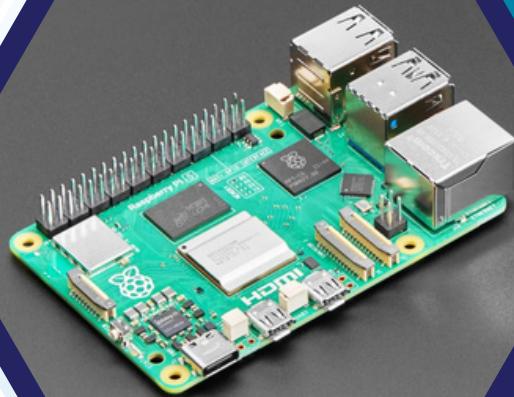
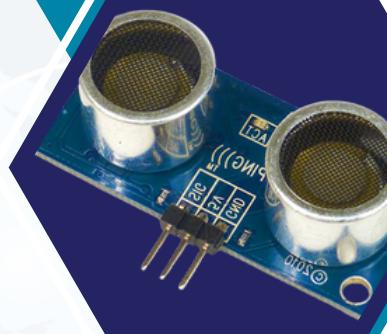


Raspberry Pi

ARAÇ PARK SENSÖRÜ

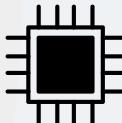


Ayoub Numan Abdallah

2020123118

Ahmed muhamed dirie

2022123158



Mikroİşlemciler dersi

İÇİNDEKİLER

1. Proje Tanıtımı	3
2. Projenin Amacı	3
3. Proje Kapsamı	3
4. Kullanılan Teknolojiler ve Araçlar	4
4.1 Raspberry Pi 3	4-5
4.2 Ultrasonik Sensör (HC-SR04)	5-6
4.3 İletişim Protokolleri ve Bağlantı	7
4.4 Yazılım Dili	7
4.5 Kullanılan Kütüphaneler	7
4.6 Çıkış ve Uyarı Mekanizmaları	8
4.7 Güç Kaynağı	8
5. Mikroişlemci ve Yazılım Modelinin İşleyışı	10
5.1 Mikroişlemci İşleyışı	10
5.2 Yazılım Modelinin İşleyışı	11-12
6. Teknik Şemalar ve Diyagramlar	12
6.1 Blok Diyagram	12
6.2 Akış Şeması	13
6.3 Devre Şeması	13
7. Beklenen Sonuçlar ve Sistem İşleyiği Tahminleri	14
7.1 Beklenen Doğruluk Oranı	14
7.2 Sistem Çıktısı	15
8. Alternatif Teknolojiler ve Seçim Nedenleri	16
8.1 Alternatif Teknolojiler	16
8.2 Teknolojilerin Seçim Nedenleri	16
9. Gerçek Hayatta Kullanım Senaryoları	17
10. KAYNAKÇA	18

1. PROJE TANITIMI:

Bu proje, bir **Raspberry Pi** mikro bilgisayarı ve **ultrasonik mesafe sensörü** kullanarak, araçların park sırasında çevredeki nesnelere olan mesafelerini ölçen ve kullanıcıya **sesli ya da görsel** uyarılar veren bir sistem geliştirmeyi hedefler. Sistem, özellikle dar park alanlarında kolaylık sağlayarak sürücülerin araçlarını güvenli bir şekilde park etmesine yardımcı olur.

2. PROJENIN AMACI:

Bu projenin amacı, araçların park sırasında çevredeki nesnelere zarar vermesini önlemek ve sürücülere güvenli, kolay bir park deneyimi sunmaktır. **Raspberry Pi ve ultrasonik sensörler kullanılarak** geliştirilecek sistem, düşük maliyetli ve etkili bir çözüm sağlarken aynı zamanda elektronik, yazılım ve IoT uygulamaları hakkında bilgi edinmek için bir öğrenme aracı olarak da kullanılacaktır.

Proje, **hem** bireysel kullanıcılar için ekonomik bir alternatif sunmayı **hem de** açık kaynak teknolojileri kullanarak özelleştirilebilir bir platform geliştirmeyi hedeflemektedir.

3. PROJE KAPSAMI:

Projenin kapsamı, araçlar için akıllı park sensörü sisteminin geliştirilmesini içerir. Bu sistem, araç çevresindeki engelleri algılamak, sürücüyü görsel ve işitsel olarak uyarmak ve park etme sürecini kolaylaştırmak amacıyla tasarlanmıştır. Proje kapsamında Raspberry Pi 3 mikroişlemcisi, ultrasonik sensörler, LED ışıklar, ve buzzer. Sistem, Python programlama diliyle geliştirilmiş bir yazılım modeli üzerinden çalışacak, mesafe ölçümlerini yaparak kullanıcıya anlık olarak bilgi sağlayacaktır. Ayrıca, sistemin tasarıımı, farklı park senaryolarına uygun şekilde çalışabilecek hassasiyette olacaktır.

4. KULLANILAN TEKNOLOJILER VE ARAÇLAR:

4.1 RASPBERRY PI 3

Raspberry Pi 3, düşük maliyetli, küçük boyutlu, ancak güçlü bir mikroişlemci kartıdır. Özellikle IoT (Nesnelerin İnterneti), otomasyon, robotik ve öğrenme projelerinde sıkılıkla kullanılır. Bu cihaz, bilgisayar tabanlı işlemleri yapabilen ve programlanabilir bir platform sunar.

Donanım Özellikleri:

- **İşlemci:** ARM Cortex-A53, dört çekirdek, 1.2 GHz.

Bu güçlü işlemci, hem hızlı veri işleme hem de gerçek zamanlı işlemler için uygundur.

- **RAM:** 1 GB.

Bu bellek kapasitesi, sensör verilerinin işlenmesi ve yazılımın çalıştırılması için yeterlidir.

- **GPIO:** 40 adet genel amaçlı giriş/çıkış pini.

Sensörler, LED'ler, buzzer ve diğer çevre birimleri ile bağlantı sağlamak için kullanılır.

- **Bağlantı:** Dahili Wi-Fi, Bluetooth, harici cihazlarla kablosuz iletişim için uygundur.

- **Görüntü Çıkışı:** **HDMI** çıkışı.

Monitör bağlayarak sistem arayüzü görüntülenebilir.

- **Güç Girişi:** 5V, 2.5A micro-USB bağlantısı.

- **Depolama:** **MicroSD** kart yuvası.

Yazılım ve veriler için hafıza sağlar.

Projedeki Rolü:

- **Veri Toplama:**

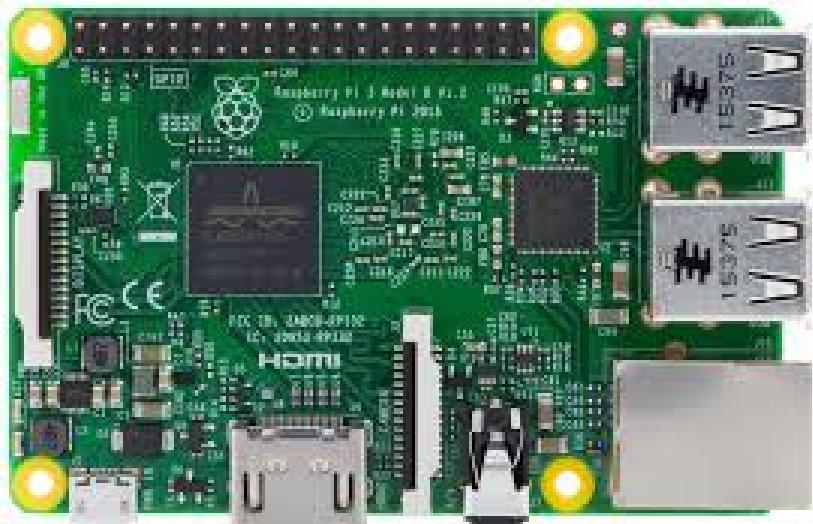
Sensörlerden gelen mesafe verilerini alır ve işleme başlatır.

- **Veri İşleme:**

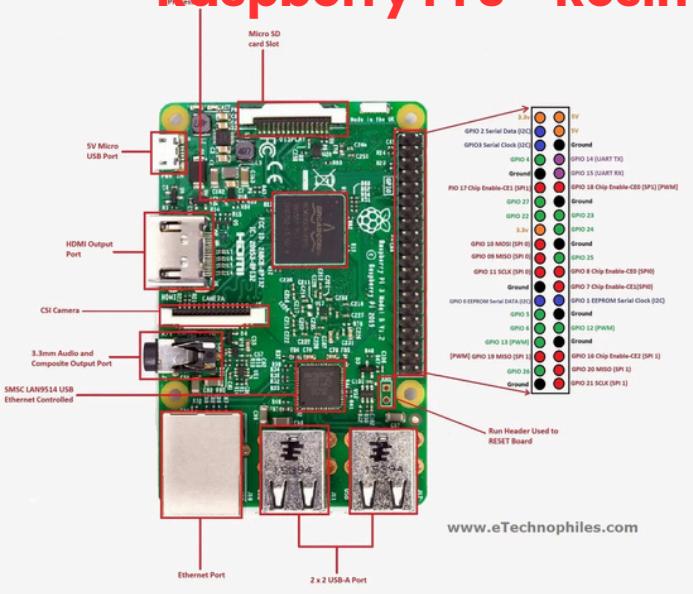
Python yazılımı kullanarak mesafeyi hesaplar.

- **Çıktı Yönetimi:**

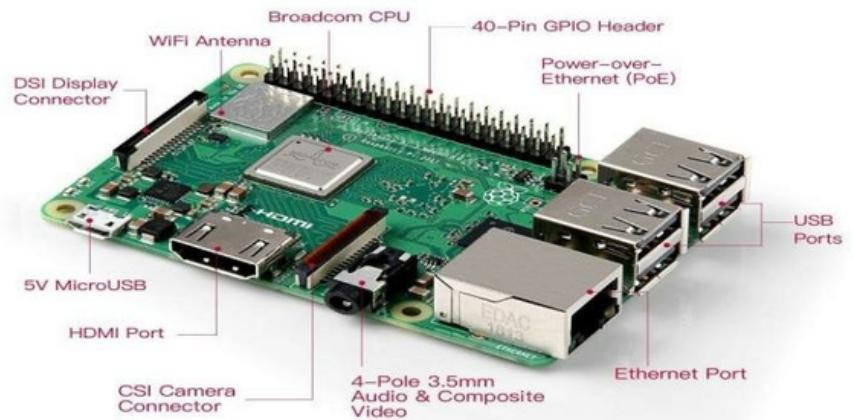
Mesafeye göre **LED'leri ve buzzeri** kontrol ederek sürücüye uyarılar verir.



Raspberry Pi 3 – Resim 1



Raspberry Pi 3 – Resim 2



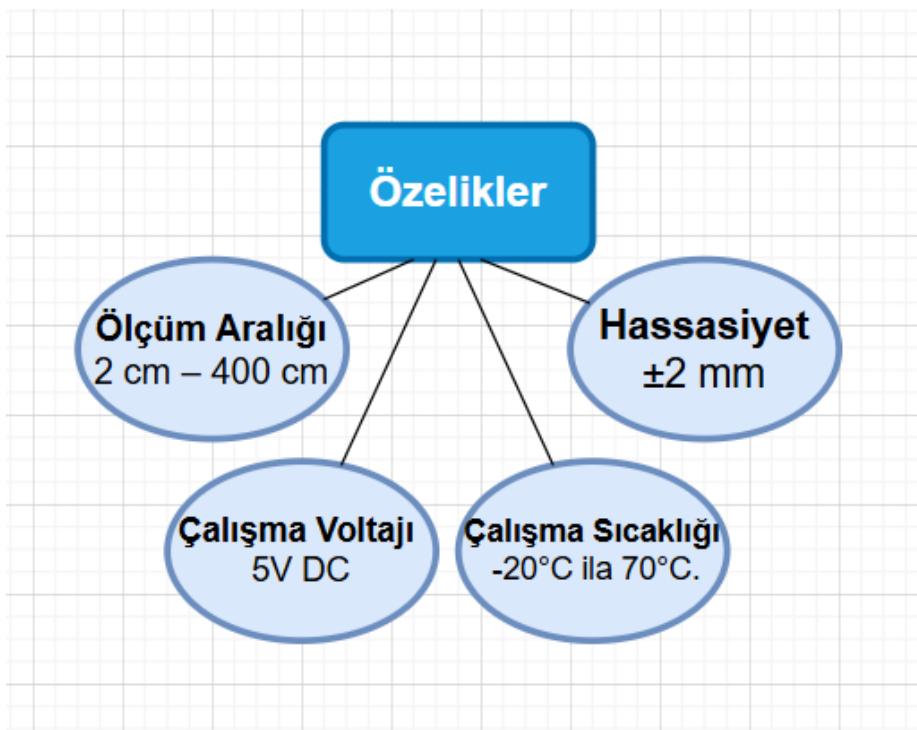
Raspberry Pi 3 – Resim 4

Raspberry Pi 3 – Resim 3

4.2 ULTRASONIK SENSÖR (HC-SR04)

HC-SR04, ses dalgalarını kullanarak mesafe ölçen bir ultrasonik sensördür. Sensör, engellerin mesafesini hassas bir şekilde ölçerek projede kullanılan veri tabanını oluşturur.

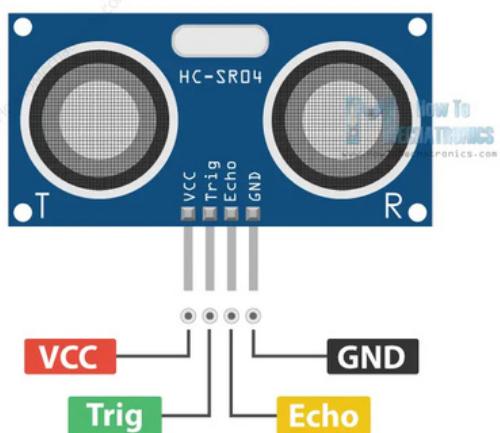
Trig ve Echo pinleri, Raspberry Pi GPIO pinlerine bağlanır.



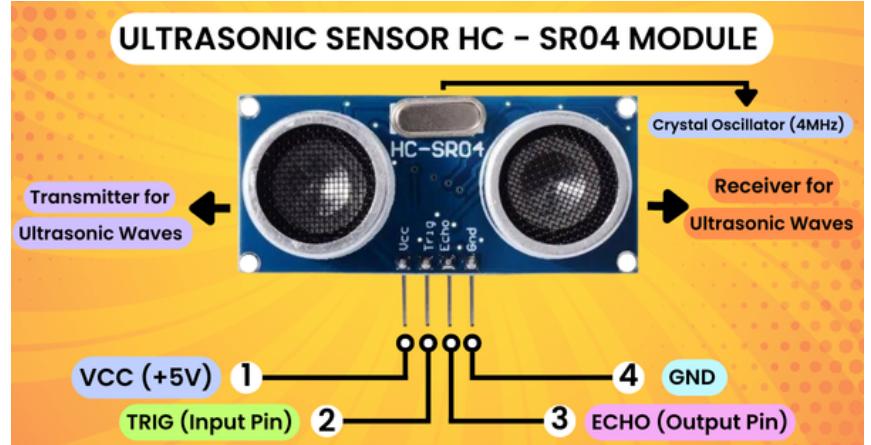
Projedeki Rolü:

Raspberry Pi'ye mesafe bilgisi göndererek park sensörünün **temel ölçüm** mekanizmasını **sağlar**.

HC-SR04 Pinout



HC-SR04 – Resim 1



HC-SR04 – Resim 2

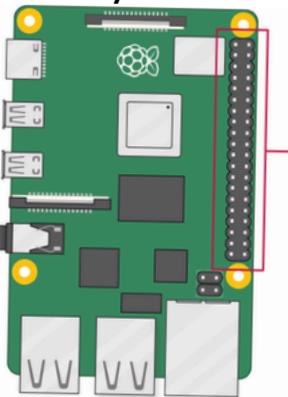
4.3 İLETİŞİM PROTOKOLLERİ VE BAĞLANTI

GPIO Kullanımı:

- **Raspberry Pi 3'ün** GPIO pinleri, sensörler ve çevre birimleriyle veri alışverişi yapılmasını sağlar.
- GPIO pinleri, hem giriş (sensör veri alımı) hem de çıkış (LED kontrolü) olarak yapılandırılabilir.

İletişim Yapısı:

- **Trig ve Echo:** Sensörle iletişim için kullanılır.
- **LED ve Buzzer Bağlantıları:** Belirli eşik mesafelerinde görsel ve sesli uyarılar sağlar.



Alternate Function	
I2C1 SDA	3.3V PWR
I2C1 SCL	GPIO 2
	3
	GPIO 3
	5
	GPIO 4
	7
GND	9
	GPIO 17
	11
	GPIO 27
	13
	GPIO 22
	15
3.3V PWR	17
SPI0 MOSI	GPIO 10
SPI0 MISO	GPIO 9
SPI0 SCLK	GPIO 11
	21
	GPIO 11
	23
GND	25
Reserved	27
GPIO 5	29
GPIO 6	31
GPIO 13	33
SPI1 MISO	GPIO 19
	35
	GPIO 26
	37
GND	39
SV PWR	1
SV PWR	3
GND	5
UART0 TX	7
UART0 RX	9
GPIO 18	10
GND	12
GPIO 23	14
GPIO 24	16
GND	18
GPIO 25	20
GPIO 8	22
SPI0 CS0	24
GPIO 7	26
SPI0 CS1	28
Reserved	30
GND	32
GPIO 12	34
GND	36
GPIO 16	38
SPI1 CS0	40
GPIO 20	39
SPI1 MOSI	41
SPI1 SCLK	42

GPIO pinleri

4.4 YAZILIM DİLİ

GPIO Kullanımı: **Python**, Raspberry Pi projelerinde en çok tercih edilen programlama dilidir. [Bunun başlıca nedenleri:](#)



- Kolay öğrenilebilir ve okunabilir olması.
- GPIO kütüphaneleri ile donanım kontrolünün kolayca yapılabilmesi.
- Sensörlerden veri toplama, işleme ve çıktı üretme süreçlerinde geniş kütüphane desteği sunması.

4.5 KULLANILAN KÜTÜPHANELER

- **RPi.GPIO:** Donanımın kontrol edilmesi ve yazılım aracılığıyla pinlerin yönetilmesi için kullanılır.
- **time:** Zaman ölçümelerini sağlar.

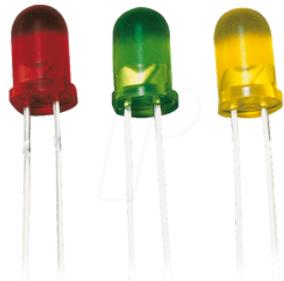
4.6 ÇIKIŞ VE UYARI MEKANİZMALARI

LED:

- Görsel uyarılar için kullanılır.
- GPIO pinlerine bağlanarak mesafe durumuna göre yanıp söner.

Buzzer:

- Sesli uyarılar sağlar.
- Özellikle tehlikeli mesafelerde sürücüyü uyarmak için kullanılır.



LED'ler



Buzzer

4.7 GÜC KAYNAĞI

Raspberry Pi 3 ve çevre birimleri için **5V, 2.5A** bir **adaptör** gereklidir.



Adaptör VE micro kablosu

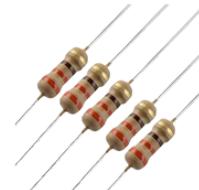
4.8 DIRENCİLER (330Ω VE $10K\Omega$)

LED'ler için uygun akım sınırlamasını sağlar. (330Ω)

Ultrasonik sensör sinyallerinin düzenlenmesinde de kullanılır. ($10k\Omega$)



($10k\Omega$)



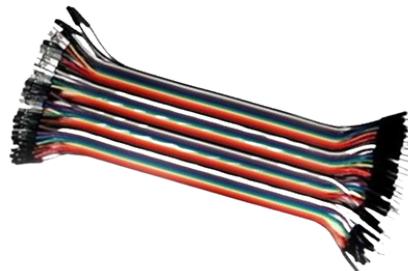
(330Ω)

4.9 DIĞER MALZEMELER:



Devre Tahtası

Sensör ve LED/Buzzer gibi bileşenlerin **Raspberry Pi'ye** kolayca bağlanmasılığını sağlar.



Jumper kablosu

Donanım bileşenlerini **Raspberry Pi'nin** GPIO pinlerine bağlamak için.



SD kartı

Raspberry Pi işletim sistemi ve proje dosyalarını depolamak için.



Wifi
(iletişim sağlamak için)



Klavye ve Fare

Raspberry Pi'yi kurulum sırasında yapılandırmak için.

5. MIKROİŞLEMCI VE YAZILIM MODELİNİN İŞLEYİŞİ

5.1 MIKROİŞLEMCI İŞLEYİŞİ

1. Donanım Bileşenleri

Mikroişlemci (Raspberry Pi) birden fazla bileşenle çalışır:

Giriş Bileşenleri:

Sensörler: Örneğin, ultrasonik park sensörü, mikroişlemciye mesafe verilerini sağlar.

Çıkış Bileşenleri:

Ekranlar: Sistemin durumu veya sensör verileri bir ekranда gösterilir.

(Sistem ekranı)

Buzzer/LED: Kullanıcıyı uyarın çıkış bileşenleri.

2. Veri Akışı

Sensörlerden Veri Toplama:

Sensör belirli bir frekansta veri toplar ve bu veriler mikroişlemciye iletilir (örneğin, **GPIO pinleri** veya **İletişim protokolleri** aracılığıyla).

Örnek: Ultrasonik sensör mesafeyi ölçer ve bu veriyi Raspberry Pi'ye gönderir.

İşleme:

Mikroişlemci, alınan ham veriyi bir algoritma veya yazılım modeli kullanarak işler.

Filtreleme: Gürültüyü azaltmak için veriler filtrelenir.

Dönüşüm: Ham veri (örneğin, **voltaj**) insan tarafından anlaşılabılır bir forma dönüştürülür (örneğin, **mesafe**: santimetre/metre).

Karar Mekanizması:

İşlenmiş veri, yazılım tarafından belirlenen kriterlere göre değerlendirilir.

Örnek: Eğer mesafe **10 cm'den azsa** bir **alarm(Buzzer)** tetiklenir.

Çıkış: Mikroişlemci, işlenmiş verileri bir ekran'a gönderir veya bir çıktı (örneğin, alarm sesi veya ışık) üretir.

5.2 YAZILIM MODELİNİN İŞLEYİŞİ

Yazılım modeli, mikroişlemcide çalışarak donanım bileşenleriyle etkileşimi sağlar. **Bu model aşağıdaki adımları içerir:**

A. Başlangıç ve Kurulum

1. Donanım İletişimi Ayarları:

Raspberry Pi, ultrasonik sensörden gelen veriyi GPIO pinleri üzerinden okur.

GPIO pinlerinin, iletişim protokollerinin ve diğer bileşenlerin tanımlanması yapılır.

2. Kütüphanelerin Yüklenmesi:

Gerekli yazılım kütüphaneleri yüklenir (örneğin, **Python'da RPi.GPIO ve time**). **Python programı** veriyi işler ve mesafeyi hesaplar.

3. Başlangıç Kontrolleri:

Donanım ve sensörlerin düzgün çalışıp çalışmadığı test edilir.

- Raspberry Pi GPIO pinlerinin programlanması.
- PWM veya dijital sinyallerin kontrolü.

B. Veri Toplama ve İşleme

1. Sensörden Veri Okuma:

Yazılım, belirli aralıklarla sensörlerden veri okur. Bu, bir döngü içinde sürekli olarak gerçekleşir.

2. Algoritma ve İşleme:

- **Veri Doğrulama:** Okunan veriler hatalara karşı kontrol edilir.
- **Karar Alma:** Önceden belirlenen eşik değerler kullanılarak gerekli aksiyonlar alınır.

C. Çıktı Yönetimi

Yazılım, işlenen veriyi belirli bir cihaz veya arayüz üzerinden kullanıcılara iletir:

- **Veri Görselleştirme:** Sensör verileri bir uygulama aracılığıyla görüntülenir.
- **Uyarılar:** **Sesli veya görsel** uyarılar tetiklenir.

D. Teknik Detaylar

- **HC-SR04 Sensörü:**

- Sensör, "Trigger" pini ile sinyal gönderir ve yankının geri dönüş süresini "Echo" pini üzerinden okur.
- Ölçülen süre, Python kodunda mesafeye dönüştürülür.

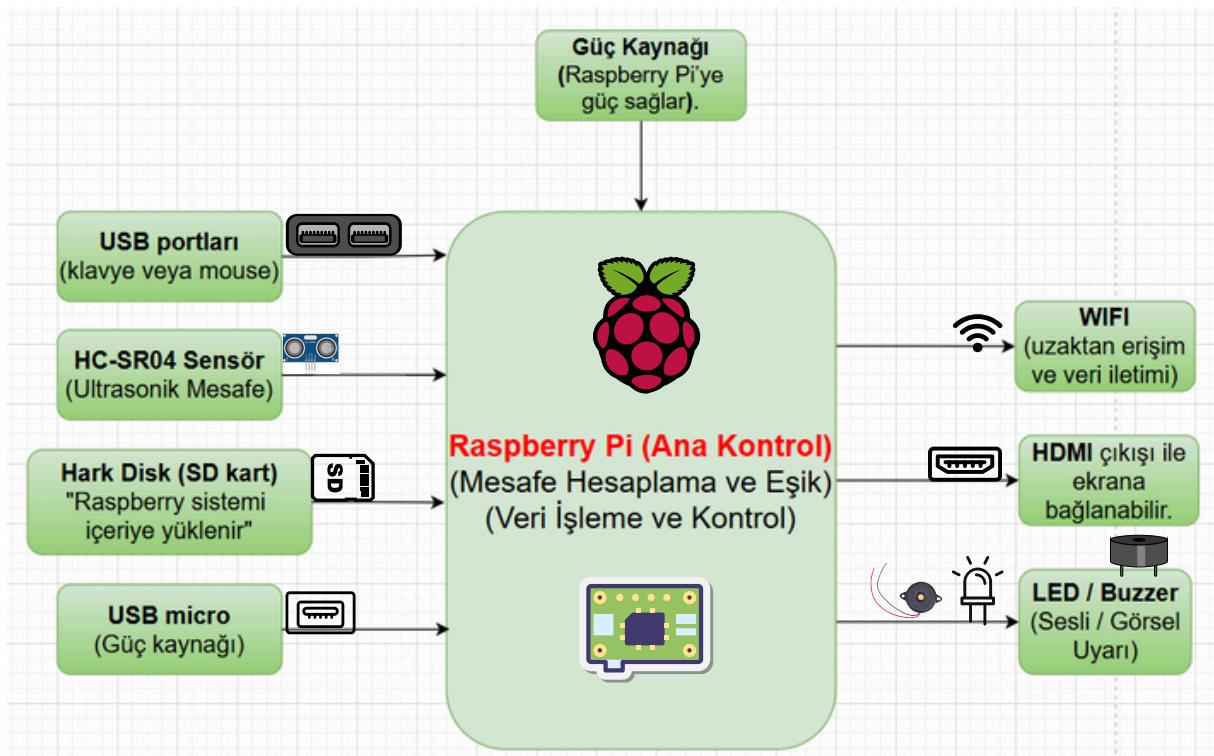
- **Raspberry Pi GPIO Kullanımı:**

- Ultrasonik sensör "Trigger" ve "Echo" pinlerine bağlanır.
- **Buzzer veya LED**, GPIO pinleri üzerinden kontrol edilir.



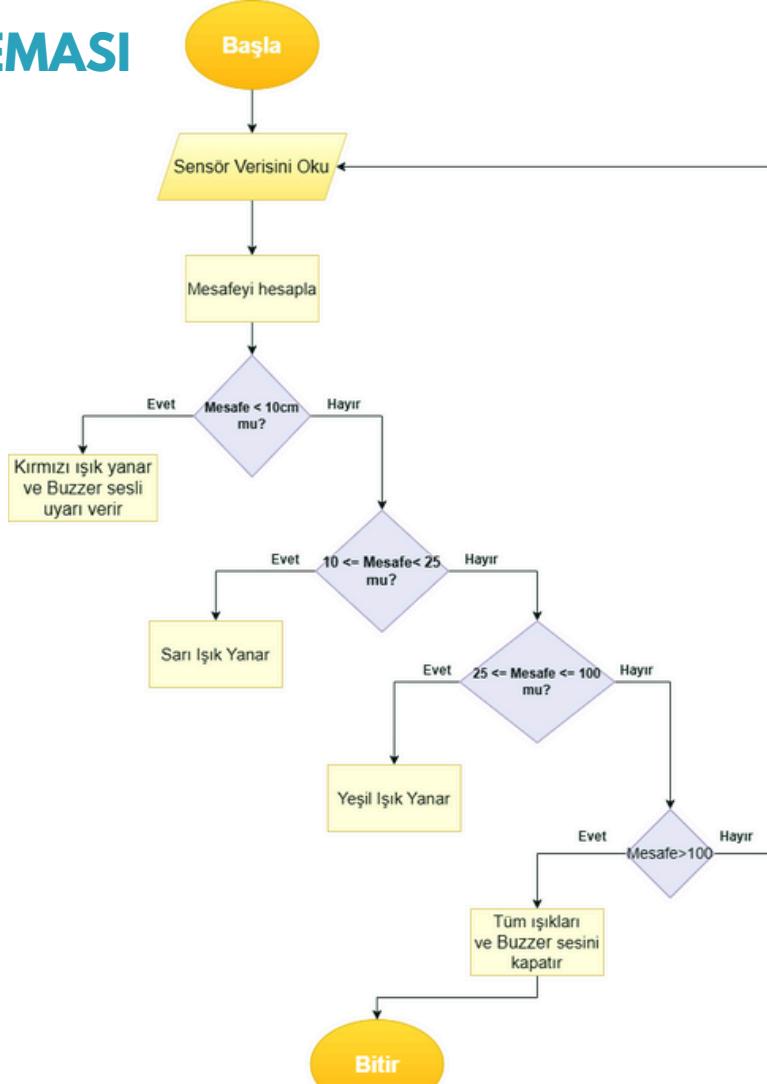
6. TEKNİK ŞEMALAR VE DIYAGRAMLAR

6.1 BLOK DIYAGRAM



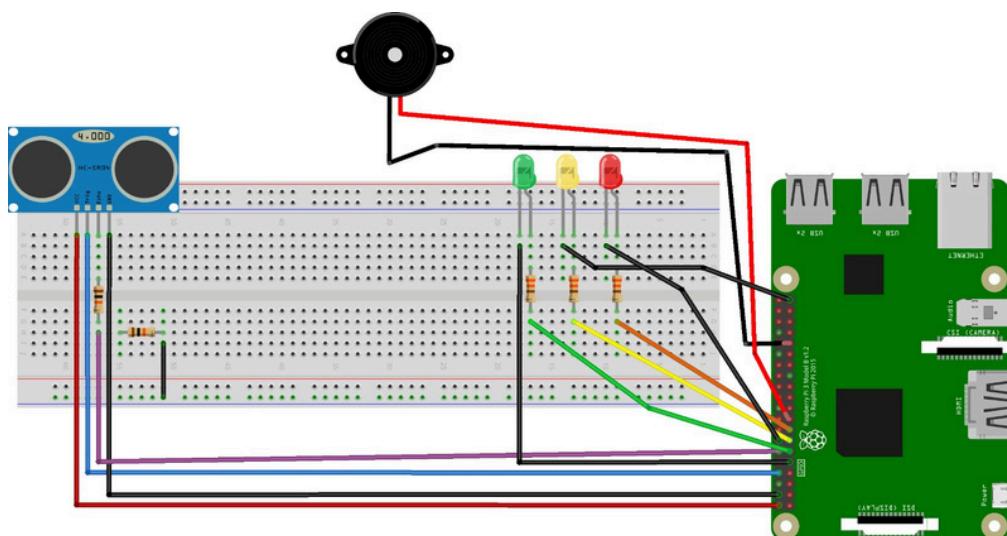
diagrams.net sitesini kullanılarak bu diyagram çizildi.

6.2 AKIŞ ŞEMASI



diagrams.net
sitesini
kullanılarak
bu şema
çizildi.

6.3 DEVRE ŞEMASI



Fritzing programı kullanılarak bu devre çizildi.

7. BEKLENEN SONUÇLAR VE SİSTEM İŞLEYİŞİ TAHMİNLERİ

Projenin tamamlanmasıyla birlikte şu sonuçların elde edilmesi beklenmektedir:

1. Beklenen Doğruluk Oranı:

- Ultrasonik sensörün doğruluğu, engelin boyutu ve yüzey türüne bağlı olarak değișebilir. **Düz yüzeylerde** sistem **%95**'e kadar doğruluk sağlayabilir.
- **Karmaşık yüzeylerde** (**eğimli** veya **yansıtıcı** yüzeyler) bu oran **%85**'e kadar düşebilir.
- Sistem yazılımında, **yanlış pozitifleri** (engelin olmadığı halde engel var gibi algılanması) ve **yanlış negatifleri** (engelin olduğu halde algılanmaması) en aza indirmek için algoritmalar geliştirilmiştir.
- Sistem **100 cm'ye** kadar mesafe ölçebilmekte ve **10 cm'den** az mesafede kullanıcıyı **sesli/görsel** olarak uyarabilmektedir.
- **Buzzer**, yalnızca **ciddi tehlike** durumlarında devreye girerek sürücünün gereksiz rahatsızlık yaşamamasını önler.
- Raspberry Pi 3, anlık olarak mesafeyi ekrana yazdırabilir.
- **Mesafeye göre** LED yanıp sönecek veya buzzer devreye girecek.
- Kullanıcı dostu arayüz, park işlemini kolaylaşıracaktır.
- Raspberry Pi tabanlı sistemin, araç aküsüne aşırı yük bindirmeyecek şekilde enerji tüketimi optimize edilecektir.
- Belirlenen mesafe sınırlarının aşılması durumunda, **sesli veya görsel** bir uyarı sistemi devreye girecektir.

2. Sistem Çıktısı:

1. Sesli Uyarı (Buzzer):

- Engel mesafesine bağlı olarak buzzer sesi artan bir frekansta çalar.

Örneğin: **<10 cm mesafe**: Sürekli yüksek frekansta uyarı.

2. Görsel Bildirimler:

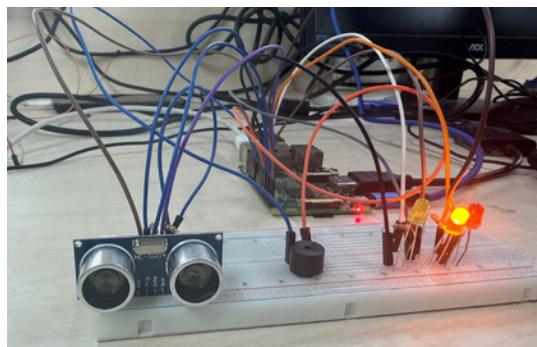
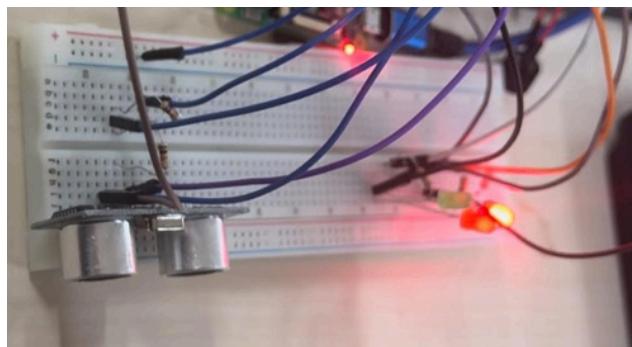
- LED Işığı:** Mesafeye bağlı olarak farklı renklerde yanabilir (ör. yeşil, sarı, kırmızı).
- Mesafe ölçümü Raspberry sistemi** aracılığıyla gösterilmektedir. **Örneğin:**

Mesafe **10 cm'den** azsa, **kırmızı** ışığı ve sesli uyarıyı açar.

Mesafe **10 ile 25 cm** arasındaysa, **sarı** ışığı açar.

Mesafe **25 ile 100 cm** arasındaysa, **yeşil** ışığı açar.

Mesafe **100 cm'den** fazlaysa, tüm LED'leri ve sesli uyarıyı kapatır.



```
1 import RPi.GPIO as GPIO
2 import time
3
4 # Disable warnings and set BCM mode
5 GPIO.setwarnings(False)
6 GPIO.setmode(GPIO.BCM)
7
8 # Pin Setup
9 TRIG = 4
10 ECHO = 18
11 GREEN = 17
12 YELLOW = 27
13 RED = 22
14 BUZZER = 23 # Define the pin for the buzzer
15
16 # Set pins as input/output
17 GPIO.setup(TRIG, GPIO.OUT)
18
19 while True:
20     GPIO.output(TRIG, 1)
21     time.sleep(0.0001)
22     GPIO.output(TRIG, 0)
23
24     GPIO.wait_for_edge(ECHO, GPIO.RISING)
25     start = time.time()
26
27     GPIO.wait_for_edge(ECHO, GPIO.FALLING)
28     end = time.time()
29
30     duration = end - start
31
32     distance = duration * 34300 / 2
33
34     print("Distance: " + str(distance) + " cm")
35
36     if distance < 10:
37         GPIO.output(BUZZER, 1)
38         GPIO.output(RED, 1)
39     else:
40         GPIO.output(BUZZER, 0)
41         if distance < 25:
42             GPIO.output(YELLOW, 1)
43         else:
44             GPIO.output(YELLOW, 0)
45             if distance < 100:
46                 GPIO.output(GREEN, 1)
47             else:
48                 GPIO.output(GREEN, 0)
```

Resim 1: mesafe **10 cm'den azalırsa**, kırmızı ışığı ve sesli uyarıyı açar.

Resim 2: mesafe **10 ile 25 arasında** olduğunda sarı LED yanar.

Resim 3: Mesafe ölçümü **Raspberry sistemi** aracılığıyla gösterilmektedir.

8. ALTERNATIF TEKNOLOJILER VE SEÇİM NEDENLERİ

8.1 ALTERNATIF TEKNOLOJILER

1. Arduino:



Daha basit ve uygun maliyetlidir; **ancak** Raspberry Pi kadar **güçlü** bir işlemciye ve çok yönlü işletim sistemine sahip değildir.

2. Lidar Sensörleri:



Daha hassas mesafe ölçümü yapar, **ancak** maliyeti yüksektir.

3. Kamera Tabanlı Sistemler:



Görüntü işleme ile engelleri algılayabilir; **ancak** bu tür sistemler, Raspberry Pi'nin işlem gücünü zorlayabilir ve daha karmaşık yazılımlar gerektirir.

4. Kızılıötesi Sensörler(IR):



Ultrasonik sensörlerle kıyasla hassasiyeti düşük, maliyeti daha uygun olabilir.

8.2 PROJEDE KULLANILAN TEKNOLOJILERIN SEÇİM NEDENLERİ

1. Raspberry Pi 3:

Raspberry Pi 3, çok yönlülüğü ve güçlü işlemci özellikleri nedeniyle tercih edilmiştir. Gelişmiş bağlantı seçenekleri, yazılım geliştirme kolaylığı ve geniş topluluk desteği de önemli faktörlereidir.

Raspberry Pi'nin Python desteği, proje geliştirmeyi hızlandırır.

2. HC-SR04:

HC-SR04 Ultrasonik Sensör, uygun maliyetli ve yeterli doğrulukta olduğu için seçilmiştir.

3. Python:

Python, kolay öğrenilebilirliği ve Raspberry Pi ile uyumu nedeniyle tercih edilmiştir.

9. GERÇEK HAYATTA KULLANIM SENARYOLARI

1. Park Etme:

Senaryo: Bir sürücü dar bir otopark alanına giriyor. Sistem, engellerin mesafesini ölçerek sürücüyü sesli ve görsel olarak yönlendirir.

2. Geri Geri Gitme:

Senaryo: Sürücü geri manevra yaparken arkasındaki çöp kutusunu fark etmez. Sistem, çöp kutusunu algılar ve sürücüyü zamanında uyarır.

3. Dar Alanlardan Geçiş:

Senaryo: Araç, iki duvar arasında park etmeye çalışıyor. Sistem, mesafeyi sürekli kontrol ederek sürücüyü yönlendirir.

4. Kör Nokta Algılama:

Senaryo: Araç hareket halindeyken yanındaki bir bisiklet algılanır. Sistem, sesli/görsel uyarı vererek sürücüyü tehlikeye karşı bilgilendirir.



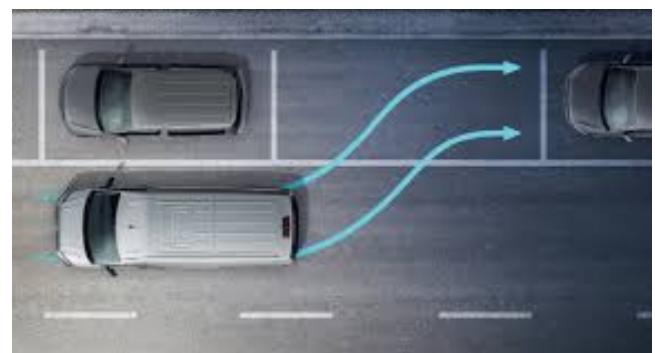
Örnek - Resim 1



Örnek - Resim 2



Örnek - Resim 3



Örnek - Resim 4

10. KAYNAKÇA

1. Raspberry Pi Foundation, Raspberry Pi Kullanım Rehberi .

<https://projects.raspberrypi.org/en/projects/physical-computing>

2. "Raspberry Pi ve Ultrasonik Sensörler Kullanılarak IoT Tabanlı Park Yardım Sistemi".

<https://thesai.org/Publications/ViewPaper?Volume=12&Issue=6&Code=IJACSA&SerialNo=23>.

3. HC-SR04 Ultrasonik Sensör Teknik Özellikleri ve Kullanımı

https://components101.com/sites/default/files/component_datasheet/HCSR04%20Datasheet.pdf

4. Raspberry Pi ile Fiziksel Bilişim Projeleri

<https://components101.com/tags/ultrasonic-sensor>

5. Raspberry Pi Python Kütüphaneleri ve GPIO Kullanımı

https://components101.com/sites/default/files/component_datasheet/HCSR04%20Datasheet.pdf

6. Raspberry Pi Ultrasonik Sensör HC-SR04 Arayüzü Eğitimi

<https://www.electronicshub.org/raspberry-pi-ultrasonic-sensor-interface-tutorial/>

7. Ultrasonik sensörlerin nasıl çalıştığını ve Raspberry Pi ile entegrasyonu

<https://newbiely.com/tutorials/raspberry-pi/raspberry-pi-ultrasonic-sensor>

8. Raspberry Pi ile Buzzer Bip

<https://www.freva.com/buzzer-on-raspberry-pi-generating-beeps/>

9. Raspberry Pi ile Ultrasonik Sensörler ve Buzzer'ların Kullanımı

<https://newbiely.com/tutorials/raspberry-pi/raspberry-pi-ultrasonic-sensor-piezo-buzzer>

10. Raspberry Pi için Python GPIO Kütüphanesi

<https://newbiely.com/tutorials/raspberry-pi/raspberry-pi-ultrasonic-sensor-piezo-buzzer>