörünün sapma açısı kullanılmalıdır. Aşağıdaki kod bu iş için kullanılabilir.

hub = PrimeHub()

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

surus\_tabani.move(30,"cm",0,80)

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

while hub.motion\_sensor.get\_yaw\_angle()<=90:

    surus\_tabani.start\_tank(80,0)

surus\_tabani.stop()

print(hub.motion\_sensor.get\_yaw\_angle())

Rehber öğretmen bu kodu çalıştırır ve sapma açısının dönme sonucunda 90 dereceden biraz daha büyük bir değere sahip olduğunu konsol ekranını ekrana yansıtarak gösterir. Öğrenciler ile bu değerin neden 90 dereceden büyük olduğu ve nasıl 90 değerine getirebileceği konusunda tartışılır.

|  |
| --- |
| Not |
| Sensör tam olarak 90 dereceyi ölçtüğü anda ölçülen verinin aktarılması ve ardından motorlara durma komutunun gönderilmesi biraz zaman alır. Ayrıca robot kendi etrafında dönerken oluşan dönme kuvveti, motorlar durdurulduktan sonra da robotun biraz daha dönmesine sebep olur. Robotun 90 dereceden biraz daha az bir açı ile dönecek şekilde programlanması bir çözüm olabilir. |

### Uygula: Kare şeklinde hareket

Bu etkinlikte bir kenarı 30 cm olan bir kare şeklinde hareket edecek olan robotun programı yazılacaktır. Bu görev için for döngüsünün içerisinde while döngüsü kullanılacaktır. Bu da bir iç içe döngüdür. Rehber öğretmen öğrencileri iç içe döngü kullanma konusunda yönlendirir. Rehber öğretmen için örnek bir program aşağıda sunulmuştur.

hub = PrimeHub()

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

for i in range(4):

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

    surus\_tabani.move(30,"cm",0,60)

    while hub.motion\_sensor.get\_yaw\_angle()<=87:#Çevresel faktörlerden

        surus\_tabani.start\_tank(50,0)           #etkilenebilir

    surus\_tabani.stop()                         #Gerekli düzeltme yapılmalıdır

|  |
| --- |
| ***Rehber Öğretmen İçeriği –Hub’ın Yönleri*** |
| *Hub’ın ön, arka, sağ, sol, üst ve alt olmak üzere altı farklı yönü bulunmaktadır. Bu yönleri belirlemek için Hub’ın butonlu yüzü cep telefonunun ekranı gibi düşünülmelidir. Hub’ın butonlu yüzü (cep telefonunun ekranı) size bakacak şekilde yere paralel hizaya getirilir. Görülen yüz ön yüzdür. Bunun tam arka tarafı size bakacak şekilde çevrilirse görülen yüz arka yüzdür. Hub’ın sağ tarafı (cep telefonunun sağ tarafı) size bakacak şekilde çevrilirse görülen yüz sağ yüz olacaktır. Hub’ın sol tarafı (cep telefonunun sol tarafı) size bakacak şekilde çevrilirse görülen yüz sol yüz olacaktır. Hub’ın alt tarafı (cep telefonunun altı) size bakacak şekilde çevrilirse görülen yüz alt yüz olacaktır. Hub’ın üst tarafı (cep telefonunun üstü) size bakacak şekilde çevrilirse görülen yüz üst yüz olacaktır. Hub’ın yüzleri aşağıdaki resimde gösterilmiştir.*    *Resim 8.4 Hub’ın Yönleri* |

### Gözle: Hub Yönünün Algılanması

Ayakları üzerinde duran bir kişinin elinde duran Hub’ın kişiye doğru bakan yüzünün bulunması için *get\_orientation* metodu kullanılır. Bu metodun kullanımı aşağıda gösterilmiştir.

hub.motion\_sensor.get\_orientation()

Bu metot altı farklı değer üretir. Bu değerler “front” (ön), “back” (arka), “rightside” (sağ), “leftside” (sol), “up” (üst) ve “down” (alt) olarak adlandırılmıştır.

Rehber öğretmen öğrencilerden aşağıdaki kodu yazıp çalıştırmalarını ister. Öğrencilerin Hub yönünü çalışan program yardımıyla anlamlandırmalarını sağlar.

hub = PrimeHub()

while True:

    print(hub.motion\_sensor.get\_orientation())

    wait\_for\_seconds(1)

### Uygula: Sürekli Yukarıyı Gösteren Ok Resimleri

Bu etkinlikte Hub’ın ekranında sürekli yukarıyı (sınıfın tavanını) gösterecek bir ok sembolü için gerekli program yazılacaktır. Robotun duruş şekli değiştikçe okun yönü de değiştirilerek robotun her konum için yukarıyı (sınıfın tavanını) göstermesi sağlanmalıdır. Rehber öğretmen için aşağıda örnek bir program verilmiştir.

hub = PrimeHub()

while True:

    print(hub.motion\_sensor.get\_orientation())

    hub.light\_matrix.off()

    if hub.motion\_sensor.get\_orientation()=="up":

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

    if hub.motion\_sensor.get\_orientation()=="down":

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

    if hub.motion\_sensor.get\_orientation()=="rightside":

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

    if hub.motion\_sensor.get\_orientation()=="leftside":

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

    if hub.motion\_sensor.get\_orientation()=="back":

        hub.light\_matrix.set\_pixel(2,2) #Orta Nokta, arkadan görünüm temsili

    if hub.motion\_sensor.get\_orientation()=="front":

        for i in range(5): #Çapraz, önden görünüm temsili

            for j in range(5):

                if i==j or i+j==4:

                    hub.light\_matrix.set\_pixel(i,j)

    wait\_for\_seconds(0.5)

### Uygula: Yunuslama Açısına Göre İleri Geri Dönen Tekerlekler

Bu etkinlikte öğrencilerden yunuslama açısına göre tekerleklerin dönüş yönünü belirleyen ve onları hızlandırıp yavaşlatan bir program yazmaları istenmektedir. Yunuslama açısı 0 ise tekerlekler duracaktır, pozitif yönlü ise ileri doğru gidecektir, negatif yönlü ise geri doğru gidecektir ve hareket hızı yunuslama açısının büyüklüğü ile orantılı olacak şekilde değişecektir. Rehber öğretmen için aşağıda örnek bir kod verilmiştir.

hub = PrimeHub()

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

while True:

    yunuslama\_acisi=hub.motion\_sensor.get\_pitch\_angle()

    surus\_tabani.start(0,yunuslama\_acisi)

|  |
| --- |
| Not |
| Bu etkinlikte öğrenciler sürüş tabanlarını ellerinde tutarak farklı yunuslama açıları oluşturacaktır. |

### Gözle: Hub ile Yön Değişimine Bağlı İşlem Yapma

Bazen yapılmak istenen işlemler Hub’ın yönü yerine, yön değişimine bağlı olarak gerçekleştirilmek istenebilir. Bu durumlarda Hub’ın yönü değişene kadar beklemeyi sağlayacak bir metot faydalı olacaktır. Bu işi için *wait\_for\_new\_orientation* metodu kullanılabilir. Bu metodun kullanımı aşağıda verilmiştir. Aşağıdaki kod kullanıldığında Hub’ın yönü değişene kadar beklenir ve yön değişikliği olduğunda elde edilen yeni yön *“yon”* değişkenine aktarılır.

yon = hub.motion\_sensor.wait\_for\_new\_orientation()

Rehber öğretmen aşağıdaki kodu öğrencilere vermeden önce *if-elif-else* seçim ifadesini öğrencilere anlatır. Ardından kodu tahtaya yansıtır ve öğrencilerden kodu yazıp, çalıştırıp sonuçlarını deneyimlemelerini ister. Kodun ne iş yaptığı sınıfça tartışılır.

hub = PrimeHub()

while True:

    print("Yeni Yön Bekleniyor...")

    yon = hub.motion\_sensor.wait\_for\_new\_orientation()

    print("Yeni Yön alındı")

    print("-------------------------------------")

    if yon=="front":

        print("Ön Yüz")

    elif yon=="back":

        print("Arka Yüz")

    elif yon=="rightside":

        print("Sağ Yüz")

    elif yon=="leftside":

        print("Sol Yüz")

    elif yon=="up":

        print("Üst Yüz")

    else:

        print("Alt Yüz")

### Uygula: Sürekli Yukarıyı Gösteren Ok Resimleri - Yeni Versiyon

Sürekli yukarıyı gösteren ok resimleri isimli etkinlikte öğrencilerden Hub’ın ekranında sürekli yukarıyı (sınıfın tavanını) gösterecek bir ok sembolü bulunan programı yazmaları istenmiştir. Işık matrisinde ok gösterme işlemi 0.5 saniye aralıklarla gerçekleştirilmiştir. Bu etkinlikte 0.5 saniye gibi bir süre beklemeden Hub’ın yönü değiştiğinde yeni ok yönü ışık matrisinde gösterilmelidir. Bu iş için *wait\_for\_new\_orientation* metodu kullanılabilir. Öğrencilerin yazdığı kodda çalışma esnasındaki ilk yön gösterilmeyebilir. Fakat bu etkinlikte ilk yönün de gösterilmesi beklenmektedir. Rehber öğretmen için örnek bir kod aşağıda verilmiştir.

sayac=1

hub = PrimeHub()

while True:

    if sayac==1:

        yon=hub.motion\_sensor.get\_orientation()

        sayac+=1

    else:

        yon=hub.motion\_sensor.wait\_for\_new\_orientation()

    hub.light\_matrix.off()

    if yon=="up":

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

    if yon=="down":

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

    if yon=="rightside":

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

    if yon=="leftside":

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

    if yon=="back":

        hub.light\_matrix.set\_pixel(2,2)

    if yon=="front":

        for i in range(5):

            for j in range(5):

                if i==j or i+j==4:

                    hub.light\_matrix.set\_pixel(i,j)

### Gözle: Hub’a Tıklama, Hub’ı Sallama ve Düşürme Hareketleri

Hareket sensörü Hub üzerine tıklama, çift tıklama, Hub’ın sallanması ve düşürülmesi gibi hareketleri algılayabilmektedir. Hub hareketlerinin algılanabilmesi için *get\_gesture* metodu kullanılabilir. Bu metodun kullanımı aşağıda örneklendirilmiştir.

hareket=hub.motion\_sensor.get\_gesture()

*Get\_gesture* metodu “tapped” (tıklandı), “doubletapped” (çift tıklandı), “shaken” (sallandı) ve “falling” (düşüyor) değerlerini üretebilir. Bunun yanında herhangi bir Hub hareketi yapılmamışsa *None* değerini döndürür. Rehber öğretmen öğrencilerin aşağıdaki kodu yazıp çalıştırmalarını ve sonuçlarını deneyimlemelerini sağlar.

hub = PrimeHub()

while True:

    hub\_hareketi=hub.motion\_sensor.get\_gesture()

    if hub\_hareketi!=None:

        print(hub\_hareketi)

Öğrenciler kodun çalışmasını deneyimlerken şu hususlara dikkat etmelidirler:

* Tıklama için biraz sert vurmak gerekmektedir. Tıklamayı (“tapped”) deneyimlemek için yumuşak bir dokunuş ile başlanıp yavaş yavaş sert vurmaya doğru ilerlenerek eşik değeri bulunabilir.
* Çift tıklama (“doubletapped) için daha önce kullanılan tıklama sertliğinde ve hızlı bir şekilde iki tıklama yapılmalıdır. Aksi takdirde çift tıklama gözlenemeyebilir.
* Düşme hareketi için öğrenciler robotu yumuşak bir zemine 20 cm gibi bir mesafeden bırakmalıdırlar. Daha yüksek değerler robota zarar verebilir.

### Uygula: Hub Hareketine Göre Ses Çıkaran Robot

Bu etkinlikte Hub’ın farklı hareketlerine göre hoparlörden farklı notalarda ses çıkaran robot programı yapılacaktır. Öğrenciler tıklama, çift tıklama, sallama ve düşme hareketleri için farklı notalar belirlemelidir. Bu görev için aşağıda örnek kod verilmiştir.

hub = PrimeHub()

while True:

    hub\_hareketi=hub.motion\_sensor.get\_gesture()

    if hub\_hareketi=="tapped":

        hub.speaker.beep(62)

    if hub\_hareketi=="doubletapped":

        hub.speaker.beep(64)

    if hub\_hareketi=="shaken":

        hub.speaker.beep(65)

    if hub\_hareketi=="falling":

        hub.speaker.beep(67)

### Gözle: Yeni Bir Hub Hareketi İçin Bekleme

Bazı durumlarda bir Hub hareketi yapılana kadar programın duraklatılması veya bekletilmesi istenebilir. Bu gibi durumlarda *wait\_for\_new\_gesture* metodu kullanılabilir. Bu metodun kullanımı aşağıda örneklendirilmiştir.

hub\_hareketi=hub.motion\_sensor.wait\_for\_new\_gesture()

Bu metot “tapped”, “doubletapped”, “shaken” ve “falling” değerlerini döndürebilir. Fakat None değerini döndürmez.

### Uygula: Hub’a Tıklandığında Harekete Başlayan İkinci Tıklamada Duran Robot

Bu etkinlikte Hub’a tıklandığında sürüş tabanı harekete başlamalıdır. Ardından Hub’a yeni bir tıklama gerçekleştirildiğinde sürüş tabanı durmalıdır. Yeniden tıklandığında hareket edip, bir daha tıklandığında durmalıdır. Bu işlem sürekli tekrarlanmalıdır. Rehber öğretmen öğrencilere çözülmesi istenen problemi anlatır. Rehber öğretmen için örnek bir program aşağıda verilmiştir.

hub = PrimeHub()

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

zamanlayici=Timer()

birinci=zamanlayici.now()

sayac=1

while True:

    hub\_hareketi=None

    ikinci=zamanlayici.now()

    if ikinci-birinci>0.5:

        hub\_hareketi=hub.motion\_sensor.wait\_for\_new\_gesture()

    print(hub\_hareketi)

    print(sayac)

    print

    if sayac==1 and hub\_hareketi=="tapped":

        surus\_tabani.start(0,50)

        sayac+=1

    elif sayac==2 and hub\_hareketi=="tapped":

        surus\_tabani.stop()

        sayac=1

        birinci=zamanlayici.now()

Öğrencilerin yazdığı programda ikinci tıklamadan sonra, sürüş tabanının durması esnasında yaptığı, ileri savrulup geri gelme hareketi Hub tarafından yeni bir tıklama olarak algılanabilir. Böylece sürüş tabanı durur durmaz yeniden harekete başlayabilir. Bu yüzden yukarıdaki programda hareketin başlayabilmesi için iki tıklama arasında 0.5 saniyelik bir zaman dilimi gerekliliği belirlenmiştir. Savrulma hareketi bu 0.5 saniyelik dilimde bitmiş olacağı için bahsi geçen hatalı durum ortadan kalkmış olacaktır.

Sınıf içerisinde savrulma esnasında durup yeniden başlama olayı gerçekleşirse öğrenciler ile bu durumun nedeni tartışılır. Eğer vakit kaldıysa rehber öğretmen yukarıdaki kodu kullanarak öğrencilerin bu istenmeyen durumdan nasıl kurtulabileceğini açıklar. Diğer etkinlikler için yeterli vakit yok ise tartışma sonrasında herhangi bir açıklama yapmadan öğrencilerin ders sonrasında bu durumun nasıl çözülebileceği üzerine düşünmeleri istenir.

## TASARLA

Hareket Sensörü ile Programlanabilen Robot

Bu etkinlikte öğrenciler programlanabilen bir robot geliştireceklerdir. Robota komutları Hub’ın yönleri kullanılarak verilecektir ve robotun bu komutları otomatik olarak yerine getirmesi sağlanacaktır. Bu işlemler için gerekli yönerge aşağıda verilmiştir:

* Sağ yön (“rightside”) robotun sağ tarafa doğru kendi ekseni etrafında 90 derece dönmesini sağlar,
* Sol yön (“leftside”) robotun sol tarafa doğru kendi ekseni etrafında 90 derece dönmesini sağlar,
* Üst yön (“up”) robotun 50 hızıyla doğrusal bir şekilde 10 cm ileri gitmesini sağlar,
* Alt yön (“down”) robotun 50 hızıyla doğrusal bir şekilde 10 cm geri gitmesini sağlar,
* Her yeni komuttan önce robot ön yönüne (“front”) doğru çevrilmelidir,
* Robotun verilen komutları yerine getirmesi için yavaşça arka yüzü (“back”) çevrilmeli, robot tekerleri üzerinde masaya bırakılmalı ve robotun üzerine çift tıklanmalıdır (“doubletapped”).

Örneğin robota iki kere ileri, bir kere sağa ve bir kere geri komutları şu şekilde verilir: üst, ön, üst, ön, sağ, ön, alt, ön, (yavaşça) arka, robotu masaya bırak, çift tıkla. Robot çift tıklama işleminin ardından iki kere ileri gidecek, bir kere sağa dönecek ve bir kere geri gidecektir.

Programı yazmaya başlamadan önce grupların tasarlama adımı için, daha önce örnekleri verilen, tanımlama ve fikir üretme sürecini gerçekleştirmeleri gerekir.

## ÜRET

Öğrenciler tasarımlarını yaptıktan sonra bilgisayar ve robot başında çalışarak istenen görevi yerine getirmelidir. Aşağıdakine benzer bir program hazırlayabilirler. Öğrenciler bu programın aynısını yapmak zorunda değildir. Aynı problem farklı yollardan çözülebilir.

hub = PrimeHub()

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

komut\_listesi=[]

baslasin\_mi=None

while True:

    komut=hub.motion\_sensor.wait\_for\_new\_orientation()

    if komut!="back" and komut!="front":

        komut\_listesi.append(komut)

    elif komut=="back":

        baslasin\_mi=hub.motion\_sensor.wait\_for\_new\_gesture()

        while baslasin\_mi!="doubletapped":

            baslasin\_mi=hub.motion\_sensor.wait\_for\_new\_gesture()

        if len(komut\_listesi)!=0:

            del komut\_listesi[-1]

        for komut in komut\_listesi:

            if komut=="rightside":

                surus\_tabani.move(90,"degrees",100,50)

            if komut=="leftside":

                surus\_tabani.move(90,"degrees",-100,50)

            if komut=="up":

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

            if komut=="down":

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

        komut\_listesi.clear()

Bu programda çeşitli iyileştirmeler yapılabilir. Örneğin robotun verilen komutları yerine getirmesi için arka yüzü çevrilmeli fakat bu işlem yavaşça yapılmalıdır. Eğer hızlı bir şekilde yapılırsa verilen son komut çalıştırılmayabilir. Başka bir örnek olarak program çalışırken ışık matrisinde hareket edilen yöne dair ok işaretleri gösterilebilir ve hareketin çeşidine göre hoparlörden farklı uyarı sesleri çalınabilir. Program üzerinden bu ve benzer alanlarda iyileştirmeler yapılabilir fakat pedagojik olarak bu seviyedeki program yeterli görüldüğü için bu iyileştirmeler yapılmamıştır. Vakit kalması durumunda sınıfça bu iyileştirmelerin nasıl yapılabileceği tartışılır ve yapılır.

## DEĞERLENDİR

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

* Aşağıdaki kodun çıktısı ne olur (Öğrenciler programı çalıştırmadan çıktıyı tahmin etmelidir)?

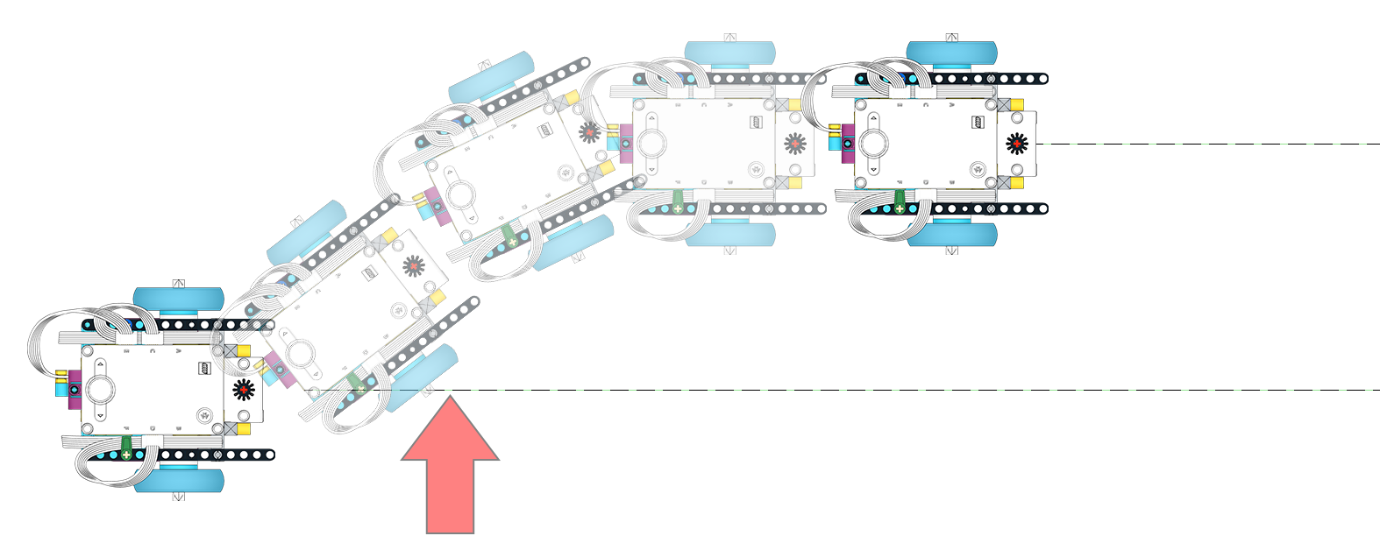
|  |
| --- |
| liste=[]  for i in range(5):      ic\_liste=[]      for j in range(5):          ic\_liste.append(i+j)      liste.append(ic\_liste)  for i in range(5):      for j in range(5):          if i==j:              print(liste[i][j]) |

* Bugün gerçekleştirdiğiniz programlama etkinliklerinde en çok zorlandığınız şeyler (örn. konu, kavram, işbirlikli çalışma ve yöntem) neler oldu? Bunları aşmak için neler yaptınız?
* Hareket sensörü günlük hayatta nerelerde kullanılabilir?
* Bir sonraki derse daha verimli bir öğrenme deneyimi elde etmek için neleri yapıp neleri yapmamanız gerekir?

## İLAVE ETKİNLİKLER

Yönünü Kaybetmeyen Robot

Bu etkinlikte öğrencilerden belirli bir yönde giderken dışarıdan yapılan müdahalelere tepki göstererek yeniden başlangıç yönüne dönen robot kodu yazılacaktır. Robot belirli bir yönde doğrusal gidecek şekilde kodlanacaktır. Ardından dışarıdan yapılan bir müdahale ile robotun yönü değiştirildiğinde Resim 8.5’te gösterildiği gibi robot kendisini toparlayıp eski yönüne yönelmelidir. Örneğin robot başlangıçta güneye doğru doğrusal ilerlerken, rehber öğretmenin müdahalesi ile yönü doğuya çevrilen robot yönünü tekrar güneye çevirmelidir.



*Resim 8.5 Yönünü Kaybetmeyen Robot*

Aşağıda bu görev için kullanılabilecek örnek bir program verilmiştir. Program sapma açısına ters yönde hareket ile tepki verecek şekilde tasarlanmıştır. Sapma açısının 3 ile çarpılması ve hızın 80 olarak belirlenmesi programın yazıldığı koşullar için geçerlidir. Öğrenciler bu değerleri kendi koşullarına göre uyarlayabilirler.

hub = PrimeHub()

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

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

hub.left\_button.wait\_until\_pressed()

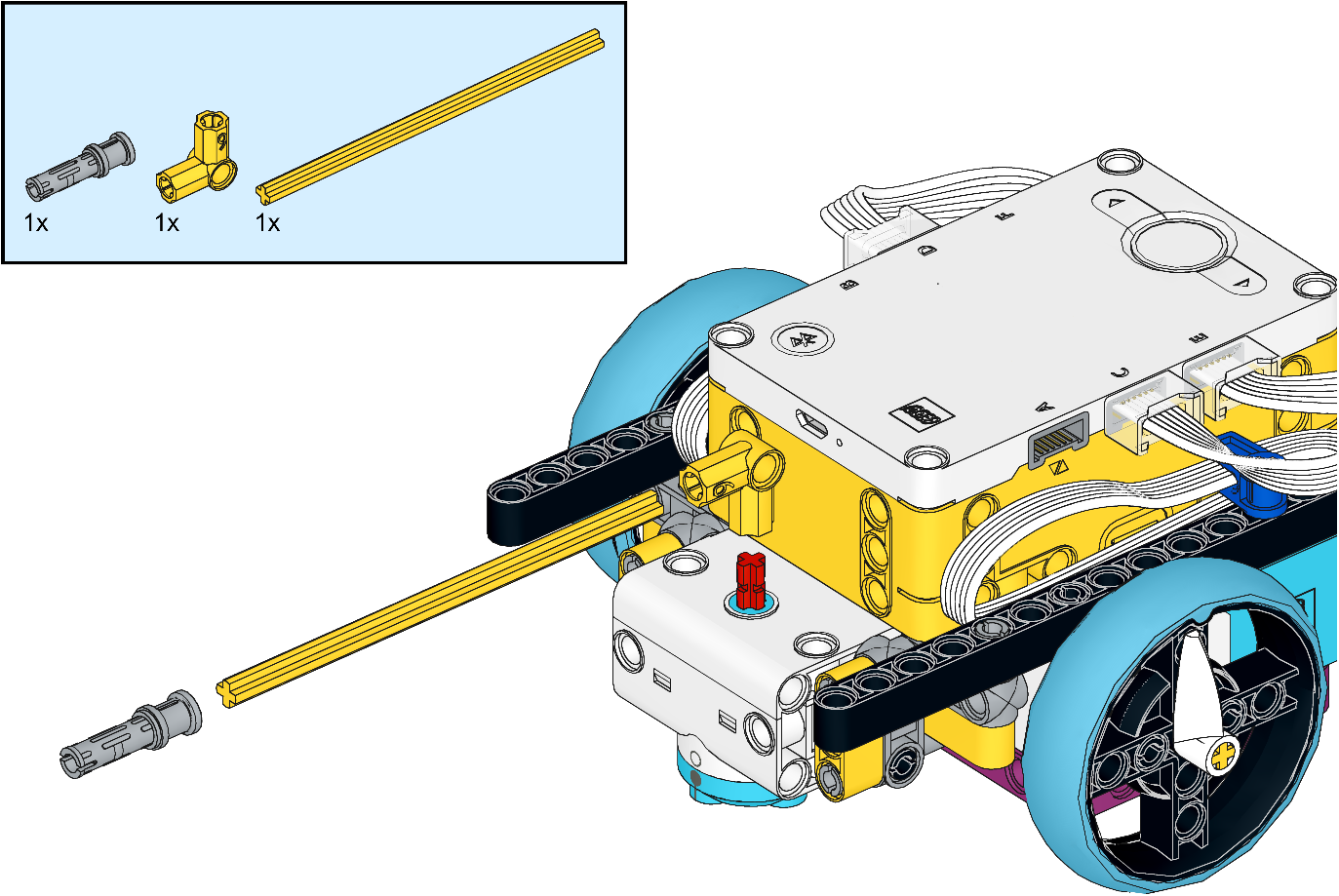
while True:

    sapma\_acisi=hub.motion\_sensor.get\_yaw\_angle()

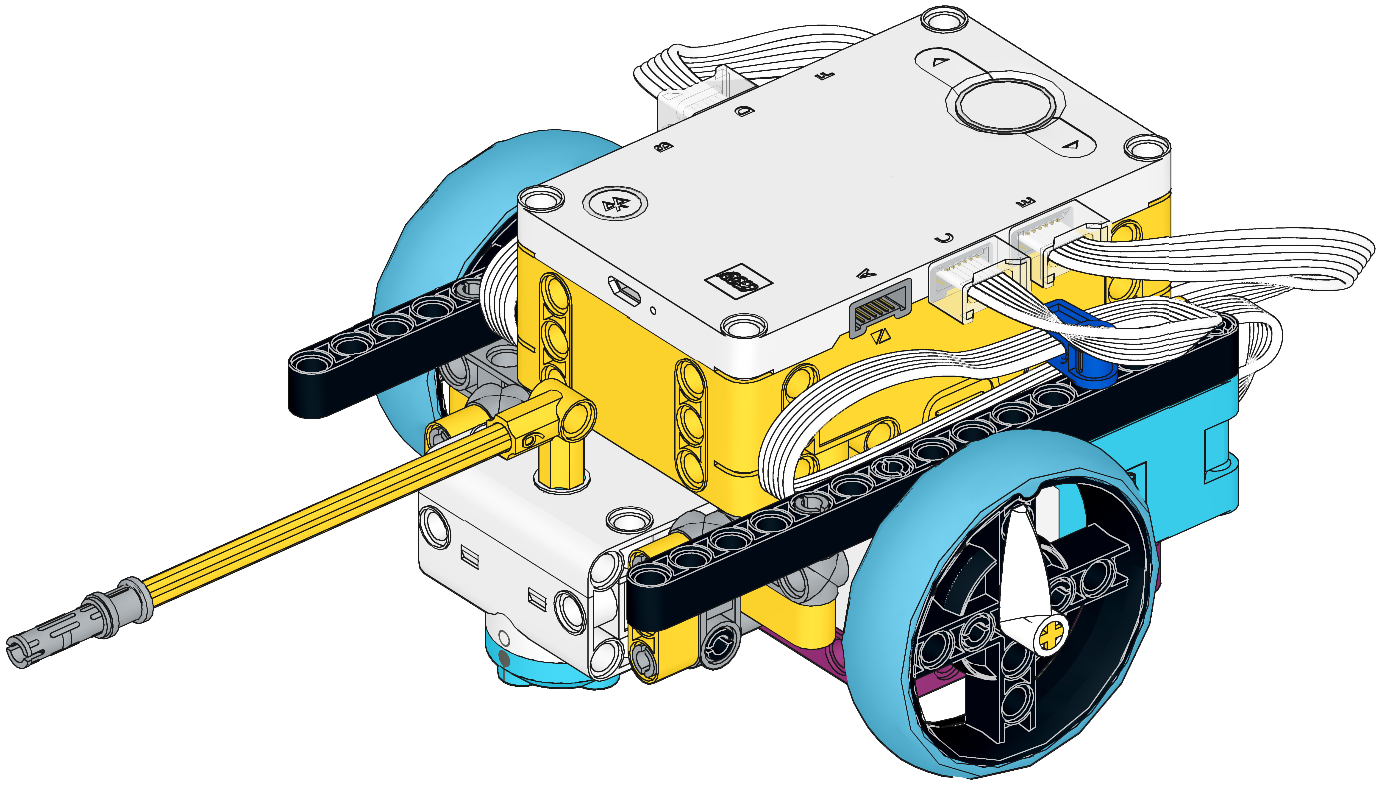
    surus\_tabani.start(-3\*sapma\_acisi,80)

Hedef Yön Koluna Yönelen Robot

Bu etkinlikte durmakta olan robot için hedef yön kolu kullanılarak bir hedef belirlenecektir. Hedefin yönü gösterildikten sonra Hub’a çift tıklanmasının ardından robotun yönünü hedefe doğru çevirmesi sağlanacaktır. Robot bu işlemi sürekli tekrarlamalıdır. Etkinliğin gerçekleştirilebilmesi için sürüş tabanında değişiklik yapılmalıdır. Rehber öğretmen öğrencilerden Resim 8.6 ve 8.7’de görüldüğü gibi sürüş tabanına hedef yön kolunu eklemelerini ister. Etkinlik sonrasında öğrenciler sürüş tabanını eski haline getirmelidir.



Resim 8. 6 Hedef Yön Kolu Montajı Birinci Adım



Resim 8.7 Hedef Yön Kolu Montajı İkinci Adım

Rehber öğretmen için örnek bir program aşağıda verilmiştir. Programda dönülecek açı için 4 derecelik bir düzeltme yapılmıştır. Bu değer farklı koşullarda değişiklik gösterebilir. Öğrenciler bu değeri kendi durumlarına göre düzenlemelidir. Sapma açısının sol tarafa doğru negatif ve sağ taraf doğru pozitif olması ve büyük motorun 0-360 arasında saat yönünde artacak şekilde değerler alması göz önünde bulundurularak dönülecek açı değeri üzerinde de düzenleme yapılmıştır. Aynı kod farklı şekillerde de yazılabilir.

buyuk\_motor=Motor("E")

hub = PrimeHub()

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

while True:

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

    hub\_hareketi=hub.motion\_sensor.wait\_for\_new\_gesture()

    while hub\_hareketi=="doubletapped":

        hub\_hareketi=hub.motion\_sensor.wait\_for\_new\_gesture()

    donulecek\_aci=buyuk\_motor.get\_position()

    if donulecek\_aci<180:

        while hub.motion\_sensor.get\_yaw\_angle()> -donulecek\_aci+4:

            surus\_tabani.start\_tank(0,50)

    else:

        while hub.motion\_sensor.get\_yaw\_angle()<=360-donulecek\_aci-4:

            surus\_tabani.start\_tank(50,0)

    surus\_tabani.stop()

    hub.speaker.beep(62,0.5)

    buyuk\_motor.run\_to\_position(0)

|  |
| --- |
| **PROJE HAZIRLIYORUM- HAZIRLIK AŞAMASI** |
| Öğrencilerin öğrendiklerini grup hâlinde uygulayabilecekleri proje geliştirme sürecinin, eğitimin yaklaşık son 4 haftasını kapsayacak şekilde planlanması beklenmektedir. Projelerin *hazırlık, tanımlama, empati, fikir üretme, geliştirme-test etme ve sunum* gibi etkinliklerle yapılması ve rehber öğretmenler gözetiminde öğrencilere ders dışı etkinliklerle destek olunması önerilmektedir. Bu hafta ile projeye hazırlık aşamasıyla başlanmış olur.  **Açıklama:** Bu haftadan itibaren gruplar iki hafta boyunca projelerini tanımlama ve tasarlama üzerinde çalışacaklardır. Daha sonra, tanımlayıp tasarladıkları robotu oluşturmaları ve gerekli programları geliştirmeleri beklenir. Öğrenciler öncelikle çevrelerinden veya günlük hayatlarından uygun bir problem belirleyip bu probleme kendi tasarladıkları bir robot ile çözüm üretirler. Öğrencilerin problem üzerinde sistematik bir şekilde çalışmaları için grupların problemi analiz etmeleri ve kendi çözüm önerilerini tasarlamaları, geliştirmeleri ve değerlendirmeleri istenir. Bu süreçte karşılaşılacak tasarım ve programlamayla ilgili problemlerde rehber öğretmenler, öğrencilere uygun zaman ve miktarda destek olurlar.  **Sonraki Haftaya Hazırlık:** Bir sonraki hafta için öğrencilerin çevrelerinden ve günlük hayatlarından robotlar kullanılarak çözülebilecek bir problemi belirlemeleri gerektiği iletilir. Uygun bir problem sorusu belirlemeden önce problemi anlamak gerekir. Belirlenen problem, projenin zorluklarını ve hedeflerini tanımlamaya yardımcı olur. Sonrasında problemle ilgili yoğun bir araştırma ve gözlem yapılması gerekir. Süreç, projenin tamamlanması için gerekli bilgiler sağlar ve ortam hakkında bilgiler sunar. Süreç sonunda çok vakit ve emek harcanarak geliştirilecek robotun, insanların işine yaraması, bir problemini çözmesi ve hayatlarını kolaylaştırması istenir. |