

# Fikri Çıktı 1

## MINECRAFT PI FİZİKSEL BLOKLARI

Çıktı: IO1-A2: MINECRAFT PI FİZİKSEL BLOKLARI



### STEM4CLIM8

29.09.2021

HeartHands Solutions LTD

Yazar: HeartHands Solutions LTD

Proje Numarası: 2020-1-UK01-KA201-079141



Co-funded by the  
Erasmus+ Programme  
of the European Union

Avrupa Komisyonu'nun bu yayının üretimine verdiği destek, yalnızca yazarların görüşlerini yansıtan içeriğin onaylanması anlamına gelmez ve Komisyon, burada yer alan bilgilerden doğabilecek herhangi bir kullanımdan sorumlu tutulamaz.

## Özet

Son zamanlarda yapılan ve 25 Avrupa ülkesini kapsayan bir OECD araştırmasında (OECD- Eğitim ve Becerilerde Bugün 2018), hemen hemen tüm ülkeler, öğretmenlerin becerilerini güncellemedeki zorluklar ile birlikte okul ihtiyaçlarını karşılamada eksikliklerini rapor etmektedir. Bununla birlikte, dijitalleşmenin öğrenme ve çalışma şeklimizi derinden değiştirmesi beklenmektedir. Bugün okula başlayan birçok çocuğun, gelecekte şu anda mevcut olmayan işlerde çalışması muhtemeldir. Bu henüz bilinmeyen işler için öğrencilerin hazırlanması, sadece doğru teknik kabiliyetlere sahip olmalarının yanında, duygusal ve sosyal kabiliyetlerini güçlendirmemiz gerektiği anlamına da gelmektedir. Değişimle başa çıkma süreci, olumsuz durumların üstesinden gelme konusundaki bireysel kapasite ve bunları kişisel gelişim için kaynak olarak kullanma kapasitesi, değişime başarılı bir şekilde ayak uydurmayı ve böylece dijital dünyamız ile aktif bir şekilde etkileşime geçebilmeyi kapsar. Aynı zamanda, gençlik için ortaya çıkan sorunlar olarak IAB (İnternet Bağımlılık Davranışı) riski ile internet bağımlılığı ve davranışlarını bilmemiz gereklidir. Ekran dışı iş birliğine dayalı aktivitelere odaklanırken fiziksel aktiviteler ile çevresel bilinç arasında köprü kuran bir STEM yaklaşımı, duygusal ve sosyal becerileri güçlendirirken, teknik kapasitelerin artırılması için mükemmel bir yöntemdir.

STEM4CLIM8 projesinin birincil amacı, çocuklarla çalışan kişilerin programlama ile meşgul olmalarına ve STEM ile ilgili becerileri geliştirmelerine yardımcı olmak amacıyla çocuklara ulaşmalarına etki edecek yaklaşımlar ve araçlar üretmektir. Buna ekran süresini artırarak değil Raspberry GPIO aracılığıyla sanal dünya ile etkileşime girecek şekilde programlanacak olan fiziksel blokları kullanarak, doğal afetlerle ilgili görevleri yerine getirerek ve aynı zamanda Minecraft modding kullanarak özel bir sanal dünya oluşturması yardımıyla erişmeyi amaçlamaktadır. Bu görevler, aynı zamanda STEM becerilerini artırırken sıklıkla iklim değişikliğiyle ilişkilendirilen doğal olayların arkasındaki bilimi ortaya çıkaracaktır ve çevresel bilinç için ilham verecektir.

### Referans:

\*OECD- Eğitim ve Becerilerde Bugün, Değişimle başa çıkma konusunda başarı- Okullar için dersler, 29 Ocak 2018, aşağıdaki adresten 18 Şubat 2021 tarihinde alındı: <https://oecdutoday.com/succeeding-with-resilience-lessons-for-schools/>

## İçindekiler

ÖZET .....	1
<b>1. MINECRAFT PI FİZİKSEL BLOKLARI .....</b>	<b>4</b>
1.1 "FISHING TOWN" MINECRAFT PI DÜNYASINI YÜKLEME .....	4
1.2 SENARYOLAR VE ETKİLEŞİMLER .....	5
1.3 DEPREM .....	5
1.3.1 Ekipman ve senaryo .....	5
1.3.2 Elektronik devre ve fiziksel blok.....	6
1.3.3 Python programlama .....	7
1.4 OBRUK .....	8
1.4.1 Ekipman ve senaryo .....	8
1.4.2 Elektronik devre ve fiziksel blok.....	9
1.4.3 Python programlama .....	11
1.5 VOLKANİK PÜSKÜRME.....	12
1.5.1 Ekipman ve senaryo .....	12
1.5.2 Elektronik devre ve fiziksel blok.....	12
1.5.3 Python programlama .....	14
1.6 SICAK DALGASI .....	15
1.6.1 Ekipman ve senaryo .....	15
1.6.2 Elektronik devre ve aktivitenin yürütülmesi .....	15
1.6.3 Python programlama .....	18
1.7 SEL .....	19
1.7.1 Ekipman ve senaryo .....	19
1.7.2 Elektronik devre ve aktivitenin yürütülmesi .....	20
1.7.3 Python programlama .....	23
1.8 SONUÇ .....	23
<b>REFERANSLAR .....</b>	<b>23</b>

## 1. MINECRAFT PI FİZİKSEL BLOKLARI

Bir dizi doğal afetin meydana geldiği bir Minecraft Pi World ile etkileşim kurmak için çeşitli "Do It Yourself-Kendin Yap" (DIY) fiziksel bloklar tasarlanmıştır. Bu bloklar Raspberry Pi GPIO aracılığıyla konsola bağlanacak ve doğal afet ve bunun yıkıcı sonuçlarını küçük bir "Fishing Town" (balıkçı kasabası) için simüle ederek Minecraft Pi dünyası ile etkileşime geçmek için Python programlama dili aracılığıyla programlanacaktır.

Eğitmenlere yönelik olan bu kılavuz, Python programlama ve Minecraft Pi kullanarak programlanmaları için ve GPIO'ya bağlanmada kullanılacak olan fiziksel blokları birleştirmek için talimatlar sunmaktadır. Eğitmenlerin kullanacağı araçlar:

- STEM4CLIM8 konsolu
- Fiziksel Bloklar
- Bu fenomenleri öğretmek için Eğitim Materyalleri

### 1.1 "Fishing Town" Minecraft Pi Dünyasını Yükleme

Doğal afet senaryolarından herhangi birini gerçekleştirmek için, öncelikle farklı afetlerin test alanı olarak kullanılacak bir balıkçı kasabasının haritasını kurmanız gerekmektedir. Bunu yapmak için önce STEM4CLIM8 konsolunuzda Raspberry Pi'nizi açmanız gerekir.

Masaüstünde "Fishing Town Map" (balıkçı kasabası haritası) adlı bir klasör göreceksiniz. Klasörün içeriğini görmek için üzerine çift tıklayın. Ardından şu basit adımları izleyin:

1. Minecraft Pi'yi açın, yeni bir dünya yaratın ve ardından Minecraft Pi'yi kapatın.
2. "Dosya Yöneticisi"ni açın.
3. "Görünüm"e tıklayın ve "Gizli Göster"i işaretleyin.
4. ".minecraft" adlı klasöre çift tıklayın.
5. ".games" adlı klasöre çift tıklayın.
6. ".com.mojang" adlı klasöre çift tıklayın.
7. ".minecraftWorlds" adlı klasöre çift tıklayın.
8. Mevcut bir "world" (dünya) klasörü içerisinden "Fishing Town Map"ın içeriğini kopyalayın.
9. Bu klasörün adını "fishingtown" olarak tekrar adlandırın.
10. Minecraft Pi'yi açın ve "Fishing Town" dünyasını yükleyin.

Bir "Fishing Town" Minecraft Pi dünyası yüklenecektir. Lütfen unutmayın, yenileme ve afetsiz bir haritaya ihtiyacınız olduğunda bu prosedürü tekrar yapmanız gerekecektir.

Bu Minecraft Pi dünyası, aşağıdaki doğal afetleri simüle etmek için kullanılacaktır:

- Deprem (earthquake)
- Volkanik püskürme (volcano eruption)
- Obruk (sinkhole)
- Sıcak dalgası (heatwave)
- Sel (flood)

## 1.2 Senaryolar ve Etkileşimler

Raspberry Pi GPIO, sensörler, elektronikler, çevresel birimler ve DIY karton yapılarının kullanımıyla öğrenciler, doğal afetler hakkında bilgi edinecek, kendi sonuçlarını test edecek ve sonuçlarını tartışacaklardır.

Öncelikle öğretmenler, sensörler ve diğer elektroniklerin Raspberry Pi'ye nasıl bağlanacağı, DIY karton yapılarının nasıl oluşturulacağı ve Python'da Minecraft Pi etkinliklerinin nasıl yürütüleceği hakkında öğrencilerine rehberlik edecektir.

Minecraft Pi dünyası, doğal afetlerin meydana geleceği küçük bir balıkçı kasabası olarak şekillendirilmiştir. Bu aktiviteler beş başlık altında sınıflandırılmıştır:

1. Deprem (earthquake)
2. Volkanik püskürme (volcano eruption)
3. Obruk (sinkhole)
4. Sıcak dalgası (heatwave)
5. Sel (flood)

Her bir aktivite için, nasıl yürütüleceğine dair özel talimatlar geliştirilmiştir. Öğrencilerinizi çeşitli gruplara ayırmanız ve her bir grubun bir hoparlörün yanı sıra bağlı bir klavye ve faresi olan en azından bir STEM4CLIM8 konsoluna erişimi olması tavsiye edilmektedir.

## 1.3 Deprem

### 1.3.1 Ekipman ve senaryo

Bu aktivite, Minecraft Pi dünyasında bir deprem oluşturacaktır. Öğrenciler Raspberry Pi GPIO'ya bir buton bağlayacak ve bunu bir kez basıldıktan sonra bir deprem oluşturacak şekilde programlayacaklar.

Amaç aşağıdaki elektronik ve çevre birimlerini kullanarak bir fiziksel blok oluşturmaktır:

1. 1 x Elektronik devre prototip deney kartı (breadboard)
2. 1 x Kapaklı basmalı buton
3. 1 x 220 Ohm direnç
4. 2 x Erkek-Dişi elektronik devre elemanları bağlantı kabloları (jumper kabloları)
5. 1 x Karton yapısı

Her şey bağlandıktan sonra, öğrenciler her bir STEM4CLIM8 konsolunun masa üstünde bulunabilen "earthquake.py" isimli Python kodunu ve "Fishing Town" haritasını yüklemelidir.

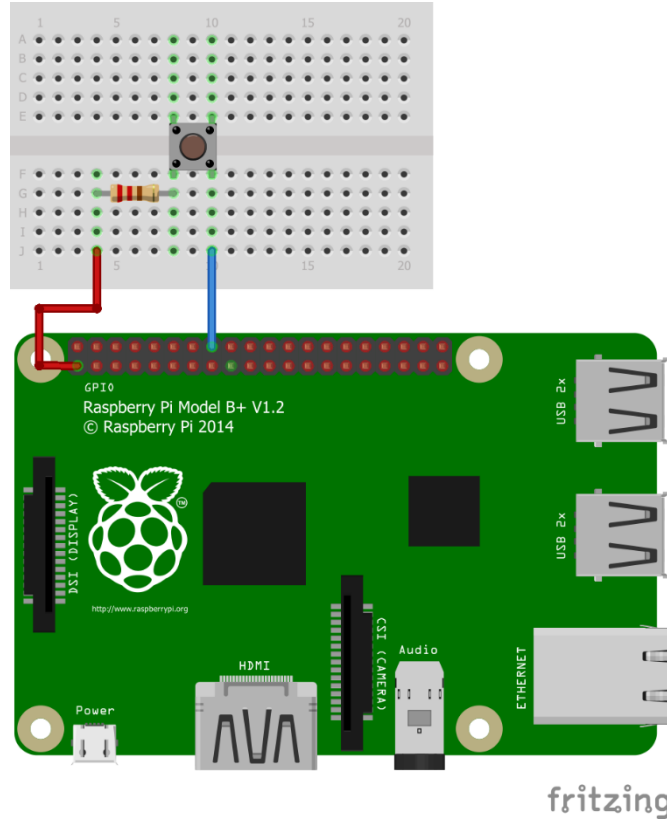
Ardından butona her basıldığında, büyük ölçekli gerçek bir depreme benzer bir şekilde "Fishing Town"nın binaları ve diğer bloklarını yok edecek olan Minecraft Pi dünyasında bir deprem olacaktır.

Aktiviteden sonra depremin etkisi ve iyi inşa edilmiş binanın ne demek olduğuna dair bir takip tartışması yapılmalıdır.

### 1.3.2 Elektronik devre ve fiziksel blok

Raspberry Pi'nin genel amaçlı giriş/çıkışlarına (GPIO) bir buton bağlanması, aşağıdaki Şekil 1'de gösterildiği gibi basit bir işlemdir:

1. Butonun sağ tarafını bir erkek-dişi jumper kablosu kullanarak pin 16 (GPIO23) girişine bağlayın.
2. Butonun sol tarafını 220 Ohm dirence bağlayın.
3. Direncin diğer tarafını 3.3V pin 1'e bağlayın.
4. Artık devreniz hazır.



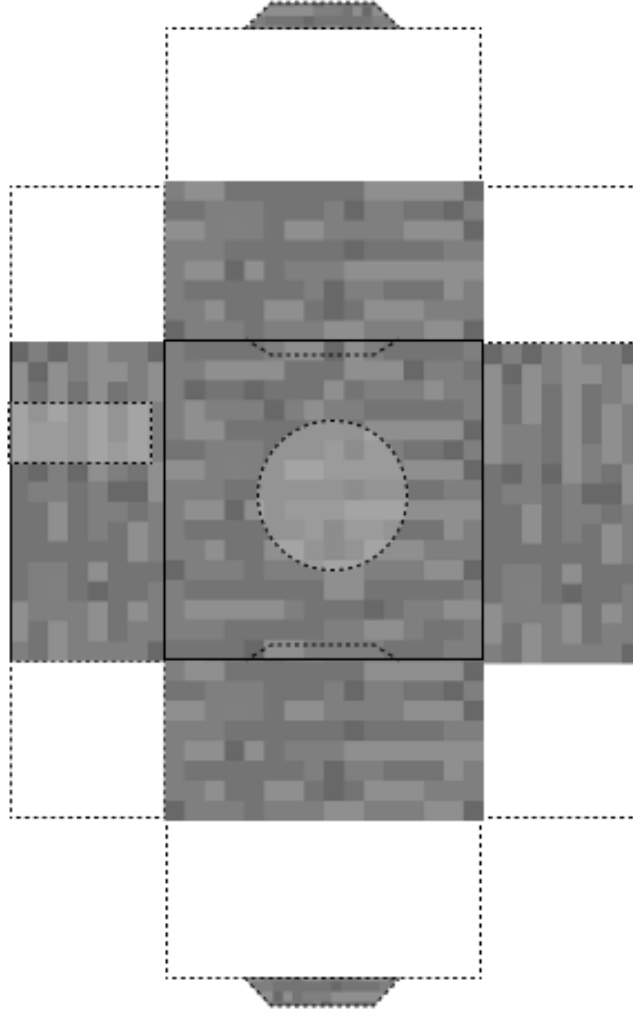
Şekil 1. Bir breadboard ve iki jumper kablosu kullanarak bir butonu GPIO'ya bağlama.

Farklı GPIO pinlerini anlamak için, her bir pinin özelliklerini açıklayan Şekil 2'yi incelemelisiniz.

5V Power	5V Power	GND	GPIO14 UART0_TXD	GPIO15 UART0_RXD	GPIO18 PCM_CLK	GND	GPIO23	GPIO24	GND	GPIO25	GPIO8 SPI0_CEO_N	GPIO7 SPI0_CEL_N	ID_SC I2C EEPROM	GND	GPIO12	GND	GPIO16	GPIO20	GPIO21																				
1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	37	38	39	40
3V3 Power	GPIO2 SDA1 I2C	GPIO3 SCL1 I2C	GPIO4 1-Wire	GND	GPIO17	GPIO27	GPIO22	3V3 Power	GPIO10 SPI0_MOSI	GPIO9 SPI0_MISO	GPIO11 SPI0_SCLK	GND	ID_SD I2C EEPROM	GPIO5	GPIO6	GPIO13	GPIO19	GPIO26	GND																				

Şekil 2. Raspberry Pi GPIO Pinleri.

Devre oluşturulduktan sonra, bir DIY karton yapısı kurmanız ve devreyi bunun içine yerleştirmeniz gerekir. Bu amaçla, aşağıdaki yazdırılabilir resmi kullanmanız gerekmektedir. Daha iyi bir deneyim için, A4 beyaz karton kağıda yapıyı yazdırdığınızdan ve ardından belirtilen kenarları kestiğinizden emin olun.



Şekil3. Butonlu fiziksel blok için yazdırılabilir karton yapısı.

### 1.3.3 Python programlama

Basit bir şekilde, Raspberry Pi masa üstünüzde Thonny Python'u açın ve ardından masa üstünde bulunan "earthquake.py" adlı kodu yükleyin veya seçtiğiniz Python derleyicisini kopyalayıp yapıştırın. Butonu farklı bir GPIO pinine bağlamaya karar vererseniz, lütfen aşağıdaki kodda yer alan GPIO numarasını değiştirdiğinizden emin olun (gri renkli satırlar).

```
import RPi.GPIO as GPIO
import mcpi.minecraft as minecraft
import mcpi.block as block
mc = minecraft.Minecraft.create()
import random, time

GPIO.setmode(GPIO.BCM)
GPIO.setup(23, GPIO.IN, pull_up_down=GPIO.PUD_DOWN)

def earthquake(x, z):
```

```
mc.postToChat('Earthquake!')
y = mc.getHeight(x, z)
endtime = time.time() + 60
nearthtime = time.time()
while endtime > time.time():
    if time.time() > nearthtime:
        nearthtime = time.time() + 5
    ppos = mc.player.getPos()
    if ppos.x < x+100 and ppos.x > x-100:
        if ppos.y < y+100 and ppos.y > -60:
            if ppos.z < z+100 and ppos.z > z-100:
                mc.player.setPos(ppos.x, ppos.y, ppos.z)
    bx = random.randint(x-100, x+100)
    by = y
    bz = random.randint(z-100, z+100)
    if mc.getHeight(bx, bz) > -50:
        by = mc.getHeight(bx, bz)
    if mc.getBlock(bx, by, bz) in [block.GLASS.id, block.GLASS_PANE.id]:
        mc.setBlock(bx, by, bz, block.AIR.id)
        continue
    mc.setBlock(bx, by, bz, block.GRAVEL.id)
    mc.setBlocks(bx, by-1, bz, bx, -60, bz, block.AIR.id)

disasters = [earthquake]
def main(disasters, mc):
    baseed = random.randint(1, 10000)
    mc.postToChat('Press button for earthquake!')
    while True:
        if GPIO.input(23) == GPIO.HIGH: #Look for button press
            t = random.randint(5, 60)
            t = 5
            time.sleep(t)
            random.seed(baseed + t)
            baseed = random.randint(1, 10000)
            random.shuffle(disasters)
            disaster = random.choice(disasters)
            ppos = mc.player.getTilePos()
            disaster(ppos.x, ppos.z)

try:
    import _thread as thread
except ImportError:
    import thread
thread.start_new_thread(main, (disasters, mc))
```

## 1.4 Obruk

### 1.4.1 Ekipman ve senaryo

Bu aktivite, bir deprem sonrasında oluşacak olan bir obruğu Minecraft Pi dünyasına sunacaktır. İlk aktiviteye benzer bir şekilde, öğrenciler Raspberry Pi GPIO'ya bir buton bağlayacak ve bunu bir kez basıldıktan sonra bir obruk oluşturacak şekilde programlayacaklar.

Bu aktivite aşağıdaki elektronik ve çevre birimlerini kullanır:

1. 1 x Breadboard
2. 1 x Kapaklı basmalı buton
3. 1 x 220 Ohm direnç
4. 2 x Erkek-Dişi jumper kabloları
5. 1 x Karton yapısı



Her şey bağlandıktan sonra, öğrenciler her bir STEM4CLIM8 konsolunun masa üstünde bulunabilen "sinkhole.py" isimli Python kodunu ve "Fishing Town" haritası yüklemelidir.

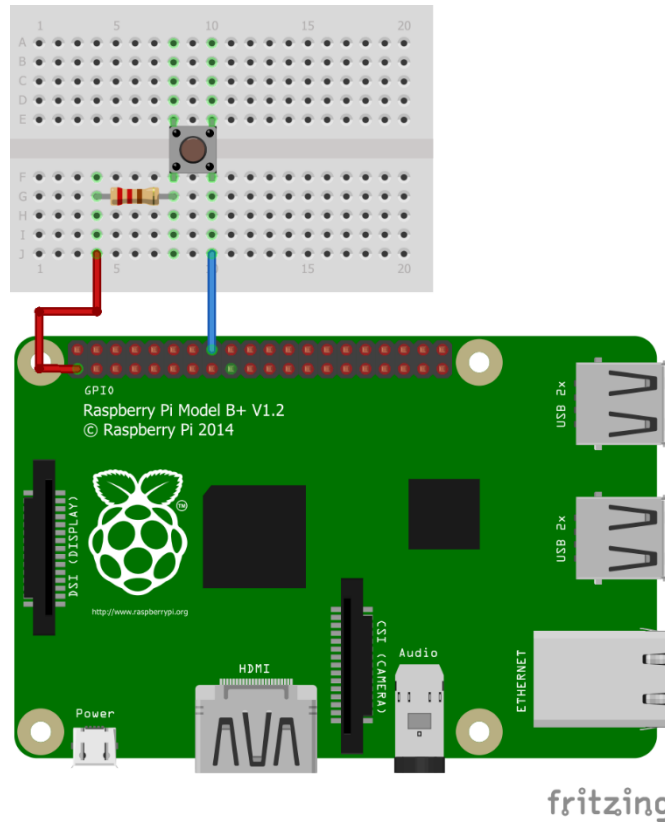
Ardından, butona her basıldığında, Minecraft Pi dünyasında bir deprem olacak ve sonrasında binaları yok edecek ve gerçek bir felaket gibi zemini batıracak bir obruk oluşacaktır.

Aktiviteden sonra depremin ve obruğun etkisine dair öğrencileriniz ile bir takip tartışması yapılmalıdır.

### 1.4.2 Elektronik devre ve fiziksel blok

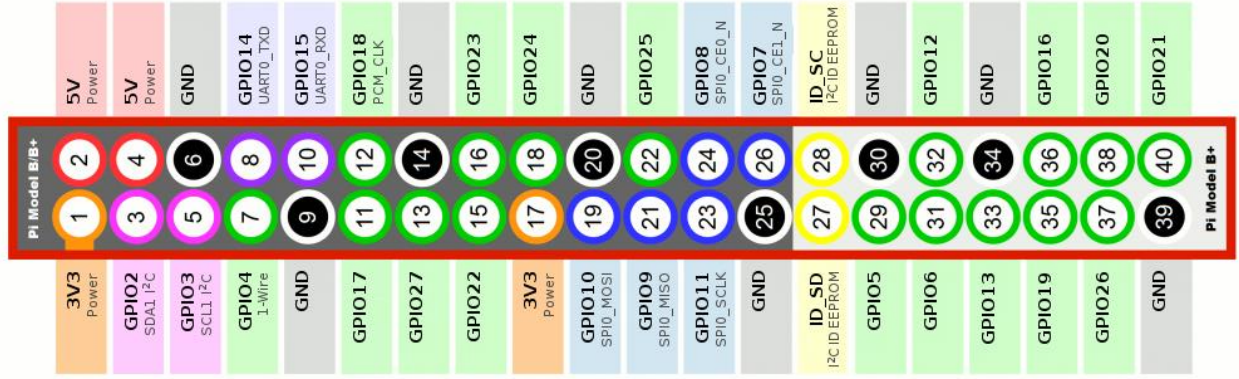
Raspberry Pi'nin GPIO'suna bir buton bağlanması, aşağıdaki Şekil 4'te gösterildiği gibi çok basit bir işlemdir:

1. Butonun sağ tarafını bir erkek-dişi jumper kablosu kullanarak pin 16 (GPIO23) girişine bağlayın.
2. Butonun sol tarafını 220 Ohm dirence bağlayın.
3. Direncin diğer tarafını 3.3V pin 1'e bağlayın.
4. Artık devreniz hazır.



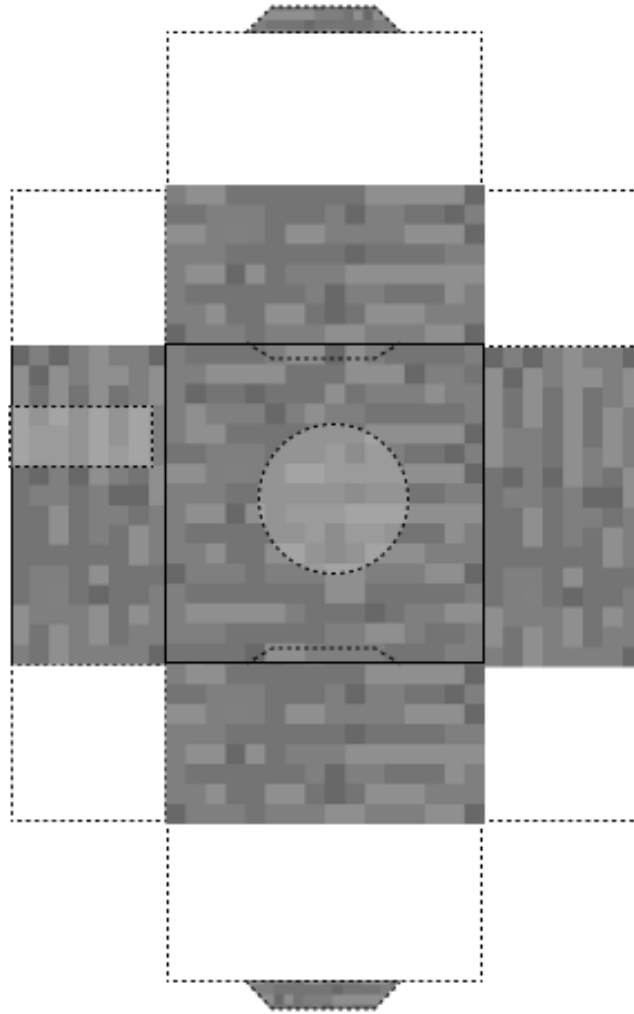
Şekil 4. Bir breadboard ve iki jumper kablosu kullanarak bir butonu GPIO'ya bağlama.

Farklı GPIO pinlerini anlamak için, her bir pinin özelliklerini açıklayan Şekil 5'i incelemelisiniz.



Şekil 5. Raspberry Pi GPIO Pinleri.

Devre oluşturulduktan sonra, bir DIY karton yapısı kurmanız ve devreyi bunun içine yerleştirmeniz gerekir. Bu amaçla, aşağıdaki yazdırılabilir resmi kullanmanız gerekir. Daha iyi bir deneyim için, A4 beyaz karton kağıda yapıyı yazdırdığınızdan ve ardından belirtilen kenarları kestiğinizden emin olun.



Şekil 6. Butonlu fiziksel blok için yazdırılabilir karton yapısı.

### 1.4.3 Python programlama

Basit bir şekilde, Raspberry Pi masa üstünüzde Thonny Python'u açın ve ardından masa üstünde bulunan "sinkhole.py" adlı kodu yükleyin veya seçtiğiniz Python derleyicisini kopyalayıp yapıştırın. Butonu farklı bir GPIO pinine bağlamaya karar verirsiniz, lütfen aşağıdaki kodda yer alan GPIO numarasını değiştirdiğinizden emin olun (gri renkli satırlar).

```
import RPi.GPIO as GPIO
import mcpi.minecraft as minc
import mcpi.block as block
mc = minc.Minecraft.create()
import random, time

GPIO.setmode(GPIO.BCM)
GPIO.setup(23, GPIO.IN, pull_up_down=GPIO.PUD_DOWN)

def sinkhole(x, z):
    mc.postToChat('Earthquake!')
    y = mc.getHeight(x, z)
    endtime = time.time() + 60
    nearthtime = time.time()
    while endtime > time.time():
        if time.time() > nearthtime:
            nearthtime = time.time() + 5
        ppos = mc.player.getPos()
        if ppos.x < x+50 and ppos.x > x-50:
            if ppos.y < y+50 and ppos.y > -60:
                if ppos.z < z+50 and ppos.z > z-50:
                    mc.player.setPos(ppos.x, ppos.y, ppos.z)
        bx = random.randint(x-50, x+50)
        by = y
        bz = random.randint(z-50, z+50)
        if mc.getHeight(bx, bz) > -50:
            by = mc.getHeight(bx, bz)
        if mc.getBlock(bx, by, bz) in [block.GLASS.id, block.GLASS_PANE.id]:
            mc.setBlock(bx, by, bz, block.AIR.id)
            continue
        mc.setBlock(bx, by, bz, block.GRAVEL.id)
        mc.setBlocks(bx, by-1, bz, bx, -60, bz, block.AIR.id)

    mc.postToChat('Sinkhole!')
    blks = []
    y = mc.getHeight(x, z)
    xdists = random.randint(1, 20)
    for bx in range(-xdists, xdists+1):
        zdists = random.randint(1, 20)
        for bz in range(-zdists, zdists+1):
            blks.append([x+bx, z+bz])
    for blk in blks:
        mc.setBlocks(blk[0], mc.getHeight(blk[0], blk[1]), blk[1], blk[0], -60, blk[1],
        block.AIR.id)
        mc.setBlocks(blk[0], -55, blk[1], blk[0], -60, blk[1], block.LAVA.id)
    for blk in blks:
        mc.setBlock(blk[0], y, blk[1], block.GRAVEL.id)

disasters = [sinkhole]
def main(disasters, mc):
    baseed = random.randint(1, 10000)
    mc.postToChat('Press button for disaster!')
    while True:
        if GPIO.input(23) == GPIO.HIGH: #Look for button press
            t = random.randint(5, 60)
            t = 5
            time.sleep(t)
            random.seed(baseed + t)
            baseed = random.randint(1, 10000)
            random.shuffle(disasters)
```

```
disaster = random.choice(disasters)
ppos = mc.player.getTilePos()
disaster(ppos.x, ppos.z)

try:
    import _thread as thread
except ImportError:
    import thread
thread.start_new_thread(main, (disasters, mc))
```

## 1.5 Volkanik püskürme

### 1.5.1 Ekipman ve senaryo

Bu aktivite, Minecraft Pi dünyasında bir volkanik püskürme oluşturacaktır. Bir kez daha, öğrenciler Raspberry Pi GPIO'ya bir buton bağlayacak ve bunu bir kez basıldıktan sonra bir volkanik püskürme oluşturacak şekilde programlayacaklar.

Bu aktivite için aşağıdaki elektronik ve çevre birimlerini kullanmalısınız:

1. 1 x Breadboard
2. 1 x Kapaklı basmalı buton
3. 1 x 220 Ohm direnç
4. 2 x Erkek-Dişi Jumper kabloları
5. 1 x Karton yapısı

Her şey bağlandıktan sonra, öğrenciler her bir STEM4CLIM8 konsolunun masa üstünde bulunabilen "volcano.py" isimli Python kodunu ve "Fishing Town" haritası yüklemelidir.

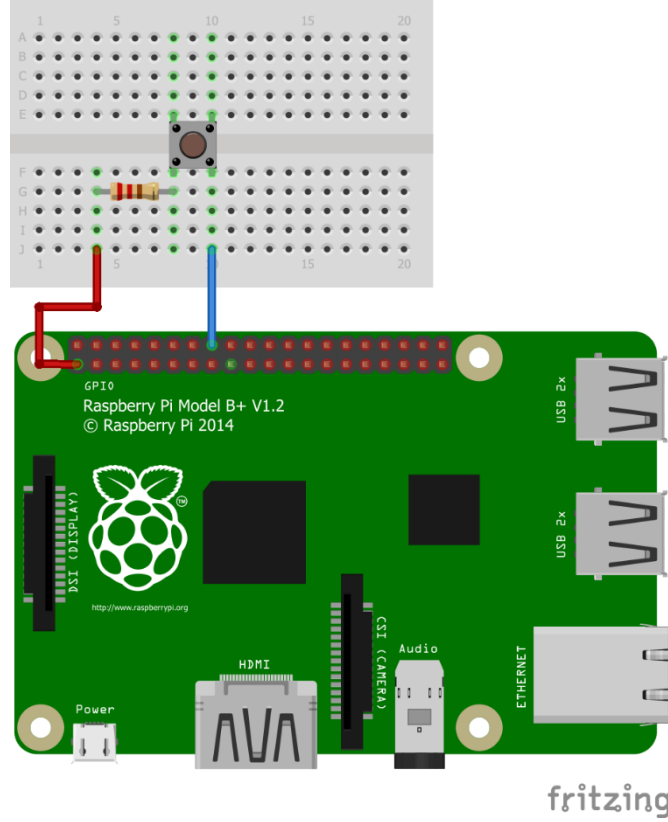
Ardından butona her basıldığında Minecraft Pi dünyasında patlayacak olan bir volkan belirecek ve lav balıkçı kasabasına doğru akmaya başlayacaktır. Bir süre sonra, lav gerçek hayattaki bir volkanik patlamaya benzer bir şekilde taşla dönüşecektir. Binalar ve diğer bloklar yok olacak ve ağaçlar ve çimenler alev alacaktır.

Aktiviteden sonra şehir sakinleri ve binaların etkisi hakkında öğrencilerinizle bir takip tartışması yapılmalıdır, çünkü doğal afetlerin sınırı yoktur ve hayatlarımızı ve yaşam durumumuzu tehdit edebilir.

### 1.5.2 Elektronik devre ve fiziksel blok

Raspberry Pi'nin GPIO'suna bir buton bağlanması, aşağıdaki Şekil 7'de gösterildiği gibi çok basit bir işlemdir:

1. Butonun sağ tarafını bir erkek-dişi jumper kablosu kullanarak pin 16 (GPIO23) girişine bağlayın.
2. Butonun sol tarafını 220 Ohm dirence bağlayın.
3. Direncin diğer tarafını 3.3V pin 1'e bağlayın.
4. Artık devreniz hazır.



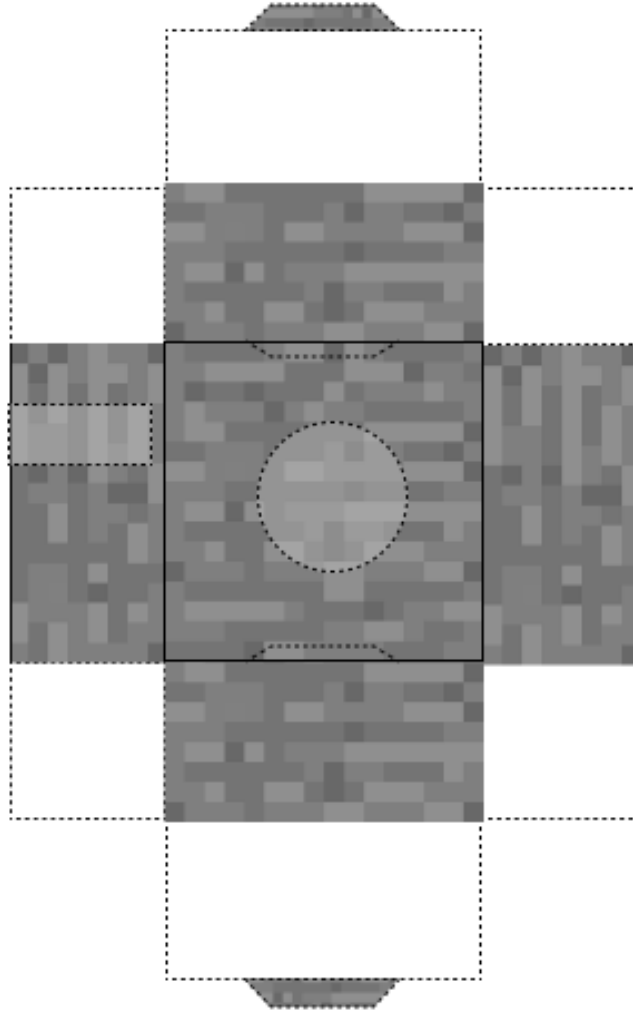
Şekil 7. Bir breadboard ve iki jumper kablosu kullanarak bir butonu GPIO'ya bağlama.

Farklı GPIO pinlerini anlamak için, her bir pinin özelliklerini açıklayan Şekil 8'i incelemelisiniz.

5V Power	5V Power	GND	GPIO14 UART0_TXD	GPIO15 UART0_RXD	GPIO18 PCM_CLK	GND	GPIO23	GPIO24	GND	GPIO25	GPIO8 SPI0_CEO_N	GPIO7 SPI0_CEL_N	ID_SD I2C EEPROM	GND	GPIO12	GND	GPIO16	GPIO20	GPIO21																				
1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	37	38	39	40
3V3 Power	GPIO2 SDA1 I2C	GPIO3 SCL1 I2C	GPIO4 1-Wire	GND	GPIO17	GPIO27	GPIO22	3V3 Power	GPIO10 SPI0_MOSI	GPIO9 SPI0_MISO	GPIO11 SPI0_SCLK	GND	ID_SD I2C EEPROM	GPIO5	GPIO6	GPIO13	GPIO19	GPIO26	GND																				

Şekil 8. Raspberry Pi GPIO Pinleri.

Devre oluşturulduktan sonra, bir DIY karton yapısı kurmanız ve devreyi bunun içine yerleştirmeniz gerekir. Bu amaçla, aşağıdaki yazdırılabilir resmi kullanmanız gerekir. Daha iyi bir deneyim için, A4 beyaz karton kağıda yapıyı yazdırdığınızdan ve ardından belirtilen kenarları kestiğinizden emin olun.



Şekil 9 Butonlu fiziksel blok için yazdırılabilir karton yapı.

### 1.5.3 Python programlama

Basit bir şekilde, Raspberry Pi masa üstünüzde Thonny Python'u açın ve ardından masa üstünde bulunan "volcano.py" adlı kodu yükleyin veya seçtiğiniz Python derleyicisini kopyalayıp yapıştırın. Butonu farklı bir GPIO pinine bağlamaya karar vererseniz, lütfen aşağıdaki kodda yer alan GPIO numarasını değiştirdiğinizden emin olun (gri renkli satırlar).

```
import RPi.GPIO as GPIO
import mcpi.minecraft as minecraft
import mcpi.block as block
import random, time

mc = minecraft.Minecraft.create()

GPIO.setmode(GPIO.BCM)
GPIO.setup(23, GPIO.IN, pull_up_down=GPIO.PUD_DOWN)

mc.postToChat('Press button for eruption!')
while True:
    if GPIO.input(23) == GPIO.HIGH: #Look for button press
        mc.postToChat("Eruption")
        x, y, z = mc.player.getPos()
```

```
mc.setBlocks(x, y, z, x+20, y+15, z+20, block.AIR)

height = 10
center = x+15, y, z+15

for i in range(height): #Create the base
    size = height - i
    mc.setBlocks(center[0] - size, center[1] + i, center[2] - size, center[0] +
size, center[1] + i, center[2] + size, block.STONE)

while True:
    mc.setBlock(center[0], center[1]+height, center[2], block.LAVA_FLOWING)
    time.sleep(1)
```

## 1.6 Sıcak dalgası

### 1.6.1 Ekipman ve senaryo

Bu aktivite, Minecraft Pi dünyasında bir sıcak dalgası oluşturacaktır. Öğrenciler bir ultrasonik sensörü GPIO'ya bağlayacak ve bunu bir nesne sensöre yaklaştığında bir sıcak dalgası oluşturacak şekilde programlayacaktır.

Bu aktivite aşağıdaki ekipmanları kullanılacaktır:

1. 1 x Breadboard
2. 1 x HC-SR04 ultrasonik sensör
3. 3 x 1k Ohm direnç
4. 4 x Erkek-Dişi jumper kabloları
5. 1 x Karton yapısı

Her şey bağlandıktan sonra, öğrenciler her bir STEM4CLIM8 konsolunun masa üstünde bulunabilen "heatwave.py" isimli Python kodunu ve "Fishing Town" haritası yüklemelidir.

"sun" (güneş) dünyaya yaklaştıkça, gerçek hayattaki felakete benzer şekilde ağaçların, bitkilerin ve çimlerin ölümüne neden olan bir sıcak dalgası fenomeni ortaya çıkacaktır.

Aktiviteden sonra sıcaklık artışı fenomeni ve bunun çevreye olan olumsuz sonuçları hakkında öğrencileriniz ile bir takip tartışması yapılmalıdır.

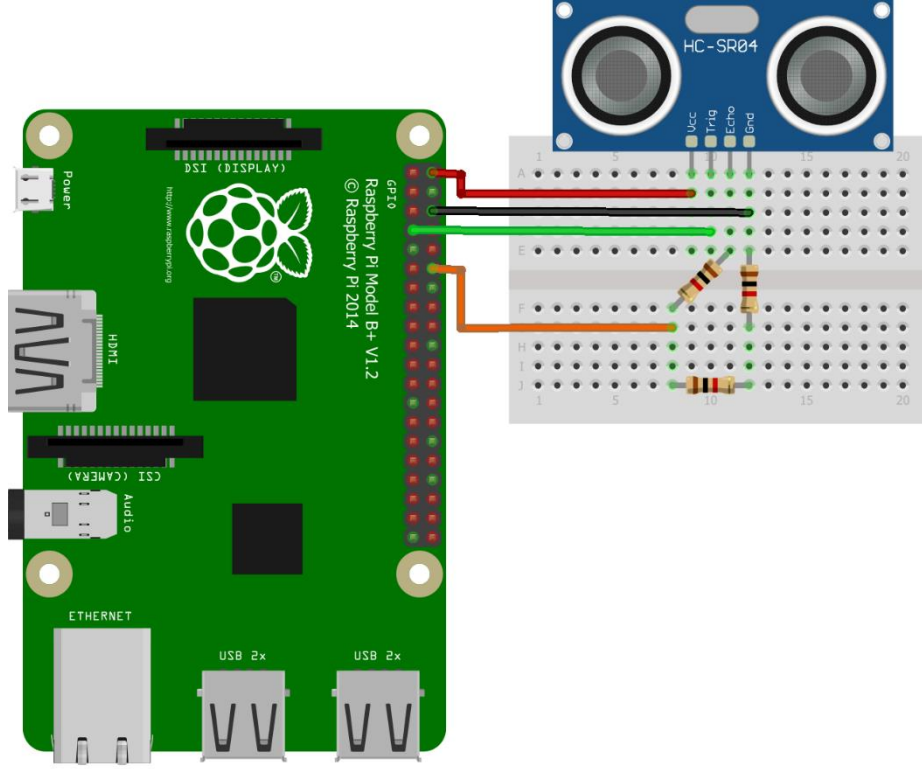
### 1.6.2 Elektronik devre ve aktivitenin yürütülmesi

Yapmanız gereken ilk şey, devreyi oluşturmak ve HC-SR04 sensörünü Raspberry Pi'nin GPIO pinlerine bağlamaktır. Devam etmeden önce Raspberry Pi'nizi kapatmanız ve devreden çıkarmanız gerekir. Devrenin tamamı aşağıda Şekil 10'da yer almaktadır.

HC-SR04 ultrasonik mesafe sensörü 4 pin ile birlikte teslim edilir: voltaj kaynağı (VCC), tetikleyici (TRIG), echo (ECHO) ve topraklama (GND). Voltaj kaynağı pini, Raspberry Pi'nin 5V pinine bağlanacaktır, tetikleyici (pin 4) çıkış olarak bir GPIO pinine atanacak, echo (pin 18) giriş olarak bir GPIO pinine atanacak ve toprak Raspberry Pi GPIO'daki topraklama pinine bağlanacaktır.

Bu program, bir nesne sensöre yaklaştığında çalışacaktır, bu sensör bir ultrason darbesi yayar ve program ekoyu geri almanın ne kadar süreceğinin zamanını tutar (başka bir deyişle, ses dalgaları yayılırken ve sensörün önündeki nesneye çarparken ve sekip geri dönerken ne kadar zaman geçtiğini tutar).

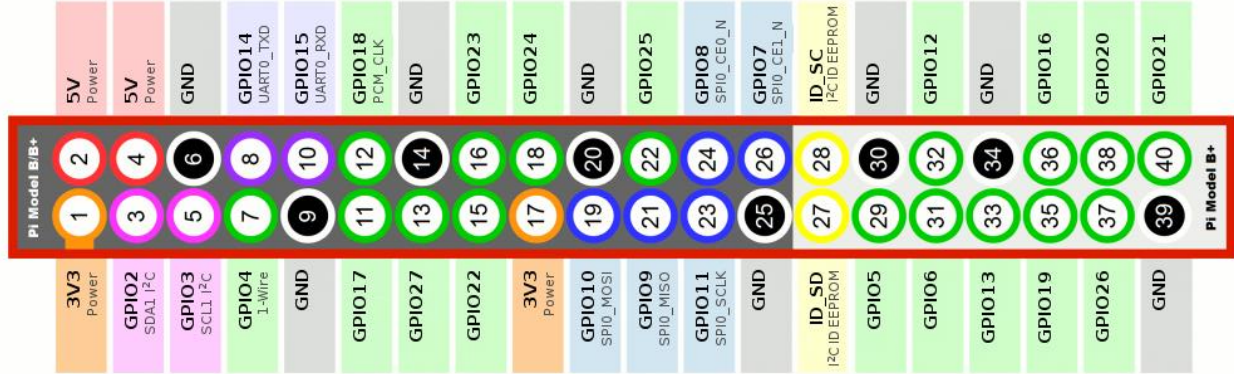




fritzing

Şekil 10. GPIO pinlerine bağlı mesafe sensörlü devrenin şematik diyagramı.

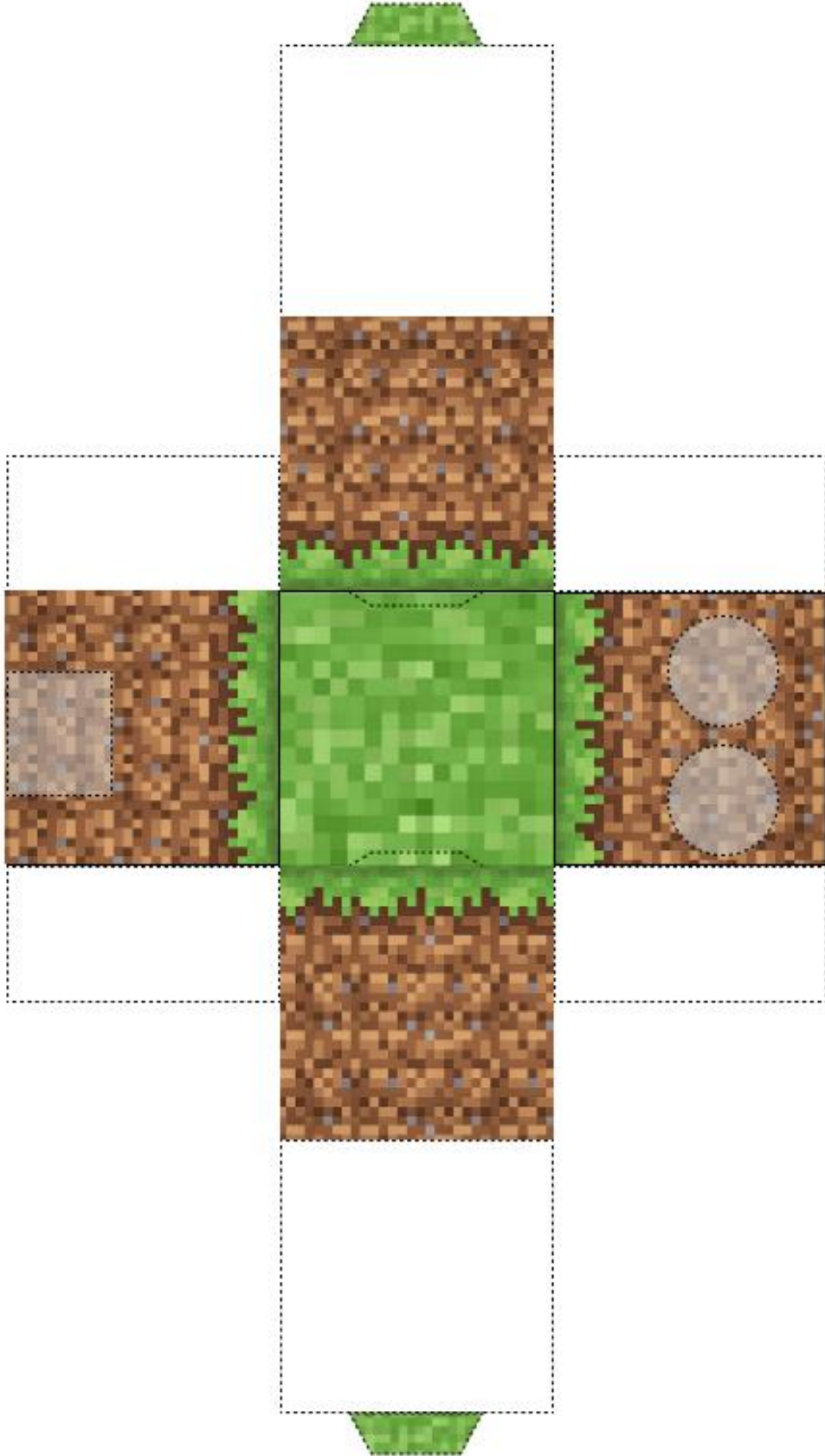
Farklı GPIO pinlerini anlamak için, her bir pinin özelliklerini açıklayan Şekil 11'i incelemelisiniz.



Şekil 11. Raspberry Pi GPIO Pinleri.

Devre oluşturulduktan sonra, bir DIY karton yapısı kurmanız ve devreyi bunun içine yerleştirmeniz gerekir. Bu amaçla, aşağıdaki yazdırılabilir resmi kullanmanız gerekir. Daha iyi bir deneyim için, A4 beyaz karton kağıda yapıyı yazdırdığınızdan ve ardından belirtilen kenarları kestiğinizden emin olun.





Şekil 12. Mesafe sensörlü fiziksel blok için yazdırılabilir karton yapı.

DIY karton yapısı "Earth"ü (dünya) temsil eder. Ayrıca "sun"ı temsil edecek benzer büyüklükte başka bir nesneye de ihtiyacınız olacak. STEM4CLIM8 paketinde, kullanabileceğiniz 3D baskılı bir küre bulacaksınız. Ek olarak, Raspberry Pi masa üstünde bulunan ilgili ".stl" dosyalarını kullanarak kendi "sun"ınızın 3D çıktısını alabilirsiniz.

### 1.6.3 Python programlama

Devreyi tamamladığınızda, Raspberry Pi'yi açabilir ve Thonny Python'u başlatabilirsiniz ve ardından masa üstünde bulunan "heatwave.py" adlı kodu yükleyin veya seçtiğiniz Python derleyicisini kopyalayıp yapıştırın. Kısacası, programı çalıştırdığınızda, mesafe HC-SR04 sensörü ultrason patlaması yaymaya başlayacaktır. "sun"ı sensöre ("Earth") her yaklaştırdığınızda, Minecraft Pi dünyasında sıcak dalgası fenomeni gerçekleşecektir. Ağaçların, çiçeklerin ve bitkilerin alev alıp kül olduğunu göreceksiniz.

```
import RPi.GPIO as GPIO
import mcpi.minecraft as minc
import mcpi.block as block
mc = minc.Minecraft.create()
import random, time

GPIO.setwarnings(False)
GPIO.cleanup()
GPIO.setmode(GPIO.BCM)
GPIO.setup(23, GPIO.IN, pull_up_down=GPIO.PUD_DOWN) #pin for push button

#defining TRIG and ECHO pins
TRIG = 4
ECHO = 18
GPIO.setup(TRIG, GPIO.OUT)
GPIO.setup(ECHO, GPIO.IN)

def heatwave(x, z):
    mc.postToChat('Heatwave!')
    y = mc.getHeight(x, z)
    endtime = time.time() + random.randint(50, 90)
    while time.time() < endtime:
        blkid = block.AIR.id
        while blkid == block.AIR.id:
            bx = random.randint(x-10, x+10)
            by = random.randint(y, y+10)
            bz = random.randint(z-10, z+10)
            blkid = mc.getBlockWithData(bx, by, bz).id
        blk = blkid
        blkd = mc.getBlockWithData(bx, by, bz).data
        if blkid == block.GRASS.id:
            blk = block.DIRT.id
            blkd = 0
        elif blkid in [block.WATER.id, block.WATER_FLOWING.id, block.WATER_STATIONARY.id]:
            blk = block.WATER.id
            blkd = 1
        elif blkid == block.LEAVES.id:
            blk = block.COBWEB.id
            blkd = 0
        elif blkid == block.WOOD.id:
            blk = block.LAVA_STATIONARY.id
            blkd = 1
        mc.setBlock(bx, by, bz, blk, blkd)

disasters = [heatwave]
def main(disasters, mc):
    baseed = random.randint(1, 10000)
    mc.postToChat('Push button for sensor to work!')
    mc.postToChat('Move "sun" closer to sensor, and wait for Heatwave!')
    #emit a burst of ultrasound
```

```
GPIO.output(TRIG, True)
time.sleep(0.00001)
GPIO.output(TRIG, False)
StartTime = time.time()
StopTime = time.time()
while True:
    if GPIO.input(23) == GPIO.HIGH:
        while GPIO.input(ECHO) == False:
            StartTime = time.time()
        while GPIO.input(ECHO) == True:
            StopTime = time.time()
        TimeElapsed = StopTime - StartTime
        distance = (TimeElapsed * 34300) / 2
        while distance < 15:
            print (distance)
            t = random.randint(5, 120)
            t = 5
            time.sleep(t)
            random.seed(baseed + t)
            baseed = random.randint(1, 10000)
            random.shuffle(disasters)
            disaster = random.choice(disasters)
            ppos = mc.player.getTilePos()
            disaster(ppos.x, ppos.z)
try:
    import _thread as thread
except ImportError:
    import thread
thread.start_new_thread(main, (disasters, mc))
```

## 1.7 Sel

### 1.7.1 Ekipman ve senaryo

Bu aktivite, Minecraft Pi dünyasında bir sel oluşturacaktır. Öğrenciler bir yağmur damlası sensörünü GPIO'ya bağlayacak ve sensör suya her dokunduğunda bir sel oluşturacak şekilde programlayacaktır.

Amaç aşağıdaki ekipmanları kullanarak bir fiziksel bloğu oluşturmaktır:

1. 1 x Breadboard
2. 1 x Yağmur damlası tespit sensörü
3. 3 x Erkek-Dişi jumper kabloları
4. 2 x Dişi-Dişi jumper kabloları
5. 1 x Karton yapısı

Her şey bağlandıktan sonra, öğrenciler her bir STEM4CLIM8 konsolunun masa üstünde bulunabilen "flood.py" isimli Python kodunu ve "Fishing Town" haritası yüklemelidir.

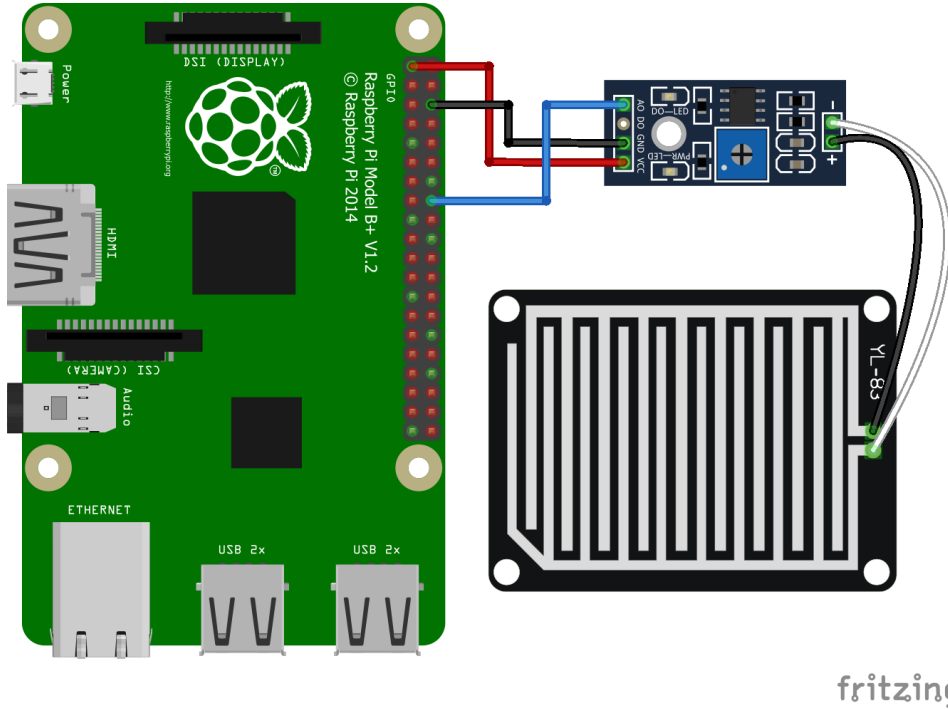
Ardından sensör üzerine bir damla düştüğünde, Minecraft Pi dünyasında her şeyin sular altında kalmasına neden olan bir sel oluşmalıdır. Şiddetli yağmur, Minecraft dünyasında bir Geyzer (sel) yaratacak bir mesajla anılacaktır.

Aktiviteden ağır bir yağmurdan sonra oluşan selin sebepleri ve yanlış uygulamalar hakkında öğrenciler ile bir takip tartışması yapılmalıdır.

### 1.7.2 Elektronik devre ve aktivitenin yürütülmesi

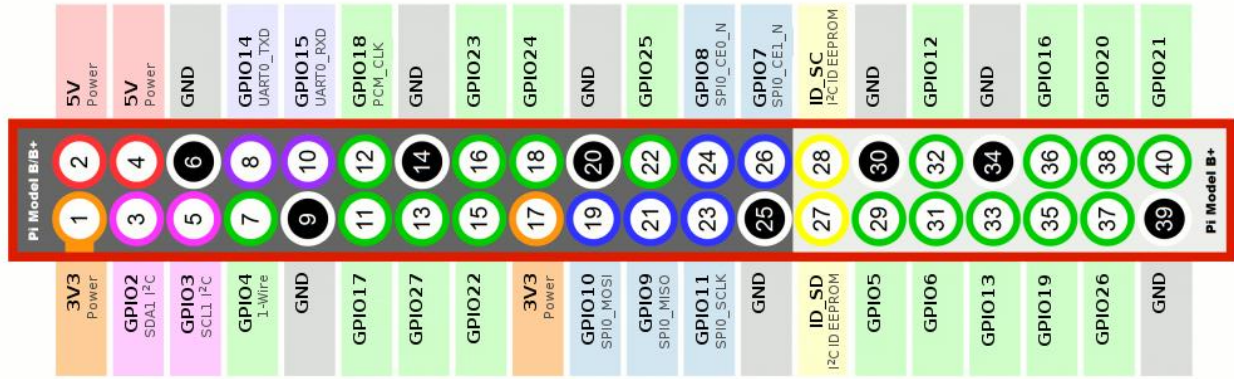
Raspberry Pi'nin GPIO'suna bir yağmur damlası sensörü bağlanması, aşağıdaki Şekil 13'te gösterildiği gibi nispeten kolay bir işlemdir:

1. Yağmur damlası sensörü kontrol panosu, bir GPIO'ya bağlanabilen 4 pine sahiptir, bunlar voltaj (VCC), topraklama (GND), dijital çıkış (DO) ve analog çıkış (AO)'dur.
2. Aşağıdakileri bir breadboard aracılığıyla veya doğrudan GPIO'ya bağlamanız gerekir:
  - a. VCC - 5V (pin 2)
  - b. GND - GND (pin 4)
  - c. DO - GPIO23 (pin16)
3. Ardından, aşağıdakileri bağlayarak iki jumper kablosu kullanarak sensör algılama panelini kontrol paneline bağlamalısınız:
  - a. Kontrol panosunun + tarafını tespit panelinin + tarafına.
  - b. Kontrol panosunun - tarafını tespit panelinin - tarafına.
4. Artık devreniz hazır.



Şekil 13. Raspberry Pi'nin GPIO pinlerine bağlı yağmur damlası sensörünün şeması.

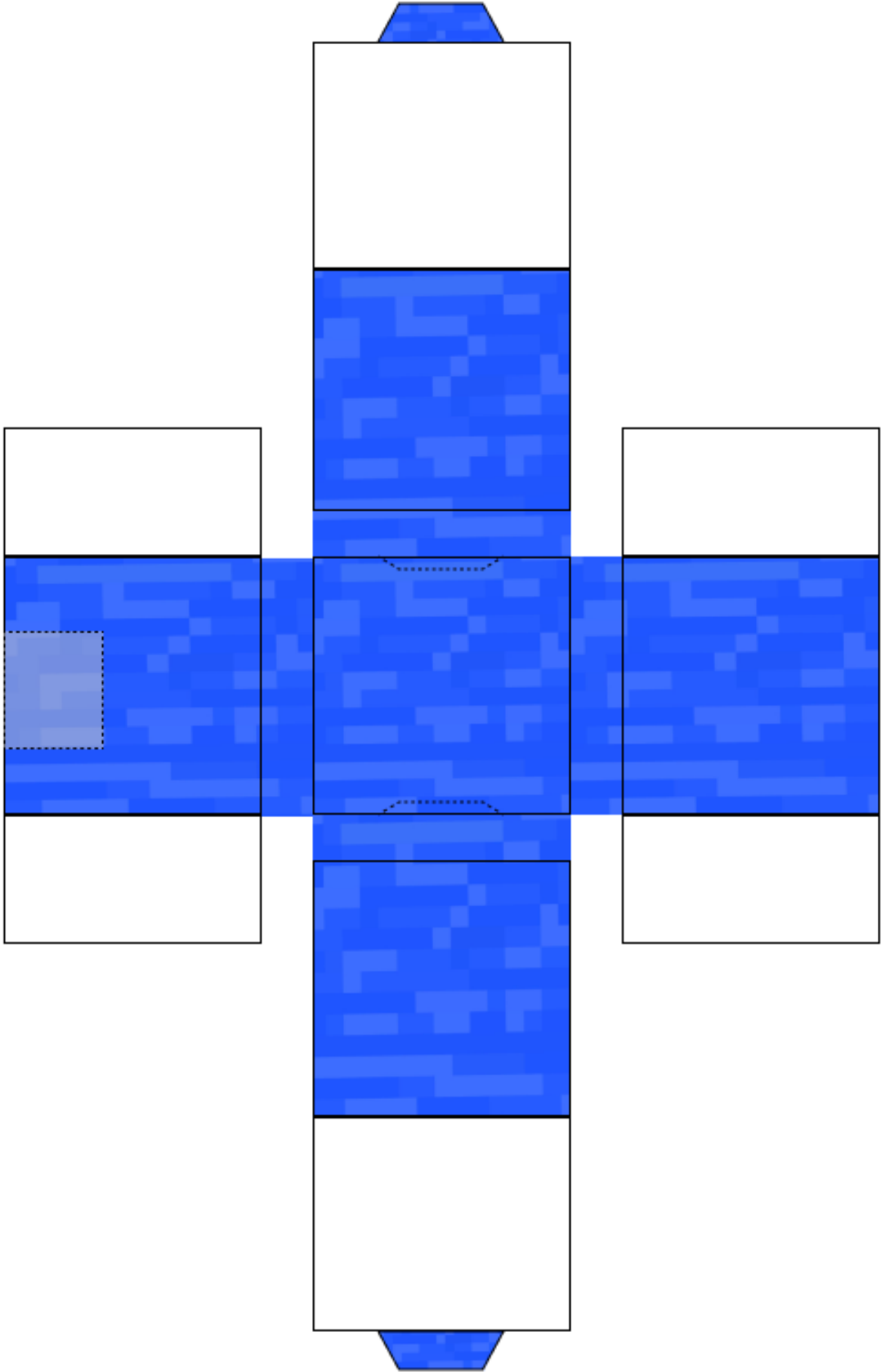
Farklı GPIO pinlerini anlamak için, her bir pinin özelliklerini açıklayan Şekil 14'ü incelemelisiniz.



Şekil 14 Raspberry Pi GPIO Pinleri.

Bir yağmur damlası sensörü hem analog hem de dijital çıkışları kontrol cihazına gönderebilir. Bu durumda, Raspberry Pi bilgisayarı sadece dijital sinyalleri okur. Yani bu aktivite için, sadece dijital çıkış gereklidir, çünkü bu program sensörün Minecraft Pi dünyasında bir sele neden olan bir yağmur damlasını tespit ettiğinde çalışacaktır.

Devre oluşturulduktan sonra, bir DIY karton yapısı kurmanız ve devreyi bunun içine yerleştirmeniz gerekir. Bu amaçla, aşağıdaki yazdırılabilir resmi kullanmanız gerekir. Daha iyi bir deneyim için, A4 beyaz karton kağıda yapıyı yazdırdığınızdan ve ardından belirtilen kenarları kestiğinizden emin olun.



Şekil 15. Yağmur damlası sensörlü fiziksel blok için yazdırılabilir karton yapı.

### 1.7.3 Python programlama

```
import RPi.GPIO as GPIO
import mcpi.minecraft as minc
import mcpi.block as block
mc = minc.Minecraft.create()
import random, time

GPIO.setmode(GPIO.BCM)
GPIO.setup(23, GPIO.IN)

def geyser(x, z):
    mc.postToChat('Flood!')
    y = mc.getHeight(x, z)
    mc.setBlocks(x-2, y+5, z-2, x+2, -60, z+2, block.WATER.id)
    time.sleep(25)
    mc.setBlocks(x-2, y+5, z-2, x+2, -60, z+2, block.AIR.id)

disasters = [geyser]
def main(disasters, mc):
    baseed = random.randint(1, 10000)
    mc.postToChat('Drop water on sensor for disaster!')
    while True:
        if GPIO.input(23) == GPIO.LOW: #Look for raindrop sensor activation
            t = random.randint(5, 120)
            t = 5
            time.sleep(t)
            random.seed(baseed + t)
            baseed = random.randint(1, 10000)
            random.shuffle(disasters)
            disaster = random.choice(disasters)
            ppos = mc.player.getTilePos()
            disaster(ppos.x, ppos.z)

try:
    import _thread as thread
except ImportError:
    import thread
thread.start_new_thread(main, (disasters, mc))
```

### 1.8 Sonuç

Bu aktivitelerin kullanımı, çocukların iklim değişikliği gibi bu olguları daha da kötüleştirmede insan aktivitelerinin rolünü anlamaları ve doğal afetleri deneyimlemesi içindir.

### Referanslar

Minecraft Pi Sürümü: HAYATTA KALMA modu benzeri. Alınan yer <https://teachwithict.weebly.com/computing-blog/minecraft-pi-edition-survival-mode-ish#sthash.S6XeUUDH.dpbs>

SunFounder Maker Education'dan Arduino & Raspberry Pi ile Yağmur Sensörü Modülü nasıl kullanılır. Alınan yer <https://www.youtube.com/watch?v=Xnisf0GP9bA&t=203s>

Raspberry Pi mesafe sensörü (ultrasonik sensör HC-SR04) kullanma. Alınan yer <https://tutorials-raspberrypi.com/raspberry-pi-ultrasonic-sensor-hc-sr04/>