# 7. Hafta – Kuvvet Sensörü

## Haftanın Amacı:

Bu hafta, Lego Spike Prime robot setinde yer alan kuvvet sensörü öğrencilere tanıtılarak, kuvvet sensörünün;  basılma, sert basılma ve bırakılma parametreleri ve ölçülen kuvvet miktarının farklı programlarda kullanılması sağlanacaktır.

## Haftanın Kazanımları:

* Öğrenciler kuvvet sensörünün basılma, sert basılma ve bırakılma durumlarını, program akışında kullanabilir.
* Öğrenciler Boolean değişkenleri bilir.
* Öğrenciler koşul ifadelerini kullanabilir.
* Öğrenciler while döngüsünü kullanabilir.
* Öğrenciler kuvvet sensörü kullanarak, kuvvet ölçümü yapabilir.

## Kullanılacak Malzemeler:

Robot seti, bilgisayar ve çalışma alanları.

## GÖZLE VE UYGULA

|  |
| --- |
| ***Rehber Öğretmen İçeriği –*** ***Kuvvet Sensörü Hakkında*** |
| *Kuvvet sensörü, EV3 robot setinde bulunan dokunma sensörüne birçok yönden benzer. Ancak kuvvet sensörü, dokunma sensörün yapabileceğinden daha fazlasını yapar. Bu sensör, kullanıcının 10 Newton'a kadar kuvveti ölçmesini sağlar. Ayrıca bir dokunma sensörü gibi basılı olma durumunu algılayabilir. Kuvvet sensörü aşağıda gösterilmiştir. Kuvvet birimi Newton olup simgesi N’dir. Kütlesi 1 kg olan bir cismin hızını, 1 saniye içerisinde 1 m/s arttırmak için o cisme uygulanması gereken kuvvet, Newton cinsinden 1N = 1kg.m/s2 olarak tanımlanır.*    *Kuvvet sensörünün kullanmak için öncelikle bir ForceSensor nesnesi oluşturulmalıdır. Bu nesne oluşturulurken ön değer olarak port bilgisi verilmelidir. Sürüş tabanında kuvvet sensörü A portuna bağlı olduğu için kuvvet sensörü aşağıdaki şekilde oluşturulmuştur.*  *kuvvet\_sensoru=ForceSensor("A")*  *Kuvvet sensörünün is\_pressed isimli bir metodu bulunur. Bu metot kuvvet sensörünün butonuna basılıp basılmadığını kontrol eder. Eğer butona basılı ise doğru (True), basılı değilse yanlış (False) değeri döndürür. Bu metodun kullanımı aşağıda gösterilmiştir.*  *kuvvet\_sensoru.is\_pressed()*  *Sözcük Blokları programlama ortamında aşağıdaki resimde gösterildiği gibi basılı, sert basılı, bırakılmış ve basınç değişti gibi dört farklı durum vardır. MicroPython bunlardan sadece basılı olma durumunu kontrol eder. Diğer durumlar ise yazılacak bir kod ile ele alınabilir.* |

### Gözle: Kuvvet Sensörünü Tanımlayıp Kullanıyorum

Bu etkinlikte amaç kuvvet sensörünü kullanmaya yöneliktir. Aşağıdaki kod kuvvet sensörü basılana kadar bip sesini çalmaktadır. Rehber öğretmen aşağıdaki kodu öğrencilere açıklayarak anlatır.

hub = PrimeHub()

kuvvet\_sensoru=ForceSensor("A")

while kuvvet\_sensoru.is\_pressed()==False:

    hub.speaker.beep()

Bu kod ile öncelikle hub ve kuvvet sönsörü oluşturulmuştur. While döngüsünde while ifadesi ile iki nokta üst üste ifadesi arasında kalan kısım (kuvvet\_sensoru.is\_pressed()==False) döngünün koşul ifadesidir. Buna test ifadesi de denilebilir. Döngünün devam edip etmeyeceği buradaki test ifadesinin sonuçlarına bakılarak karar verilir. Test ifadesi doğru olduğu müddetçe döngü tekrar eder ve test ifadesi yanlış olduğu an döngüden çıkılır. Bu test ifadesi ile kuvvet sensörüne basılmama durumu kontrol edilmektedir. Kuvvet sensörüne basılmadığı müddetçe döngü gövdesini tekrar etme işlemi sürdürülecektir. Döngü gövdesini aşağıdaki komut oluşturmaktadır.

hub.speaker.beep()

Döngü gövdesini oluşturan komutlar bir veya birden fazla satırda olabilir. Gövdede bulunan komutları belirlemek için bir Tab büyüklüğünde girinti yapılır. Sonuç olarak kuvvet sensörüne basılmadığı süre boyunca arka arkaya bip sesi çalınacaktır. Kuvvet sensörüne basıldığında ise gövdenin tekrarı biter ve bip sesi kesilir.

Rehber öğretmen yukarıdaki kodu çalıştırıp öğrencilere anlattıktan sonra “for” döngüsü ile “while” döngüsü arasındaki farkları sorar. Grupların görüşlerini aldıktan sonra “for” döngüsünde range komutunda belirtilen sayıda döngü gövdesinin tekrar edildiğini, while döngüsünde ise koşul ifadesi doğru olduğu müddetçe döngü gövdesindeki komutların tekrar edildiğini vurgular.

### Gözle: Koşul İfadeleri

Koşul ifadeleri ile belirli bir koşul gerçekleştiğinde çalışan programlama yapısından bahsedilir. Örnek vermek gerekirse ders notu 60 üzerinde olan öğrencilerin geçip altında olan öğrencilerin kaldığı bir puan sistemi içi öğrencinin toplam notunu değerlendirerek, onun geçip geçmediğini bulan bir program için koşul ifadeleri kullanılabilir. Aşağıda bu görev için sözde kod verilmiştir.

eğer toplam\_puan>60:

print(“Geçti”)

değilse:

print(“kaldı”)

Spike Prime’dan bir örnek vermek gerekirse kuvvet sensörüne basıldığında ışık matrisinde kuzey yönlü ok basılmadığında ise güney yönlü ok gösterilmesi istenen program için aşağıdaki sözde kod kullanılabilir.

eğer kuvvet\_sensoru.is\_pressed():

hub.light\_matrix.show\_image(“ARROW\_N”)

değilse:

hub.light\_matrix.show\_image(“ARROW\_S”)

Python’da bir koşulun gerçekleşip gerçekleşmediğini kontrol etmek için if-else programlama yapısı kullanılır. Koşulun gerçekleşme durumu *if* kısmında ele alınırken gerçekleşmeme durumu *else* kısmında ele alınır*. If-else* programlama yapısının kullanım şekli aşağıda gösterilmiştir.

if koşul:

Koşul doğruysa çalıştırılacak komutlar

else:

Koşul doğru değilse çalıştırılacak komutlar

Rehber öğretmen öğrencilere yukarıda anlatılanları kullanarak *if-else yapısını* açıklar. Koşul ifadelerinde de bir *tab* girintisi yapıldığını vurgular. Ardından aşağıdaki kodu öğrencilerin yazıp çalıştırmasını ister. Bu kod ile kuvvet sensörüne basıldığında ışık matrisinde kuzey yönlü ok basılmadığında ise güney yönlü ok gösteren programın yapılması tasarlanmıştır. Fakat kod bir kerelik çalıştığı için istenen görevi yerine getirmez. Öğrencilerin kodun neden görevi yerine getirmediğine dair tartışması sağlanır. Tartışma sonucunda rehber öğretmen sebebi öğrencilere açıklar.

hub = PrimeHub()

kuvvet\_sensoru=ForceSensor("A")

if kuvvet\_sensoru.is\_pressed():

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

else:

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

### Uygula: Sürekli Tekrar ve Koşul İfadesinin Birlikte Kullanımı

Yukarıdaki örnekte koşul ifadesi sürekli tekrarlanmadığı için istenen görevi gerçekleştirmez. İstenen görevin gerçekleştirilmesi için *while döngüsü* ile sürekli tekrarlama işlemi gerçekleştirilmelidir. Rehber öğretmen bu ipucunu öğrencilere verdikten sonra onlardan kuvvet sensörüne basıldığında ışık matrisinde kuzey yönlü ok, basılmadığında ise güney yönlü ok gösterilmesi istenen programı yazmalarını ister. Bu işe için örnek kod aşağıda verilmiştir.

while True:

    if kuvvet\_sensoru.is\_pressed():

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

    else:

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

Rehber öğretmen daha sonra öğrencilerden kuvvet sensörüne basıldığında aynı zamanda bip sesi çıkartmalarını ister. Bu iş için aşağıdaki kod örnek olarak verilebilir.

while True:

    if kuvvet\_sensoru.is\_pressed():

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

        hub.speaker.beep()

    else:

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

### Uygula: Döngü, Koşul ve Değişkenin Bir Arada Kullanımı

Yukarıdaki kod ile kuvvet sensörü basılı olduğu durumda sürekli bip sesi çalmaktadır. Öğrenciler bu programı geliştirmelidir. Kuvvet sensörüne her basıldığında sadece bir kere bip sesi çalınmalıdır.

### Gözle: Miktar Belirtmeden Sürüş Tabanını ve Motorları Hareket Ettirme

Daha önceki derslerde sürüş tabanındaki motorların veya tek tek motorların nasıl hareket ettirileceği anlatılmıştır. Bu hareket belirli bir süre veya miktarda yapılmaktadır. Hareketi sensörlerden gelen veriye bağlı olarak yapmak için süre veya miktardan bağımsız olarak yapmak gerekmektedir. Sürüş tabanının veya bir motorun ne kadar süre veya uzunlukta hareket edeceği belirtilmeden de hareket etmesi sağlanabilir. Aşağıda gösterilen start metodu ile sürüş tabanı istenilen yön ve hızda hareket ettirilebilir.

start(yön, hız)

Görüldüğü üzere *start* metodunda hareket miktarına yönelik parametre bulunmamaktadır. Örneğin aşağıdaki kod çalıştırılırsa sürüş tabanı sürekli ileri doğru 20 hızında hareket edecektir.

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

surus\_tabani.start(0,20)

Sadece bir motorun istenen hızda harekete başlaması isteniyorsa yön parametresini kullanmaya gerek yoktur. Aşağıdaki kod kullanılırsa C portuna bağlı olan motor sürekli ileri doğru 20 hızında hareket edecektir.

motor=Motor("C")

motor.start(20)

Sürüş tabanı için sol ve sağ motorlara ayrı ayrı hız verilmesi de sağlanabilir. Bu iş için move\_tank metodu kullanılır. Bu metodun kullanımı aşağıdaki gibidir.

move\_tank(sol motor hızı, sağ motor hızı)

Sürüş tabanının veya bir motorun durdurulması için *stop()* metodunun kullanılması gerekir. Fakat *start* metodunun hemen arkasından stop kullanılırsa motorların hareketi gerçekleşmez. Durdurma işlemini belirli bir sensörden gelen değere veya bir süreye (wait\_for\_seconds() gibi) bağlamak yerinde olacaktır.

Aşağıdaki kod ile sürüş tabanı 50 hızıyla ileriye doğru ilerler. Kuvvet sensörüne basıldığında sürüş tabanı duracaktır. Rehber öğretmen bu kodu öğrencilere açıklayarak anlatır.

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

kuvvet\_sensoru=ForceSensor("A")

surus\_tabani.start(0,50)

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

    pass

surus\_tabani.stop()

Yukarıdaki kod içerisinde ilk defa kullandığımız iki programlama yapısı bulunmaktadır. Bunlar *not ve pass* isimli yapılardır. Not mantıksal bir operatördür. İfadenin mantıksal olarak değilini alır. Yani eğer bir ifade *doğru (True*) ise değili *yanlış (False) yanlış ise değili* doğru olacaktır. Aşağıda verilen ifade kod içerisinde kullanılmıştır.

not kuvvet\_sensoru.is\_pressed()

Burada *not* sonrasında bulunan ifade daha önce anlatıldığı gibi kuvvet sensörüne basıldığında doğru, aksi durumda yanlış değerini üretecektir. Bu ifadenin başına *not* getirilirse kuvvet sensörüne basıldığında yanlış basılmadığında ise doğru değerini üretecektir. Böylece kuvvet sensörüne basılmadığı müddetçe döngü gövdesindeki komutlar çalıştırılacaktır. Kuvvet sensörüne basıldığında döngüden çıkılır. Döngü gövdesinde sadece *pass* ifadesi bulunmaktadır. Bu ifade içinde bulunduğu blokta hiçbir işlem yapılmayacağı zaman kullanılır. Yani döngü çalıştığı müddetçe hiçbir işlem gerçekleştirme denilmektedir. Tüm döngü yeniden gözden geçirildiğinde kuvvet sensörüne basılmadığı müddetçe hiçbir işlem yapılmayacaktır, yani kuvvet sensörüne basılana kadar bekleyecektir. Aslında bu iş için Python’da aşağıdaki *wait\_until\_pressed* metodu bulunmaktadır ve kullanımı aşağıdaki şekildedir. Bu komut ile yukarıdaki while döngüsünün işlevi aynıdır. Programcı istediğini kullanabilir.

kuvvet\_sensoru.wait\_until\_pressed()

Yukarıda yazılan kod bütün olarak incelendiğinde kuvvet sensörüne basılana kadar sürüş tabanı 50 hızıyla doğrusal bir şekilde ileri yönlü hareket yapar. Kuvvet sensörüne basıldığında ise sürüş tabanı durdurulur.

### Gözle ve Uygula: Beklet metodunu kullanıyorum

Kuvvet sensörüne basılana kadar bekleten metot olduğu gibi kuvvet sensörü bırakılana kadar bekleten metot da bulunmaktadır. Bu metodun kullanımı aşağıda verilmiştir.

kuvvet\_sensoru.wait\_until\_released()

Rehber öğretmen bu kodun kullanımını öğrencilere anlatır. Ardından öğrencilere *while* döngüsü kullanarak bu komut ile benzer göver gören kodu yazılmasını ister. Bu iş için aşağıdaki kod kullanılabilir.

while kuvvet\_sensoru.is\_pressed():

    pass

### Gözle ve Uygula: Engele Çarpınca Geri Giden Robot

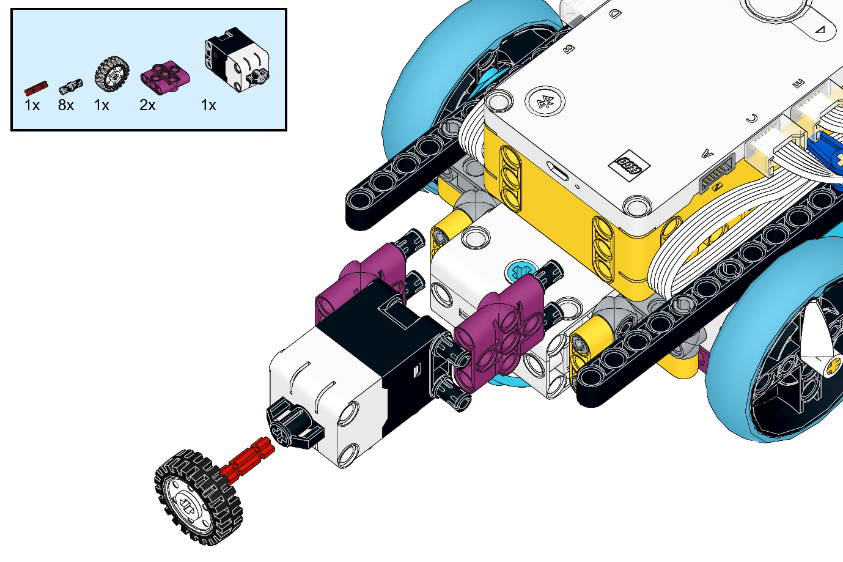
Öğrencilerden engele çarpınca geri dönen bir robot programlamaları istenir. Robot bu iş için aşağıdaki adımları takip etmelidir:

(i) Robotun bir engele çarpıncaya kadar düz gider.

(ii) Robot engele çarpınca bir miktar geri gider.

(iii) Robot son olarak kendi etrafında dönme hareketini yaparak durmalıdır.

Bu görev için sürüş tabanı Resim 7.1 ve 7.2 de gösterildiği gibi yeniden oluşturulmalıdır.



Resim 7. 1 Kuvvet sensörü montajı 1. adım



Resim 7. 2 Kuvvet sensörü montajı 2. adım

Bu görev için aşağıdaki kod kullanılabilir.

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

kuvvet\_sensoru=ForceSensor("A")

surus\_tabani.start(0,40)

kuvvet\_sensoru.wait\_until\_pressed()

surus\_tabani.stop()

surus\_tabani.move(-180,"degrees",0,40)

surus\_tabani.move(1,"rotations",-100,40)

### Uygula: Sürekli olarak Engele Çarptığında Geri Dönen Robot

Bu etkinlikte öğrencilerden önceki etkinliğe benzer bir robot programını yazmaları istenir. Fakat bu defa robot engele çarptığı her seferde geriye dönme işlemini yapmalıdır. Yani önceki görevde bir kez gerçekleştirilen görev sürekli gerçekleştirilmelidir. Bunun için öğrencilere sürekli tekrarla döngüsünün kullanılabileceği söylenebilir. Bu görev için örnek kod aşağıdadır.

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

kuvvet\_sensoru=ForceSensor("A")

while True:

    surus\_tabani.start(0,40)

    kuvvet\_sensoru.wait\_until\_pressed()

    surus\_tabani.stop()

    surus\_tabani.move(-180,"degrees",0,40)

    surus\_tabani.move(1,"rotations",-100,40)

### Gözle: Kuvvet Sensörüne Her Basıldığında Bir Kere Bip Sesi Çalan Program

Bu etkinlikte öğrencilerden kuvvet sensörüne her basıldığında, basılma süresinden bağımsız olarak, sadece bir kere bip sesi çıkaran programı yazmaları istenmektedir. Bu programın yazılabilmesi için değişkenler kullanılarak kuvvet sensörünün önceki ve şimdiki durumu modellenebilir. Programda sadece sensöre basılı olup olmadığı kontrol edilirse, sensöre basılı olduğu müddetçe bip sesi çalınır. Fakat soruda sensöre her basıldığında bir kere bip sesinin çalınması istenmektedir. Bu durumda basılmama durumundan basılma durumuna geçiş kontrol edilmeli ve bu durum oluştuğunda bip sesi çıkarılmalıdır. Sensörün önceki durumunun basılı değil (*False*) ve şimdiki durumunun ise basılı (True) olması bize bu geçişi göstermektedir. Aşağıdaki kod bu düşünce temel alınarak yazılmıştır. Rehber öğretmen verilen kodu öğrencilere açıklayarak anlatır. Kodu açıklamadan önce öğrencilere *Boolean* yani sadece *True ve False* değeri alabilen değişkenleri anlatır. Bunun yanında aşağıdaki kod içerisinde *and* operatörü kullanılmıştır. Rehber öğretmen etkinlik öncesinde and operatörün kullanımını öğrencilere anlatır arından etkinliğe geçer.

hub = PrimeHub()

kuvvet\_sensoru=ForceSensor("A")

onceki\_durum=False  #sensöre henüz basılmamış

simdiki\_durum=False #sensöre henüz basılmamış

basili=False

while True:

    simdiki\_durum=kuvvet\_sensoru.is\_pressed()  #sensör değeri okunuyor

    if onceki\_durum==False and simdiki\_durum==True: #sensöre basıldı

        basili=True

    onceki\_durum=simdiki\_durum #durum güncellendi

    if basili:

        hub.speaker.beep()

    basili=False  #yeniden basılma işlemi için gerekli

### Uygula: Kuvvet Sensörüne Her Basılıp çekildiğinde Bir Kere Bip Sesi Çalan Program

Önceki etkinlikte sensöre her basıldığında sadece bir kere bip sesi çalınmıştır. Bu etkinlikte bir sesinin çalışma koşulu değiştirilmiştir. Kuvvet sensörüne basıldığında değil, sensöre basılıp çekildiğinde “bip” sesi çıkaran program yazılmalıdır Sensörün önceki ve şimdiki hâli farklı şekilde modellenerek problem çözülebilir. Bu etkinlik için örnek kod aşağıda verilmiştir.

hub = PrimeHub()

kuvvet\_sensoru=ForceSensor("A")

onceki\_durum=False

sonraki\_durum=False

basilip\_cekildi=False

while True:

    simdiki\_durum=kuvvet\_sensoru.is\_pressed()

    if onceki\_durum==True and simdiki\_durum==False:

        basilip\_cekildi=True

    onceki\_durum=simdiki\_durum

    if basilip\_cekildi:

        hub.speaker.beep()

    basilip\_cekildi=False

### Gözle ve Uygula: Uygulanan Kuvveti Ölçüp Işık Matrisinde Yazdırıyorum

Kuvvet sensörü ile bu sensöre uygulanan kuvvet miktarı Newton cinsinden ve yüzde olarak ölçülebilir. Kuvvet sensörü ile 10 Newton’a kadar kuvvet ölçümü yapılabilir. 10 N üzerindeki değerler ölçülemez. Kuvvet ölçümü için *get\_force\_newton* metodu kullanılmaktadır. Bu metot uygulanan kuvvet miktarını döndürür. Metodun kullanımı aşağıda verilmiştir.

kuvvet\_sensoru.get\_force\_newton()

Kuvvet sensörü ile ölçüm yüzde olarak yapıldığında 0-100 arasında değerler üretilir. 0N için 0, 10 N için 100 ve 0-10 N arasında kalan kuvvet için bu orantısal değerler üretilir. Yüzde olarak yapılan ölçümlerde *get\_force\_percentage* metodu kullanılır. Bu metodun kullanımı aşağıda gösterilmiştir.

kuvvet\_sensoru.get\_force\_percentage()

Rehber öğretmen aşağıdaki kodu öğrencilere açıklayarak anlatır. Ardından öğrencilerden yüzde cinsiden ölçüm yaptırarak elde edilen değeri ışık matrisinde yazdırmaları istenir.

hub = PrimeHub()

kuvvet\_sensoru=ForceSensor("A")

while True:

    kuvvet\_miktari=kuvvet\_sensoru.get\_force\_newton()

    hub.light\_matrix.write(kuvvet\_miktari)

    wait\_for\_seconds(0.5)

### Uygula: Kuvvet Miktarına Göre Değişen Bip Sesi

Bu etkinlikte amaç kuvvet miktarı değiştikçe değişik notalar ile bip sesi çalan programı yazmaktır. Bip sesi için kullanılabilecek nota numaraları 44 ile 123 arasında değişmektedir. Öğrenciler bu etkinlikte 0-100 arasında elde edilen kuvvet yüzdesi değerini 44-123 arasına dönüştürmelidir (Not: Öğrenciler kuvvet ölçümü yaparlarsa dönüşüm 0-10 ile 44-123 arasında olacaktır). 0 ile 100 aralığında n tam sayısı alındığında bunun 44-123 aralığındaki karşılığını bulmak için nota=44+n\*(79/100) olarak tanımlanmış dönüşüm fonksiyonu kullanılabilir. Fakat bu fonksiyon ondalık değer *(float)* üretirken beep metodu tam sayı *(int)* değerler kabul etmektedir. Bu yüzden elde edilen sonuç tam sayıya çevrilmelidir. Bu görev için *int()* fonksiyonu kullanılabilir; nota=44+int(n\*(79/100)) istenen dönüşümü sağlayacaktır. Rehber öğretmen etkinlik öncesinde öğrencilere ondalık sayıyı tam sayıya nasıl çevireceğini anlatır fakat dönüşüm fonksiyonu hakkında bilgilendirmez. Öğrenciler dönüşüm fonksiyonunu bulmak için çaba göstermelidir. Rehber öğretmen öğrencilere yeterli bir süre verdikten sonra dönüşüm fonksiyonunu bulamayanlara yardım eder. Bu görev için kullanılabilecek kod aşağıda verilmiştir.

hub = PrimeHub()

kuvvet\_sensoru=ForceSensor("A")

while True:

    n=kuvvet\_sensoru.get\_force\_percentage()

    nota=44+int(n\*(79/100))

    print(nota,n)

    hub.speaker.beep(nota,0.2)

### Gözle: İki Olay Arasında Geçen Süreyi Saniye Cinsinden Hesaplıyorum

Python’da time modülü içerisinde time isimli bir fonksiyon bulunmaktadır. Bu fonksiyon 1 Ocak 1970 gece yarısından itibaren şimdiye kadar geçen süreyi saniye cinsinden vermektedir. Bu süreyi bulmak üzere aşağıdaki kod kullanılabilir.

import time

print(time.time())

Görüldüğü üzere time metodunu kullanmak için öncelikle time modülü *import* komutu ile programa dâhil edilmelidir. Aşağıdaki program sürüş tabanının tekerlerinin harekete başlamasından itibaren kuvvet sensörüne basılıp tekerlerin durdurulması arasında geçen süreyi saniye cinsinden ölçerek konsola yazmaktadır. Rehber öğretmen aşağıdaki kodu öğrencilere açıklayarak anlatır.

import time

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

hub = PrimeHub()

kuvvet\_sensoru=ForceSensor("A")

surus\_tabani.start(0,50)

birinci=time.time()

kuvvet\_sensoru.wait\_until\_pressed()

surus\_tabani.stop()

ikinci=time.time()

gecen\_sure=ikinci-birinci

print(gecen\_sure)

*time()* fonksiyonu milisaniye cinsinden ölçümler yapamamaktadır. Daha hassas ölçümler yapılmak isteniyorsa bu fonksiyon kullanılmamalıdır. Bunun yerine *time.ticks\_ms()* fonksiyonu kullanılabilir. Bu fonksiyon kullanılarak milisaniye cinsiden geçen süre hesaplanabilir. Bu fonksiyon tam sayı değerler üretmektedir. Süre farkını saniye cinsiden bulmak için elde edilen sonuç 1000 ile bölünmelidir. Rehber öğretmen öğrencilerden aynı görevi *ticks\_ms* fonksiyonunu kullanarak yerine getirmesini ister. Bu iş için gerekli program aşağıda verilmiştir.

import time

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

hub = PrimeHub()

kuvvet\_sensoru=ForceSensor("A")

surus\_tabani.start(0,50)

birinci=time.ticks\_ms()

kuvvet\_sensoru.wait\_until\_pressed()

surus\_tabani.stop()

ikinci=time.ticks\_ms()

gecen\_sure=ikinci-birinci

print(gecen\_sure)

Rehber öğretmen son olarak öğrencilerden *time ve ticks\_ms* fonksiyonlarını aynı programda kullanarak ölçümler yapmalarını ister. Öğrenciler ölçümlerin sonucunu karşılaştırarak iki metot arasındaki farkı tam olarak kavrar.

### Uygula: Sürüş Tabanının Maksimum Hızını Buluyorum

Sürüş tabanının hareketi için motorlara verilen hız miktarı sürüş tabanının gerçek hızını vermemektedir. Bu etkinlikte sürüş tabanına maksimum güç verildiğinde (100) ve doğrusal olarak ileriye doğru hareket ederken gerçek hızının cm/saniye cinsinden bulunması istenmektedir. Rehber öğretmen öğrencilerden gerekli kodu yazmalarını ister. Rehber öğretmen için örnek bir kod aşağıda verilmiştir.

import time

hub = PrimeHub()

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

birinci=time.ticks\_ms()

surus\_tabani.move(100,"cm",0,100)

ikinci=time.ticks\_ms()

gecen\_sure=(ikinci-birinci)/1000

hiz=100/gecen\_sure

print(hiz)

## TASARLA

Kuvvet Sensörü Oyunu

Bu etkinlikte kuvvet sensörüne basılma miktarına orantılı olarak ilerleyen bir robot kodu yazılmalıdır. Oyunun oynanabilmesi için her bir öğrenci grubu kendi kodunu yazmalıdır. Bu kod çalıştırıldığında:

1. Işık matrisinde gülen yüz şeklinde ışıklar yanacaktır.
2. Sürüş tabanına sabitlenmiş kuvvet sensörüne basıldığında ışık matrisinde 3 ten geriye doğru sayılmaya başlanacaktır.
3. Bu süre zarfında sensre basılı tutulmalıdır. Sıfır anında kuvvet sensöründen ölçüm yapılacaktır.
4. Ölçümün ardından hoparlörden bip sesi çalınacaktır. Bip sesi çalındığında kuvvet sensörü serbest bırakılmalıdır.
5. Bip sesi bittiği anda sürüş abanı kuvvet sensöründen ölçülen basınç miktarının on katı santimetre ileri gidecektir. Hareket esnasında ölçülen kuvvetin Newton cinsinden değeri ekranda görülecektir.

Rehber öğretmen 10 cm’nin katları olacak şekilde bir hedef uzaklık belirleyip (örneğin 70 cm) belirlenen uzaklığa bir Lego parçası yerleştirecektir. Öğrenciler ise programı çalıştırıp kuvvet sensörüne uygulayacakları kuvvet ile sürüş tabanının istenen miktarda giderek hedefe ulaşmasını sağlayacaklardır. Robotu hedefe en çok yaklaştıran grup yarışmayı kazanır. Yarışma öncesinde gruplar hedefe yaklaşma denemeleri yapabilirler fakat yarışma esnasında her grubun sadece bir hakkı bulunmaktadır. İki grup arasında eşitlik olması durumunda hedefe daha hızlı varan grup birinci sayılacaktır. Hızlı grup sürüş tabanlarının hareket hızına bakılarak karar verilebilir.

Tanımlama: Bu aşamada istenilenler belirlenmeli ve sürüş tabanı kodu ile neler yapılacağı tanımlamalı ve bunlar maddeler halinde sıralamalıdırlar. Rehber öğretmen için aşağıda bir örnek verilmiştir.

* Geri sayım için kuvvet sensörüne basılması beklenmeli,
* Bip sesi ile birlikte kuvvet sensöründen ölçüm alınmalı,
* Alınan ölçüm büyüklüğü kullanılarak sürüş tabanı hareket ettirilmeli,
* Sürüş tabanının ani bir şekilde durması sağlanmalı,
* Sürüş tabanının hızı olabildiğince fazla olacak şekilde belirlenmeli. Fakat gereğinden fazla belirlenecek hız miktarı robotun görevi yerine getirmesini engelleyebilir.

Fikir üretme: Bu aşamada öğrencilerden istenen kodun yazılabilmesi için, kullanılacak programlama yapılarına ve algoritmaya dair fikir üretmeleri beklenmektedir. Rehber öğretmen için aşağıda örnek bir liste verilmiştir.

* *wait\_until\_pressed* metodu kullanılarak kuvvet sensörüne basılması beklenebilir,
* *for* döngüsü kullanılarak geri sayım işlemi gerçekleştirilebilir,
* get\_force\_newton metodu ile ölçüm Newton cinsinden yapılabilir. Elde edilen sonuç 10 ile çarpılarak ölçüm cm cinsinden mesafeye dönüştürülebilir,
* Sürüş tabanı keskin bir duruş için *hold* ile durdurulabilir,
* Sürüş tabanının hızı 60 civarında olabilir. Fakat bu değer deneme yanılma yöntemiyle daha net olacak şekilde belirlenebilir.

## ÜRET

Üret bölümünde öğrenciler tasarla bölümünde ortaya koydukları fikirleri kullanarak ilgili programı yazmalıdırlar. Diğer üret etkinliklerinde olduğu gibi kod yazma sorumluluğu öğrencilere aittir ve öğrenciler aktif rol üstlenmelidir. Rehber öğretmen doğrudan anlatan ve problem çözümünde aktif rol üstlenen bir yöntem izlememelidir. Bunun yerine öğrencileri yönlendirerek onlara rehberlik etmelidir. Öğrenciler grup olarak bilgisayar ve robot başında çalışarak gerekli kodu geliştirirler. Aşağıda rehber öğretmen için örnek bir kod verilmiştir.

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

surus\_tabani.set\_stop\_action("hold")

hub = PrimeHub()

kuvvet\_sensoru=ForceSensor("A")

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

kuvvet\_sensoru.wait\_until\_pressed()

for i in range(3,-1,-1):

    hub.light\_matrix.write(i)

    wait\_for\_seconds(1)

kuvvet=kuvvet\_sensoru.get\_force\_newton()

hub.speaker.beep(62,1)

hub.light\_matrix.write(kuvvet)

surus\_tabani.move(kuvvet\*10,"cm",0,60)

## DEĞERLENDİR

Günün sonunda rehber öğretmen öğrencilere aşağıdaki soruları yönelterek öğrenilen konuların birlikte özetlenmesini sağlar.

* Robot setinde bulunan kuvvet sensörünün özellikleri nelerdir?
* Günlük hayatta kullanılan dokunma/kuvvet/ağırlık sensörlerine örnekler veriniz?

(Örneğin arabanın kapısı açık kaldığında uyarı amacıyla ses vermesi, asansörlerde yük fazla olması durumunda uyarı vermesi, arabalarda lastik basıncının ölçülmesi).

* Bugün öğrendiğiniz komutlar hangileridir? Görevleri nelerdir?
* Sizce bu hafta en zor görev hangisidir? Zorluğun üstesinden nasıl geldiniz?

## İLAVE ETKİNLİK

Rehber öğretmen öğrencilere kuvvet sensörü oyununda uygulanan kuvvet oranında hareket sağlandığını belirtir. Bu oyunda sürüş tabanı uygulanan kuvvet ve kuvvet sensörü bırakılıncaya kadar saniye cinsinden geçen sürenin çarpımı kadar ileri gidecek şekilde geliştirmelidir. Takip edilmesi gereken adımlar aşağıda verilmiştir.

* Kuvvet sensörüne basılınca geri sayım işlemi başlayacaktır. Kuvvet sensörü serbest bırakılmamalıdır.
* Geri sayım işlemi bitince ışık matrisinde YES (✓) şekli belirmelidir. Bu şekil belirdiği andan itibaren süre ölçülmeye başlanacaktır.
* Kuvvet sensörü serbest bırakılınca toplam süre hesaplanacaktır (bunun için kronometre kullanılmamalıdır).
* Bip sesinin çalmasının ardından sürüş tabanı kuvvet\*süre cm kadar ileri gitmelidir (kuvvet Newton ve süre saniye cinsinden).

Bu oyunda ulaşılacak cismin yerini 10’un katları olacak şekilde belirlemeye gerek yoktur. Rehber öğretmen bir hedef uzaklık belirleyip örneğin 75 cm belirlenen uzaklığa bir Lego parçası yerleştirecektir. Öğrenciler ise programı çalıştırıp kuvvet sensörüne uygulayacakları kuvvet ve belirledikleri süre ile sürüş tabanının istenen miktarda giderek hedefe ulaşmasını sağlayacaklardır. Robotu hedefe en çok yaklaştıran grup yarışmayı kazanır. Gruplar arasında eşitlik olması durumunda hızlı olan grubun robotu yarışmayı kazanır. Yarışma öncesinde gruplar hedefe yaklaşma denemeleri yapabilirler fakat yarışma esnasında her grubun sadece bir hakkı bulunmaktadır. Rehber öğretmen için aşağıda örnek bir kod verilmiştir.

import time

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

surus\_tabani.set\_stop\_action("brake")

hub = PrimeHub()

kuvvet\_sensoru=ForceSensor("A")

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

kuvvet\_sensoru.wait\_until\_pressed()

for i in range(3,-1,-1):

    hub.light\_matrix.write(i)

    wait\_for\_seconds(1)

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

birinci=time.ticks\_ms() #time() kullanılırsa sadece 10'un katları

kuvvet=kuvvet\_sensoru.get\_force\_newton()   #cinsinden hareket

kuvvet\_sensoru.wait\_until\_released()       #sağlanabilir

ikinci=time.ticks\_ms()

gecen\_sure=(ikinci-birinci)//1000 #saniye elde edilir

hub.speaker.beep(62,1)

surus\_tabani.move(kuvvet\*gecen\_sure,"cm",0,60)