

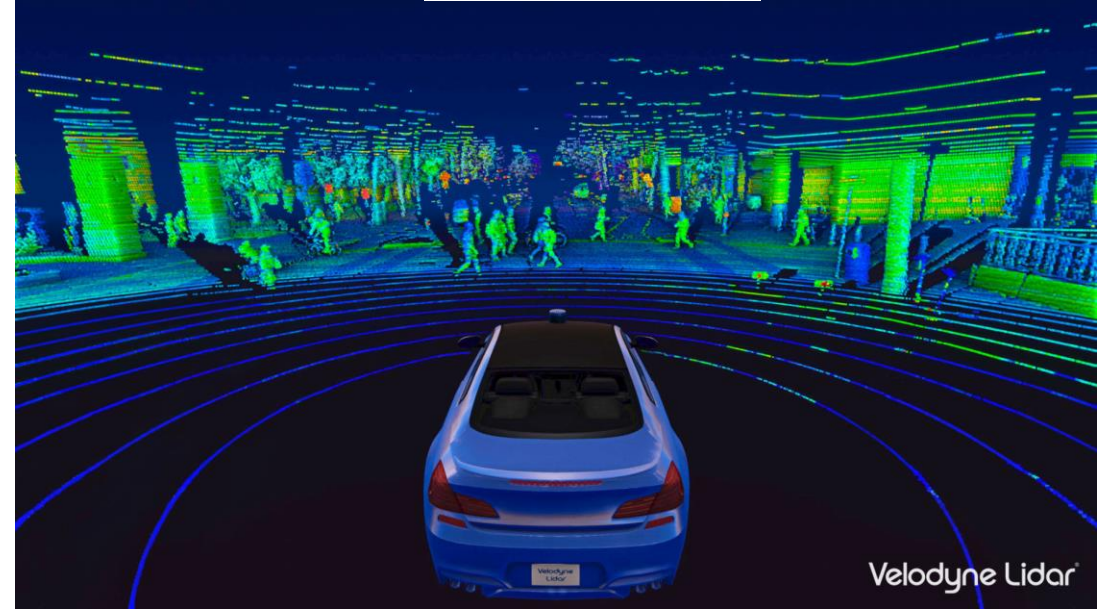


# RASPBERRY PI İLE ARALIK ÖLÇME

Ömer Mutlu Türk KAYA, Arş. Gr.

[omer.kaya@yildiz.edu.tr](mailto:omer.kaya@yildiz.edu.tr)

# MESAFE ÖLÇÜM SENSÖRLERİ

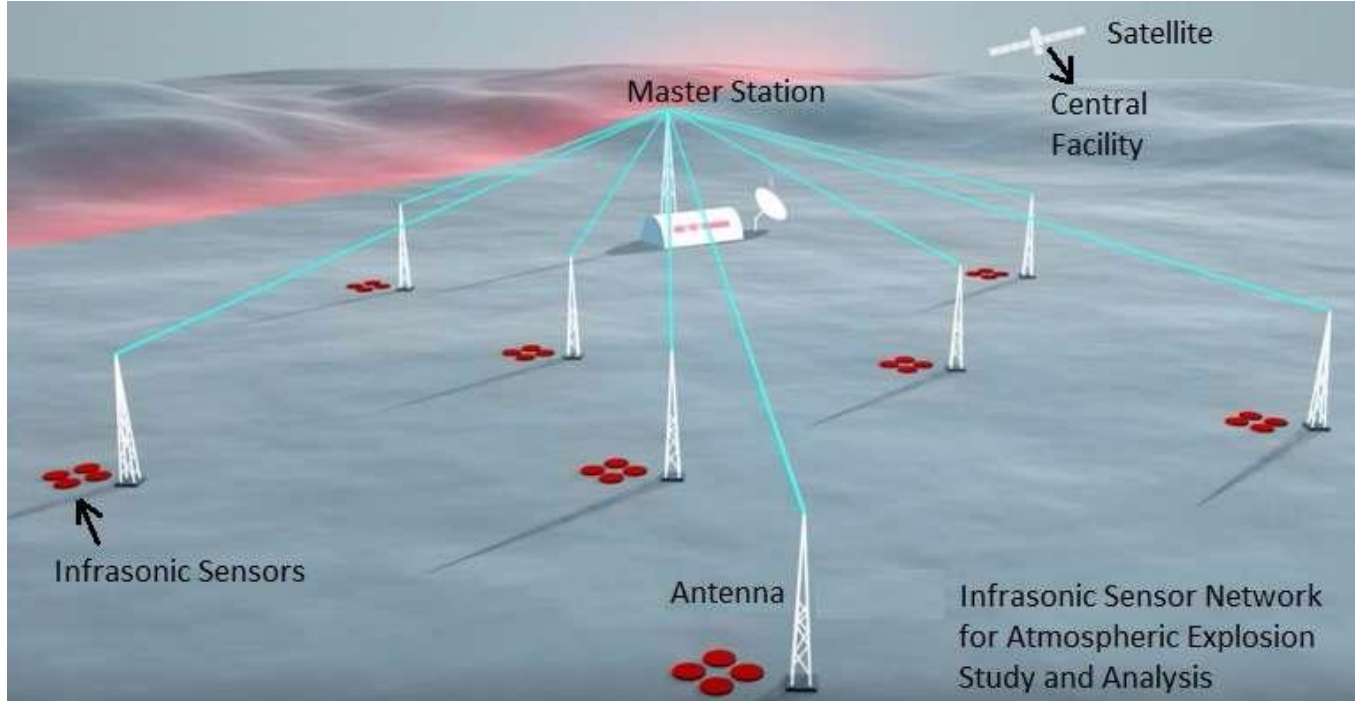


# Kullanılacak Bileşenler

• Raspberry Pi	x	1 tane
• microSD Card	x	1 tane
• Raspberry Pi Güç Adaptörü	x	1 tane
• Fare	x	1 tane
• Klavye	x	1 tane
• Ekran	x	1 tane
• HC SR04 Ultrasonik Sensör	x	1 tane
• 12 V Güç Kaynağı	x	1 tane

# KAVRAMLARI ANLAMAK

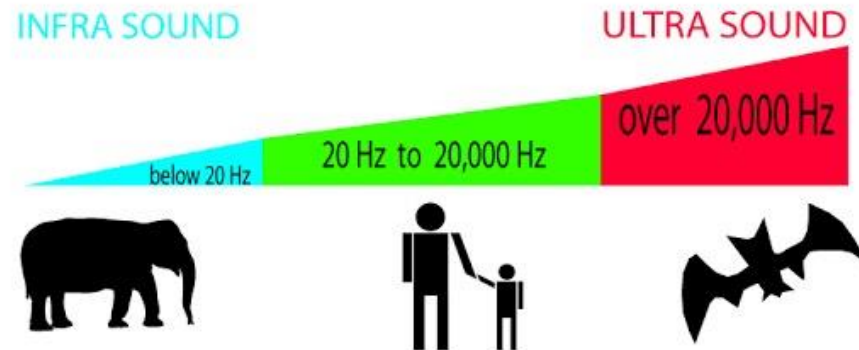
Ses nedir?      Ses üstü nedir?      Ses altı nedir?





# SES, SES ÜSTÜ (ULTRASONIC), SES ALTI (INFRASONIC) NEDİR?

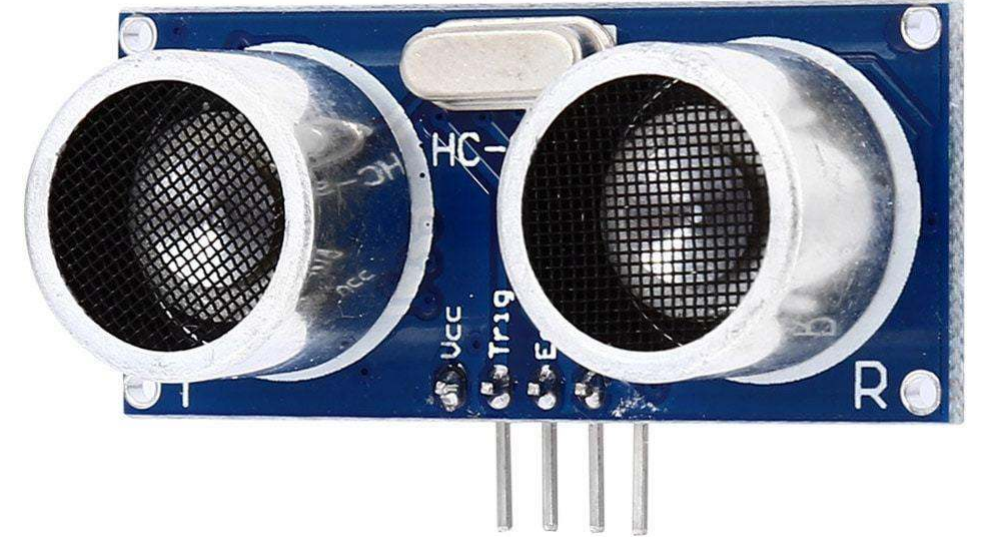
- **Ses**, canlıların işitme organlarıyla duyabileceği düzenli basınç değişimleri yani dalgalarıdır.
- Sağlıklı bir insanın duyabileceği ses **20 Hz** ile **20 kHz** aralığındadır.
- İnsanların duyabileceği üst ses sınırı olan **20 kHz'den yüksek** sıklıktaki sesler **ses üstü (ultrasonic)** olarak tanımlanmaktadır.
- Aynı biçimde insanların duyabileceği alt ses sınırı olan **20 Hz'den düşük** sıklıktaki sesler de **ses altı (infrasonic)** olarak tanımlanmaktadır.



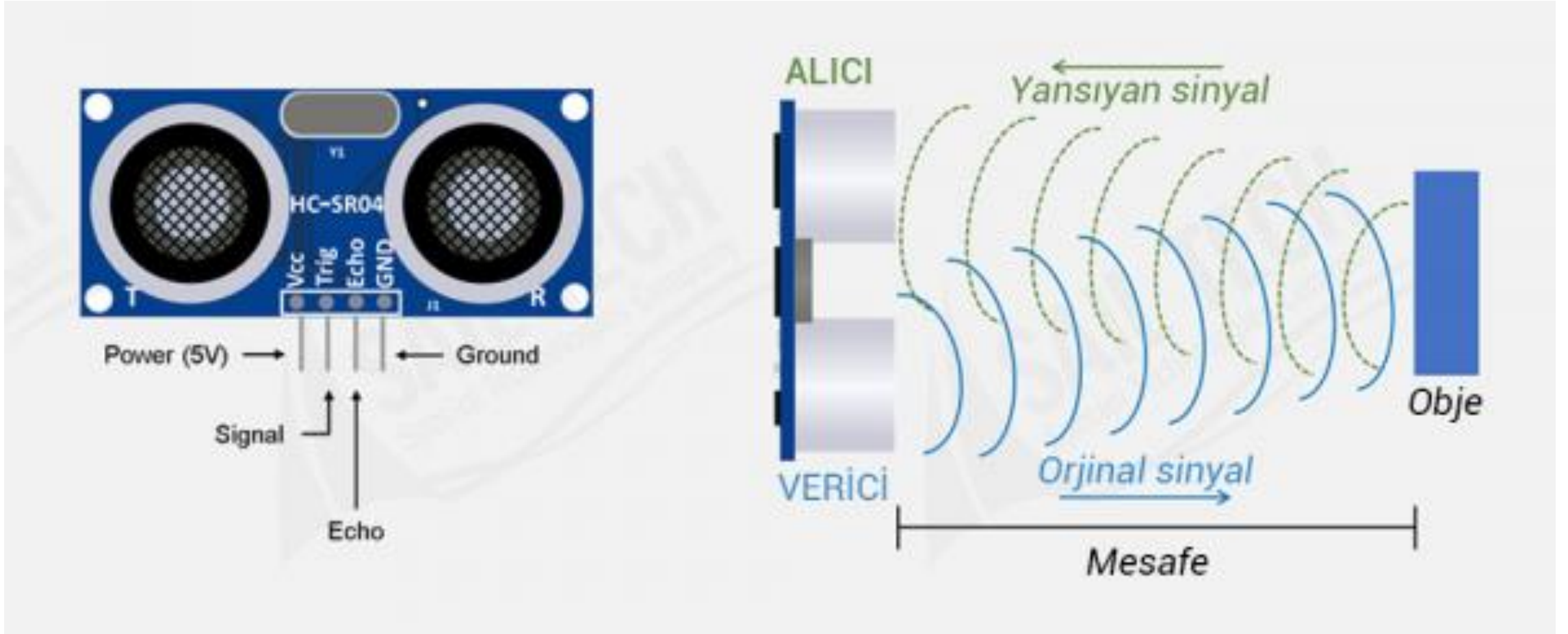
# HC SR04 SES ÜSTÜ (ULTRASONİK) SENSÖR

- Teknik Özellikleri:

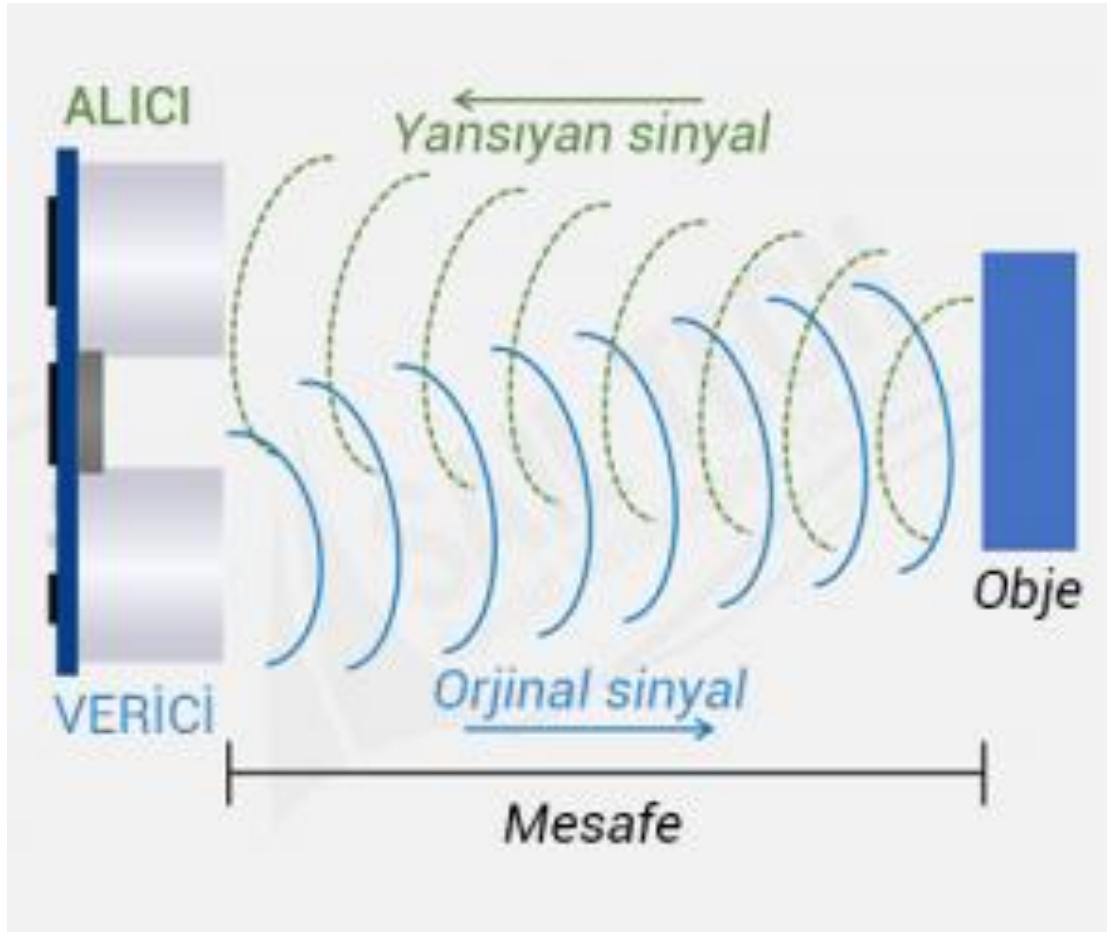
- Çalışma Gerilimi : DC 5V
- Çektiği Akım : 15 mA
- Çalışma Frekansı : 40 Hz
- Maksimum Görme Menzili : 4m
- Minimum Görme Menzili : 2cm
- Görme Açısı : 15° (Yatay)

PDF

# HC SR04 SES ÜSTÜ SENSÖR NASIL ÇALIŞIR?



# HC SR04 SES ÜSTÜ SENSÖR NASIL ÇALIŞIR?



- 20° kuru havada ses hızı 343 m/sn'dir.
- $Mesafe = ses\ hızı \times \frac{sesin\ seyahat\ süresi}{2}$

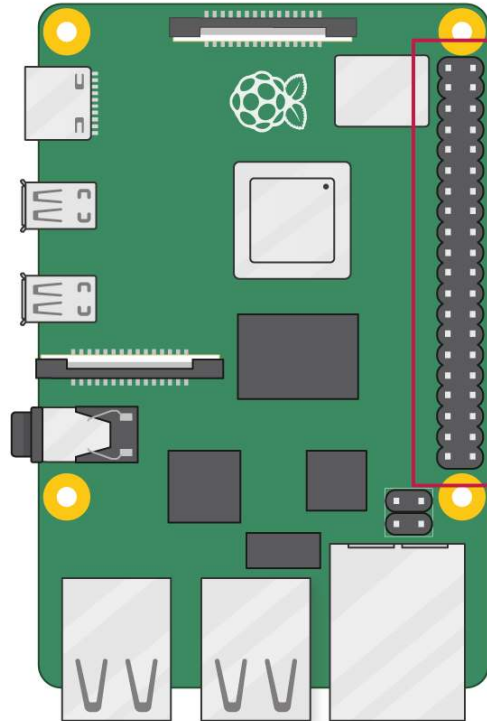


# ELEKTRONİK BAĞLANTILAR VE DEVRE

## Bacak dizilimleri:

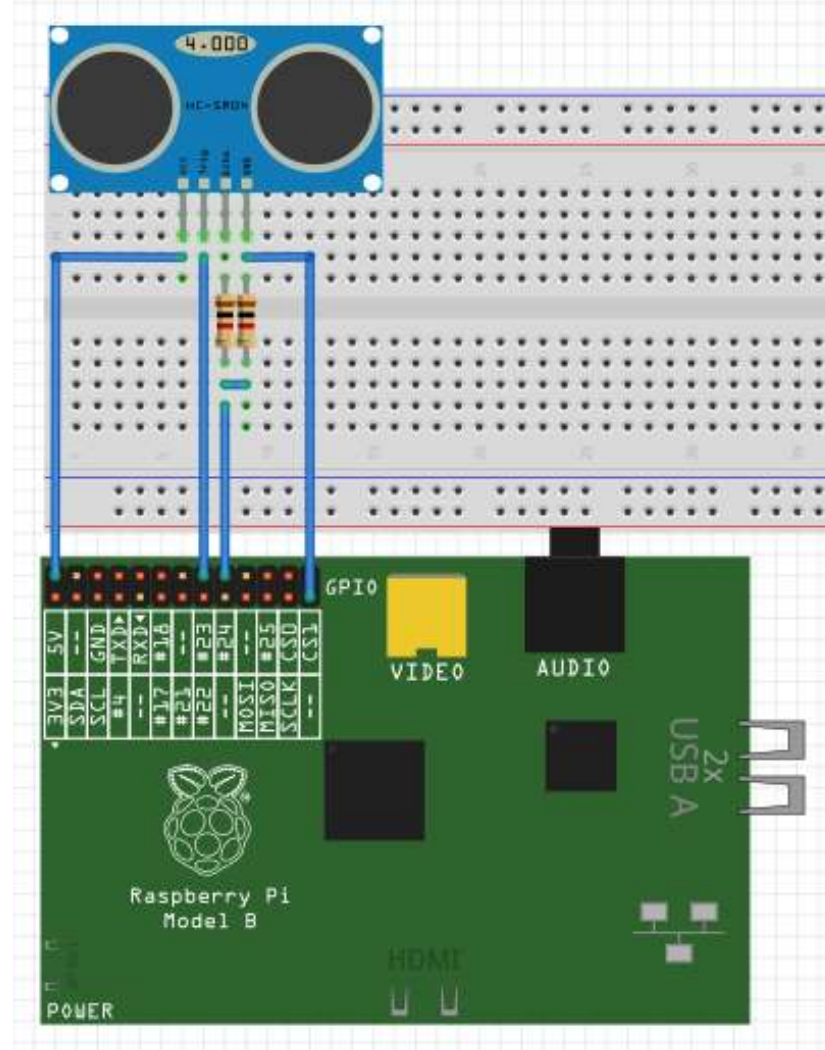
- **BCM dizilimi** : Pinlere verilen GPIO numaralarından oluşmaktadır. Bunlar sıralı numaralar değildir.

- **BOARD dizilimi**: Pinlerin fiziksel numaralandırılmasıdır. 1'den başlayıp 40'a kadar devam eden sıralı sayılardan oluşur.



3V3 power	1	2	5V power
GPIO 2 (SDA)	3	4	5V power
GPIO 3 (SCL)	5	6	Ground
GPIO 4 (GPCLK0)	7	8	GPIO 14 (TXD)
Ground	9	10	GPIO 15 (RXD)
GPIO 17	11	12	GPIO 18 (PCM_CLK)
GPIO 27	13	14	Ground
GPIO 22	15	16	GPIO 23
3V3 power	17	18	GPIO 24
GPIO 10 (MOSI)	19	20	Ground
GPIO 9 (MISO)	21	22	GPIO 25
GPIO 11 (SCLK)	23	24	GPIO 8 (CE0)
Ground	25	26	GPIO 7 (CE1)
GPIO 0 (ID_SD)	27	28	GPIO 1 (ID_SC)
GPIO 5	29	30	Ground
GPIO 6	31	32	GPIO 12 (PWM0)
GPIO 13 (PWM1)	33	34	Ground
GPIO 19 (PCM_FS)	35	36	GPIO 16
GPIO 26	37	38	GPIO 20 (PCM_DIN)
Ground	39	40	GPIO 21 (PCM_DOUT)

# ELEKTRONİK BAĞLANTILAR VE DEVRE



**GPIO 23 : TRIG**  
**GPIO 24 : ECHO**

# UYGULAMA KODLARI

```
1 import RPi.GPIO as GPIO
2 import time
3
```

- Kütüphanelerin ve metotların eklenmesi

# UYGULAMA KODLARI

```
1 import RPi.GPIO as GPIO
2 import time
3
4 GPIO.setmode(GPIO.BCM)
5 GPIO.setwarnings(False)
6 TRIG = 23
7 ECHO = 24
8 GPIO.setup(TRIG,GPIO.OUT)
9 GPIO.setup(ECHO,GPIO.IN)
10
```

- Pin isimlendirme modunun belirlenmesi
- Pin atamalarının yapılması
- Ataması yapılan pinlerin giriş mi çıkış mı olacağının belirlenmesi



# UYGULAMA KODLARI

```
1 import RPi.GPIO as GPIO
2 import time
3
4 GPIO.setmode(GPIO.BCM)
5 GPIO.setwarnings(False)
6 TRIG = 23
7 ECHO = 24
8 GPIO.setup(TRIG,GPIO.OUT)
9 GPIO.setup(ECHO,GPIO.IN)
10
11▼ while True:
```

- Kodumuzun sürekli çalışabilmesi için bir sonsuz döngü oluşturuyoruz.

# UYGULAMA KODLARI

```
1 import RPi.GPIO as GPIO
2 import time
3
4 GPIO.setmode(GPIO.BCM)
5 GPIO.setwarnings(False)
6 TRIG = 23
7 ECHO = 24
8 GPIO.setup(TRIG,GPIO.OUT)
9 GPIO.setup(ECHO,GPIO.IN)
10
11 while True:
12     GPIO.output(TRIG, False)
13     time.sleep(2)
14     GPIO.output(TRIG, True)
15     time.sleep(0.00001)
16     GPIO.output(TRIG, False)
```

10 mikrosaniye

# UYGULAMA KODLARI

```
1 import RPi.GPIO as GPIO
2 import time
3
4 GPIO.setmode(GPIO.BCM)
5 GPIO.setwarnings(False)
6 TRIG = 23
7 ECHO = 24
8 GPIO.setup(TRIG,GPIO.OUT)
9 GPIO.setup(ECHO,GPIO.IN)
10
11 while True:
12     GPIO.output(TRIG, False)
13     time.sleep(2)
14     GPIO.output(TRIG, True)
15     time.sleep(0.00001)
16     GPIO.output(TRIG, False)
17
18     while GPIO.input(ECHO)==0:
19         darbe_baslangici = time.time()
20
21     while GPIO.input(ECHO)==1:
22         darbe_bitisi = time.time()
23
24     darbe_seyahat_suresi = darbe_bitisi - darbe_baslangici
25
```

# UYGULAMA KODLARI

```
1 import RPi.GPIO as GPIO
2 import time
3
4 GPIO.setmode(GPIO.BCM)
5 GPIO.setwarnings(False)
6 TRIG = 23
7 ECHO = 24
8 GPIO.setup(TRIG,GPIO.OUT)
9 GPIO.setup(ECHO,GPIO.IN)
10
11 while True:
12     GPIO.output(TRIG, False)
13     time.sleep(2)
14     GPIO.output(TRIG, True)
15     time.sleep(0.00001)
16     GPIO.output(TRIG, False)
17
18     while GPIO.input(ECHO)==0:
19         darbe_baslangici = time.time()
20
21     while GPIO.input(ECHO)==1:
22         darbe_bitisi = time.time()
23
24     darbe_seyahat_suresi = darbe_bitisi - darbe_baslangici
25
26     aralik = (darbe_seyahat_suresi / 2) * 34300
27     aralik = round(aralik, 2)
```

- 20° kuru havada ses hızı 343 m/sn
- Bu da 34300 cm/sn yapar.
- 2'ye bölüm sesin seyahat ettiği yolun yarısını yani sensör ile cisim arasındaki mesafeyi bulmak içindir.



# UYGULAMA KODLARI

```
1 import RPi.GPIO as GPIO
2 import time
3
4 GPIO.setmode(GPIO.BCM)
5 GPIO.setwarnings(False)
6 TRIG = 23
7 ECHO = 24
8 GPIO.setup(TRIG,GPIO.OUT)
9 GPIO.setup(ECHO,GPIO.IN)
10
11 while True:
12     GPIO.output(TRIG, False)
13     time.sleep(2)
14     GPIO.output(TRIG, True)
15     time.sleep(0.00001)
16     GPIO.output(TRIG, False)
17
18     while GPIO.input(ECHO)==0:
19         darbe_baslangici = time.time()
20
21     while GPIO.input(ECHO)==1:
22         darbe_bitisi = time.time()
23
24     darbe_seyahat_suresi = darbe_bitisi - darbe_baslangici
25
26     aralik = (darbe_seyahat_suresi / 2) * 34300
27     aralik = round(aralik, 2)
28
29     if aralik > 2 and aralik < 400:
30         print "Mesafe:",aralik,"cm"
31     else:
32         print "Hatali olcum"
```

- Sensörün ölçüm menzili  
2 – 400 cm aralığıdır.

# Teşekkürler

